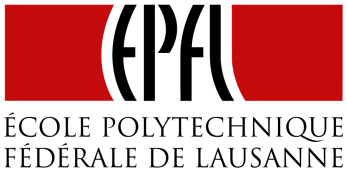


UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Incorporating Human Expertise in Robot Motion Learning and Synthesis

Hang Yin

Supervisors:	**Doctor Ana Maria Severino de Almeida e Paiva**
	Doctor Aude Billard
Co-Supervisor:	**Doctor Francisco António Chaves Saraiva de Melo**

Thesis approved in public session to obtain the PhD Degree in

Electrical and Computer Engineering

Jury final classification: Pass with Distinction

2018

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Incorporating Human Expertise in Robot Motion Learning and Synthesis

Hang Yin

Supervisors:	**Doctor Ana Maria Severino de Almeida e Paiva**
	Doctor Aude Billard
Co-Supervisor:	**Doctor Francisco António Chaves Saraiva de Melo**

Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering

Jury final classification: Pass with Distinction

Jury

Chairperson:	**Doctor José Alberto Rosado dos Santos Victor**	**Instituto Superior Técnico, Universidade de Lisboa**
Members of the Committee:		
	Doctor Aude Billard	**School of Engineering, École Polytechnique Fédérale de Lausanne, Switzerland**
	Doctor José Alberto Rosado dos Santos Victor	**Instituto Superior Técnico, Universidade de Lisboa**
	Doctor Manuel Fernando Cabido Peres Lopes	**Instituto Superior Técnico, Universidade de Lisboa**
	Doctor Cristina Manuela Peixoto Santos	**Escola de Engenharia, Universidade do Minho**
	Doctor Sylvain Calinon	**IDIAP Research Institute, Switzerland**

Funding Institutions

Fundação para a Ciência e a Tecnologia
National Centres of Competence in Research, Switzerland

2018

Acknowledgements

First of all, I would like to express my greatest gratitude to Prof. Aude Billard and Prof. Ana Paiva, for offering me the opportunity to pursue my PhD in their groups. During my study, I enjoyed much freedom of exploring research ideas that interest me and insightful guidance when I needed it. This thesis would not have been possible without their patience, comments and continuous supports. A special thank you goes to my co-supervisor Prof. Francisco S. Melo, for his scientific advices and inspirations. I also thank all my supervisors for their efforts and time spent on reading, commenting and revising my reports and paper manuscripts.

I would also like to take this opportunity to thank Prof. José Santos-Victor, Prof. Manuel Lopes, Prof. Cristina Santos and Dr. Sylvain Calinon, for their kindness in being part of my jury committee and constructive comments on the draft of this thesis.

I was really fortunate to spend these fruitful years in two different but equally amazing countries. Beyond experiencing those natural and cultural things, it was extremely lucky for me to be surrounded by so many wonderful fellow colleagues from both of the two labs. The day I arrived in Lausanne happened to be a cold one in late November, but the reception from LASA was a warm one. I would like to thank Miao for the help from his family, our collaborative work and countless discussions, on both scientific and non-scientific topics. Thanks also to Ravin for his advice as a senior IST-EPFL student and many fun moments in our office sharing. I thank Nicola for hosting those memorable world cup & movie watching nights and thank Lucia for exchanging our feelings and encouragements throughout this endeavor. I owe a special thank you to senior lab members, Basilio, Sahar and especially to Mohammad, Seungsu, Suphi, Klas and Ashwini for always being helpful with scientific questions and coding is-

sues. Thanks to the colleagues I have worked and hung out with, Guillaume de Chambrier, Ajay, Silvia, Felix, José, Sina, Iason, Nadia, Mahdi, Joel, Guillaume Pihen, Luka, Bidan, João, Ajung and Prof. Kenji Tahara, for being such a diverse and energetic team. My deepest gratitude also goes to my GAIPS friends at IST, University of Lisbon. I would like to thank Rui and Filipa, for their help with so many things since I started my life in Lisbon and kind words when I was down. I thank Shruti and Patrícia for their collaborations and supports in my research work. Thanks to Alexis for our discussions, sometimes debates, on math problems and sharing geek jokes. Thanks to him, both Miguels, Faria and Vasco, as well as Ali for the energy and fun they brought to my lives inside and outside the office. Thanks to Kim for hosting many events, academic or non-academic, including my visit to Brooklyn. Thanks to Sofia for her kind encouragements and ideas about historical games and Talasnal. Thanks to Tiago, Carla, Fernando, Maria, Raul, Mojgan, Catarina, Joana, Raquel, Chloe, Ramona, Elmira, Pedro and Samuel, for their accompany in this trip.

I would also like to thank Prof. Pierre Dillenbourg, Séverin, Wafa, Thibault and Deanna for their comments and help on my research works in the CoWriter project. I thank IST-EPFL Joint Initiative, NCCR and FCT for generously funding this project and my PhD study. A thank you note also goes to the administrative staff, Mrs. Joanna Erfani from LASA, Mrs. Sandra Sá from GAIPS, and especially the EDRS program secretary Mrs. Corinne Lebet for their assistance and efforts from the begin to the end.

Finally I am truly thankful to my parents for their understanding and mental supports all along this journey. This thesis is dedicated to them.

Lisbon, 15 June 2018 Hang Yin

Abstract

With the exponential growth of robotics and the fast development of their advanced cognitive and motor capabilities, one can start to envision humans and robots jointly working together in unstructured environments. Yet, for that to be possible, robots need to be programmed for such types of complex scenarios, which demands significant domain knowledge in robotics and control. One viable approach to enable robots to acquire skills in a more flexible and efficient way is by giving them the capabilities of autonomously learn from human demonstrations and expertise through interaction. Such framework helps to make the creation of skills in robots more social and less demanding on programing and robotics expertise. Yet, current imitation learning approaches suffer from significant limitations, mainly about the flexibility and efficiency for representing, learning and reasoning about motor tasks. This thesis addresses this problem by exploring cost-function-based approaches to learning robot motion control, perception and the interplay between them.

To begin with, the thesis proposes an efficient probabilistic algorithm to learn an impedance controller to accommodate motion contacts. The learning algorithm is able to incorporate important domain constraints, e.g., about force representation and decomposition, which are nontrivial to handle by standard techniques. Compliant handwriting motions are developed on an articulated robot arm and a multi-fingered hand. This work provides a flexible approach to learn robot motion conforming to both task and domain constraints.

Furthermore, the thesis also contributes with techniques to learn from and reason about demonstrations with partial observability. The proposed approach combines inverse optimal control and ensemble methods, yielding a tractable

learning of cost functions with latent variables. Two task priors are further incorporated. The first human kinematics prior results in a model which synthesizes rich and believable dynamical handwriting. The latter prior enforces dynamics on the latent variable and facilitates a real-time human intention cognition and an on-line motion adaptation in collaborative robot tasks.

Finally, the thesis establishes a link between control and perception modalities. This work offers an analysis that bridges inverse optimal control and deep generative model, as well as a novel algorithm that learns cost features and embeds the modal coupling prior. This work contributes an end-to-end system for synthesizing arm joint motion from letter image pixels. The results highlight its robustness against noisy and out-of-sample sensory inputs. Overall, the proposed approach endows robots the potential to reason about diverse unstructured data, which is nowadays pervasive but hard to process for current imitation learning.

Key words : learning from demonstrations ; inverse optimal control ; robot motion synthesis and control ; deep generative model.

Resumo

O crescimento exponencial da robótica associado ao rápido desenvolvimento das capacidades cognitivas e motoras dos robôs, permite antever que humanos e robôs venham a conseguir executar trabalho conjunto em ambientes não estruturados. No entanto, para tal ser possível, os robôs necessitam de ser programados para funcionar nesses cenários complexos, o que requer conhecimentos profundos no domínio da robótica e do controlo. Uma alternativa viável é dotar os robôs de mecanismos de aprendizagem automática que permitam, de uma forma flexível e eficiente, aprender a realizar novas tarefas com base em demonstrações feitas durante a interação com humanos. Tal abordagem permite tornar a criação de competências nos robôs num processo mais social e, principalmente, menos dependente de programadores especializados. Contudo, as abordagens atuais para aprendizagem por imitação apresentam ainda limitações significativas, principalmente no que diz respeito à flexibilidade e à eficiência nos processos de representação, aquisição e raciocínio sobre tarefas motoras. Esta tese aborda esse problema, explorando abordagens baseadas em funções de custo para a aprendizagem quer do controlo de movimento, quer da percepção, quer da interação entre as duas componentes.

Numa primeira parte, a tese propõe um algoritmo probabilístico eficiente para a aprendizagem de um controlador de impedância de forma a acomodar contatos durante o movimento. O algoritmo incorpora restrições essenciais, por exemplo no que diz respeito à representação e decomposição de forças, restrições essas que não são triviais de incorporar utilizando técnicas standard. O algoritmo proposto é exemplificado num cenário em que um manipulador dotado de uma mão robótica com dedos individuais aprende a escrever manualmente. O método desenvolvido para aquisição de movimento a partir de demonstrações

permite lidar tanto com restrições específicas da tarefa como do domínio.

De seguida, a dissertação contribui novas técnicas de aprendizagem e raciocínio baseadas em demonstrações com observabilidade parcial. A abordagem proposta combina controlo ótimo inverso e métodos ensemble, permitido obter um processo de aprendizagem tratável com base em funções de custo com variáveis latentes. Este método permite também a incorporação de informação prévia sobre a tarefa, por exemplo, acomodando informação sobre a cinemática humana, resultando num modelo que sintetiza escrita manual dinâmica, rica e credível. Este método acomoda informação prévia sobre o comportamento dinâmico das variáveis latentes, o que facilita a inferência em tempo real sobre a intenção humana e permite uma adaptação online do movimento do robô em tarefas colaborativas.

Finalmente, a tese estabelece uma ligação entre as duas modalidades exploradas: controlo motor e percepção. É oferecida uma análise onde se estabelece a relação entre controlo ótimo inverso e um modelo de geração profundo. A partir desta análise, é proposto um novo algoritmo que permite a aprendizagem de *features* da função de custo incorporando conhecimento prévio sobre o acoplamento modal. Assim, a tese contribui com um sistema completo, capaz de sintetizar o movimento das várias juntas de um manipulador a partir de imagens de letras. Os resultados obtidos realçam a robustez do sistema face a inputs sensoriais com ruído e fora da amostra. No seu todo, a abordagem proposta dota robôs com o potencial de raciocinar sobre dados não-estruturados de natureza diversa, frequentemente encontrados em diversas áreas e aplicações mas que oferecem significativa dificuldade de processamento para os atuais algoritmos de aprendizagem por imitação.

Palavras-chaves: Aprendizagem por demonstração; controlo ótimo inverso; síntese e controlo de movimento de robô; modelo de geração *com aprendizagem profunda.*

Résumé

Face à l'avancée exponentielle de la robotique et au développement rapide de leur capacités cognitives et moteurs, nous pouvons dors-et-déjà envisager les robots et les hommes travaillant ensemble sur une tâche partagée, dans le chaos d' environements non structurés. Pour l'instant, afin de rendre cela possible, les robot doivent être programmés pour de tels types de sénarios complexes, ce qui demande chez l'utilisateur des competances avancées en robotique et contrôle. Une approche viable pour apporter aux robots la faculté d'aquérir des capacités d'une manière à la fois flexible et efficace consiste à leur donner la possibilité d'apprendre de façon autonome à partir de démonstrations faites par l'homme et à force d'experimenter les interactions. Un tel cadre favoriserait la création de nouvelles capacites chez des robot plus sociaux et réduirait le besoin d'expertise en programmation et robotique chez l'homme. Jusqu'ici, cette approche d'apprentissage par imitation souffre de limitations significatives, principalement en ce qui concerne la flexibilité et l'efficacité du robot à se représenter, à apprendre et à raisonner sur sa tâche. Cette thèse de doctorat contribue à résoudre ce problème en proposant une approche basée sur des fonctions de coût pour l'apprentissage de la gestuelle, pour la perception, et pour l'adaptation du geste à la perception.

Pour commencer, cette thèse propose un algorithme probabiliste efficace pour l'apprentissage d'un contrôle basé sur un modèle d'impédance pour l'adaptation d'un mouvement à des contacts physiques. L'algorithme d'apprentissage est cappable d'incorporer d'important domaines de contraintes, e.g. la représentation et la décomposition d'une force, ce qui n'est pas trivial à prendre en compte avec les techniques habituelles. La gestuelle liée à l'écriture manuscrite conforme est implémentée pour un bras articulé de robot et pour une main

robotique à plusieur doigts. Ce travail présente une approche flexible pour l'apprentissage moteur des robots, qui s'adapte à la fois aux contraintes de la tâche et du domaine.

En outre, cette thèse propose une approche pour apprendre et raisonner à partir de démonstrations partiellement observées. L'approche combine des methodes de contrôle inversé avec des méthodes de modelisation par des ensembles de fonctions, optimisant des fonctions de coût au travers de variables latentes. Deux a-priori sur la tâche sont alors incorporés. Le premier est un a-priori sur la cinématique d'un mouvement humain, qui résulte d'un modèle synthétisant une écriture manuscrite riche et convaincante. Le second a-priori impose la dynamique du la variable latente et facilite la compréhension de l'intention de l'homme et l'adaptation à cette intention pour une tâche collaborative en temps réel.

Finalement, cette thèse trace le lien entre le contrôle et la perception des modalités. Cette partie présente d'une part une analyse qui relie le contrôle optimal inversé et les modèles génératifs profonds, et d'autre part un nouvel algorithme qui apprend des caractéristiques de coût en vue d'incorporer un a-priori sur le couplage lié aux modalités. Elle propose enfin un système complet pour synthétiser les mouvements des articulations d'un bras méchanique à partir d'images de lettres pixellisées. Les resultats mettent en valeur sa robustesse, émergant d'un percept pourtant chaotique et inintelligible. Globalement, l'approche proposée dotte les robots d'une capacité à raisonner sur des données diverses et non-structurées, aujourd'hui omniprésentes mais encore bien difficiles à traiter dans le cadre actuel de l'apprentissage par imitation.

Mots clefs : apprentissage par imitation, contrôle optimal inversé, synthèse et contrôle de mouvement robotiques, modèles génératifs profonds.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Embodied agents such as robots promise great economical and social benefits for the humanity. The uniqueness of robots lies in their capabilities of affecting the environment through physical motion effects. In the last decades, the deployment of robotic systems, especially industrial ones, has largely relived human labors from repetitive, tedious or hazard tasks. Recently, as robots that work outside the factory cages, light-weight manipulators are emerging thanks to the maturity of new actuation techniques (162, 3). This trend of soft robotics opens a possibility for robots to work with humans in a close proximity, envisioning not only small-patch manufacturing but also human-centered service and assistance. However, for these applications, the hardware itself is not the only barrier. Unlike the cases in factories, the tasks and environments in human-centered applications are highly diverse and unstructured, soliciting substantially improved robot skill repertoires and adaptability. Current solutions are inadequate here: most robots nowadays are meticulously hand-programmed, which often requires extensive efforts and task domain knowledge. It is thus necessary to investigate new strategies of synthesizing robot motion to bridge this

gap.

By contrast, humans exhibit remarkable mastery and versatility in terms of motor skills, ranging from nimbly manipulating objects in hand to harmoniously twitching whole-body muscles in sprint. While this superiority highlights the biomechanical properties of the human body, established sensorimotor research has also attached great importance to the notation of internal model (230). An internal model encodes the prior knowledge about motor commands and the motion result. The encoded knowledge is exploited in the so-called active inference (58) for both perception (107) and motion control (219). For instance, to swing a racket and hit a ball, humans are instructed and practise to attain knowledge about the body and racket movement under the motor command. Skillful motion is developed, enabling humans to adapt to rackets of different weights and to hit the ball with the whole body balanced. Meanwhile, previewing the ball position helps humans to anticipate the hitting impulse, and as such, to timely stiffen the arms for a strike with the expected angle and velocity. To this end, progressively learning and refining an internal model is important in human skill acquisition. Studies in sensorimotor learning identifies two main ways of achieving this, including interacting with the environment and observing others' behaviors (235).

As the relevant counterpart in agents, Machine Learning (ML) techniques explore data-driven approaches to reason about and work out perceptual and decision-making tasks. While ML has achieved significant successes in tasks like image classification and game playing, the application in robotics faces some unique challenges. To begin with, robotic tasks are executed by an integrated system, which often involves multiple sensory and actuation modules. Thus, the ML methods need to be tailored to deal with different types of data and subtasks. Secondly, robot learning rarely has the access to a massive labeled dataset. Specifically, data instances with informative labels, e.g., success in executing the task, are lacking. Gathering successful instances by exploring in the physical world is expensive and even risky for robots. Synthesizing data from simulators is relatively cheap but the accuracy of simulating certain effects, e.g., physical contacts, is still unsatisfying. In that sense, human demonstrations are worth to be exploited because they contain direct and dense information signaling how

to execute the task. Thirdly, robotics and human motion science possess much well-established research. The design of ML methods can benefit from merging these pieces of research. Also, the incorporation of domain priors are useful for learning from small dataset. Last but not the least, the computational cost of ML techniques is critical in many robotic applications. With a rapid algorithm, it is potential for robots to adapt by incrementally learning new data. An efficient inference is also entailed because of the request of reasoning about sensory data in real-time.

This thesis is concerned with the research question:

> how can a robot incorporate human expertise to facilitate its motion control, perception and the interplay of the two.

The main contents and contributions of the thesis are placed in the domain of Learning from Demonstrations (LfD). The LfD paradigm enables robotic agents to acquire desired behaviors based on expert demonstrations. The human expertises include both task demonstrations and domain priors. More specifically, the thesis focuses on (inverse) optimal control and generative model, which respectively situate in robotics and ML. Both of techniques realize LfD in a similar way. The general idea is to interpret data with scalar functions or statistical moments, which, for example, make the demonstrations incur low function values or high data likelihoods. Learning demonstrated behaviors boils down to estimating the function or moments. The task synthesis can then be shaped to generate samples that are subject to the same functions or moments, hence imitating or learning from the demonstrations.

Learning motion control from demonstrations needs to consider domain knowledge such as task-dependent constraints. For instance, controlling a reaching movement requires identifying the reaching point and applying appropriate corrections around the point to accommodate disturbances. When a trajectory is of interest, the robot needs to extract the motion reference and decompose the control directions along the trajectory. These constraints are useful from the robotics point of view. However, as identified in Chapter 3, incorporating task-dependent constraints sometimes makes the learning problem ill-posed to standard techniques. Thus it is necessary to adopt new methods to address this challenge.

While a local trajectory control provides certain robustness to small disturbances, humans demonstrate an adaptability beyond that. In fact, humans can exploit the redundancy of performing the task, e.g., taking different paths to reach an object and grasp it, to adapt to their preferences or contextual conditions, such as the existence of an obstacle. However, the preferences or conditions might not be observable due to the limited robot perception capability. In that sense, the robot needs to learn from incomplete demonstration data and discern the contextual conditions in execution. Current techniques solve this through expensive numerical optimizations without explicitly considering the unobservables. Efficient learning and inference techniques are desired to reason about this type of demonstrations.

Finally, learning and linking robot perception and control often resorts to handcrafting data features for each modules. Usually, this is tedious and not straightforward for sensory modalities like images. Thus, it necessitates an approach to automate the feature engineering process, as such boost the productivity and flexibility of the LfD approaches. Progresses have been made in representation learning to enable agents to abstract important features that are relevant to the task. Leveraging these progresses in the LfD framework can facilitate learning from complex types of data and devising the control loop in an end-to-end manner.

1.2 Approaches

The main techniques explored in the thesis are optimal control (and its inverse problem) as well as generative modeling. The following sections introduce basic principles, applications and the specific variants that are employed in the thesis.

1.2.1 Optimal Control and the Inverse Form

Imagine you start stretching your arm from a certain posture to touch a spot on the table. Such basic movement actually coordinates multiple joints and muscles of the human body, implying a plethora of possible ways to execute this task. Yet, it has been demonstrated that, though humans barely think over this movement before acting, their behavior patterns are highly stereotyped.

For instance, the motions are stereotyped in terms of consistent features such as velocity profiles. This seemingly contradicting fact implies regularities and structures that drive us to take selective actions. Research has suggested that the possible principle behind is optimality: we choose to adopt and control a motion trajectory that is optimal with respect to certain performance criteria. The identified criterias include the motor effort (226, 5, 93) and the motion variation under sensorimotor noises (74, 215). Interestingly, the applicability of optimality principle is beyond neurophysiology. Even before the success of calculus of variations in solving the brachistochrone curve problem, the early optimal control ideas helped in describing physics phenomenas such as light reflection and refraction (199), and eventually evolved to the Pontryagin maximum principle and correlated to more general topics including Hamiltonian and quantum mechanics.

Optimization-based control has long been the workhorse method in robot planning and control. After all, it is much more intuitive to design high-level task metrics than to explicitly program commands for many robot degree-of-freedoms (DOFs). Modern solvers based on direct optimization like Sequential-Quadratic-Programming (SQP) can generate an optimal trajectory within a sub second or even millisecond interval. Thus, real-time model predictive control is possible in sophisticated robot systems such as quadcopter (61, 7) and humanoid robots (115, 116). Instead of such a direct approach, this thesis bases most of the control derivation on an indirect approach. An indirect approach relies on the Hamilton-Jacobian-Bellman (HJB) equation and can provide a regulator or a feedback control besides the optimal trajectory. The downside is that it is often more expensive to apply indirect approaches to general problems, unless the integral of system dynamics is simple and efficient. One seminal example about this is the work from Kalman (88), who proposed an efficient algorithm to obtain an optimal feedback controller for Linear Quadratic Regulator (LQR) systems. In order to relax the constraints about the system form, differential dynamic programming (44, 212) and iterative LQR (220, 228) advocate to approximately solve the problem in a successive manner. The HJB equation in these indirect approaches correlates to a natural probabilistic interpretation, which will be exploited throughout the thesis. Meanwhile, the indirect ap-

proach is also fundamental to the adopted cost/cost-to-go function structure in the inverse problem.

The inverse problem of optimal control is simply the opposite of searching the trajectories that incur the minimum costs: with respect to which cost function are a set of trajectories (locally) optimal. From a behavioral perspective, the goal is to infer the driving forces or the motivations given the observation of the agent behavior, assuming the agent is following the optimality principle. If the estimated cost function is accurate, another agent can simply develop its own behavior guided by the same goal via a forward optimal control. To this end, Inverse Optimal Control (IOC) allows to imitate or transfer task skills among agents, so it is of interest to the central topic of the thesis. The early IOC work is again pioneered by Kalman, who discussed an "inverse LQR": under which LQ system a given linear feedback controller is optimal (89). The progress about a more general formulation is, however, much more recent. Relevant works include apprentice learning (1) and the duality of nonlinear control-affine systems (217, 91), opening much research ranging from the probabilistic formulation of the inverse reinforcement learning to the recent efforts of interpreting the IOC problem as a generative adversarial network (56). This thesis is generally rooted in the probabilistic formulation of the IOC problem. This formalization, specifically the linearly-solvable system and its variations (216, 52), provides a general and sound foundation to bridge the HJB-based control synthesis and the human prior embedding. Similar to (56), the thesis also takes an eye on the connections and impacts of representation learning around IOC. However, the motivation is rooted in robotics and the connection is established to a different deep generative model.

1.2.2 Generative Model Learning

In machine learning, a generative model observes presented samples, extracts hidden structures and synthesizes samples that are similar to the observed ones. Therefore, the inherent problem of learning a generative model also concerns the notion of imitation. In most cases, the purpose of having a generative model is to unconditionally create plausible samples. Thus it differs from a discriminative model, which makes predictions conditioning on an input. Learning generative

models is gaining much momentum these days because the desire of processing a huge amount of data and the high expense of exhaustively labeling them. Also, comprehending the process of pattern generation appears to be important for analyzing and understanding the pattern itself, just as stated by a famous quote:

> What I cannot create, I do not understand.
>
> —Richard Feynman

Statistical learning searches a generative model in the hypothesis space to match certain empirical evidences. Data likelihood is a common choice if the data is assumed to be truly sampled from the candidate distribution. However, natural data and standard distributions with nice properties often violate this assumption. In light of this, advanced models choose to adopt non-trivial structures, e.g., by adding hidden variables, and to surrogate the true likelihood or similarity to the true model. Probabilistic models like Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) have been vastly used in numerous applications because of a good trade-off between the model capacity and efficiency. On one hand they allow hidden variables for modeling complex data, while on the other hand the efficiency is retained with well-behaved hidden variables and their posterior distributions. In the recent renaissance of neural models, the intersection between the probabilistic model and representation learning has become one of the research spotlights. Typical approaches include generative adversarial network (GAN) (64) and variational auto-encoders (VAE) (100). In these approaches, the randomness is separated as a simple prior distribution, such as an isotropic Gaussian. Models build their high capacity upon a non-trivial posterior with a complex and differentiable feature mapping. These so-called deep generative models have been shown effective for synthesizing highly unstructured patterns such as images (164, 170) and audios (38, 229).

This thesis resorts to generative model techniques for learning and synthesizing robot motion. The idea of hidden variables is adopted to tackle the computational challenge in inverse optimal control as well as empirical applications such as learning from incomplete demonstration data. Moreover, the thesis also incorporates the progress in the deep generative model research to

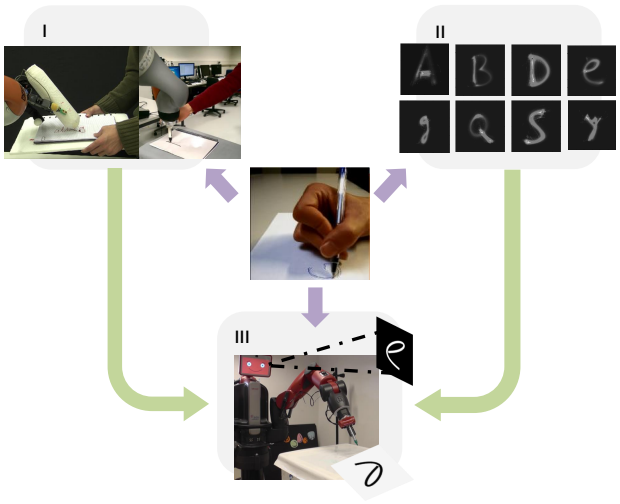


Figure 1.1: Thesis structure concerning different robotics aspects: i) impedance-based robot control; ii) encapsulating and interpreting motions of different modes; iii) association between control and perception. Learning from human demonstrations is central throughout all these aspects while the task is decoded and instantiated on different robots.

deal with raw sensory data.

1.3 Thesis Structure and Contributions

This thesis is organized in alignment with the stated main research questions about motion control, perception and sensory-motor association (Figure 1.1). The next chapter starts by discussing the research background, including a section of pinning the interested topics on the grand picture, a review of related literatures and a brief description about the background CoWriter project. Before a final summary and discussion in Chapter 6, the main contributions for each sub research question are presented. Large portions of the thesis work have been published on or submitted to peer-reviewed conferences or journals. A short summary about the main contributions and relevant publications are given below.

Chapter 3 focuses on composing a motion controller based on learning from human demonstrations. The main result is an efficient sampling-based IOC algorithm that learns structured cost-to-go functions with the human-designed

constraints embedded. The robotics application demonstrates that the approach can be used to derive feedforward reference trajectory and gain parameters for a compliance controller, of which the compliance parameter is described in a moving reference frame. This part of work has appeared in the publications of (238), (239) and (242).

Chapter 4 extends the first piece of work to model a skill repertoire rather than a single reference trajectory. Similar to the first part, human inspired priors are also incorporated. The presented ensemble approach is shown as an efficient way of modeling multi-mode human behaviors, with the applications including synthesizing human-like dynamical handwriting and online human intention inference. This part of work has been presented as a conference paper (240) and a journal paper (243).

A further extension about the LfD/IOC framework is presented in Chapter 5, where the difficulty of engineering the data features is alleviated. The robotics motivation lies in the challenge of embedding the association among various sensor modalities, most of which are not easy to be represented with a hand-crafted feature. The key novelty is an idea of factorizing the LfD model for an efficient inference upon high-dimensional data. As a result, the robot features the capacity of synthesizing a motion trajectory from a raw visual input, thus works as an end-to-end system. The chapter also contributes with the approaches of data augmentation and posterior trajectory optimization. These contributions tackle limited and corrupted sensory data, which are empirical challenges to the implementation on a robot. Techniques about human-like motion synthesis and inference, which are developed in the prior chapters, are reused as part of these approaches. The main contents have been included in the publication (241) and part of results are also reported in (242).

Apart from a robotics-oriented view, the thesis organization can also be understood as a strand of ML algorithms with an increasing complexity (Figure 1.2). Specifically, Chapter 3 extends the basic IOC framework by incorporating priors (the dashed loop) about the interested task feature. In Chapter 4, latent variables are introduced to relax the assumption of full observability. Chapter 5 further eliminates the necessity of knowing the relation between the original sensor representation and task-relevant features. Meanwhile, the learning algo-

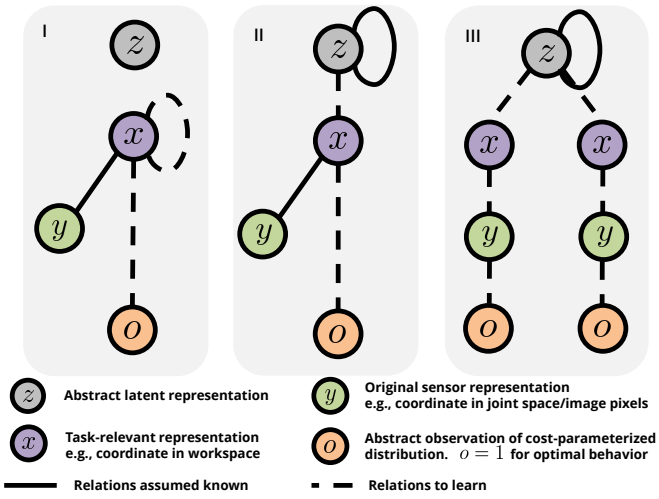


Figure 1.2: Thesis structure from the machine learning point of view: the chapters progressively extend the IOC framework with added structures and complexities.

rithm in Chapter 5 supports to learn from data with multiple modalities that are conditionally independent on the task.

The contents in this thesis are also used or related to co-authored works on other topics. (130) focuses on robot hand grasping and proposes to extract the motion compliance under a set of adhoc constraints. These constraints effectively adapt the principled form presented in Chapter 3. The adaptation removes the prior of representing task in a moving frame of reference, because the motivated bulb insertion task only considers 1D rotational stiffness along a fixed axis. The algorithms of modeling and synthesizing multi-mode handwriting are used in papers (31) and (30). These pieces of research eye on activity design in the context of human-robot interaction. Specifically, human studies are performed to assess how children perceive the robot learning capability and if a smarter robot will engage the children more, and as such improve children's learning gains in a handwriting tutoring activity. In these studies, the synthesis algorithm in Chapter 4 is used to efficiently generate legible dynamical handwriting samples. At the same time, the experiment conditions about the robot capability are intuitively controlled by specifying different levels of model perturbation.

2

Background

This chapter focuses on the thesis research background. The first section 2.1 comes with an overview of robot motion control and learning. The main purpose of this part is to provide a brief introduction about some loosely relevant topics and to situate the thesis work in the grand landscape. Detailed reviews about the closely related works will be given in Section 2.2, along three main axes: compliant motion learning, learning from demonstrations and representation-learning-based sensorimotor control. Then technical preliminaries about the approaches and a glossary of notations are in Section 2.3. The chapter closes with Section 2.4. Setting up the scene of application of the thesis, CoWriter project, which aims to build a robot agent system to help children acquire handwriting skills, is discussed to provide more background information about the thesis work.

2.1 Robotic Motion Control and Learning: Setting the Scene

2.1.1 Robot Motion Control: Feedback and Feedforward

Robot motion execution faces some substantial challenges due to conditions in the real physical world. Specifically, the dynamics of most robot manipulators often exhibits nonlinearity and coupling across multiple joints. Also, uncertainties resulted from the unmodeled effects such as frictions or environment disturbances add more difficulties. A simple yet widely adopted scheme exploits kinematic relations to represent and regulate motion in the joint space (42). The joint space is often controlled through a linear PD or PID controller, which simply assumes a local linearity and a weak joint dependency to cope with the model complexities. In practice, the task specification might not be in the joint space thus it can be insufficient to close the control loop in the joint space. Operational space control (99) addressees this by directly expressing the task dynamics in the operational space. The representations are transformed through the Jacobian of kinematics. This scheme features more task dynamics intuitions to the designers, while it requires invertible Jacobians and can be complex to implement as a centralized system.

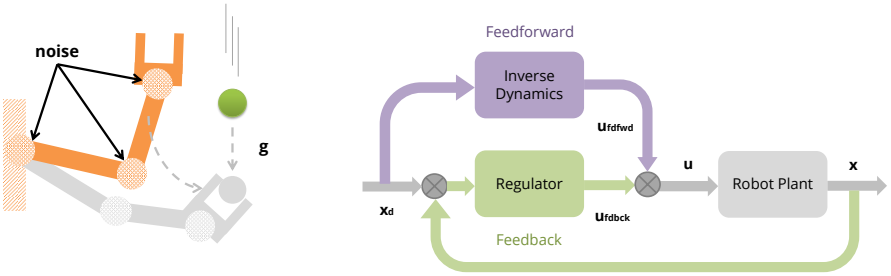
Many kinds of uncertainties source from the physical contacts between a robot and the external environment. The collision event might drastically change the dynamics mode and the operating point, leading to erroneous or even unstable behaviors. Hence, an appropriate force accommodation is critical for contact-rich applications. In industrial assembly, an early solution is a mechanical device called remote center of compliance (RCC) (47). The device is a mechanical part providing a passive compliance to absorb the impact from the rigid environments. Software solutions resort to a controller and the algorithm design to address this problem. Direct force control approaches monitor and track contact force by mounting force/torque sensors on the end-effector. However, the conflict between force and position control loop implies that the position and force tracking errors cannot be concurrently eliminated in the same direction (135). This spurred the research about decomposing the task directions according to the importance of force or position tracking. The controllers

along these orthogonal directions were then superimposed as a hybrid force-position controller (166). Another approach avoided an explicit specification about the task dimensions by applying parallel homogeneous controllers. To resolve the target conflict, the parallel force-position controller (36) used an integral loop to prioritize the tracking of force component.

One of the notable alternatives to direct force control is stiffness control (182). This approach was extended in the seminal paper (76), which pivoted a trilogy concerning the framework of impedance control. The impedance control, instead of tracking the force signal in a direct way, argues to take the dynamic relation between force and motion as the control objective. Specifically, the casual relation between the velocity and torque was emphasized to design an appropriate impedance and admittance pair for a stable interaction. For instance, when interacting with a stiff environment, the robot oughts to behave as an impedance, generating the reactive force with a positional input. This is in accordance with the principle of RCC which also adds compliance units to the robot. Meanwhile, the force-position relation can be exploited in the other way around. Formalized from such an idea is admittance control, in which the robot generates motion under a driving force. When the force measurement is available, admittance control is more feasible to realize an accurate low impedance behavior. This is because, as argued in (151), the robot dynamical behavior is dominated by its inertia, as such inherently resembles an admittance. The main application of admittance control is physical human-robot interaction, such as powered exoskeletons (95).

Most of these approaches, at least of their basic formalizations, assume a predefined trajectory and focus on the feedback design. Yet for large deviations and systematic uncertainties, it is inadequate to assume errors are rooted from small disturbances and can be corrected through a local feedback. In that sense, feedforward control addresses this by synthesizing the control reference based on a prior model. One of the illustrative examples is that humans can preview the muscle activation for catching a dropping ball based the experience and knowledge about the object gravity (Figure 2.1).

Feedforward control has been widely investigated in the literature on advanced robot motion control. Typical approaches include inverse dynamic and



(a) (b)

Figure 2.1: (a): the robot manipulator anticipates the ball trajectory and derives a feedforward motion based on this prior knowledge. Feedback control regulates the executed motion under disturbances and noises. (b): a common architecture containing both feedforward and feedback components in the robot motion control.

computed-torque control (193, 189), where model-based dynamical terms are computed to cancel nonlinear effects. Adaptive control estimates the unknown model parameters (e.g., the inertia of external workloads) and then derives control based on the predicted values. It has been shown that, for the system inertia uncertainties, one can obtain a linear parameterized inertia wrench so as to design a stable estimator (192). Meanwhile, research efforts have been made for synthesizing feedforward trajectories in an autonomous manner. A large portion of motion planning literatures explored searching (94, 119) and optimal control (220, 136, 210) to compose admissible or low-cost trajectories given task criteria and constraints. Moreover, when models are unavailable, one can explore a motion trajectory via trial-and-error, and as such learn the feedforward control. Among robotics control, a renowned approach based on this principle is iterative learning control (ILC) (41, 9). Specifically, ILC assumes the robot motion can be operated cheaply and repetitively, and suggests a PID-like rule to update the feedforward input. For a control-affine dynamics and a linear observation model, the iterations are guaranteed to converge to a minimized trajectory error in stationary tasks (8). In (237), ILC was adopted and extended to also adapt the gain scheduling in a human-like way. Consequently, both reference trajectory and feedforward impedance profile were learned in a repeated force regulation task.

This thesis falls in the domain of feedforward control. In particular, the focus is on the representation, evaluation and derivation of feedforward motion from prior internal models or signals from other modalities, such as vision. These priors are acquired by learning from humans and embedding the human motion characteristics into the process. In the sense of learning-based methods, the techniques developed in the thesis are in the similar vein as the ILC-like approaches. However, the main thesis work concerns a human-guided learning and correlates to more recent advancements in machine learning. A discussion about motion synthesis in the modern machine learning will be expanded below. The thesis also extensively exploits the results in optimal control and open-loop impedance control for applications involving both humans and robots.

2.1.2 Robot Motion Learning: Discriminative and Generative

In machine learning, depending on the source of data labels, motion synthesis is mainly addressed through two categories of approaches: reinforcement learning (RL) and learning from demonstrations (LfD) [1]. Although the thesis is built upon the latter paradigm, both will be reviewed since RL directly correlates to optimal control, its industrial counterpart, and was fundamental for developing the thesis approaches.

2.1.2.1 Reinforcement Learning: Value-based and Policy-based Approaches

Reinforcement learning is promoted for solving a relatively general AI problem: how can an agent take a sequence of actions to maximize the received rewards [2]. The foundations of general RL approaches can trace back to dynamic programming and Bellman principle (14). Practical learning algorithms emerge as an intersection of the Monte-Carlo method and boostraping for the estimation of value functions (201, 203). Alternative value-based algorithms include renowned Q-learning (232, 196) and SARSA (179). For a continuous state space, function approximators could be utilized (18, 202, 172, 144). The probabilistic stability was proved for the case of learning a linearly parameterized Q-function under certain feature conditions (139). In these works, actions

1. Also known as "programming by demonstrations" or "imitation learning".
2. Or equivalently, minimize the incurred costs.

are implicitly derived from the learned value functions. Q-learning, for example, needs to select the action which causes an optimal value to instantiate the implied policy. Astounding achievements have been made in the applications involving enumerable discrete actions, such as playing Atari (145) and board games (211, 191).

Much of the research in robotics-oriented RL, however, relies on a direct policy optimization: estimating the mapping between a control and a sensory input. The main argument of policy optimization against the value-based approaches lies in its merits for naturally dealing with continuous action space and control constraints, both of which are pervasive in robotics (104). Successful applications include locomotive robots (148, 209, 213, 48, 75), object manipulation (103, 128, 129, 117, 69) and synthetic characters (131).

The policy optimization can be categorized as derivative-free and policy gradient approaches. Derivative-free approaches iteratively fit and sample from a stochastic policy proxy to make good performed rollouts more likely. Exploring a variety of sampling and weighing mechanisms, relevant techniques include finite difference method (152), expectation-maximization (45, 158, 108), cross-entropy-like methods (73, 46, 105), approximate inference control (AICO) (221) and path-integral-based methods (90, 213, 198). Latest research shows that, although a large amount of samples is desired for a low-variance black-box optimization, a smart distributed cross-entropy-like approach is still scalable and competitive for optimizing a high-dimension policy (181).

Gradient-based approaches are based on scoring the cost sensitivity under the policy perturbation, with a similar idea developed in the early works (4, 177) and a fundamental formulation well known as REINFORCE (234). A connection between the REINFORCE and importance-sampling was revealed in (206). Departing from the vanilla formulation, one of the research concerns is determining an appropriate learning rate when applying the gradient. This is especially of the interest in robotics, because the rollouts might be risky and expensive to obtain so the chance of an overshoot should be minimized. Relevant work includes natural gradient (159) and trust-region policy optimization (TRPO) (184). The main idea is evaluating and constraining the policy shift according to certain metrics, for which Fisher information matrix was used in (159) while

(184) exploited the Kullback-Leibler divergence. Other research focused on reducing the gradient variance of the vanilla REINFORCE. As proposed in the original work (234), the key is enforcing an informative baseline to evaluate the advantage of the rollout performance. This connects to the research on actor-critic algorithms (68), where value functions are also learned at the same time to bias the policy gradient. Henceforth, the actor-critic paradigm is somehow a blend of value-based and policy-based approaches. The incorporation of value (critic) learning has been well acknowledged in plenty of state-of-the-art RL algorithms such as asynchronous actor-critic agents (A3C) (146) and generalized advantage estimation (GAE) (185). The application of these algorithms ranges from controlling the robot joint motion to synthesizing abstract agent actions. It is notable that non-parameteric approaches, such as PILCO (48), were also employed for an efficient policy gradient approximation.

2.1.2.2 Learning Policy from Examples

Learning an RL-based agent can be challenging because it requires to attribute delayed observations to a sequence of actions in history, as such solving a credit assignment problem (140). In particular, when the rewarding event is rare, the agent might have to explore exhaustively to gather informative signals. Much like the critic component in the policy optimization, one way to improve the learning efficiency is to bias the exploration with a guidance from (potentially) good examples. In light of this, a model-based approach called guided policy search was proposed in (125) and its integration with path-integral RL was discussed in (32). The idea was repeatedly searching the high performed trajectories through optimal control and fitting the optimal state-control pairs with a neural network policy. This effectively turns the hard policy learning problem into a comparatively easier supervised learning problem. Such an idea was also explored in a grasping synthesis problem, where solutions from a static optimization were fit to a Gaussian Mixture Model (79).

2.1.2.3 Learning Policy from Human Demonstrations

Apart from optimization solvers, one can easily imagine another source of the expert guidance: human demonstrations. As an independent research domain, the idea of programming robots based on human demonstrations has been

explored for decades. The pace of LfD research is roughly synchronous with the development of mainstream AI techniques. A line of the research originated in the eighties focused on automating the robot motion planning through a symbolic representation and graph-like connections (133, 186, 2, 134). Similar to the typical expert systems, if-then rules were used to compose a high-level policy before the concrete geometry motion planning (15).

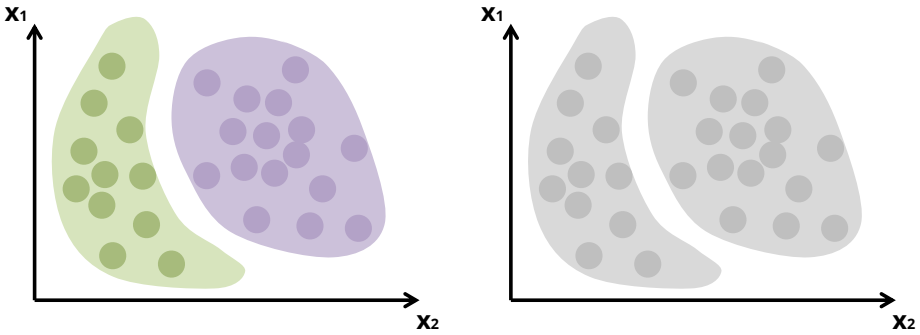
Other recent approaches describe tasks with more details. Leveraging nonlinear regression techniques, demonstrations in these approaches were encoded as state trajectories or parameterized dynamical systems (DS). Early works focused on fitting a time-dependent state trajectory with nonlinear basis functions, such as splines (225, 224). A more formal DS-based treatment was proposed as Dynamic Movement Primitives (DMP) in (83, 183) and gained much popularity for its learning efficiency and the flexibility of encoding both discrete and rhythmic movements. As a canonical system, DMP does not have an explicit dependency on time. It comprises of a linear system with established attractors and a nonlinear term that shapes the trajectory profile. A shared phase variable is the factor of the nonlinear term and decays as the system progresses. Hence, after the nonlinear fluctuation vanishes, the system is dominated by the linear damping component so the motion stability is guaranteed. It is worth noting, however, that the state variable of DMP is not completely autonomous. The reason is that the evolution of the phase variable is exclusively governed by the time so an implicit time dependency still exists [3].

Beyond representing a single trajectory, many works in the last decade focused on probabilistic dynamics, which provides a natural way of handling the demonstration variations. (187) employed Gaussian Process (GP) and Correlated Component Analysis (CCA) to build mappings between human and robot joint DOFs via a latent variable. In (24), Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) were respectively used to encode the temporal and spatial demonstration correlations. A GMM-centric approach was explored in (25, 27). Much like the DMP works, the temporal information is encoded as a covariate in the state space so the demonstration variabilities are captured by time-dependent covariance matrices. Variations of GMM approaches proposed

3. According to (83), one can of course introduce a state-dependent feedback, although it is nontrivial to assure the stability in this case.

to embed structures to the covariance matrices. Typical examples include (23), where the covariance entries were correlated by assuming the data dimensions are resulted from the views in different frames of reference. Reported as another example in (207), the covariances across the GMM components were tied to assume less model parameters. Both works showed an improved generalization performance. As another popular statistical model, Probabilistic Movement Primitives (ProMPs) modeled a trajectory distribution by estimating the parameter statistics (156). The trajectories are often parameterized by linear function approximators, which allow for an efficient trajectory adaptation. The common choice about the statistics is the mean and variance of a multivariate Gaussian, although multi-mode distributions like GMM can also be used (54). Similar to the sparse GMM works, (40) also researched the dimension reduction of ProMPs parameters.

More recently, this line of research also concerned fitting an autonomous dynamical system. This formulation, be it deterministic or stochastic, is useful when the robot is expected to learn a time-invariant policy. A relevant approach was explored in (157), where the DMP is incorporated with a state feedback for a sensor-based trajectory adaptation. (66) provided a more explicit autonomous DS formulation, based upon a GMM for modeling the demonstrated position and velocity pairs. This research was followed by variations that enforce the desired system properties with various constraints. A notable example is Stable Estimator of Dynamical Systems (SEDS) in (97). SEDS exploited the well-established research on Lyapounov stability and added relevant constraints for training a GMM. A global asymptotic stability was assured for the resulting policy. In spite of these appealing properties, a constrained GMM like SEDS is known as much more difficult to train in comparison of fitting DMP or a time-dependent system (112). Besides the work of SEDS, other structures were also explored. (188) adapted the training of Support Vector Machine (SVM) for modeling a policy with multiple attractors. Local modulation matrices were introduced in (114) to facilitate a GP-based incremental learning.



(a) $p(x|y = \text{“green”})$ or $p(x|y = \text{“purple”})$ (b) $p(x) = \int p(x, y)dy$

Figure 2.2: Discriminative and generative models: (a)a discriminative model learns a conditional distribution with an explicit labeling of y. (b)a generative model estimates the structure and distribution with the labels of y as unknown latent variables.

2.1.2.4 Learning Discriminative Policy vs. Generative Model

Most of the works reviewed above learn a reactive policy that predicts an action on the given arguments (be it a state variable or a time index). This can also be understood as a kind of behavior cloning (161). From the statistical machine learning perspective, this type of imitation estimates a conditional distribution [4] or learns a discriminative model, as such works as supervised learning. This shares some similarities with the policy-based approaches against the value-based ones in the RL literatures. Hence, a natural question is if it is possible to perform LfD via a value-based approach. Relevant ideas for robot motion synthesis have been investigated in the early works (98, 106), where a scalar potential function was used to shape the task dynamics. Formal learning-based methods are named as Inverse Optimal Control (IOC) or Inverse Reinforcement Learning (IRL). The main task of IOC is, similar to the value-based RL, estimating a task-relevant value function (or equivalently a cost-to-go or instantaneous cost/reward), with the given samples assumed to be high-performed ones. Playing a same role in the actor-critic RL, the value function evaluates the preference of the states and provides an implicit guidance in developing the policy or control. As such, this type of implicit LfD works as a

4. Though GMM and the model in (187) learn full probability distributions, they are mainly used for regression and conditional inference in aforementioned works.

generative model and can be considered as unsupervised learning (Figure 2.2). The advantage of learning a value function can be argued from the perspectives of policy robustness and data efficiency. Concretely, a generative model is a complete probabilistic model of the data and one can conduct a robust Bayesian inference via priors. For instance, if a state is evaluated as a low-rewarding or high-cost one, the agent will tend to escape from it to the ones appear more frequently in the expert demonstrations. Without such global information, a discriminative policy will blindly predict an action and not take the optimality of the resulting state into account. When demonstrations are not sufficient for a good data coverage, the errors might be accumulated, possibly leading to a catastrophic behavior (error cascading, see: 12).

The thesis focuses most of its algorithmic contributions on learning cost-to-go functions to develop the robot control and perception mechanisms. Therefore, the topic in this thesis belongs to implicit LfD and generative model. Section 2.2.2 will detailedly review closely related IOC approaches and spell out the difference of the thesis contributions.

2.2 Related Work

This section reviews the literature that is directly related to the research problems and contributions, including learning-based compliance motion synthesis, inverse optimal control methods and the works about representation-learning-powered robot control.

2.2.1 Learning Compliant Robot Motion

The necessity of robot compliance stems from the need to deal with tasks requiring contact accommodation or force exertion. Direct force control or impedance control, as reviewed in the above Section 2.1.1, are viable solutions for these types of tasks. However, devising a proper force/impedance profile is nontrivial, at least not as explicit as the case of position control which relies on intuitive geometry constraints. This fact motivates a learning-based approach to automate the design based on human or robot experience.

A large body of research proposes to derive the feedforward force or impedance parameters via an iterative learning or adaptation. In (231), an iterative learn-

ing method was designed to realize a target impedance. As with other works in iterative learning, a zero impedance error could be theoretically guaranteed and the method was demonstrated to be robust against system and sensory uncertainties. As another example, a bio-inspired adaptation rule was adopted in (60). The performance was principally formalized as a combination of the tracking error and the muscle activation. When the control is assumed to be linear with the activation, the general adaptive control law (192) was obtained in the muscle space. The law was then applied to all relevant robot control terms including the feedforward reference trajectory, force and impedance. As the learning progresses, the accuracy of feedforward motion and force improved gradually. Meanwhile, the impedance, correlating to the muscle activation, decreased hence a human-like modulation emerged. In (237), the same approach was presented in the context of adapting interaction force under perturbations, with an additional convergence proof provided. (67) drew a closer relation to this thesis because the feedforward control and reference force/position were learned from demonstrations. Autonomous dynamical systems were estimated to generate stable trajectories terminating at a target. However, the impedance parameters were not explicitly learned but predefined and subject to the online adaptation, according to a similar rule in (60, 237).

As a more general formulation, optimal control was utilized as a principle to design compliant behaviors (141). The advantage of an optimal-control-based approach is twofold. First, beyond the tracking errors in ILC, more flexible task objectives can be specified, such as maximizing the speed of links with variable stiffness joints (71). Secondly, optimal control can exploit the model structure of different systems. Exploiting the passive dynamics has been demonstrated as crucial to generate highly dynamic, powerful and agile movements (20).

When the model is unknown, reinforcement learning (RL) can be used for searching variable impedance policies. In (213, 22), path-integral RL was used to explore an impedance profile, which was represented as an additional policy output alongside the reference position. The trajectories of each independent DOF were parameterized as DMPs. During the learning iterations, the DMP parameters were randomly perturbed and estimated based on the episodic performance, such as the locomotion distance or the success of jumping over ob-

stacles. (197) adopted the same RL approach in various simulated force-field tasks. The study showed that, a robot arm learned to adapt the feedforward command in face of a predictable external force, while chose to increase the motion impedance when the disturbance was unstable. As a result, interesting human-like modulations were developed from the principle of minimizing errors and control efforts.

LfD-based approaches have also been investigated for learning the robot compliance. Early efforts relied on the mounted sensors to record a direct measurement of the task force. By analyzing the force data, (11) proposed to decompose task dimensions into force and position control components, hence obtaining a hybrid controller from demonstrations. In (10), neural networks were used to represent a nontrivial compliance relation with a nonlinear mapping between the force and position data. Much of the research, on the other hand, advocated to imitate the compliant behavior in the impedance control framework. For example, the parameters of stiffness control were estimated from demonstrations in (190). The implementation difficulty was that the stiffness control parameters are redundant so one could not have a unique estimation by solely looking at the trajectory data. The solution proposed in (190) enforced constraints and heuristics to deal with this issue. The similar challenge was also addressed in (223). Specifically, the authors assumed quasi-static human movements and a constant impedance within a small time interval. The robot adjusted the damping parameters in proportional to the estimated human impedance to ensure a stable physical interaction.

The research on learning impedance parameters from humans was then followed by exploiting the trajectory statistics. This line of work is illustrated in the research by (26) and (109). (109) learned the force profile as a feedforward control in addition to the estimated impedance parameters. The heuristics used to determine the impedance is based on the trajectory variance: the robot should adopt a high stiffness in the directions resembling low variance, and a more compliant movement when the trajectory distribution is more flat. The implicit goal behind the heuristics can be understood as tracking a trajectory under the minimum intervention principle, in which the robot places less importance and efforts on rejecting disturbances when the deviation is not impacting

the task performance. A similar idea was also exploited in (113), where a haptic interface for human users to exert force variabilities and explicitly convey the desired impedance. (227) applied the same principle to both force and position demonstrations and analyzed the variabilities across different reference frames. As a result, the task dimensions and temporal segments were again possible to be decomposed to prioritize impedance and position control. Moreover, as a task-parametrized formalization, the variances represented in the reference frames helped to identify the critical scene marks, hence improved the generalization under a new task configuration. The trajectory variabilities could also be captured as an autonomous GMM-based dynamical system, as was exemplified in (96). The resulting control was a mixture of a set of impedance controllers, whose weights were state dependent.

As reviewed above, there exists an optimality principle behind the impedance shaping based on statistical heuristics. A more formal treatment in optimal control was presented in (138). The authors characterized the trajectory consistency with a quadratic cost function, whose weighting matrix was inversely proportional to the motion variance. Moreover, a risk-aware formulation was introduced, with the disturbance measured as the deviation from a recorded force profile. Such a formulation provides a unified way to deal with the conflict between position tracking and force yielding. Specifically, when the robot is risk-sensitive, it will take a negative attitude to the external force disturbance, hence adopting an increased stiffness to eliminate tracking errors. On the other hand, a risk-seeking behavior will tend to increase the importance of force regulation and yield to the external disturbance. (174) used a task-parameterized GMM (TPGMM) to fit the demonstration trajectories and estimated the impedance parameters separately. In the task reproduction, the impedance for each trajectory segment depended on the similarity to each GMM component. TPGMM was also explored in (29, 175, 176). In these works, a quadratic cost function was parameterized by the regression mean and covariance then an impedance controller was resolved from a finite-horizon LQR. (121) applied a Bayesian estimation to the covariance matrix. Only diagonal positive-definitive matrix was allowed due to the constraint from the prior, so the impedance controllers of joint DOFs were independent. Similar covariances were also learned, however,

in a different manner in (178). The proposed method featured a quadratic cost function and a sub optimal control system, which were termed as a planning movement primitive in the paper. The quadratic cost function and a linear system were fit based on the rollouts scored on an extrinsic signal. Finally, research has been done to facilitate the cost design through inverse optimal control approaches. Relevant work on the robot motion synthesis (87) used a sampling-based to estimate the linear cost parameters. Impedance parameters were explicitly transfered in (78), where apprentice learning was used to estimate, again, a linear parameterized cost function.

The first thesis contribution mainly focuses on learning an impedance controller, through inverse optimal control like (78) and (87). The cost/cost-to-go parameters are nonlinearly correlated. With respect to the trajectory, the cost-to-go function is a similar quadratic form resembling a clear statistical intuition as (138) and (29). Like (121) and (178), a structure about the parameters will be enforced in the cost-to-go function. Much different from learning a diagonal covariance, however, the work in this thesis assumes a representation in the local frame of reference, in the same spirit of (149). This structure raises certain challenges for the standard gradient-based inverse optimal control. The thesis work exploits the problem duality and proposes a sampling-based inference method like (87), while with an adapted trajectory parameterization.

2.2.2 Implicit Learning from Demonstrations

In the early work as (89), Kalman investigated the problem about what quadratic cost function makes a given linear control optimal. In particular, a mono input system was discussed in the paper and a necessary and sufficient condition for such a cost function was established in the frequency domain. The same problem, with a less constrained control penalty matrix, was addressed in (19) through the linear matrix inequality (LMI). Concretely, the optimality and feasibility were respectively captured by an adapted Riccati equation and a constraint from the Lyapounov stability. The resulting formulation searched the cost matrices and the existence of an auxiliary matrix under the LMI constraints, yielding a convex optimization problem.

Focusing on empirical applications, the machine learning community de-

velops similar pieces of research under the motivation of understanding and imitating agent behaviors. Inverse reinforcement learning (IRL) was formally introduced in (153), proposing a condition that the expert performance should be no worse than any alternatives. This is, however, a necessary condition and there might exist rewards, such as a constant function, that fulfill the condition while encodes no interesting information. In light of this, the authors suggested an additional constraint to enlarge the performance gap between the actions following the expert policy and the non-expert ones.

Apprenticeship learning in (1) proposed to use IRL for an agent to perform nearly as good as the given expert policies. Apprenticeship learning exploited the linearity of reward parameters and the integral operation, allowing to match the apprentice policy through a feature expectation. The learning runs as an adversarial game involving two competing modules. On one hand, a reward parameter was searched as to maximize the discrepancy between expert and apprentice policies. On the other hand, the other module tried to shrink the gap by mixing a new optimal policy derived from RL. The game would terminate until the feature expectation error is within a predefined threshold. Successes were demonstrated in modeling and learning a car-driving behavior (1) and maneuvering a robot helicopter (39). Following this method, (204) proposed an extended multiplicative weights apprenticeship learning, which advocated to estimate an $\epsilon-$optimal policy through linear programming (163). In addition, the authors also exploited a dual form to obtain a stationary policy based on the counts of state visitation, while a mixture of policies was returned in the original apprenticeship learning.

In (168), the policy optimization was first replaced by its dual form, which implied a value function and a Bellman inequality. Slack variables were searched to be as small as possible. On the other hand, the variables need to ensure a sufficiently large performance gap between the expert examples and the optimal policies with respect to an augmented loss and the Bellman-flow constraint. For an efficient optimization, the problem was then transformed into a form with a hinge-loss so the subgradient method could be used. Such a formalization amounts to the maximum margin prediction like Support Vector Machines, hence named as maximum margin planning. All the methods reviewed above

include a forward policy optimization subroutine. Actually this is one of the general challenges for IOC/IRL approaches, although sometimes it appears as an equally difficult problem, e.g., evaluating a partition function in the probabilistic models.

Apprenticeship learning was formalized as a Bayesian approach in (167). The regularization of parameters in (153) was then understood as a Laplacian prior. More importantly, the authors formulated the greedy policy as a probabilistic distribution parameterized by a state-action value function. The policy and the reward were updated by sampling and rejecting states based on the distinction to the expert trajectories. This is also different from the standard apprenticeship learning where a set of mixed policies are used. (150) presented a similar approach with a value-parameterized policy. However, unlike the derivative-free algorithm in (167), gradients were derived and evaluated through an empirical estimation of the partition function. In (246, 247, 245), this type of models was principally identified as a class of maximum entropy distributions (MaxEnt). (248) made an extension and proposed a maximum causal entropy model whose actions were only depending on part of the prior observations. As a probabilistic approach, MaxEnt IOC/IRL naturally handles the demonstration noise and intuitively interprets the imitation learning as maximizing the likelihood of expert trajectories.

Much of the research about probabilistic models focused on the computational challenge of partition function evaluation. While most pioneering IOC/IRL literatures were developed on agents with a discrete state-action space, robotic applications often face a high-dimensional continuous space, making a discretization impractical. One way to address this is approximating the integral with a tractable probabilistic density. In (126), the cost function was approximated with a quadratic form along the demonstration trajectories, hence obtaining a Laplacian approximation from the probabilistic point of view. The quadratic approximation was also implicitly used in (78), where iterative LQR (220) was used to search the Laplacian mode. Another type of approximation relies on sampling-based methods. (16) proposed to approximate the original probabilistic model with a proposal one and regulate the difference to the empirical distribution via a relative entropy. In (87), a group of samples centered at the mean

in the trajectory parameter space was taken to approximate the full probability. The partition evaluation was thus solved through path-integral RL with a local optimality. This effectively estimated the partition integral under a Gaussian distribution, instead of the uniform one in the original MaxEnt formalization. As a third way, non-parameteric approaches in Reproducing Kernel Embeddings Hilbert Space (RKHS) could be used to realize a closed-form evaluation (194). An example of such work is (169), which performed a one-shot path integral with an RKHS embedding. In spite of the appealing theoretical properties, kernel methods often suffer from a nearly cubic complexity as the number of data increases. Hence, practical implementations often approximate the kernel operations with linear parameterized random features (165). Lastly, the structure of the planning problem can also be exploited to facilitate the evaluation. The work of (50), for example, decomposed the transfer of a multi-target reaching task into goal and trajectory prediction stages. The predicted goal could help discriminate other unlikely trajectories under an optimality assumption so as to efficiently derive a good mode approximation.

Other research works also explored variations in terms of the function parameterization and MaxEnt principle. Gaussian Process was used in (127) as a non-parametric representation. The partition function evaluation, however, still resorted to a local optimization. (37) proposed another non-parameteric representation, which contained compositional kernels for a feature selection in the cost function learning. Furthermore, a hierarchical IOC was presented in (110). The hierarchy lied in a decomposition of the original task into subtask segments, which were revealed through a GMM model. Then the state was augmented with a variable indicating the GMM membership and the task progress. Such a formalization helps for a sparse or delayed reward signal, such as a $\{0, 1\}$ setting for encoding the success and failure of the task execution.

In (52), the kernel width was also learned in additional to the linear parameters. This resulted a non-convex problem, which was proposed to be addressed by alternating the optimization steps. More importantly, the authors of (52) identified the MaxEnt model as a special case of a broader linearly-solvable system framework. The framework was developed on the basis of the stochastic optimal control of a control-affine dynamical system. Specifically, it showed that

when the dynamics is a continuous one and Gaussian-noised, the model knowledge could be exploited to bias the MaxEnt sampling in the partition function evaluation.

Finally, more recent works reported learning a cost function parameterized by deep neural networks (236, 57). A connection between a Generative Adversarial Network (GAN) and the MaxEnt IOC (or broadly speaking, a Boltzmann energy model) was established in (56). The discriminator network effectively trains the cost parameters in a way like the critic in the actor-critic methods. Correspondingly, the refinement of the generator network proceeds like the actor, which involves searching a good proposal distribution for evaluating the partition function.

All the thesis chapters develop IOC approaches, which differ from the aforementioned ones in that the motivation and adaptation are rooted in learning human-robot applications. In the first part, the MaxEnt model and a sampling-based partition function evaluation like (87) are used. However, unlike the popular linear parameterization, the thesis explores a structured cost-to-go for realizing an intuitive and human-like force/motion control decomposition. The second part of the thesis takes a unique way to deal with the partition function in that it intentionally to adopt a simple quadratic cost form to make the linear-solvable framework tractable. The loss of the expressiveness is compensated with an aggregation of these "weak" models, as such introducing the ensemble principle into the IOC approaches. This part also incorporates human kinematics features and interpretable parameters for motion synthesis and adaptation. The final contribution is relevant to the latest deep-learning-powered IOC works. However, the thesis method is developed based on variational auto-encoders, enjoying a straightforward probabilistic interpretation and stable training in comparison with the GAN-style methods. Also, this part of work motivates a factored distribution in light of learning from redundant and unstructured demonstrations, which was less explored in other LfD literatures.

2.2.3 Representation Learning in Sensorimotor Control

Representation learning differs from general machine learning techniques in that it promises an easier way to handle unstructured patterns, which often

desire a laborious feature engineering and domain knowledge. This appears compelling for the robotic sensorimotor control since such kind of data presents in many sensor modalities.

The main instantiation of representation learning is often formed as connectionist models such as neural networks (NN). One of the pioneering works about modeling an NN controller is (161), in which a real-time video stream was fed in an autonomous vehicle task to keep the car on the track. However, the early efforts of implementing an NN controller were often limited to small-scale models with a careful design (82). This is due to the fact that learning large-scale neural networks often requires a great amount of data and computational power which had not been available by then.

In recent, witnessing the encouraging success in pattern recognition and generation (111, 200), roboticists have regained the enthusiasm towards this type of controller. A successful story was reported in (123), where the authors considered a robot cutting task and a model predictive control (MPC) approach based on the latent feature learning. The necessity of inducing the latent features was argued, that the MPC should account for dynamics variations due to different materials and cutting stages. Importantly, the success of this framework was ascribed to carefully-tailored feature structures and recurrent latent units for capturing a long-term time dependency. Offline unsupervised learning was performed for a good initialization of the latent features. An analogous pretraining and domain regularization design were presented in (124), where the graspable regions of an object were identified from an RGB-D image.

In a large body of works, NNs powered by convolutional operations (CNN) (59, 120) were used to reason about the raw image inputs and develop visuomotor controllers. As a seminal work in this line of research, (128) presented a practical system in which the robot executed various contact-involved manipulation tasks with pixel-based visual feedbacks. Stacked convolutional layers were used as detectors to filter out a representation corresponding to the 2D coordinate of the interested operating point. The representation was then concatenated with the robot configuration as the neural network input to predict the desired torques. Much like the MPC work (123), pre-training data was collected. Concretely, the object poses were labeled so it avoided optimizing

trajectories in the pixel space. The collected vision-pose pairs were used to extract the interested representation, which then replaced real poses in predicting the torque trajectory. In general, an offline prior training is important to the success of a neural controller or policy. The extraction of informative representations often requires large amount of data as is demonstrated in other deep learning works. However, acquiring data through real physical explorations tends to be expensive and risky. Besides the prior data collection, the learning stability is another concern. (128) utilized guided policy search which turned to fit a supervised learning model on trajectories from a model-based optimal control, as such alleviating the difficulty of tunning high-dimension policy parameters based on delayed signals. Also, algorithms which update the policy with certain guarantees, such as trust region and natural gradients, perform better in benchmarking tasks (51) and expect to have an improved data efficiency. (244) attempted to realize a cheap data acquisition via exploring in a simulation environment. However, negative results were reported by the authors that the trained controller, even though generalized well in the simulated scenarios, failed with a zero success rate under the real-world camera input.

A recent trend in robot learning focuses on approaches that address the challenges from the data starving problem. The first solution lies in a distributed architecture with multiple homogeneous robots to parallelize the data acquisition. (129) realized such a system with about 10 robots to learn picking and grasping objects based on mono-camera inputs. The images and a kinematic motor command were combined to predict if a successful grasp could be achieved. The cross-entropy optimization was performed to determine the action to take when a test image was presented. It took about two months to collect 80, 000 trials before the emergence of a controller with a satisfying success rate. In (69), the A3C RL (146), which allows for an asynchronous policy update for multi-agents, was utilized in a group of two robot manipulators. The authors showed a boosted learning efficiency by sharing the experience between the robots, which learned to open a door in around 2.5 hours. The second avenue taken by researchers is associating the simulation data to the real world. Exploring such an idea, (72) proposed to improve the fidelity of a simulator by adjusting its parameters to fit the collected rollouts on a real humanoid robot.

The real-world data could be task irrelevant and the interested task policy was optimized in the adjusted simulation environment. With a sufficient training in the simulator, a humanoid robot achieved a faster walking velocity in comparison with an off-shelf strategy. In another work (180), the authors first trained a robot skill with an encoder representation in the simulator. The encoded information was then used to bias the training of another model on the real robot. Exploring a more principled way, other research eyes on how the new task learning can be facilitated by reusing the prior task knowledge. A two-stage approach was proposed in (6). In the first step, a mapping that bridges different task spaces was established with an unsupervised manifold alignment algorithm. The mapping was then used in the second step to initialize the policy searching for the target task. In (70), a common feature space between tasks was learned to facilitate the target task learning. Importantly, the reuse of the task knowledge was implemented across multiple robots, which potentially differed in their embodiments. The invariant feature space was learned through a proxy task which should be mastered by both source and target agents. It is worth noting that, this third avenue belongs to transfer learning, which is currently an active research topic and mostly focuses on patterns like images in the general machine learning. As a last type of paradigm, (205) explored a layer design for a better generalization performance, thus less demanding about the data volume. Specifically, the authors noticed the equivalence between the value iteration in RL and the convolution operation in CNN, and proposed to stack convolutional layers to embed an implicit planning computation. Empirical results showed that a policy with the induced structure generalized well in a collision-free path planning task.

Instead of directly modeling a neural controller, other researchers embrace networks models as a good complement to the probabilistic modeling. The argument is that one can exploit the expressiveness and differentiable structure of NNs to represent complex yet tractable statistical moments (143). This implies the possibility of building a full probability model and conducting various inference tasks, linking to a broader topic of probabilistic programming (118, 222). Relevant works in a robotics scenario includes (33) and (233). In (33), auto-encoders and DMPs were combined to obtain a compact and structured latent

space for whole-body joint movements. (233) proposed to learn a latent representation from high-dimension unstructured observations (e.g., image pixels). A policy in the extracted low-dimension space was efficiently searched for balancing an inverted pendulum with the pixel feedback. Much like (233), a similar method with a more rigorous derivation was proposed and validated in the pixel inverted pendulum task (92).

As a nonparameteric option, kernel machines were also applied in robotics. As was pointed in (165), a certain type of kernel representation (e.g., a radial basis kernel) could be cast as an equivalence to a random projection feature, which somehow justified the effectiveness of the so-called extreme learning machine (ELM) (80). Such models in effect suggest a linear parameterized neural network, with randomly chosen intermediate layers and only the output layer tuned. This is beneficial in some situations. One can easily, for instance, achieve a stable online learning by exploiting the kernel representation in a ridge regression. Employing such a method, an incremental robot dynamics learning was reported in (63). Moreover, linear parameters are generally favored for the system analysis and synthesis. As an example, (122) utilized an ELM to learn a vector field to model handwriting motion with a locally ensured Lyapounov stability.

The thesis extends the inverse optimal control framework in the third part, in alignment with the progress of NN-based frameworks. The proposed adaptation is similar to (33) and (233), which emphasize the importance of learning a latent space rather than the direct policy. Meanwhile, the relevant chapter also shares some similarities with (6) and (70) in terms of learning an overlapped manifold or space, although here the motivation is not mapping between tasks but sensor modalities. Also, unlike (6), the representation is simultaneously extracted alongside the task learning. The thesis also reports a practical end-to-end system, while most of the preceding works were showcased in simulators with virtual visual inputs. To achieve this, the data-starving problem is also alleviated in a different way. The image data in this work is obtained from the synthesis of the other modality. The quality of synthetic data is ensured by the other contribution (240), which incorporates kinematic features for a human-like variability. The thesis demonstrates the generalization to new tasks as well.

This is achieved with the sampling-based trajectory optimization proposed in (238). The method in effect solves a variant of the cross-entropy optimization, whose standard form was also employed in (129) to infer the motion from a neural model.

2.3 Technical Preliminaries

This section gives a brief overview about the main technical foundations. The terminology and notations will also be established as the section expands. Section 2.3.1 reviews the topic of impedance control with its most basic formulation. The core method about the forward and inverse optimal control is introduced in 2.3.2, targeting the particular type of linearly-solvable dynamical system used in the thesis. Gaussian Mixture Models and its task-parameterized variant are reviewed in 2.3.3. Including these materials will provide background information about the algorithm connection and experiment implementation in Chapter 4. The adopted representation learning technique, variational auto-encoders, is introduced in 2.3.4 with an aim to support Chapter 5.

2.3.1 Impedance Control

Impedance control concerns steering a dynamical system (DS) with respect to a desired dynamic relation between the physical effort and effect. Taking the example from robotics, impedance control is usually used to regulate the force effort and motion effect. Let $\boldsymbol{u} \in \mathbb{R}^d$ denote a d-dimension force input to the robot system and $\boldsymbol{x} \in \mathbb{R}^d$ represent a d-dimension robot state, such as the coordinate in the Cartesian space or joint space. The goal is to control the robot so as to follow a dynamics like

$$\boldsymbol{H}\frac{d^2\boldsymbol{x}}{dt^2} + \boldsymbol{D}\frac{d\boldsymbol{x}}{dt} + \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}^r) = \boldsymbol{u} \tag{2.1}$$

where $\boldsymbol{H}$, $\boldsymbol{D}$, $\boldsymbol{K} \in \mathbb{R}^{d\times d}$ are desired inertia, damping and stiffness matrices and $\boldsymbol{x}^r$ denotes the reference state. The dynamic relation is fully governed by the matrices and the reference state $\boldsymbol{x}^r$, which are not necessarily constant. Hence, the central task of an impedance control is to choose proper $\boldsymbol{H}$, $\boldsymbol{D}$, $\boldsymbol{K}$ matrices to indirectly accommodate the exerted force $\boldsymbol{u}$. In practice, the inertia term is often ignored due to the difficulty of obtaining an accurate acceleration

estimation. The remained two terms in effect emulate the behavior of a damped spring, which generates proportional forces according to the displacement and velocity of the endpoint. The stability can be assured for certain matrices when the reference is fixed (regulation), although tracking a slowly-varying reference also works fine in practice. The thesis implements an impedance control by learning both the stiffness matrix $\boldsymbol{K}$ and reference $\boldsymbol{x}^r$. The damping matrix is determined according to $\boldsymbol{K}$ and the critical damping ratio. A control rule can be derived from the relation in 2.1 with an additional term to compensate the gravity:

$$\boldsymbol{u^c = G(x) - K(x - x^r) - D\frac{dx}{dt}} \tag{2.2}$$

with $G(x)$ denoting the state dependent gravity from the manipulator mass. One can apply the rule to a standard robot dynamics model subject to an external wrench u^e:

$$\boldsymbol{M}(\boldsymbol{x})\frac{d^2\boldsymbol{x}}{dt^2} + \boldsymbol{C}(\boldsymbol{x}, \frac{d\boldsymbol{x}}{dt})\frac{d\boldsymbol{x}}{dt} + \boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{u}^c + \boldsymbol{u}^e \tag{2.3}$$

where M and C denote the robot inertial and Coriolis terms. As a result, a second-order dynamics will be obtained as

$$\boldsymbol{M}(\boldsymbol{x})\frac{d^2\boldsymbol{x}}{dt^2} + [\boldsymbol{C}(\boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}) + \boldsymbol{D}]\frac{d\boldsymbol{x}}{dt} + \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x}^r) = \boldsymbol{u}^e \tag{2.4}$$

Therefore, when the system is stable, the free-space robot motion (with $\boldsymbol{u}^e = 0$) could converge to the reference $\boldsymbol{x}^r$. When an external wrench exists, the reference $\boldsymbol{x}^r$ becomes a virtual target, which can be modulated together with $\boldsymbol{K}$ to accommodate the contact. Note that in impedance control, the system input $\boldsymbol{u}^c$ is wrapped by the spring-damping law, hence the high-level algorithm interfaces with the system by specifying the virtual target and the desired compliance.

2.3.2 Optimal Control and Inference for Linearly-Solvable Dynamical System

Nonlinear Dynamical System (DS) is an important tool for modeling robot dynamics. This section will briefly review a special type of nonlinear DS used in the thesis. Among different forms, the thesis specifically considers the discrete-time nonlinear DS with a continuous state and system input as

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{2.5}$$

which subscripts the system state and input with the time index t. The thesis considers a control-affine variant of this general form, by separating the transformation into two parts according to their dependencies on $\boldsymbol{u}$:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + \boldsymbol{B}\boldsymbol{u}_t \tag{2.6}$$

The independent nonlinear transformation $f : \mathbb{R}^d \to \mathbb{R}^d$ is called passive dynamics as it captures how the dynamics proceeds in absence of the control input. The input $\boldsymbol{u}$ linearly applies to the system with a gain matrix $\boldsymbol{B} \in \mathbb{R}^{d\times d}$. It is worth noting that a more general formalization allows a state dependent gain matrix. The thesis takes the constant $\boldsymbol{B}$ as a simplification, though the main conclusions still hold for the general form. Also, the $\boldsymbol{B}$ could be non-square for an under-actuated system. However, $\boldsymbol{u}$ is assumed to be of the same dimension as the state $\boldsymbol{x}$ hence a fully controlled system is considered. This control-affine formulation is sufficient to describe many practical systems, such as the robot dynamics model in Equation (2.3).

The dynamics can be steered by a scalar function assigning scores to the state and input at each time step. In particular, one can accumulate the instantaneous scores to evaluate a rollout as:

$$\mathcal{J}_\varsigma(\boldsymbol{x}_0, t_0) = \sum_{t=t_0}^{T} C(\boldsymbol{x}_t, t) + \frac{1}{2}\boldsymbol{u}_t^T \boldsymbol{R}\boldsymbol{u}_t \tag{2.7}$$

where the rollout is denoted as a sequence of the state and input $\varsigma = \{\boldsymbol{x}_t, \boldsymbol{u}_t\}_{t=0:T}$. $C(\cdot)$ denotes a state dependent cost for each time step and the additional term on $\boldsymbol{u}_t$ penalizes a large input magnitude through $\boldsymbol{R} \in \mathbb{R}^{d\times d}$. The $\mathcal{J}_\varsigma$ is termed as the cost-to-go function, as it summarizes an accumulated value for a rollout starting at $\boldsymbol{x}_0$ and following ς [5]. A control or policy $\boldsymbol{u} = \{\boldsymbol{u}_t\}_{t=0:T-1}$ or its resulting rollout is regarded as optimal if:

$$\varsigma_{\boldsymbol{u}}^* = \underset{\boldsymbol{u}}{\operatorname{argmin}}\, \mathcal{J}_{\varsigma_{\boldsymbol{u}}} \tag{2.8}$$

Therefore seeking an optimal control aims to minimize the cost-to-go along the state trajectory. It is known that, for a control-affine system and a cost-to-go function like Equation (2.6) and (2.7), the optimal control can be derived as:

$$\boldsymbol{u}_t^* = -\boldsymbol{R}^{-1}\boldsymbol{B}\frac{\partial \mathcal{J}_{\varsigma^*}(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{x}_{t+1}} \tag{2.9}$$

5. The time horizon can also be indefinite for the general first-exit problem. Most parts of the thesis consider a finite horizon case.

Note the control here is not explicit due to its dependency on the future state. Nonetheless, it can be efficiently solved through a backward sweeping or equivalently, solving a linear differential equation in the continuous time setting, hence named as a linearly-solvable dynamical system (216). Moreover, the solution could be efficient and less implicit under a linear-quadratic assumption, resulting in an LQR problem as its special case:

$$\begin{aligned} f(\boldsymbol{x}_t, t) &= \boldsymbol{A}_t \boldsymbol{x}_t \\ C(\boldsymbol{x}_t, t) &= \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{r}_t)^T \boldsymbol{Q}_t (\boldsymbol{x}_t - \boldsymbol{r}_t) \\ \mathcal{J}_\varsigma(\boldsymbol{x}_t) &= \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Lambda}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t) \end{aligned} \tag{2.10}$$

where $\boldsymbol{A}_t$ denotes a linear state transformation. $C(\cdot)$ takes a quadratic form with $\boldsymbol{r}_t$ as the reference state and $\boldsymbol{Q}_t$ is a positive-definite (PD) weight matrix. Thus the sum-up of these instantaneous costs will yield another quadratic cost-to-go $\mathcal{J}_\varsigma$, with the remaining constant term ignored. $\boldsymbol{\Lambda}_{t+1}$ is the corresponding PD matrix which can be computed from the Riccati equation [6]. $\boldsymbol{\mu}_t$ denotes the reference state which takes the feed-forward reference trajectory $\{\boldsymbol{r}_t\}$ into account. In this case, an optimal controller depending on the current state is explicitly given by:

$$\boldsymbol{u}_t^* = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{A}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t). \tag{2.11}$$

It is worth pointing out that, this controller is much like the impedance control in (2.2) if the velocity of the regulation point is augmented into the system state.

An inverse optimal control problem can be cast as inferring the unknown parameters in $\mathcal{J}(\cdot)$ or $C(\cdot)$ given a set of optimal rollouts $\{\varsigma_i\}_{i=1:N}$ as the demonstrations. Taking the quadratic case as an example, the candidate cost can be parameterized as the ones in Equation 2.11, with an unknown parameter $\boldsymbol{\theta} = \{\boldsymbol{r}_t, \boldsymbol{Q}_t\}$ or $\boldsymbol{\theta} = \{\boldsymbol{\mu}_t, \boldsymbol{\Lambda}_t\}$. In this thesis, the inverse problem is solved under a stochastic formalization, with the consideration of handling noisy demonstrations. Also, the thesis focuses on learning the cost-to-go function for incorporating constraints about a trajectory. In a stochastic form, the control-affine dynamics in Equation (2.6) is first adapted by adding a white noise:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + \boldsymbol{B}\boldsymbol{u}_t + d\boldsymbol{W}_t \tag{2.12}$$

6. Namely, by recursively evaluating $\boldsymbol{\Lambda}_t = \boldsymbol{Q}_t + \boldsymbol{A}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{A}_t - \boldsymbol{A}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{B}_t (\boldsymbol{B}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{B}_t + \boldsymbol{R}_t)^{-1} \boldsymbol{B}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{A}_t$ for a finite-horizon problem, with $\boldsymbol{\Lambda}_T = \boldsymbol{Q}_T$.

where $d\boldsymbol{W}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma}_0)$ and the covariance $\boldsymbol{\Sigma}_0$ is inversely proportional to $\boldsymbol{R}$ as $\boldsymbol{\Sigma}_0 = \boldsymbol{B}^T \boldsymbol{R}^{-1} \boldsymbol{B}$. Prior research (52) has shown that the stochastic optimal control of Equation (2.12) can be derived from the follow probabilistic model over the system rollouts:

$$p(\varsigma|\boldsymbol{x}_0, \boldsymbol{\theta}) = \frac{p_0(\varsigma|\boldsymbol{x}_0) \exp[-\sum_{t=0}^{T} C(\boldsymbol{x}_t, \boldsymbol{\theta})]}{\int_{\varsigma'|\boldsymbol{x}_0} p_0(\varsigma'|\boldsymbol{x}_0) \exp[-\sum_{t=0}^{T} C(\boldsymbol{x}'_t, \boldsymbol{\theta})] d\varsigma'} \tag{2.13}$$

or in a factorized form:

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{\theta}) = \frac{p_0(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) \exp[-\mathcal{J}_\varsigma(\boldsymbol{x}_{t+1}, \boldsymbol{\theta})]}{\int_{\boldsymbol{x}'_{t+1}} p_0(\boldsymbol{x}'_{t+1}|\boldsymbol{x}_t) \exp[-\mathcal{J}_\varsigma(\boldsymbol{x}'_{t+1}, \boldsymbol{\theta})] d\boldsymbol{x}'_{t+1}} \tag{2.14}$$

The factorization exploits the Bellman equation and softening the maximization with a log-exp-sum operator:

$$\mathcal{J}(\boldsymbol{x}_t) = C(\boldsymbol{x}_t) + \log \int p_0(\boldsymbol{x}'_{t+1}|\boldsymbol{x}_t) \exp[-\mathcal{J}(\boldsymbol{x}'_{t+1})] d\boldsymbol{x}'_{t+1} \tag{2.15}$$

Here p_0 denotes the stochastic passive dynamics with $f(\cdot)$ as the deterministic part. When the passive propagation is uniformly distributed, which assumes no control penalty, one can obtain a Boltzmann distribution over the state trajectories:

$$p(\varsigma|\boldsymbol{\theta}) = \frac{\exp[-\sum_{t=0}^{T} C(\boldsymbol{x}_t, \boldsymbol{\theta})]}{\int_{\varsigma'} \exp[-\sum_{t=0}^{T} C(\boldsymbol{x}'_t, \boldsymbol{\theta})] d\varsigma'} \tag{2.16}$$

As is revealed in (52), this form is in accordance with the maximum-entropy (MaxEnt) IRL (246). The trajectory cost can be interpreted as the statistic moments, which generate the optimal trajectories with a high probability. Therefore, the inverse optimal control can be solved as inferring the distribution parameters through maximizing the demonstration likelihood, hence transforming the original formulation into an unsupervised statistical learning problem.

2.3.3 Gaussian Mixture Models and Task Parameterization

A Gaussian Mixture Model (GMM) represents a probability density over the interested data, be it a single state or an entire trajectory. The GMM combines multiple Gaussians to describe complex data distributions that a single Gaussian

fails to model. This is achieved by introducing a latent variable z, which is constrained to be categorical as $z = 1, ..., K$ for a tractable posterior evaluation. A marginal data distribution with z integrated out can be written as:

$$p(\boldsymbol{x}) = \int_z p(\boldsymbol{x}|z)p(z) = \sum_{k=1}^{K} w_{z=k}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.17}$$

where $w_{z=k}$ parameterizes a multinomial prior distribution $p(z)$ and indicates the probability of generating $\boldsymbol{x}$ from the k-th Gaussian component. The mean and covariance of the component are denoted as $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. Therefore, fitting a GMM estimates the parameters $\boldsymbol{\theta} = \{w_{z=k}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1,...,K}$. This is often realized through maximizing a lower bound of the log-likelihood:

$$\begin{aligned} \log p(\boldsymbol{x}) &= \log[\sum_k p(z=k)p(\boldsymbol{x}|z=k)] \\ &\geq \sum_k p(z=k)\log p(\boldsymbol{x}|z=k) = \mathbb{E}_{p(z)}[\log \frac{p(\boldsymbol{x},z)}{p(z)}] \end{aligned} \tag{2.18}$$

which uses Jensen's inequality to swap the logarithm and expectation operations and separate Gaussian statistic moments. The expectation is not directly evaluable since $p(z)$ is unknown. One can, however, take the expectation with respect to another distribution $q(z)$, which is assumed to be in the same family as $p(z)$. The Equation (2.18) is re-written as:

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(z)}[\log \frac{p(\boldsymbol{x},z)}{q(z)}] = \mathbb{E}_{q(z)}[\log \frac{p(\boldsymbol{x})p(z|\boldsymbol{x})}{q(z)}] = \log p(\boldsymbol{x}) - \mathrm{KL}[q(z)\|p(z|\boldsymbol{x})] \tag{2.19}$$

Hence, a proper choice of $q(z)$ should minimize the gap between the actual likelihood and the lower bound, implying a minimized Kullback-Leibler (KL) divergence $\mathrm{KL}[q(z)\|p(z|\boldsymbol{x})] = 0$. Applying the Bayes rule to the posterior $p(z|\boldsymbol{x})$, the optimal $q(z)$ can be explicitly written as:

$$q(z=k) = p(z=k|\boldsymbol{x}) = \frac{p(z=k)p(\boldsymbol{x}|z=k)}{\sum_{k'} p(z=k')p(\boldsymbol{x}|z=k')} = \frac{w_{z=k}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} w_{z=k'}\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}'_k, \boldsymbol{\Sigma}'_k)} \tag{2.20}$$

Training a GMM model thus takes iterative steps of

- evaluating the expectation of the full probability $\mathbb{E}_{q(z)}[\log p(\boldsymbol{x}, z)]$ with the estimation of $\boldsymbol{\theta}_t$ in the last step.
- maximizing $\mathbb{E}_{q(z)}[\log p(\boldsymbol{x}, z)]$ to obtain a new $\boldsymbol{\theta}_{t+1}$ given that the Gaussian component likelihoods are factored and $q(z)$ is independent of $\boldsymbol{\theta}_{t+1}$.

This can be summarized as the Expectation-Maximization (EM) algorithm (49).

Equation (2.20) means that, with a trained model, one can infer the latent variable via the Bayes rule. Similarly, the missing dimensions could also be part of $\boldsymbol{x} = [\boldsymbol{x}^o, \boldsymbol{x}^u]^T$, and inferred upon the observable dimensions $\boldsymbol{x}^o$:

$$\begin{aligned} p(\boldsymbol{x}^u|\boldsymbol{x}^o) &= \sum_k p(z=k|\boldsymbol{x}^o)\frac{P(\boldsymbol{x}^o,\boldsymbol{x}^u)}{p(\boldsymbol{x}^o)} \\ &= \sum_k \frac{w_{z=k}\mathcal{N}(\boldsymbol{x}^o|\boldsymbol{\mu}_k^o,\boldsymbol{\Sigma}_k^o)}{\sum_{k'} w_{z=k'}\mathcal{N}(\boldsymbol{x}^o|\boldsymbol{\mu}_{k'}^o,\boldsymbol{\Sigma}_{k'}^o)}\mathcal{N}(\boldsymbol{x}^u|\boldsymbol{\mu}_k^{u|o},\boldsymbol{\Sigma}_k^{u|o}) \end{aligned} \tag{2.21}$$

where $\boldsymbol{\mu}_k^o$ and $\boldsymbol{\Sigma}_k^o$ denote the Gaussian mean and covariance of the observable dimensions. $\boldsymbol{\mu}_k^{u|o}$ and $\boldsymbol{\Sigma}_k^{u|o}$ are the conditional Gaussian means and covariances:

$$\begin{aligned} &\boldsymbol{\mu}_k = \begin{bmatrix}\boldsymbol{\mu}_k^o \\ \boldsymbol{\mu}_k^u\end{bmatrix} \quad \boldsymbol{\Sigma}_k = \begin{bmatrix}\boldsymbol{\Sigma}_k^o, \boldsymbol{\Sigma}_k^{ou} \\ \boldsymbol{\Sigma}_k^{uo}, \boldsymbol{\Sigma}_k^u\end{bmatrix} \\ &\boldsymbol{\mu}_k^{u|o} = \boldsymbol{\mu}_k^u + \boldsymbol{\Sigma}_k^{uo}(\boldsymbol{\Sigma}_k^o)^{-1}(\boldsymbol{x}^o - \boldsymbol{\mu}_k^o) \\ &\boldsymbol{\Sigma}_k^{u|o} = \boldsymbol{\Sigma}_k^u - \boldsymbol{\Sigma}_k^{uo}(\boldsymbol{\Sigma}_k^o)^{-1}\boldsymbol{\Sigma}_k^{ou} \end{aligned} \tag{2.22}$$

As a result, the mean of Equation (2.21) could be determined by taking a weighted combination of multiple Gaussian linear regressions, hence obtaining the name of Gaussian Mixture Regression (GMR).

The GMR is often used to predict the control or the desired state from the observation, after a GMM has been fit over the demonstrated state pairs $[\boldsymbol{x}^o, \boldsymbol{x}^u]$. An extension called task-parameterized GMM (TPGMM) allows the prediction to take account of the importance of each state dimensionality (23), and improves the generalization performance under new task configurations. Concretely, the GMM is trained on the data with an augmented state, which involves descriptions relative to the configuration parameters. For instance, the robot pose can be measured from the perspective of M landmarks, such as:

$$\bar{\boldsymbol{x}} = \begin{bmatrix}\boldsymbol{x}^1 \\ ... \\ \boldsymbol{x}^M\end{bmatrix} = \begin{bmatrix}\boldsymbol{T}_w^{1\,T}\boldsymbol{x}^w - \boldsymbol{b}^1 \\ ... \\ \boldsymbol{T}_w^{M\,T}\boldsymbol{x}^w - \boldsymbol{b}^M\end{bmatrix} \tag{2.23}$$

where $[\boldsymbol{T}^m, \boldsymbol{b}^m]$ (orientation and offset) indicate the m-th landmark pose expressed in an inertial reference frame. Although the resulting $\bar{\boldsymbol{x}}$ is an augmented high-dimension variable, its mean and covariance possess structures since the descriptions are redundant. Specifically, the TPGMM assumes the demonstration variations are independent with respect to landmarks so the Gaussian covari-

ances are block diagonal:

$$\boldsymbol{\Sigma}_k = \text{diag}(\boldsymbol{\Sigma}_k^1, ..., \boldsymbol{\Sigma}_k^M) \tag{2.24}$$

This could further factorize the Gaussian components in GMM/GMR when the pose in the inertial reference frame is of the interest, such as:

$$\begin{aligned} &\mathcal{N}(\boldsymbol{x}^w|\boldsymbol{\mu}_k^w, \boldsymbol{\Sigma}_k^w) = \prod_{m=1}^{M} \mathcal{N}(\boldsymbol{x}^w|\boldsymbol{\mu}_k^{wm}, \boldsymbol{\Sigma}_k^{wm}) \\ &\boldsymbol{\mu}_k^{wm} = \boldsymbol{T}_w^m \boldsymbol{\mu}_k^m + \boldsymbol{b}^m \\ &\boldsymbol{\Sigma}_k^{wm} = \boldsymbol{T}_w^m \boldsymbol{\Sigma}_k^m (\boldsymbol{T}_w^m)^T \end{aligned} \tag{2.25}$$

Therefore the estimation of $\boldsymbol{x}^w$ becomes a fusion of estimations from different reference perspectives, yielding another Gaussian:

$$\boldsymbol{\Sigma}_k^w = [\sum_{m=1}^{M} (\boldsymbol{\Sigma}_k^{wm})^{-1}]^{-1} \quad \boldsymbol{\mu}_k^w = \boldsymbol{\Sigma}_k^w \sum_{m=1}^{M} [(\boldsymbol{\Sigma}_k^{wm})^{-1} \boldsymbol{\mu}_k^{wm}] \tag{2.26}$$

The above equation implies the means associated with smaller covariances will be more significant in the final linear combination. This is in accordance with the intuition of assigning more importance to predictions with less uncertainties.

The thesis derives an inverse optimal control formalization that draws a connection to the popular GMM model. TPGMM will be used as a way to handle task configurations in a robot experiment.

2.3.4 Deep Generative Model: Variational Auto-encoders

Generative models like GMM provide powerful tools to represent various data distributions. However, high-dimension unstructured data such as images or audios are often notoriously hard to learn with a GMM. In particular, GMM is inherently a local linear model and its covariance matrices will be extremely large as the data dimension increases. To address this shortcoming, recent generative models incorporate representation learning to enrich the model capacity. Typical examples include Variational Auto-encoders (VAE) (100) and Generative Adversarial Networks (GAN) (64). This section focuses on the background knowledge about VAE, which is utilized and adapted in the thesis for its clear probabilistic interpretation and training stability .

In a similar spirit of GMM, the VAE models capture complex data with a

continuous latent variable $\boldsymbol{z} \in \mathbb{R}^{d_z}$:

$$p(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_0(\boldsymbol{z}) d\boldsymbol{z} \tag{2.27}$$

where p_0 denotes the prior. $p_{\boldsymbol{\theta}}$ is analogous to the Gaussian distribution in the GMM while its parameters are determined with a continuous mapping instead of a categorical one. Facing the same difficulty of evaluating the likelihood, a variational lower bound is derived. The VAE model proposes to use a $\boldsymbol{\phi}$-parametrized proposal distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ to approximate the real posterior $p(\boldsymbol{z}|\boldsymbol{x})$. The approximation is again regulated through a KL divergence:

$$\begin{aligned} \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}|\boldsymbol{x})] =& \mathbb{E}_{q_\phi}[\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) - \log p(\boldsymbol{z}|\boldsymbol{x})] \\ =& \mathbb{E}_{q_\phi}[\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) - \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) - \log p_0(\boldsymbol{z}) + \log p(\boldsymbol{x})] \end{aligned} \tag{2.28}$$

Applying Bayes rule and noticing that total probability $p(\boldsymbol{x})$ is independent of $\boldsymbol{z}$, the above equation can be rearranged as:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}) =& \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}|\boldsymbol{x})] - \log p(\boldsymbol{x}) \\ =& \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \| p_0(\boldsymbol{z})] - \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] \end{aligned} \tag{2.29}$$

Because of the non-negativity of KL-divergence, the right hand side can be viewed as an upper bound of the negative logarithm of (2.27). Hence $\mathcal{L}$ can be used as a valid surrogate to optimize the original data likelihood when (2.28) is small. Usually, $q_{\boldsymbol{\phi}}$ and $p_{\boldsymbol{\theta}}$ are parameterized by nonlinear mappings like deep neural networks, hence named as recognition and generation networks. Parameterizing nonlinear mappings allows for a rich representation and an improved modeling power. This, however, trades-off the necessity of evaluating the expectation term via sampling-based method, which might suffer from the high variance of gradient evaluation. Specifically, unlike the categorical posterior in the GMM, here the optimal $q_{\boldsymbol{\phi}}$ is not readily available. In that sense, one has to also evaluate the gradient of the expectation with respect to the recognition network parameter $\boldsymbol{\phi}$. If the gradient is evaluated in a standard way like REINFORCE:

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})] \tag{2.30}$$

The estimation might be quite poor when $q_{\boldsymbol{\phi}}$ is far from the real $p(\boldsymbol{z}|\boldsymbol{x})$ and the quality itself depends on the parameter $\boldsymbol{\phi}$.

(100) adopted a reparameterization trick to alleviate this issue. The trick is to rewrite the stochastic $\boldsymbol{z}$ as a combination of a deterministic part and a random variable whose distribution does not depend on $\boldsymbol{\phi}$:

$$\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})\epsilon \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{2.31}$$

As such, the $\boldsymbol{\phi}$-parameterized recognition network actually outputs the statistic moments of a Gaussian latent encoding. The prior p_0 is often chosen as an isotropic Gaussian to obtain a closed-form KL evaluation. Similarly, the generation network p_θ can be constructed as a Gaussian whose mean is determined by $\boldsymbol{z}$ in a nonlinear way [7]:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \propto \exp(-\frac{\|\boldsymbol{x} - g_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2}{2}) \tag{2.32}$$

The logarithm in the expectation thus leads to a squared reconstruction loss.

Stochastic gradient descent with an adaptive moment estimation (ADAM) (101) was proposed to adjust the learning rate in training the network parameters. ADAM keeps a decayed average of the history gradient $\boldsymbol{m}_t$ and its square $\boldsymbol{v}_t$:

$$\begin{aligned} \boldsymbol{m}_t &= \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\nabla_t \\ \boldsymbol{v}_t &= \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)(\nabla_t)^2 \end{aligned} \tag{2.33}$$

and the gradient is eventually estimated as:

$$\hat{\nabla}_t = \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}_t} + \lambda}\hat{\boldsymbol{m}}_t \qquad \hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t} \qquad \hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t} \tag{2.34}$$

The network parameters are thus optimized with a quasi-second-order update. The hyper parameters are suggested as fixed values by the authors of ADAM: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\lambda = 10^{-8}$ and $\eta = 10^{-4}$. These default settings are thoroughly used in the thesis and lead to a good empirical performance.

2.4 About CoWriter

Part of the thesis work situates in the background project of CoWriter, which aims to build a robotic companion that helps children to acquire the handwriting skill. Unlike an instructor who provides a direct guidance, the robot is assigned with the role of a learner, exhibiting writing difficulties and

7. Similarly, a Bernoulli density can be used for binary data like image pixels.

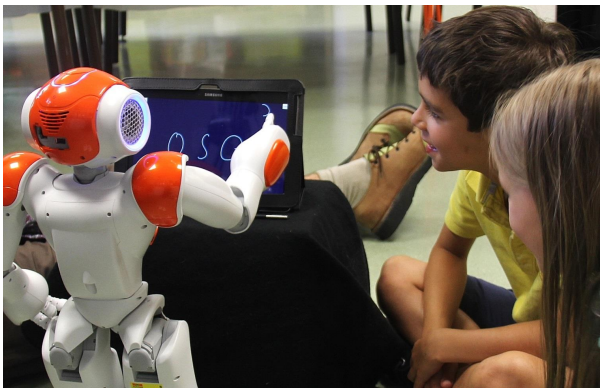


Figure 2.3: CoWriter: a robotic companion interacts with children and facilitates the development of their handwriting skills. The robot plays the role of a learner, engaging the children to practising the skill and improving their self-esteem.

requesting children's assistance. This so-called "learning by teaching" paradigm is believed as an effective approach to motivate and engage children learners in education activities (173).

The central scenario of the CoWriter project is an interaction activity between children and robots. For example, the robot can demonstrate a character sample which could be poorly written in the initial, while gradually improved under the help of children. Such an activity has been prototyped and its social and technical feasibility has been validated in (Figure 2.3, 77). Many research works, ranging from the robot control to the high-level activity design, are worth an effort to improve the system autonomy, the behavior effects, and as such, the ultimate learning gains of children. Among many research dimensions, the thesis focuses on the representation, formation and control of the handwriting skills, contributing to the project as a technical foundation and exploration.

Based upon the project motivation, handwriting is used as a running example to highlight the technical contributions throughout the thesis, even though most proposed algorithms are of a general purpose and their applications on other tasks are also included. Besides that, handwriting is also a motor skill that involves all the concerning aspects in the thesis. First, unlike a free-space movement, handwriting involves many contacts: the robot needs a careful balancing of the finger grips, and at the same time, an appropriate force accommodation on

the writing surface to generate legible characters. Hence, an impedance control, as is researched in the first part, is desired in this motor task. Secondly, human handwriting samples exhibit so many variabilities and regularities that the robot needs a proper representation for an efficient modeling and a diversified synthesis. The second part of the thesis eyes on this challenge from the broader view of modeling human behavior modes. Moreover, the proposed algorithm is extended with the features incorporating human movement characteristics so the character deformities can be intuitively controlled. This is shown to be helpful in the CoWriter interactions by generating more autonomous and richer robot handwriting samples (31). Finally, the handwriting proficiency is relevant to the development of both cognitive and motor capabilities (85). Therefore, the importance should be attached to task modalities beyond the motor movement. The third part of the thesis takes steps in the direction of jointly considering the character image and the motion generation, exploring the technical potential of introducing more sensor modalities in human-robot interactions similar to the CoWriter project.

3

Learning Structured Cost Functions and Controllers

3.1 Introduction

This chapter considers incorporating human priors in learning and synthesizing controllers. As reviewed in Section 2.1, the specification of motion trajectory and impedance is central and challenging for robots with many degrees of freedoms (DOFs). The learning from demonstrations (LfD) framework can mitigate this difficulty. On the other hand, the research about robot task decomposition and human movement has identified valuable properties for a convenient robot implementation and motion representation. For instance, the hybrid and parallel force/position control (166, 36) decompose a task specification into multiple orthogonal directions. Representing task parameters in a moving reference frame will be convenient to describe this decomposition. The description is similar to the natural curve representation, which is also applied in expressing the human movement regularities (81). One natural question is thus how to incorporate domain structures, such as a representation in the local reference frame, into LfD to synthesize controllers with the desired properties. This request,

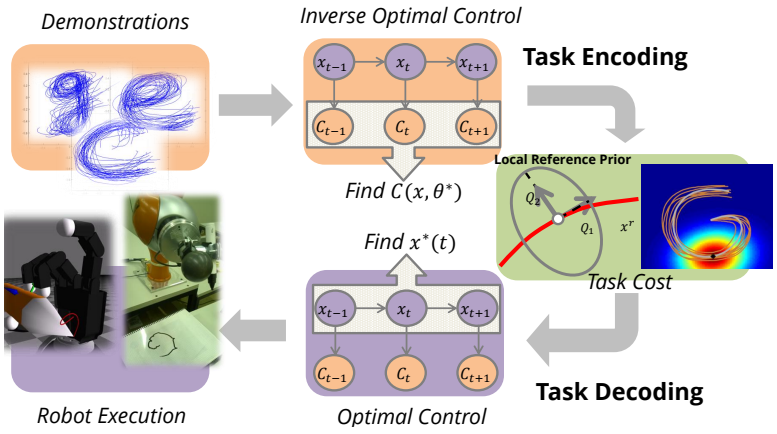


Figure 3.1: Learning a compliant robot motion through inverse optimal control. The motion trajectory and variability, together with the prior of a local reference representation, are encoded as a structured cost function. The task decoding derives an optimal impedance controller implemented as a robotic handwriting task on both single and multiple robot manipulator systems.

as shown below, sometimes adds complications that are not straightforwardly tractable for a conventional formulation. Henceforth, the central research question of this chapter, from both robotics and machine learning perspectives, can be summarized as:

- **Robotics**: how to synthesize the robot motion and impedance profiles with the certain constraints fulfilled.
- **Machine learning**: how to learn a structured task representation with the incorporated human priors.

Concretely, this chapter considers the IOC problem of extracting a tracking trajectory as well as the deviation penalty defined in the local frame of reference. This problem leads to a nonlinear parameterization, different from the popular IOC assumption that the cost function is linear with the unknown parameters. Also, the commonly used gradient-based methods are ill-suited for solving the non-convex optimization problem in this case, as they tend to end up with poor local optima. As another difficulty, the gradient derivation to explore the feature design for each model is error-prone and not applicable under a model-free setting.

Based on the optimal inference duality, this chapter proposes to use the cross-entropy method, a stochastic optimization algorithm to tackle the IOC

problems. The cross-entropy method evaluates samples without knowing the explicit model, which resembles a model-free approach. Importantly, the cross-entropy-method is flexible and efficient to incorporate the correlation of model parameters with a structured sampling. The sampling and learning is further facilitated by adopting a cost reparameterization. These novelties lead to an efficient approach with the desired compliance behavior encapsulated. Figure 3.1 illustrates the overall flow of our approach .The main contributions of this part are:

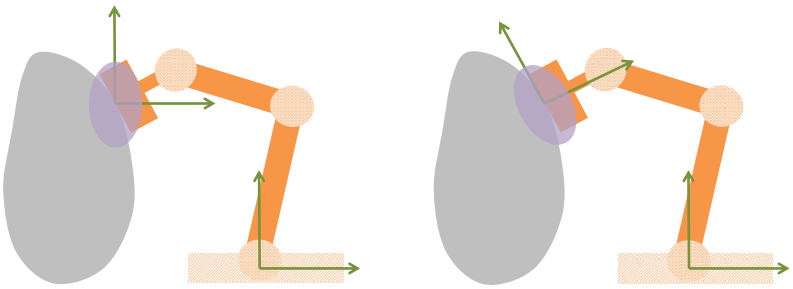
- A parameterization that naturally describes the impedance in the local moving reference frame, which connects the task decomposition in orthogonal control directions.
- A cross-entropy-like method for a model-free cost function learning with a nonlinear parameterization form.
- A nullspace sampling schema that embeds task priors and facilitates the trajectory optimization in the task decoding phase.

Most of the contents in this chapter have appeared in the publications (238, 239). Section 3.3 extends the published works with a more detailed analysis about the connection between cost-to-go and impedance parameters. The results (Section 3.6) also include some samples that were omitted due to the page limit.

3.2 Problem Statement

Following the notations in Section 2.3.2, this chapter considers the problem of transferring skills to a robot with demonstrated trajectories $\varsigma^* = \{\boldsymbol{x}_t^*\}$, where $\boldsymbol{x}_t^*$ denotes the pose of the interested frame, e.g., the robot end-effector. The star indicates the motions are optimal with respect to the underlying task goal. The goal is implicit and can be abstracted as a sum-up of the cost function $C(\boldsymbol{x}, \boldsymbol{\theta})$ along the trajectory $\varsigma = \{\boldsymbol{x}_t\}$. $\boldsymbol{\theta}$ is the parameter to infer for encoding the task. Note that the index t is a phase variable indicating the task progression. The skill transfer requires the robot to derive its own compliance behavior, mimicking the demonstrations as an impedance controller like Equation (2.2):

$$\boldsymbol{u}_t = \boldsymbol{G}(\boldsymbol{x}_t) - \boldsymbol{K}_t(\boldsymbol{x}_t - \boldsymbol{x}_t^r) - \boldsymbol{D}\frac{d\boldsymbol{x}_t}{dt} \tag{3.1}$$



(a) Global reference frame (b) Local reference frame

Figure 3.2: Representing the motion compliance in a global or a local reference frames: (a) the stiffness ellipse is aligned with a fixed global reference frame so the compliance description fails to consider the geometry of the interaction space. (b) varying the control stiffness in a local reference frame which moves according to the interaction surface. The local representation is desired as an intuitive way to decompose the control design for implementing the standard hybrid force/position scheme.

where $\boldsymbol{G}(\boldsymbol{x}_t)$ is a robot model dependent feedforward control, such as the gravity compensation term. $\boldsymbol{K}_t$ and $\boldsymbol{x}_t^r$ are the control parameters subject to the design or learning. In addition, much like the classical force/position control schemes (166, 36), the stiffness matrix $\boldsymbol{K}_t$ is expected to decompose the control directions and defines the local compliance behavior with respect to the motion trajectory. Effectively, this implies that the impedance behavior is described in the local or Frenet reference frame, as is shown in Figure 3.2.

The advantage of having a local representation lies in its intuitiveness for synthesizing and interpreting the controlled behavior. An example of the benefits can be demonstrated through a polishing task depicted in the Figure 3.2. In this case, it is desired to decouple the control directions in a way that one can orthogonally modulate the exerted forces in the normal and tangential directions. Adopting a global reference frame like Figure 3.2a ignores the geometry of polishing surface, describing and interpreting the task in a less explicit manner.

However, unlike the standard hybrid force/position control setup, here the reference trajectory is unknown and needs to be extracted from the noisy demonstrations. The problem can thus be divided into two phases. The first part which aims to reveal the unknown cost can be formulated as an inverse optimal con-

trol problem. In general, this problem is ill-posed as there are ambiguous results (e.g., constant cost) that always fulfill the optimality of demonstrations. One elegant way to address this, as is reviewed in Section 2.3.2, is the maximum-entropy framework (MaxEnt) (246), where trajectories are assumed to be subject to a Boltzmann distribution. By extending this concept, the estimation of the cost parameters effectively maximizes the demonstration likelihood under this distribution and a parameter prior:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\varsigma^*|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \frac{\exp(-\mathcal{J}(\varsigma^*, \boldsymbol{\theta}))}{\int_{\varsigma} \exp(-\mathcal{J}(\varsigma, \boldsymbol{\theta}))} p(\boldsymbol{\theta}) \tag{3.2}$$

where $\varsigma^* = \{\boldsymbol{x}^*_{1:T}\}$ and $\varsigma = \{\boldsymbol{x}_{1:T}\}$ denote demonstrated and all possible trajectories with a time horizon of T, respectively. $\mathcal{J}(\varsigma) = \sum_{t=1}^{T} C(\boldsymbol{x}_t, \boldsymbol{\theta})$ defines the accumulated cost-to-go the along trajectory ς. The incorporated prior, such as the local reference representation, is encoded as $p(\boldsymbol{\theta})$, whose concrete form is nontrivial and will be discussed in the following sections.

The second stage is to derive a robot optimal trajectory under its own dynamics given the learned cost. The remarks below about the robot dynamics are given as the additional problem assumptions:

- The execution upon a real robot dynamics desires a smooth variation of the impedance parameter, for both the force magnitude and exerting direction. The model parameter $\boldsymbol{\theta}$ thus also needs to take this into account.
- The MaxEnt formulation effectively assumes a stochastic dynamics with a uniform noise. Hence the learning stage is agnostic to the robot dynamics. Nevertheless, the control derivation can still exploit the concrete robot dynamics, which could be known as Equation (2.9), learned from the data or be unknown in a model-free trajectory optimization.
- The construction of the state feature $\boldsymbol{x}_t$ varies according to the robot dynamics. For instance, the recorded demonstrations might be featured in joint space but the feature of the cost function might be the trajectory of the end-effector or a manipulated object. Here the forward/inverse kinematics is assumed to be available to convert back and forth between the state feature $\boldsymbol{x}_t$ and the robot configuration.

3.3 Optimal Impedance Controller with Structured Cost Functions

The compliance design is determined by the stiffness matrix $\boldsymbol{K}$, which is in turn implied by the estimated cost-to-go function. For an illustrative purpose, let the interested operating point be a 2D particle with the state variable $\hat{\boldsymbol{x}} = [\boldsymbol{x}, \dot{\boldsymbol{x}}]$ representing its combined position and velocity. The control u is the acceleration or the scaled applied force. The motion dynamics for a unit mass can be written as:

$$\underbrace{\begin{bmatrix}\boldsymbol{x}_{t+1}\\ \dot{\boldsymbol{x}}_{t+1}\end{bmatrix}}_{\hat{\boldsymbol{x}}_{t+1}} = \underbrace{\begin{bmatrix}\boldsymbol{I} & dt\boldsymbol{I}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix}\boldsymbol{x}_t\\ \dot{\boldsymbol{x}}_t\end{bmatrix}}_{\hat{\boldsymbol{x}}_t} + \underbrace{\begin{bmatrix}\frac{1}{2}dt^2\boldsymbol{I}\\ dt\boldsymbol{I}\end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{u}_t \tag{3.3}$$

where $\boldsymbol{I}$ denotes a 2×2 identity matrix. According to the optimal LQR control reviewed in the background chapter, one can obtain:

$$\begin{aligned}
\boldsymbol{u}_t^* &= -(\boldsymbol{R} + \boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}_{t+1}\boldsymbol{B})^{-1}\boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}_{t+1}\boldsymbol{A}_t(\hat{\boldsymbol{x}}_t - \hat{\boldsymbol{\mu}}_t)\\
&= -(\boldsymbol{R} + \boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}_{t+1}\boldsymbol{B})^{-1}\begin{bmatrix}\frac{1}{2}dt^2\boldsymbol{I} & dt\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_{t+1} & \boldsymbol{0}\\ \boldsymbol{0} & \dot{\boldsymbol{\Lambda}}_{t+1}\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & dt\boldsymbol{I}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_t - \boldsymbol{\mu}_t\\ \dot{\boldsymbol{x}}_t - \dot{\boldsymbol{\mu}}_t\end{bmatrix}\\
&= -(\boldsymbol{R} + \boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}_{t+1}\boldsymbol{B})^{-1}\begin{bmatrix}\frac{1}{2}dt^2\boldsymbol{I} & dt\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_{t+1}(\boldsymbol{x}_t - \boldsymbol{\mu}_t) + dt\boldsymbol{\Lambda}_{t+1}(\dot{\boldsymbol{x}}_t - \dot{\boldsymbol{\mu}}_t)\\ \dot{\boldsymbol{\Lambda}}_{t+1}(\dot{\boldsymbol{x}}_t - \dot{\boldsymbol{\mu}}_t)\end{bmatrix}\\
&= -(\boldsymbol{R} + \boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}_{t+1}\boldsymbol{B})^{-1}[\frac{1}{2}dt^2\boldsymbol{\Lambda}_{t+1}(\boldsymbol{x}_t - \boldsymbol{\mu}_t) + (\frac{1}{2}dt^3\boldsymbol{\Lambda}_{t+1} + dt\dot{\boldsymbol{\Lambda}}_{t+1})(\dot{\boldsymbol{x}}_t - \dot{\boldsymbol{\mu}}_t)]
\end{aligned} \tag{3.4}$$

where $\hat{\boldsymbol{\Lambda}} = \text{diag}(\boldsymbol{\Lambda}, \dot{\boldsymbol{\Lambda}})$ is the block-diagonal weight matrix for the cost-to-go $\mathcal{J}(\hat{\boldsymbol{x}}) = (\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}})^T\hat{\boldsymbol{\Lambda}}(\hat{\boldsymbol{x}} - \hat{\boldsymbol{\mu}})$. Note that here $\dot{\boldsymbol{\Lambda}}$ is a bit abused for denoting the weight matrix for the velocity term $\dot{\boldsymbol{x}}$ instead of the time derivative of $\boldsymbol{\Lambda}$. Comparing with the feedback term of Equation (3.1) and fixing the velocity reference to zero, one can reveal a design for the impedance parameters:

$$\begin{aligned}
\boldsymbol{K}_t &= \frac{1}{2}dt^2(\boldsymbol{R} + \boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}_{t+1}\boldsymbol{B})^{-1}\boldsymbol{\Lambda}_{t+1}\\
\boldsymbol{D}_t &= \frac{1}{2}dt(\boldsymbol{R} + \boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}_{t+1}\boldsymbol{B})^{-1}(dt^2\boldsymbol{\Lambda}_{t+1} + 2\dot{\boldsymbol{\Lambda}}_{t+1})
\end{aligned} \tag{3.5}$$

On the other hand, when the cost-to-go is formalized as a quadratic one like above, the Boltzmann distribution in Equation (3.2) boils down to a Gaussian distribution with block-diagonal covariance matrices. When the cost parameters are estimated as the Gaussian statistics, both $\boldsymbol{R} + \boldsymbol{B}^T\hat{\boldsymbol{\Lambda}}\boldsymbol{B}$ and $\hat{\boldsymbol{\Lambda}}$ are at least semi-positive-definitive. So the impedance control is stable given the derived stiffness and damping matrices.

The relation between the cost parameter $\hat{\mathbf{\Lambda}}$ and the impedance design provides a perspective on a widely used heuristic, that the stiffness is designed to be inversely proportional to the trajectory covariance. To see this, consider the stiffness matrix with the cost parameter expanded:

$$\begin{aligned} \boldsymbol{K}_t &= \frac{1}{2}dt^2(\boldsymbol{R} + \frac{1}{4}dt^4\mathbf{\Lambda}_{t+1} + dt^2\dot{\mathbf{\Lambda}}_{t+1})^{-1}\mathbf{\Lambda}_{t+1} \\ &= \frac{1}{2}dt^2[(\boldsymbol{R} + dt^2\dot{\mathbf{\Lambda}}_{t+1})\mathbf{\Lambda}_{t+1}^{-1} + \frac{1}{4}dt^4]^{-1} \end{aligned} \tag{3.6}$$

where it can be seen that the stiffness co-varies with the inverse of the Gaussian covariance $\mathbf{\Lambda}_{t+1}$. In fact, a positive-definitive matrix parameterizes an ellipse (or an ellipsoid in the high-dimensional case). The cost parameter thus controls the impedance profile via two orthogonal dimensions: the magnitude of the ellipse axes which correlate to the reaction force; the orientation of the ellipse which specifies the control directions. From the point of view of a cost function, the magnitude and direction define which task dimensions are more sensitive to the disturbances. The preference of reducing the control effort refrains the stiffness magnitude along the less important dimensions. This is in accordance with the minimum intervention principle, and yields a compliance controller which is not only systematically synthesized but also optimal in terms of its impedance parameters.

For a uniform prior $p(\boldsymbol{\theta})$, the estimation of $\boldsymbol{\theta}$ is efficient as fitting a Gaussian trajectory distribution. In that case, the trajectory reference and variability are decoupled and the description of the impedance ellipse is independent of the desired movement. When considering a description in the local reference frame (Section 3.2), the parameters are correlated and result in a less tractable form. To see this, the connection between the ellipse orientation and the trajectory local reference frame is written as:

$$\begin{aligned} &\alpha = \arctan(\dot{x}_2, \dot{x}_1) \\ &\mathbf{\Lambda} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \end{aligned} \tag{3.7}$$

where the principal compliance components $[\Lambda_1, \Lambda_2]$ are decoupled from the original weight matrix and α denotes the angle of the local frame with respect to the world reference. One way to express this prior is to write $p(\boldsymbol{\theta})$ in a factored form:

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}_\Lambda|\boldsymbol{\theta}_\mu)p(\boldsymbol{\theta}_\mu) \tag{3.8}$$

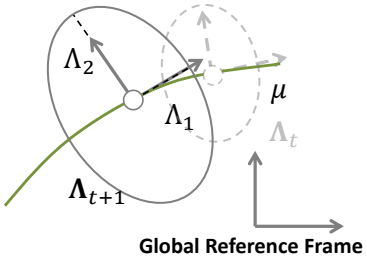


Figure 3.3: Varying impedance ellipse represented in the local reference frame. This is utilized for an intuitive force/position hybrid task specification, where the length and orientation of the principle axes correlate to the force magnitude and control direction.

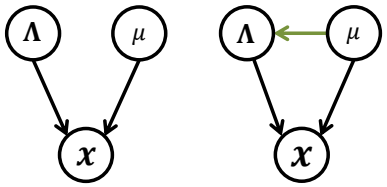


Figure 3.4: Graphical model of the observed variable $\boldsymbol{x}$ and the model parameters $\{\boldsymbol{\mu}, \boldsymbol{\Lambda}\}$. Left: the inference of model parameters is independent and relevant to the heuristics of variable impedance design based on the demonstration variability. Right: representing the impedance ellipse in the local reference frame yields a structured parameter prior so the cost parameters cannot be independently inferred.

where $p(\boldsymbol{\theta}_\Lambda|\boldsymbol{\theta}_\mu)$ conditions the weight matrix with the above constraint. The task prior eventually imposes a structure into the space of parameter $\boldsymbol{\theta}$. Synthesizing the target impedance controller requires the IOC problem to learn with a structured cost-to-go function. Apparently, in this case, $p(\boldsymbol{\theta}_\Lambda|\boldsymbol{\theta}_\mu)$ is not of a standard form for an efficient learning like fitting a Gaussian trajectory distribution. Moreover, the nonlinear parameter structure raises a few challenges to the standard gradient-based IOC approaches. On one hand, these approaches often learn a linearly parameterized cost function in order to guarantee a convex optimization. The estimation might be poor under a nonconvex optimization with respect to the nonlinear parameterization. On the other hand, it could be error-prone to derive the gradient from the parameter constraint and an approach based on less customizations is desired for handling other general task priors.

3.4 Cost Reparameterization

The thesis proposes a sampling-based probabilistic inference to address the dependency between cost parameters, as identified in Equations (3.7) and (3.8). Before its development, this section discusses a reparameterization of the cost

to facilitate the sampling procedure and the practical implementation.

In order to achieve this, the reparameterization encodes the evolution of reference states, eigen value and vectors of the weight matrices as parameter trajectories. The parameter trajectories are proposed to be represented with linear function approximators. Sampling from a featured trajectory space alleviates the issue of learning in a high dimension space. The linearity of the approximator parameter allows for an efficient sampling from the null space of the trajectories, hence handling constraints on the via-points. Moreover, a variety of basis functions could be adopted to enforce a smooth prior to the parameter variation. This is advantageous for a robust learning from noisy demonstrations. The smoothness prior could also prevent a drastic and impulsive change to the variable impedance, ensuring a safe robot implementation.

Concretely, a trajectory can be approximated with a linear combination of M normalized Radial Basis Function (RBF) plus a linear feature, taking the reference state $\boldsymbol{x}_t^r$ as an example:

$$\boldsymbol{x}_t^r(\boldsymbol{\omega}) = \boldsymbol{\omega}^T \boldsymbol{\Phi}(t) = \sum_{i=1}^{M} \omega_i \frac{\exp(-\gamma(t-t_i)^2)}{\sum_{j=1}^{M} \exp(-\gamma(t-t_i)^2)} + \omega_{M+1} t \tag{3.9}$$

where t indicates the phase variable for a general representation. The extra linear feature ensures a sparse representation to encode a straight line. For the nonlinear terms, when the phase variable is defined within the interval of $[0.0, 1.0]$, the basis center t_i can be selected to uniformly distribute the basis functions in the interval. γ shapes the width of the basis function and then entries of $\boldsymbol{\omega}$ weigh the contributions of each basis component, as shown in Figure 3.5.

Sometimes one might expect the sampled trajectories to fulfill some constraints, e.g., to pass through a specific point. This is especially useful in trajectory optimization when all the samples are supposed to start from an initial state x_0 or to fix both their boundary points. Such constraints can be imposed by sampling in the nullspace of the feature parameter space. Concretely, let $\boldsymbol{\omega}$ be constrained to generate trajectories passing through a set of points $\boldsymbol{X}_{const}^r$

$$\boldsymbol{\omega}^T [\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_c] = \boldsymbol{X}_{const}^r = [\boldsymbol{x}_{const}^1, ..., \boldsymbol{x}_{const}^c] \tag{3.10}$$

A linear transformation matrix $\boldsymbol{U}$ can be found through the Singular Value

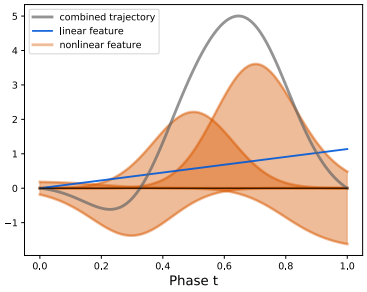


Figure 3.5: Representing a trajectory with a function approximator combining nonlinear and linear features. The parameters of the function approximator are the weight of each component: magnitude of the Gaussian RBF and slope of the linear feature.

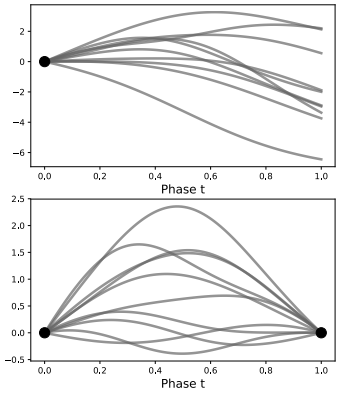


Figure 3.6: Sampling constrained trajectories from the nullspace of the approximator parameter. Top: fixing the start point; Down: fixing two end points.

Decomposition (SVD) to ensure

$$(\boldsymbol{\omega} + \boldsymbol{U}\delta\boldsymbol{\omega})^T[\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_c] = [\boldsymbol{x}^1_{const}, ..., \boldsymbol{x}^c_{const}] \tag{3.11}$$

to hold for any $\delta\boldsymbol{\omega}$ sampled in the subspace of the feature parameter space. In that sense, the parameter trajectories are efficiently explored without needing to reject those that violate the constraints. Figure 3.6 shows sampled trajectories with fixed end points. Also, when $\delta\boldsymbol{\omega}$ is sampled as Gaussian noise, the perturbed trajectory parameter $\boldsymbol{\omega} + \boldsymbol{U}\delta\boldsymbol{\omega}$ is still subject to a normal distribution in that $\boldsymbol{U}$ is a linear transformation.

Like the reference trajectory $\boldsymbol{x}^r_t$, other variables that control the weight matrix entries are encoded as:

$$\begin{aligned} \tan\alpha_t &= \frac{\sin\alpha_t}{\cos\alpha_t} = \frac{\dot{x}_2}{\dot{x}_1} = \frac{\boldsymbol{\omega}^T_{\alpha_2}\boldsymbol{\Phi}'(t)}{\boldsymbol{\omega}^T_{\alpha_1}\boldsymbol{\Phi}'(t)} \\ \boldsymbol{\Lambda}_t &= \begin{bmatrix} \cos\alpha_t & \sin\alpha_t \\ -\sin\alpha_t & \cos\alpha_t \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}^T_{\Lambda_1}\boldsymbol{\Phi}(t) & 0 \\ 0 & \boldsymbol{\omega}^T_{\Lambda_2}\boldsymbol{\Phi}(t) \end{bmatrix} \begin{bmatrix} \cos\alpha_t & -\sin\alpha_t \\ \sin\alpha_t & \cos\alpha_t \end{bmatrix} \end{aligned} \tag{3.12}$$

where $\boldsymbol{\Phi}'(\cdot)$ denotes the derivative of the basis function with respect to the phase variable, yielding another nonlinear basis for the function approximation. Comparing with Equation (3.7), all the unknown parameters are now defined (up to a constant scale) in the form of trajectory function approximators. Henceforth,

the cost learning is reparameterized to estimate $\boldsymbol{\theta} = \{\boldsymbol{\omega}, \boldsymbol{\omega}_\alpha, \boldsymbol{\omega}_\Lambda\}$.

3.5 Sampling-based Inference

The duality of optimal control and the probabilistic IOC like Equation (3.2) motivate to address the cost parameter optimization as an inference problem. The inference consists of two stages, each of which needs a sampler. The first sampling step takes samples from the parameter prior $p(\boldsymbol{\theta})$ to evaluate the posterior demonstration likelihood. The second routine samples $\boldsymbol{x}$ to estimate the likelihood denominator. Note that the latter in effect performs a trajectory optimization thus can also be used to derive the optimal control on another agent to execute the transferred task. Therefore the first stage of optimizing the cost parameter can be considered as task encoding while the trajectory optimization in the second stage decodes the task under the cost representation. Here both the task encoding and decoding are uniquely addressed through a cross-entropy-like method under the importance sampling scheme.

The importance sampling scheme suggests take samples from a proposal distribution when the target distribution is not of an easy form to take samples from. For instance, the posterior

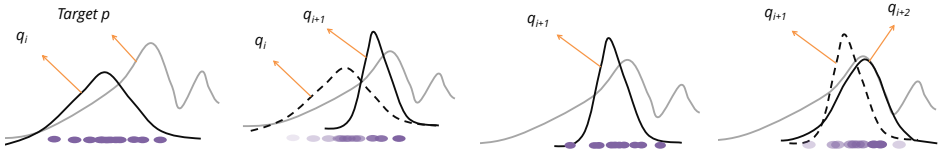
$$p(\boldsymbol{\theta}|\varsigma^*) \propto p(\varsigma^*|\boldsymbol{\theta})p(\theta) \tag{3.13}$$

is intractable for its component of the general Boltzmann form. The cross-entropy method usually uses a multivariate Gaussian $q(\boldsymbol{\theta}|\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$ as the proposal to approximate the intractable distribution. The $q(\boldsymbol{\theta}|\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$ is iteratively estimated based on the weighted samples $\{\hat{\boldsymbol{\theta}}_i\}$:

$$\boldsymbol{\mu}^*_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}^*_{\boldsymbol{\theta}} = \underset{\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}}{\operatorname{argmax}} \sum_i \mathbb{I}(\hat{\boldsymbol{\theta}}_i) \log q(\hat{\boldsymbol{\theta}}_i|\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}) \tag{3.14}$$

where $\mathbb{I}(\cdot)$ denotes the sample importance. The importance function $\mathbb{I}(\cdot)$ is subject to the user design. For example, in a standard cross-entropy method (46), $\mathbb{I}(\cdot)$ is a binary function screening out top performed samples, which construct a so-called elite set. In a similar spirit of the path-integral approaches (90, 213), the method presented here defines the importance function as:

$$\mathbb{I}(\hat{\boldsymbol{\theta}}_i) = \frac{\exp(-\eta \mathcal{L}(\hat{\boldsymbol{\theta}}_i))}{\sum_j \exp(-\eta \mathcal{L}(\hat{\boldsymbol{\theta}}_j))} \tag{3.15}$$



(a) (b) (c) (d)

Figure 3.7: Iterative sampling and importance evaluation in cross-entropy-like inference: (a)(c) - samples are taken from the current estimated proposal distribution q (black curve); (b)(d) - evaluating the importance of samples with respect to the target distribution p and fit a new proposal q. The transparency of dots indicates the sample importance evaluated under the target p (gray curve).

where $\mathcal{L}(\cdot)$ denotes the target cost function, e.g., the negative logarithm of the likelihood in Equation (3.2). In contrast with the binary $\mathbb{I}(\cdot)$ in the standard form, this importance function defines a soft elite-set membership, which is influenced by the Boltzmann temperature η. Such an importance assignment strategy has been demonstrated to be effective in robotics-related stochastic optimization (198).

Solving $\{\boldsymbol{\mu}_{\boldsymbol{\theta}}^*, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^*\}$ in Equation (3.14) simply fits a Gaussian distribution with weighted samples:

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{\theta}}^* &= \sum_i \mathbb{I}(\hat{\boldsymbol{\theta}}_i)\hat{\boldsymbol{\theta}}_i \\ \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^* &= \sum_i \mathbb{I}(\hat{\boldsymbol{\theta}}_i)(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\mu}_{\boldsymbol{\theta}}^*)(\hat{\boldsymbol{\theta}}_i - \boldsymbol{\mu}_{\boldsymbol{\theta}}^*)^T \end{aligned} \tag{3.16}$$

where the covariance estimation is a biased one and in practice the evolution of its eigen values are often truncated to assure a stable search and the chance of exploration in all dimensions. The general cross-entropy-like inference thus alternates between the score evaluation of the proposal samples and the estimation of the new distribution under weighted samples, as illustrated in Figure 3.7.

3.5.1 Learning Cost Function for Task Encoding

The cross-entropy method and feature sampling presented above are employed to learn the task cost by solving Equation (3.2). The partition function is evaluated with K locally sampled trajectories from another proposal distribution r (e.g., a Gaussian centered at the optimal solution). This is eventually

Algorithm 1 Encoding - An iteration step for learning the cost function based on cross-entropy stochastic optimization

Require: $\varsigma^*, \boldsymbol{\theta} = \{\boldsymbol{x}_t^r, \boldsymbol{\Lambda}_t\}, q(\boldsymbol{\mu}_{\boldsymbol{\theta}_{\boldsymbol{\omega}}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{\boldsymbol{\omega}}}), r(\hat{\varsigma}), K, N$ - Number of parameter and trajectory samples, $\boldsymbol{D}$ - Demonstrations of T length

Ensure: $\boldsymbol{\theta}^{New}$, $q(\boldsymbol{\mu}_{\boldsymbol{\theta}_{\boldsymbol{\omega}}}^{New}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{\boldsymbol{\omega}}}^{New})$

for all i in 1:N **do**

$\hat{\boldsymbol{\theta}}_{\boldsymbol{\omega}}^i \leftarrow q(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$ ▷ Sample parameters according to current distribution. Apply the nullspace projection if necessary.

$\hat{\boldsymbol{\theta}}^i = \{\hat{\boldsymbol{x}}_t^r, \hat{\boldsymbol{\Lambda}}_t\} \leftarrow$ Equation (3.9) ▷ Recover the reference trajectory and weight matrices from the feature space.

for all k in 1:K **do**

$\hat{\varsigma}_k = \{\boldsymbol{x}_t^k, t = 1, ..., T\} \leftarrow r(\hat{\varsigma})$ ▷ Sample locally perturbed trajectories for evaluating partition function.

end for

$\mathcal{L}_i \leftarrow -\sum_{j=1}^{D} \log \frac{\exp(-\mathcal{J}(\varsigma_j^*, \hat{\boldsymbol{\theta}}^i))}{\sum_{k=1}^{K} \frac{1}{\gamma(\hat{\varsigma}_k)} \exp(-\mathcal{J}(\hat{\varsigma}_k, \hat{\boldsymbol{\theta}}^i))}$

end for

$\{\hat{\boldsymbol{\theta}}_{\boldsymbol{\omega}}^i\}_{elite} \leftarrow \text{EliteSet}(\{\hat{\boldsymbol{\theta}}_{\boldsymbol{\omega}}^i, \mathcal{L}_i\})$ ▷ Construct the elite set.

$\boldsymbol{\theta}^{New}, \boldsymbol{\mu}_{\boldsymbol{\theta}_{\boldsymbol{\omega}}}^{New} \leftarrow \text{Mean}(\{\hat{\boldsymbol{\theta}}_{\boldsymbol{\omega}}^i\}_{elite})$

$\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{New} \leftarrow \text{Covar}(\{\hat{\boldsymbol{\theta}}_i\}_{elite})$ ▷ Update parameters through Equation (3.16).

solving the forward trajectory optimization and the sampling-based algorithm is given in the next section [1]. Assuming a uniform prior $p(\boldsymbol{\theta})$, Equation (3.2) is rewritten as minimizing the negative log-likelihood

$$\boldsymbol{\theta}_{\boldsymbol{\omega}}^* = \underset{\boldsymbol{\theta}_{\boldsymbol{\omega}}}{\text{argmin}} - \sum_{i=1}^{D} \log \frac{\exp(-\mathcal{J}(\varsigma^*, \boldsymbol{\theta}_{\boldsymbol{\omega}}))}{\sum_{k=1}^{K} \frac{1}{r(\hat{\varsigma}_k)} \exp(-\mathcal{J}(\hat{\varsigma}_k, \boldsymbol{\theta}_{\boldsymbol{\omega}}))} \tag{3.17}$$

where $\hat{\varsigma}_k = \{\hat{\boldsymbol{x}}_{1:T}^k\}$ is the locally sampled trajectory. $\boldsymbol{\theta}_{\boldsymbol{\omega}} = \{\boldsymbol{\omega}, \boldsymbol{\omega}_\alpha, \boldsymbol{\omega}_\Lambda\}$ are the learning parameters in the feature space of the function approximator. D denotes the number of demonstrations. The proposal sampler for the parameter distribution $q(\boldsymbol{\theta}_{\boldsymbol{\omega}})$ is factorized as $q(\boldsymbol{\theta}_{\boldsymbol{\omega}}) = q(\boldsymbol{\omega}_{\boldsymbol{\Lambda}}) q(\boldsymbol{\omega}_\alpha | \boldsymbol{\omega}) q(\boldsymbol{\omega})$, with $q(\boldsymbol{\omega}_\alpha | \boldsymbol{\omega})$ defined as a deterministic mapping or a Dirichlet distribution. The remained components follow the Gaussian assumption in the standard cross-entropy method to assure efficient distribution sampling and fitting in the Algorithm 1.

1. One can calculate a closed-form solution for a Gaussian distribution and quadratic function. However, a sample-based evaluation is used here to be consistent with the decoding algorithm.

Algorithm 2 Decoding - An iteration step for deriving trajectory based on cross-entropy stochastic optimization

Require: $\boldsymbol{\omega_y}, \boldsymbol{\Psi}, r(\boldsymbol{\mu}_{\boldsymbol{\omega}_y}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}_y}), C(\boldsymbol{x}_t, t), N$ - Number of samples

Ensure: $\{\boldsymbol{y}_t\}$, $r(\boldsymbol{\mu}^{New}_{\boldsymbol{\omega}_y}, \boldsymbol{\Sigma}^{New}_{\boldsymbol{\omega}_y})$

for all i in 1:N **do**

$\hat{\boldsymbol{\omega}}^i_{\boldsymbol{y}} \leftarrow r(\boldsymbol{\mu}_{\boldsymbol{\omega}_y}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}_y})$ ▷ Sample trajectory parameters according to current distribution. Apply the nullspace projection if necessary.

$\varsigma_i = \{\boldsymbol{x}_t\}_i \leftarrow (\hat{\boldsymbol{\omega}}^i_{\boldsymbol{y}})^T \boldsymbol{\Psi}(t)$ ▷ Take a rollout by following the trajectory for each DOF to obtain task-featured states.

$\mathcal{L}_i \leftarrow \mathcal{J}(\varsigma_i) = \sum_{t=1}^{T} C(\boldsymbol{x}_t, t)$

end for

$\{\hat{\boldsymbol{\omega}}^i_{\boldsymbol{y}}\}_{elite} \leftarrow EliteSet(\{\hat{\boldsymbol{\omega}}^i_{\boldsymbol{y}}, \mathcal{L}_i\})$ ▷ Construct the elite set.

$\{\boldsymbol{y}_t\}, \boldsymbol{\mu}^{New}_{\boldsymbol{\omega}_y} \leftarrow \text{Mean}(\{\hat{\boldsymbol{\omega}}^i_{\boldsymbol{y}}\}_{elite})$

$\boldsymbol{\Sigma}^{New}_{\boldsymbol{\omega}_y} \leftarrow \text{Covar}(\{\hat{\boldsymbol{\omega}}^i_{\boldsymbol{y}}\}_{elite})$ ▷ Update parameters through Equation (3.16).

3.5.2 Generating Motion Trajectory as Task Decoding

A state trajectory $\varsigma = \{\boldsymbol{x}\}$ can be derived to facilitate the partition function evaluation in Algorithm 1. This also effectively solves the trajectory optimization given the learned cost function, as such, decoding the task representation. Sample-based inference, such as the cross-entropy method, can approach the problem as a model-free method. This property is desired because the task relevant state $\boldsymbol{x}$ might depend on other actuated states. For instance, the underlying motion is exercised in the joint space as $\{\boldsymbol{y}_t\}$, and can be converted to the interested state space of the cost function through $\kappa(\boldsymbol{y}_t)$. Note that the complexity of κ depends on the robot embodiment as well as the task definition. It can be a kinematic function for characterizing the joint movement of a single manipulator in the Cartesian workspace, or other nontrivial forms, e.g., consider a $\kappa(\cdot)$ that correlates the joint trajectory of a anthropomorphic hand to the motion of a manipulated object.

Similar to the encoding algorithm, the iteration step of trajectory optimization is given as Algorithm 2. It works as a cross-entropy method with function approximators $\boldsymbol{y}_t = \boldsymbol{\omega}^T_{\boldsymbol{y}} \boldsymbol{\Psi}(t)$ for all of the actuated robot DOFs [2]. The only extra requirement is the measurement of the task state $\boldsymbol{x}_t = \kappa(\boldsymbol{y}_t)$ though the feature mapping $\kappa(\cdot)$ itself could be unknown.

2. $\boldsymbol{\Psi}(\cdot)$ could be same as or different from the task trajectory feature $\Phi(\cdot)$. A new symbol is used here to differentiate the feature design of the cross-entropy optimization in the decoding stage.

3.6 Implementation and Results

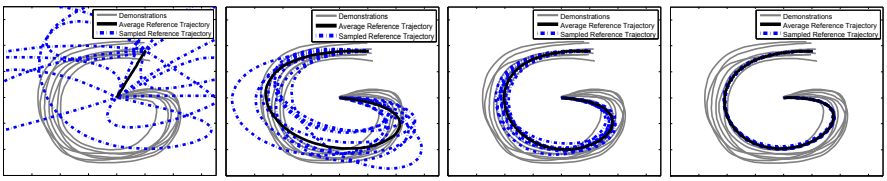
This section reports the implementation of the algorithms and obtained results on a robotic handwriting task. The first part illustrates how the structured cost parameters are learned under the proposed cross-entropy like inference. The robotic handwriting motion is then developed on both simulated and real robots, including implementations on both a single manipulator and a mult-fingered robotic hand. In the last experiment, the impedance controller is examined in a contact-involved motion to validate the learned compliant behavior.

3.6.1 Encoding Task Cost for Letter Trajectories

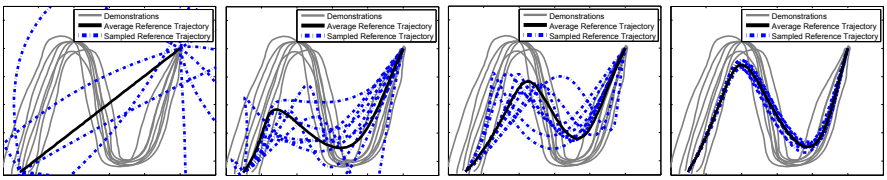
In this experiment, Algorithm 1 is implemented to learn handwritten letter trajectories. The purpose of this experiment is to extract from demonstrations an informative cost as the task representation. The cost will be further exploited to reproduce the writing task on robot agents with different embodiments.

The letter trajectories are from the dataset reported in (97). Only position coordinates are considered, thus the data consists of a series of 2D coordinates. In this experiment, the trajectories are aligned to the same time horizon by curve fitting and subsampling. All letter coordinates are within a comparable range and defined with respect to the trajectory end points.

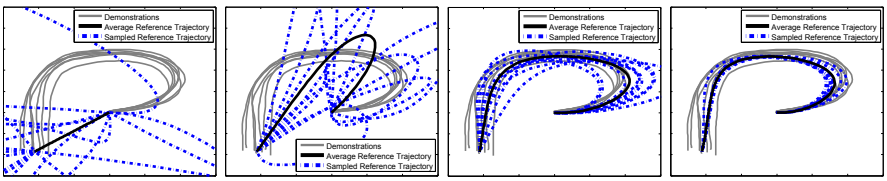
Figure 3.8 illustrates some particular iteration steps of the learning process for letters "G", "N" and "P", where for each letter seven demonstrations are used as the training data. For all these letter examples, the reference trajectory is naively initialized as a straight line, and the initial sampling distribution is set with a variance of 0.05 to ensure that a large enough parameter space is explored. We use 9 RBF basis functions to approximate the reference trajectory and modulation of eigen values of the precision matrix. The function approximator is set to represent trajectories with both the two end points fixed and such a constraint can be observed from all of the samples throughout the iteration steps of importance sampling, for which 15 parameter samples are used. As a result, the learned reference trajectory $\{\boldsymbol{x}_t^r\}$, which is encoded by the mean parameter of sampling distribution, rapidly converges to capture the profile of demonstrated trajectories. Only tens of steps are needed to achieve this even the naive initial guess might be far from the demonstration data. On the other



(a) Iteration Step = 1 (b) Iteration Step = 10 (c) Iteration Step = 15 (d) Iteration Step = 30



(e) Iteration Step = 1 (f) Iteration Step = 5 (g) Iteration Step = 10 (h) Iteration Step = 50



(i) Iteration Step = 1 (j) Iteration Step = 5 (k) Iteration Step = 15 (l) Iteration Step = 70

Figure 3.8: Evolution of the reference trajectory as the learning algorithm iterates. The iterations begin with a straight line as a tentative initial guess. The average trajectory evolves towards demonstrated profile to increase the likelihood of demonstrations. The proposal distribution converges as covariance of the sampled trajectories shrinks at the final stage.

hand, it can be seen that, the variance of the sampling trajectories also decreases as the iteration evolves. This implies that the sampling distribution shrinks its entropy thus the estimation of the reference trajectory tends to be certain.

The other cost parameter dimension is shown in Figure 3.9. Here, the varying weight matrix $\mathbf{\Lambda}_t$ is highlighted. The positive definite matrix is illustrated as a heating ellipse whose center is located at the current reference point, and the axes represent principle directions and the inverse of eigen values. Taking the letter "G" as an example, it is clear that the direction of the principle axes varies with respect to the local reference frame along the tracking trajectory. Also, the length of principle axes, which indicates weight parameter in the corresponding direction, captures the sensitivity of deviance from reference trajectory at each regulation point. Similar to the heuristic of variable impedance design based on trajectory variance, the IOC algorithm encapsulates this as the structured cost parameter. As an example, the ellipse expands its length of axis along the

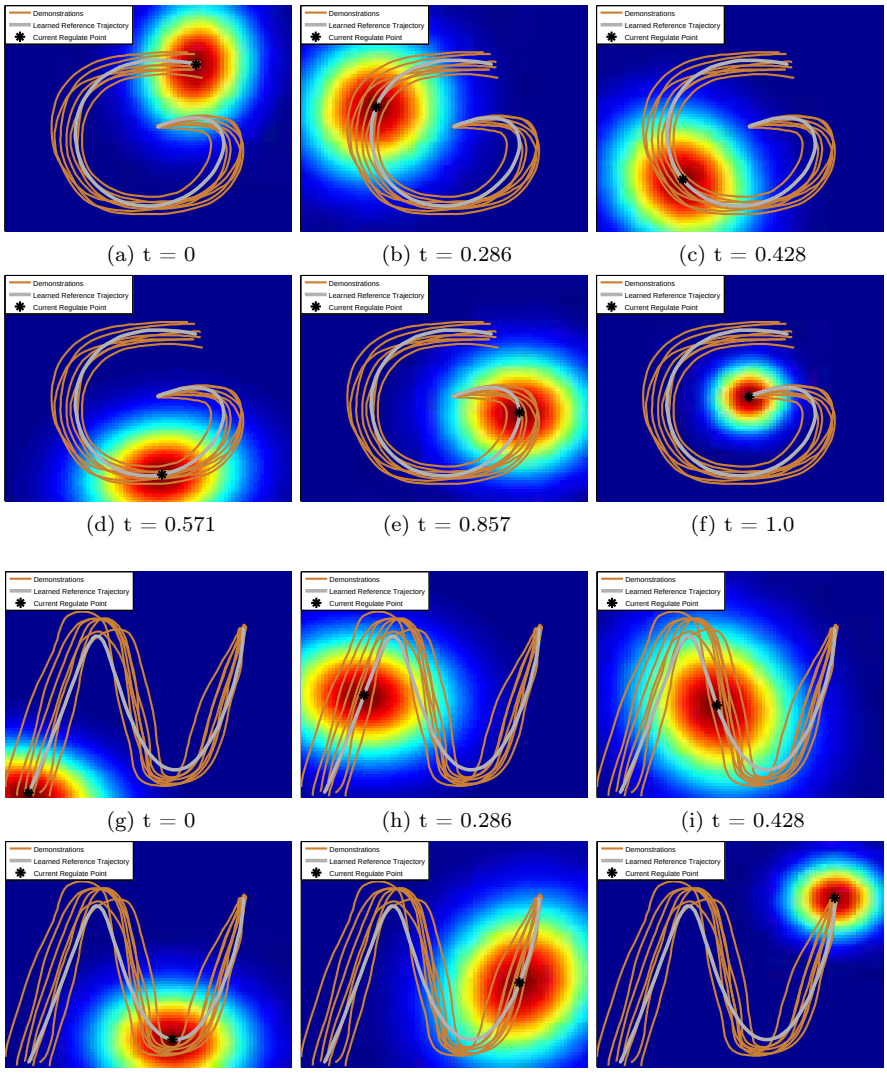
radial direction of the curve in 3.9b. On the contrary, in 3.9d, the ellipse shrinks its axis length along the radial direction as the demonstrated trajectories are more consistent within these sections. Moreover, since the demonstrations are aligned with respect to the termination point. The ellipse size is minimized corresponding to the truncated covariance eigen values in the cross-entropy method. Indeed, the deviation along these directions will incur a large cost penalty and the reference trajectory is expected to be well tracked. Note that at certain positions, such as Figure 3.9b, the ellipsoid is almost circular so the orientation of local reference frame is not very obvious. This is because the demonstrations are widely distributed in this section so the motion is flexible along different directions. Also, the RBF basis functions implicitly assume a smooth variation of cost parameters. This indeed results in a biased estimation so that the parameters are not fully determined by the data under the original Equation 3.2. However, during the execution, the noise of data might lead to a drastic impedance change, which can be harmful to the robot hardware. Embedding a dynamical parameter in the space spanned by these basis functions suppresses such drastic changes, establishing a smooth transition of the ellipsoid shape from Figure 3.9a to 3.9c.

Following the derivation of an optimal impedance controller in Section 3.3, it is thus natural to transfer a varying stiffness profile to the robot agents. The concrete compliant behavior subject to an external human interaction will be demonstrated in the robot experiments.

3.6.2 Decoding Task Cost: Robot Handwriting Motion

In this experiment, robot handwriting motion is derived as a decoding of the learned task cost. Although the quadratic cost pertains to a trajectory-based representation, it is still flexible to incorporate different task-relevant features and additional models such as inverse dynamics control.

To see this, the Algorithm 2 is instantiated on an anthropomorphic robot hand, on which the feature $\boldsymbol{x}$ is constructed with the $\kappa(\cdot)$ of a non-trivial form. The 16-DOFs Allegro (Figure 3.10) can be considered as a system consisting of multiple manipulators. The interested feature $\boldsymbol{x}$ is the pose of the manipulated pen and has to be realized by coordinating the joint motion of the involved



(a) t = 0 (b) t = 0.286 (c) t = 0.428

(d) t = 0.571 (e) t = 0.857 (f) t = 1.0

(g) t = 0 (h) t = 0.286 (i) t = 0.428

(j) t = 0.571 (k) t = 0.857 (l) t = 1.0

Figure 3.9: Results of learning a variable weight matrix as the task proceeds. The inverse of the matrix $\mathbf{\Lambda}_t$ is illustrated as a moving heating ellipse by evaluating cost value over the entire state space. The task phase horizon is scaled between 0 and 1.0.

Figure 3.10: 16-DOFs Allegro robotic hand with 4 joints for each finger manipulator.

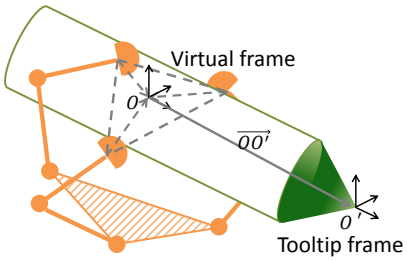


Figure 3.11: Approximating the object pose through a virtual frame. The interested feature $\boldsymbol{x}$, the tooltip frame, is then defined in an object-level reference.

fingers. To simulate the mapping function κ to get the task feature from the finger joint motion, a virtual object frame, which is commonly used in the grasping and dexterous manipulation community, is adopted here. As shown in Figure 3.11, the virtual frame is statically defined by the position vector of the tips. For the case of three fingers, the origin (O in Figure 3.11) of the virtual frame is the average position of involved end-effectors, and the orthogonal axes can be determined with the cross products of relative position vectors. The pen tip (O' in Figure 3.11) is assumed to be fixed, with respect to this virtual frame via a known transformation along the pen axle. Note that κ is designed for evaluating the cost and it is not known to the algorithm. More details about the principle and application of the virtual frame is off the main thesis topic and interested readers can refer to (130).

In the experiment of writing a letter "e", $N = 15$ samples are sufficient for exploring an optimal result. As per the parameterization of the function approximators, candidate trajectories are initialized as straight lines in the joint space. 15 samples are used in the importance sampling process. The evolution of cost values within 1000 iterations is shown in Figure 3.12. Indeed, the proposed algorithm is effective for the trajectory optimization, as the cost monotonically decreases to a relatively stable level within a few hundred iterations. Also, the variability (gray area) of the costs of sampled trajectories decreases as the samples tend to be identical, implying the exploration variance vanishes and as such, a convergence to a near optimal solution is achieved.

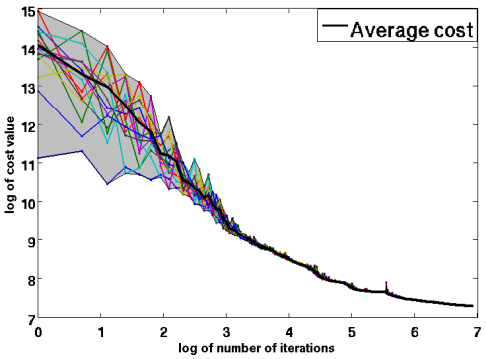


Figure 3.12: Logarithm of the cost in the iterations of running the decoding algorithm. 15 samples (curves of different colors) are taken to evaluate the rollouts and re-fit the proposal parameter distribution. The black bold curve indicates the averaged performance.

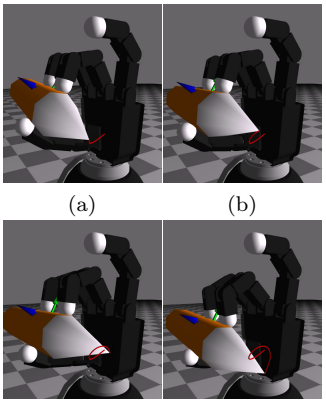


Figure 3.13: Multi-finger joint motion for writing a letter "e" via the pen-tip.

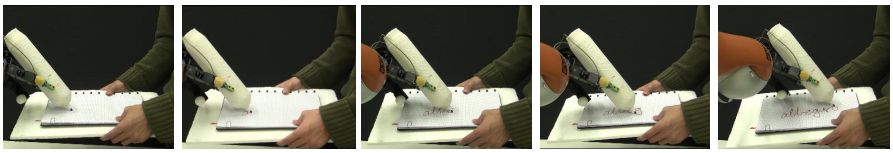
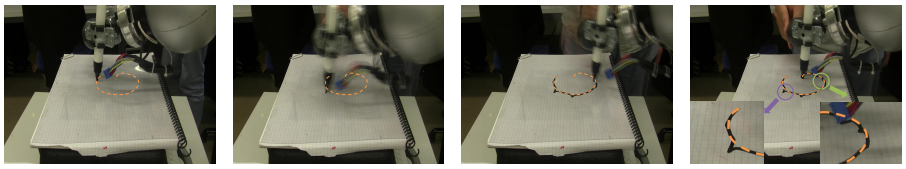


Figure 3.14: Cursive handwriting motion implemented on a manipulator with the pen compliantly grasped by the Allegro hand.

As a more general example, cursive handwriting is implemented on a real 7-DOFs KUKA LWR arm, hence applying the algorithm to a different kinematic structure and task feature. In this experiment, the pen is held by the Allegro hand mounted on the robot arm, and the motion is realized as writing a word "allegro" on a board grabbed by a human. Figure 3.14 demonstrates the success of derived motion. Besides the motion itself, the learned compliance is also approximately specified through the object-level impedance controller on the Allegro hand. As a result, the motion exhibits certain robustness to accommodate unmodeled uncertainties, which include the surface texture and more importantly, a varying board orientation under the human manipulation.



(a) Start writing (b) Large deviation under perturbation (c) Small deviation under perturbation (d) Finishing writing

Figure 3.15: Snapshots of writing "G" with the developed impedance parameters: (b) Low stiffness along radial direction - large deviation and vibration incurred under the perturbation; (c) and (d) High stiffness - small oscillation amplitude under perturbation; Reference trajectory is illustrated as the dash line and the perturbed sections are shown in detail in (d). Note to compare with the weight ellipse shape from Figure 3.9a to 3.9f

3.6.3 Decoding Task Cost: Handwriting Impedance Control

In this experiment the developed impedance is examined through a closer observation. Concretely, the end-effector motion compliance in Cartesian space is implemented by the 7-DOFs KUKA LWR robot. The encapsulated compliance is validated by subjecting the robot to disturbances during the writing execution. Figure 3.15 shows robot's compliant behavior with the developed varying impedance parameter. As expected, the robot exhibits relatively compliant behavior to perturbation in Figure 3.15b. This property can be understood by revisiting the learned cost in Figure 3.9b. Note that in Figure 3.9b, the heating ellipse indicates the inverse of weight matrix $\mathbf{\Lambda}_t$ thus a smaller axis length implies a larger desirability to keep the motion on track. In contrast to this, the robot is comparatively stiff in the radial direction in Figure 3.15c and one can observe even more resistance under perturbation in Figure 3.15d. Correspondingly, this can be explained by a larger $\mathbf{\Lambda}_t$ in these sections, with an increased impedance parameter developed.

Finally, the learned weight matrices are applied for writing other letters, with the aim of showing the generality of the learned cost function. As is shown in Figure 3.16, letters "N" and "W" are written with the impedance trajectory derived from the local structure of $\mathbf{\Lambda}_t$ of "G". The eigen values of $\mathbf{\Lambda}_t$ is independent from the reference trajectory. These values hence encapsulate the knowledge about how to shape the stiffness ellipse in the motion tangential

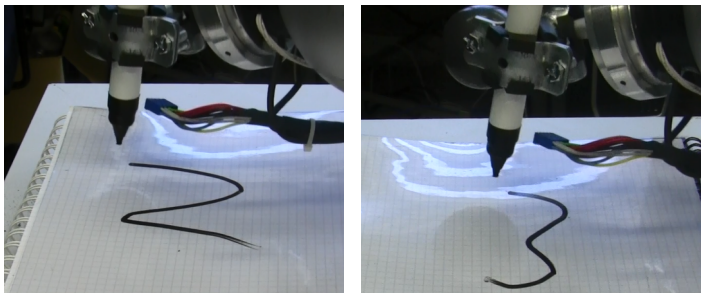


Figure 3.16: Generalizing the cost parameters to other letters: writing "N" and "W" with the impedance by exploiting the local $\mathbf{\Lambda}_t$ extracted from "G".

and radial directions. The robot is then enabled to still execute a modified trajectory by overcoming the friction, which is the main disturbance along the motion velocity direction.

3.7 Discussion

The approach presented in this section addresses learning and decoding structured cost-to-go functions. The special function structure incorporates the local reference frame representation, a robotics-related prior, while also results in tractability issues for a standard IOC solution. The presented algorithms take a probabilistic inference perspective to tackle the original problem, securing efficient computations for learning and reasoning about the task objective. As an answer, the discussed approaches employ the structured cost to address the identified domain research questions:

- **Robotics**: the task constraints, e.g. the impedance variation with respect to the local motion, can be incorporated as the dependencies within cost-to-go parameters, e.g., the correlation between the reference and weight matrices. Shaping the control synthesis is achieved by optimizing this structured cost-to-go function.
- **Machine learning**: the duality between optimal control and inference can be utilized to solve the IOC as a probabilistic inference problem, as such incorporating the parameter priors in the form of a structured distribution for the sampling-based inference.

The cross-entropy-based inference is a general stochastic optimization thus the cost-to-go and trajectories are not limited to be quadratic or linearly parameterized. The adopted cost form is advantageous for this specific task in two aspects: on one hand it explicitly draws a connection to existing heuristics about impedance design; on the other hand, it exploits the structure (dependency between the reference trajectory and local frame) to improve the sampling efficiency. The principle of inference-based IOC itself is applicable to a broader range of cost parameterizations and applications, as long as the interested structure can prompt an efficient sampling process.

The presented cost parameterization and trajectory approximation allow to learn a variable impedance profile while assuming a single reference trajectory. The question arising from this limitation is how one can learn multiple adaptable reference trajectories. This is interesting from the robotics point of view in that the human motor control not only modulates the limb impedance but also, in certain cases, systematically adapts the motion trajectory. From a machine learning perspective, the presented IOC algorithms assume similar demonstrations and only capture the data with a single dynamics or behavior mode. This assumption helps to simplify the hypotheses space while faces difficulties in modeling diversified human data. Some of these limitations will be discussed and addressed in the following chapters.

4

Modeling Latent Behavior Modes

4.1 Introduction

The learning from demonstration (LfD) approach presented in the last chapter models data variabilities around a reference trajectory. Sometimes the variabilities should be interpreted as the consequence of other task parameters, instead of the ignorable factors such as motor noise. For instance, humans might perform an identical task in their own preferred ways, exhibiting different behavior modes. The variety of demonstrations could be driven by the personal intention, contextual cues or social factors. The behaviors can be less ambiguous with these factors labeled. However, usually the labels are implicit due to the limitation of robot perception capability. As a result, the robot might have to learn from demonstrations that are not completely observable.

This chapter tackles the problem of learning from human data with latent behavior modes. An LfD approach is developed for programming rich behaviors without the need of labeling each demonstration. Reasoning about an incomplete task observation solicits the inference of what is unknown from what is

known. In light of this, the LfD model can be used in a pipeline that complements the perception and then derives the control. As an example, in a collaboration task, the intended behavior mode of a human operator might be implicit to the robot. A model about these latent behavior modes can be leveraged for the robot to resolve the perceptual uncertainty and act cooperatively, as such achieve an improved task performance.

Modeling diversified behaviors entails an LfD algorithm that disambiguates local and global distinctions. This requires estimating a multi-modal demonstration distribution. When the distribution is parameterized with nonlinear cost functions, the estimation is feasible under the standard probabilistic IOC framework. However, the generality of the standard formulation trades-off a high computational cost and approximation arising from the partition function evaluation. Also, the interpretability of the popular parameterization, which linearly combines a set of nonlinear basis functions, is not explicit for understanding the mode of an observed behavior. Noting these challenges, the research questions are set from both the robotics and machine learning (ML) perspectives:

- **Robotics**: how to facilitate the robot perception and adaptation by reasoning about multi-mode task demonstrations.
- **Machine learning**: how to efficiently learn an IOC model from demonstrations with unobservable modes.

The proposed approach exploits the problem structure about the behavior mode, which is cast as a discrete latent variable. A divide-and-conquer strategy is adopted to break the problem into efficient pieces that deal with similar demonstrations. The difficulty of grouping similar demonstrations is mitigated by bagging a collection of naive models. Therefore, the idea is leveraging ensemble principle and aggregating simple cost-to-go representations to yield a powerful model. The validity of learning simple models is ensured by focusing on locally consistent data. The data grouped in the subset is labeled with a discrete latent state that can be cast as the mode of these demonstrations. The posterior estimation of the latent variable is efficient, leading to an online mode inference and supporting realtime motion adaptation. As a summary, the main contributions are:

– An Ensemble IOC algorithm based on the linear-solvable system for learning cost-to-go functions and tackling incomplete demonstrations.
– A new perspective on Gaussian Mixture models (GMM) in the context of IOC. The results shed light on what GMMs actually learn (local MaxEnt models) and how can they be used as a guaranteed approximation.
– Integration of the task dynamics with the latent state to handle the challenge from incomplete state observation, for which a direct multi-mode policy encoding fails. The augmented dynamics provide a strategy to exploit the task redundancy to accommodate the disturbances or human intervention on-the-fly.

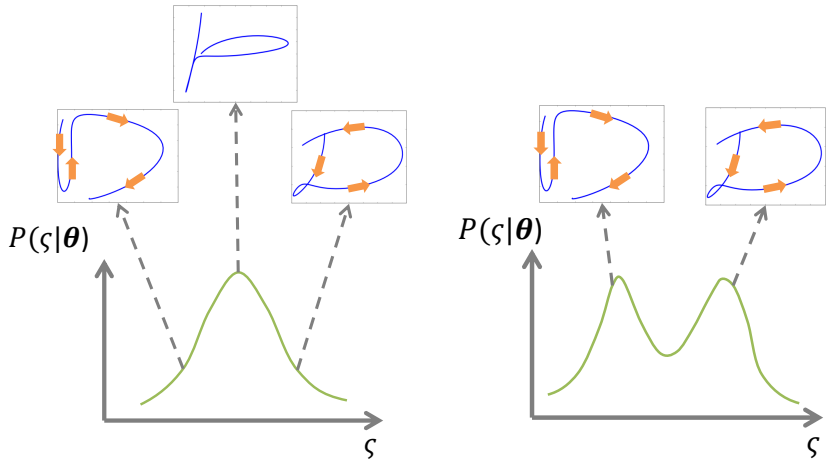
This chapter is based on the published work (240) and a submitted journal paper. Most of the sections are based on journal submission, which encompasses and extends the approach presented in (240). The contents of Section 4.5 and 4.7.2 only appear in (240). These sections focus on unique kinematics features and results of synthesizing human-like handwriting motion, pertaining to the main application of (240).

4.2 Problem Statement

4.2.1 Learning and Synthesizing Multi-mode Behaviors

Let the expert demonstrations be a dataset $\mathcal{D} = \{\varsigma^i\}$ with i as the data index. Taking the handwriting task as an example, the demonstrated data could be a set of trajectories that form different styles of written letters in the Cartesian space, with the planar position coordinate the features $\boldsymbol{x}_t^i \in \varsigma^i$. Similar as the trajectory parametrization in Section 2.3.2 and Chapter 3, the subscript t refers to the phase index. The demonstrated trajectories can be aligned by scaling the horizon to the same phase interval, e.g., from 0.0 to 1.0.

Unlike Chapter 3, the demonstrated trajectories are not necessarily with a distribution of one mode. Indeed, the driving factor of forming a stereotyped trajectory is abstracted as a discrete variable $z^i \in \mathbf{N}$, which is, however, not explicitly observed in $\mathcal{D}$. To put it into perspective, the latent variable z^i indicates a particular way of executing the task. Taking Figure 4.1b as an example, the stroke direction of forming the circle in writing the two types of "D" depends on the global style instead of the local geometry. Depending on



(a) Unimodal (b) Multi-modal

Figure 4.1: Unique and multiple modes of demonstration trajectories to execute a task, with handwriting motion as an example. 4.1a is a poor model to encapsulate the diversity and redundancy of styles in forming the letter "D". Actually, the unique mode, which approximately represents the mean trajectory, is not legible, and should be assigned with low probability (high cost value) instead. Also, the state itself (the point coordinate on the arc) is not sufficient to determine the next desired position.

the context, z^i is interchangeably interpreted as "style" or "mode" throughout this chapter.

The human and robotic agents are constrained by their corresponding dynamical models, as the linearly-solvable dynamical system reviewed in Section 2.3.2 :

$$\begin{aligned} \boldsymbol{x}_{t+1} &= f(\boldsymbol{x}_t) + \boldsymbol{B}(\boldsymbol{u}_t + d\boldsymbol{w}) \\ d\boldsymbol{w} &\sim \mathcal{N}(0, \boldsymbol{\Sigma}_0) \end{aligned} \tag{4.1}$$

where $d\boldsymbol{w}$ is the additive noise. The parameters of the dynamical system are assumed to be known or empirically determined.

The cost-to-go function that steers the desired behavior, is of a similar form in 2.3.2 but now dependent on $\boldsymbol{z}$:

$$\mathcal{J}_\varsigma(\boldsymbol{x}_0, \boldsymbol{z}, \boldsymbol{\theta}, t_0) = \sum_{t=t_0}^{T} C(\boldsymbol{x}_t, \boldsymbol{z}, \boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{u}_t^T \boldsymbol{R}\boldsymbol{u}_t \tag{4.2}$$

Learning multi-mode behaviors is thus estimating the parameter of the condi-

tional distribution:

$$P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{z}, \boldsymbol{\theta}) = \frac{P_0(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)e^{-\mathcal{J}_\varsigma(\boldsymbol{x}_{t+1}, \boldsymbol{z}, \boldsymbol{\theta})}}{\int_{\boldsymbol{x}'_{t+1}} P_0(\boldsymbol{x}'_{t+1}|\boldsymbol{x}_t)e^{-\mathcal{J}_\varsigma(\boldsymbol{x}'_{t+1}, \boldsymbol{z}, \boldsymbol{\theta})}d\boldsymbol{x}'_{t+1}} \tag{4.3}$$

where P_0 denotes the stochastic dynamics in Equation (4.1) without an active control. This likelihood cannot be directly evaluated as $\boldsymbol{z}$ is not observable. Merging $\boldsymbol{z}$ and $\boldsymbol{\theta}$ for estimating a joint variable is viable for fitting the likelihood. However, this might result in a nontrivial partition function evaluation because of the general form of $\mathcal{J}$. On the other hand, it would be beneficial to disentangle $\boldsymbol{z}$ from the unknown parameters for the efficiency of recognizing a given trajectory ($p(\boldsymbol{z}|\varsigma)$) and synthesizing motion of a specified mode ($p(\varsigma|\boldsymbol{z})$).

4.2.2 Our Approach

This chapter takes a divide-and-conquer strategy to approach the problem. It is based on the results in Chapter 3, where one or a set of simple quadratic cost-to-go function can be used to model locally consistent demonstrations. The problem then boils down to grouping trajectories of the same mode. Clustering-based preprocessing is an option to achieve this, for which the cost-to-go functions themselves serve as a natural metric: a pair of trajectories are similar if both of them are locally optimal with respect to quadratic cost-to-go functions. For sake of efficiency, the proposed approach develops an ensemble method. The key idea is to randomly group trajectories in a suboptimal while quite efficient way. An improved performance is then obtained by aggregating a set of such "naive" models.

The followings of this chapter first develop the IOC result under the weak quadratic cost-to-go function. Random subspace embedding is then employed to realize the suggested trajectory grouping. The chapter continues with the incorporation of human kinematics features and latent dynamics. These extensions target practical applications about synthesizing handwriting and inferring motion intention, both of which are correlated to modeling and reasoning about the latent behavior modes.

4.3 Quadratic Cost Learning under a Linearly-solvable System

Let the discrete $\boldsymbol{z}$ be considered as a known variable. In that sense, it can be seen that the integral of the denominator in Equation 4.3 can be efficiently evaluated if $\mathcal{J}$ is of a quadratic form. A quadratic cost-to-go implies that the demonstrations, which quantitatively expect a low entropy Gaussian probabilistic model in (4.3), roughly follow a unique behavior mode. By exploiting this fact, the demonstrations labeled with the same $\boldsymbol{z}$, when factored as state pairs, can be modeled by setting $\mathcal{J}_\varsigma(\boldsymbol{x}_t, \boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Lambda}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t)$ in (4.3), yielding

$$\begin{aligned} P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{z}) &= \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}')^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_{t+1} - \boldsymbol{\mu}')}, \\ \boldsymbol{\mu}' &= \boldsymbol{\Sigma}[\boldsymbol{\Sigma}_0^{-1} f(\boldsymbol{x}_t) + \boldsymbol{\Lambda}_{t+1} \boldsymbol{\mu}_{t+1}], \\ \boldsymbol{\Sigma} &= (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{\Lambda}_{t+1})^{-1}, \end{aligned} \tag{4.4}$$

where $\boldsymbol{\Sigma}_0$ is covariance of the Gaussian noise of the passive dynamics. $\boldsymbol{\Sigma}$ is the covariance matrix, which depends on $\boldsymbol{\Lambda}_{t+1}$, and d denotes the state dimension. Therefore, the likelihood in (4.3) can be written in an explicit way, thanks to the closed-form evaluation of the integral of the product of two Gaussian functions:

$$\begin{aligned} &\int \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_0|}} e^{-\frac{1}{2}[\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t)]^T \boldsymbol{\Sigma}_0^{-1} [\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t)]} e^{-\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}_{t+1})^T \boldsymbol{\Lambda}_{t+1} (\boldsymbol{x}_{t+1} - \boldsymbol{\mu}_{t+1})} d\boldsymbol{x}_{t+1} \\ =& \frac{\sqrt{|\boldsymbol{\Lambda}_{t+1}^{-1}|}}{\sqrt{|\boldsymbol{\Lambda}_{t+1}^{-1} + \boldsymbol{\Sigma}_0|}} e^{-\frac{1}{2}[f(\boldsymbol{x}_t) - \boldsymbol{\mu}_{t+1}]^T (\boldsymbol{\Lambda}_{t+1}^{-1} + \boldsymbol{\Sigma}_0)^{-1} [f(\boldsymbol{x}_t) - \boldsymbol{\mu}_{t+1}]} \end{aligned} \tag{4.5}$$

Moreover, a maximum-entropy (MaxEnt) formulation implies a standard Gaussian distribution $\boldsymbol{x}_{t+1} \sim \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$, with the stochastic dynamics tends to be uniform with $\|\boldsymbol{\Sigma}_0\| \to \infty$. It is apparent the maximum likelihood estimation of this approximation is even more trivial. This is because $\boldsymbol{y}$ is dependent on $\boldsymbol{Z}$ in Equation 4.4 thus estimating the original $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ requires an iterative optimization. Given these observations, the MaxEnt result appears as a reasonable starting point to guess the model or decouple the parameters. In fact, such a surrogate has a following guarantee:

Proposition 1. *The optimal estimation of* $\{\boldsymbol{\mu}_t, \boldsymbol{\Lambda}_t\}$ *for a MaxEnt formalization ensures a lower bound of the original likelihood (4.4) and the gap depends on* $\boldsymbol{\Sigma}_0$. *In particular, the gap decreases as* $\|\boldsymbol{\Sigma}_0\| \to \infty$.

See Appendix A.1 for the proof. The above conclusion means, if the assumption for learning quadratic cost-to-go function holds, the estimation can be efficiently performed through a MaxEnt approximation.

Note that the learning considers identifying cost-to-go functions as the local IOC problem because it is arguably more efficient than learning a cost function (See discussions about OptQ in (52)). Another advantage of having a cost-to-go function is a local controller can be immediately derived, as is shown in Section 2.3.2:

$$\boldsymbol{u}_t^* = -\boldsymbol{R}^{-1}\boldsymbol{B}\frac{\partial \mathcal{J}_{\zeta^*}(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{x}_{t+1}} \tag{4.6}$$

The quadratic cost-to-go functions can be either time-independent or time-dependent for modeling time-invariant task and finite-horizon trajectories. It is known that, for first-exit problems, the cost-to-go function corresponds to the cost function in the Bellman equation:

$$C(\boldsymbol{x}_t) = \mathcal{J}(\boldsymbol{x}_t) + \log \int P_0(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) e^{-\mathcal{J}(\boldsymbol{x}_{t+1})} d\boldsymbol{x}_{t+1} \tag{4.7}$$

In (52), the relation is suggested to be used for the inference of the cost function. This is, however, not exploited in the thesis and the focus is about the development of learning cost-to-go functions.

4.4 Ensemble IOC with a Random Subspace Embedding

The efficiency for learning a quadratic cost function based on the linear-quadratic (LQ) assumption is useful. Indeed, this fact motivates to address the original problem in two phases. In the first stage, similar trajectories are grouped to ensure the applicability of the LQ assumption. The following-up learning can then exploit the problem structure for a rapid estimation over grouped demonstrations. The grouping subroutine is expected to be cheap so the overall pipeline can save computational cost comparing with tackling the original problem.

There exist numerous clustering techniques for the preprocessing purpose. For example, a simple and rapid method such as K-means could be a possible option. However, the performance of a clustering algorithm usually relies on a proper metric characterizing the data similarity. The popular Euclidean distance

in the standard K-means might work when the task is time-invariant and the latent variable $\boldsymbol{z}$ depends on the state $\boldsymbol{x}$ ($p(\boldsymbol{z}|\boldsymbol{x})$). However, for certain tasks, the style of a demonstration might depend on the global trajectory feature ($p(\boldsymbol{z}|\varsigma)$). As discussed in Section 4.2.1, handwriting exemplifies such a kind of task. The challenge raised is that the Euclidean distance might no longer be viable for state trajectories, which are often of a high dimension. Section 4.7.3.2 will demonstrate a general trajectory task where the similarity metric is nontrivial.

Here the thesis proposes an approach which is, on one hand flexible for the dependency on both local and global features, and on the other hand, still simple and efficient for its implementation. The approach works in an iterative manner by recursively dividing the dataset. Take the trajectory grouping as the example, each iteration of the algorithm seeks to maximize the information gain from introducing a partition on the current dataset:

$$\Delta H(\mathcal{D}, \phi(\cdot)) = H(\mathcal{D}) - [H(\mathcal{D}^{\phi(\varsigma)\geq 0}) + H(\mathcal{D}^{\phi(\varsigma)<0})], \tag{4.8}$$

where H denotes the entropy of the data trajectories under a probabilistic model. $\mathcal{D}^{\phi(\varsigma)\geq 0}$ and $\mathcal{D}^{\phi(\varsigma)<0}$ are the partitioned subset based the criterion $\phi(\varsigma) = 0$. The reduction of entropy implies the partitioning reveals useful structure from the data space. Upon noting the simplicity of a Gaussian entropy, a MaxEnt model with the quadratic parameterization can be used to evaluate the entropy.. ϕ defines the function to decide the membership of each demonstration. For an efficient searching, this function is often constrained with a simple form. Existing research (43) provides popular options to obtain decision boundaries of different levels of complexity. The optimization in searching ϕ can be further relaxed by randomly selecting the effective features and the candidate solutions, as is suggested in (62). Among these options, the thesis employs a naive form, letting $\phi(\varsigma) = \varsigma_{t,l} - \eta$ where $\varsigma_{t,l}$ denotes the l-th dimension of the t-th state $\boldsymbol{x}_t$ in trajectory ς. η is the intercept to be decided together with t and l in the random search. This in fact explores in a family of axis-aligned decision boundaries in the temporal and spatial space of the trajectories.

The above process can be performed recursively to obtain K subsets, as is demonstrated in Figure 4.2. The recursive process can be terminated when the dividing violates the constraints of the minimum number of demonstrations

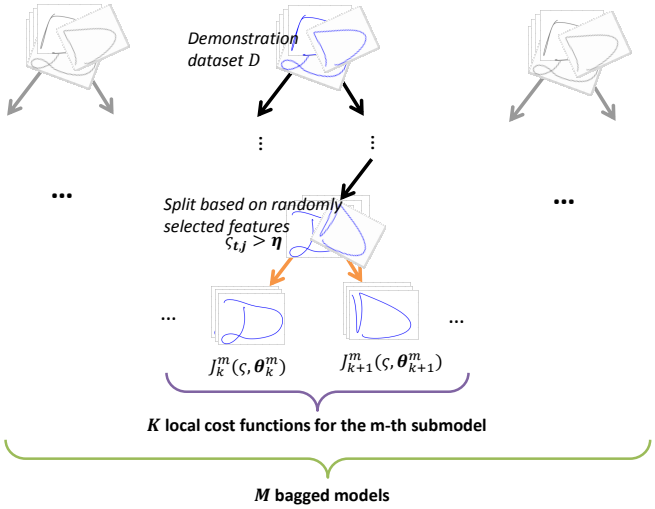


Figure 4.2: An ensemble of cost-to-go functions over partitioned datasets through random feature bagging. The demonstrations are grouped according to suboptimal yet efficient decisions, resulting in trajectories with consistent styles so that a simple IOC model is plausible.

N_D^{min} in the subsets. By randomly searching in a constrained parameter space, the formation of partitions is efficient and effective in grouping demonstrations with a similar style (low entropy distribution). The pseudocode for this recursive partitioning subroutine is given as Algorithm 3. The algorithm returns K subsets $\mathcal{D}_{k=1:K}$ taking as input the complete demonstration set $\mathcal{D}$. Further explanation about the other parameters will be given later.

Local cost-to-go functions can be estimated based on the each subset of the demonstrations as shown in (4.4). However, the estimation is unstable as the local learning depends on the results of data partitioning, which only considers the data correlation in a suboptimal way. An idea to mitigate this undesired effect is to replicate the partitioning for multiple times to build an ensemble of M models. This strategy is called bagging, which is widely accepted and applied as a scheme to reduce model variance (21). For a bagged model ensemble, there exist multiple mechanisms for generating the ultimate prediction: estimating the unknown cost-to-go function. One standard option is to take a weighted

Algorithm 3 RandomSubSpace - Partitioning dataset through feature bagging

Require: $\mathcal{D}$, N_x, N_D^{min}
Ensure: $\mathcal{D}_{k=1:K}$
  $\mathcal{D}_{k=1:K} \leftarrow \textsc{Split}(\mathcal{D}, N_x, N_D^{min})$
  function $\textsc{Split}(\mathcal{D}_{in}, N_x, N_D^{min})$
    $\{\varsigma_{t,l}^i\}_{i=1:N_x} \leftarrow \text{RandomSelect}(\varsigma)$
    $j, \eta_j^* \leftarrow \underset{i,\eta_i}{\text{argmax}}\, \Delta H(\mathcal{D}_{in}, \{\varsigma_{t,l}^i\}, \eta_i)$
    if $|\mathcal{D}_{in}^{\varsigma_{t,l}^j \geqslant \eta_j^*}| \geqslant N_D^{min}$ and $|\mathcal{D}_{in}^{\varsigma_{t,l}^j < \eta_j^*}| \geqslant N_D^{min}$ **then return** Concatenate($\textsc{Split}(\mathcal{D}_{in}^{\varsigma_{t,l}^j \geqslant \eta_j^*})$, $\textsc{Split}(\mathcal{D}_{in}^{\varsigma_{t,l}^j < \eta_j^*})$)
    else return $\mathcal{D}_{in}$ ▷ Discard this split
    end if
  end function

log-sum over local predictions with a similar form as (218):

$$\mathcal{J}^*(\boldsymbol{x}) \approx -\log \sum_{m=1}^{M} \sum_{k=1}^{K_m} w_k^m e^{-\mathcal{J}_k^m(\boldsymbol{x})} \tag{4.9}$$

where $\mathcal{J}^*$ is the target cost-to-go approximated by the ensemble of quadratic $\{\mathcal{J}_k^m\}$. The state trajectory ς was omitted and m indexes the models in the ensemble. $\{w_k^m\}$ denotes the weight of each local model (4.4). The weights can be defined as $\{w_k^m\} = \{\frac{card(\mathcal{D}_k^m)}{card(\mathcal{D})M}\}$, with $card(\cdot)$ denoting the cardinality of dataset.

The above ensemble strategy resembles a mixture of multiple simple probabilistic IOC models. The indices of $\{m, k\}$ can be understood as discrete latent variables, which loosely corresponds to trajectory styles s. It can be seen that, the number of subsets is a partially controlled value from the random partitioning. In some cases, one might like this value to be deterministic, e.g., when the number of clusters is known. In fact, the above random partitioning result is flexible to be used to enforce this model prior. To see this, one can consider the memberships of all subsets as a one-hot encoding of the data. In that sense, the random partitioning embeds the original data into a manifold, yielding a high dimensional but sparse representation. Thus, the result of random partitioning can also be used as a random trees embedding (62), which hashes the input features and constructs a non-Euclidean affinity matrix. Applying the affinity to standard techniques like K-means or spectral learning, the trajectories can be assigned into a given number of clusters with a nonlinear feature embedding.

With the cost-to-go functions estimated, the control synthesis can be retrieved through standard backward passing or solving an invariant point problem. For instance, under a finite horizon LQR condition, the backward Ricatti iteration allows for efficiently deriving the reference trajectory together with the local feedback gain, in a similar way as Section 3.3.

For a further understanding of the above model, it is also worth remarking its relation to other approaches:

– One way to explain the cost evaluation (4.9) is to see it as a soft version of pointwise minimum of a collection of cost-to-go functions. With such an evaluation, (4.6) yields:

$$\begin{aligned} \boldsymbol{u}_t^* &= -\boldsymbol{R}^{-1}\boldsymbol{B}\frac{\partial \mathcal{J}_{\varsigma^*}(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{x}_{t+1}} = \\ &-\sum_{m,k}\Big[\frac{w_k^m e^{-\mathcal{J}_k^m(\boldsymbol{x}_{t+1})}}{\sum\limits_{m',k'} w_{k'}^{m'} e^{-\mathcal{J}_{k'}^{m'}(\boldsymbol{x}_{t+1})}}\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{\Lambda}_k^m(\boldsymbol{x}_{t+1}-\boldsymbol{\mu}_k^m)\Big] \end{aligned} \tag{4.10}$$

The control can thus be explained as a combination of state dependent local impedance controllers, which are analogous to the form proposed in (96). The thesis, however, will adopt another type of control based on the most probable cost-to-go model.

– As another way, the local cost-to-go models depending on $\boldsymbol{z}$ encode different potential action modes that are applicable to the task. If the model weights $\{w_k^m\}$ can be adaptively estimated, the most plausible mode $\boldsymbol{z}$ can be inferred with certain decision-making mechanisms, such as $\boldsymbol{z}^* = \max_{\boldsymbol{z}} p(\boldsymbol{z}|\varsigma)$. This observation offers the possibility of trajectory adaptation in face of unmodeled disturbances. See Section 4.6.

– GMM can be cast as a special case of the ensemble with a MaxEnt assumption (4.4). Hence this framework can interpret GMM from the inverse optimal control perspective. Actually, the framework extends the standard GMM by enforcing the passive dynamics, which is arguably important for physical plausibility (52). Conversely, the connection to GMM implies a possible model parameter refinement through the expectation-maximization iteration though this is not formally explored here.

The complete learning algorithm is presented as Algorithm 4. The algorithm receives demonstrations and parameters for both global trajectory clustering

and local state partitioning. The partitions are used to obtain an approximated MaxEnt estimation of parameters $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Lambda}}$, as well as the partition weights w_k^m. The parameterized model Equation (4.4) can then be used to evaluate the data membership to each local model:

$$\mathbb{I}_k^m(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t) = \frac{w_k^m P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \hat{\boldsymbol{\mu}}_k^m, \hat{\boldsymbol{\Lambda}}_k^m)}{\sum_{k'=1}^{K_m} w_{k'}^m P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \hat{\boldsymbol{\mu}}_{k'}^m, \hat{\boldsymbol{\Lambda}}_{k'}^m)} \tag{4.11}$$

In turn, the new parameters for each local model are solved the MaxEnt relaxation of likelihood (4.4), with $\mathbb{I}(\cdot)$ as the weight of data.

The algorithm relies on a few arguments to trade-off the modeling power and the computational overhead. M_ς and M_x denote the number of aggregated models in the ensemble. Like other randomized methods, the performance of model ensemble improves monotonically as the ensemble size grows (21). N_ς and N_x define the number of features that are involved to decide a split (Also see Algorithm 3). N_D^{min} specifies the minimum size of a set for the next split. These arguments can be modulated to control the model complexity. A practical way of choosing N_ς or N_x is to take the square of the feature dimension (62). Intuitively, a smaller N_D^{min} leads to finer partitioning, implying a reduced bias while an increased variance and computational cost.

4.5 Cost Parameterization with Human Kinematics Features

The Algorithm 4 estimates a cost-to-go function over the original trajectory feature. Similar to Section 3.4, the function can be reparameterized to incorporate priors about the trajectory formation. Specifically, this section considers embedding character trajectories into a representation inspired from the log-normal model, which is based on the research of natural human movement (160, 155).

The log-normal model is based on the observation that the velocity magnitude of human motion stroke is of a asymmetric bell shape. It is shown that the shape can be described by a Gaussian function over the logarithmically transformed time index. A further assumption is that the path curvature remains constant within one stroke. Specifically, for a planar motion, the trajectory pro-

Algorithm 4 Learning - Learning cost-to-go ensembles from demonstrations

Require: $\mathcal{D} = \{\varsigma^i\}$, M_ς, M_x, N_ς, N_x, N_D^{min}, M(optional)
Ensure: $\mathcal{D}_{m=1:M}$, $\boldsymbol{\theta}_k^m$, $k = 1, ..., K_m$, $m = 1, ..., M$
  $\mathcal{D}_{m=1:M} \leftarrow$ RandomSubSpace($\mathcal{D}, N_\varsigma, N_D^{min}$) with M_ς model ensemble
  for all m in 1:M **do**
    $\mathcal{D}^x \leftarrow$ StatePairs($\mathcal{D}_m$)
    $\mathcal{D}^x_{k=1:K_m} \leftarrow$ RandomSubSpace($\mathcal{D}^x, N_x, N_D^{min}$) with M_x model ensemble
    for all k in 1:K_m **do**
      $\hat{\mu}_k^m, \hat{\Lambda}_k^m \leftarrow \underset{\boldsymbol{\theta}}{\text{argmax}} \sum_{i=1}^{|\mathcal{D}_k^x|} \log P_{MaxEnt}(\boldsymbol{x}^i|\boldsymbol{\theta})$
      $w_k^m \leftarrow \frac{|\mathcal{D}_k^m|}{|\mathcal{D}|}$
    end for
    for all $\{\boldsymbol{x}_{t+1}, \boldsymbol{x}_t\}$ in $\mathcal{D}^x$ **do**
      $\hat{\mathbb{I}}_k^m(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t) \leftarrow w_k^m P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mu_k^m, \Lambda_k^m)$ ▷ Membership of data to each partition under the MaxEnt approximation
    end for
    $\mathbb{I}_k^m(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t) \leftarrow$ Normalize($\hat{\mathbb{I}}_k^m(\boldsymbol{x}_{t+1}), \boldsymbol{x}_t$
    for all k in 1:K_m **do**
      $\mu_k^m, \Lambda_k^m \leftarrow \underset{\boldsymbol{\theta}}{\text{argmax}} \sum_{i=1}^{|\mathcal{D}^x|} \mathbb{I}_k^m(\boldsymbol{x}_{t+1}^i, \boldsymbol{x}_t^i) \log P_{MaxEnt}(\boldsymbol{x}^i|\boldsymbol{\theta})$ ▷ Approximately solving (7) with the data weight $\mathbb{I}(\cdot)$
    end for
  end for
  $\boldsymbol{\theta}_k^m \leftarrow \{\frac{w_k^m}{M}, \mu_k^m, \Lambda_k^m\}$

file is reconstructed from velocity and angular position, which are respectively calculated as:

$$\varsigma_t = \sum_{j=1}^{N} |v(t)| \begin{bmatrix} \cos(\phi_j(t)) \\ \sin(\phi_j(t)) \end{bmatrix} \tag{4.12}$$

$$|v(z)| = \sum_{j=1}^{N} \frac{A_j}{\sqrt{2\pi}\sigma_j(t - t_0^j)} \exp(-\frac{(\ln(t - t_0^j) - \mu_j)^2}{2\sigma_j^2}) \tag{4.13}$$

$$\phi_j(z) = \alpha_s^j + \frac{\alpha_e^j - \alpha_s^j}{2}(1 + \text{erf}(\frac{\ln(t - t_0^j) - \mu_j}{2\sigma_j})) \tag{4.14}$$

where the time index is generalized as the phase variable t. The velocity profile is estimated by combining N log-normal models, as show in (4.13). Similar to the radial basis function approximators, t_0^j and μ define the location of the impulse and σ_j defines the basis function width. The angular positions can be revealed by interpolating between the start and end positions α_s^j and α_e^j.. According to the constant curvature assumption, the angular displacement depends on the integral of the log-normal function, resulting in the Gaussian error function erf($\cdot$). Figure 4.3 depicts the velocity profile $v(t)$ for each log-normal stroke.

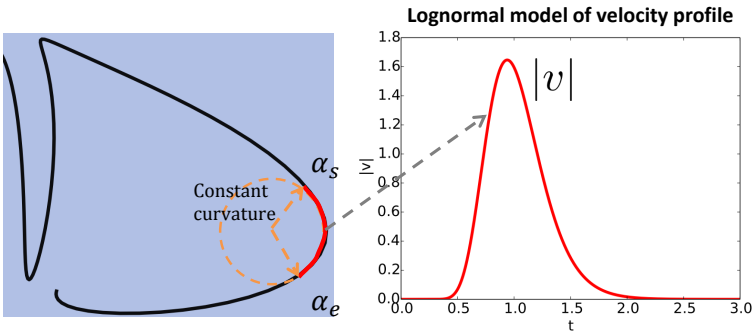


Figure 4.3: Modeling handwriting motion with curvature and lognormal velocity profile: the trajectory section is parameterized with a bell-shaped velocity magnitude and a constant curvature.

It is easy to see that certain model parameters shape the resulting motion in an interpretable way. For instance, A affects the velocity magnitude; α_s and α_e impact the stroke alignment and straightness. Thus the original local cost-to-go $\mathcal{J}^k(\varsigma)$ can be re-parameterized with respect to model parameters $\hat{\varsigma} = \{A^k, z_0^k, \mu^k, \sigma^k, \alpha_s^k, \alpha_e^k\}$.

Embedding a trajectory into the model parameter space is achieved through the *RXZERO* estimation (155). This routine first roughly segments a trajectory at inflection points and fit a log-normal model for each segmentation. The initial log-normal models are recursively refined through nonlinear optimizations to minimize the reconstruction error of the velocity and position profiles. Extra log-normal models will be added or subtracted to eliminate the residual errors. The challenge of directly estimating the statistics of $\hat{\varsigma}$ is the representation of different trajectories might not be of a same length because the number of log-normal models depends on the trajectory. Here, an approach that directly converts the parameters of $\mathcal{J}^k(\varsigma, \boldsymbol{\theta})$ to the ones of $\mathcal{J}^k(\hat{\varsigma}, \hat{\boldsymbol{\theta}})$ is considered. The $\hat{\boldsymbol{\mu}}$ in $\hat{\boldsymbol{\theta}}$ is retrieved from the reference trajectory $\{\boldsymbol{x}_t\} = \boldsymbol{\mu}_t$ derived from $\boldsymbol{\theta}$. Because the trajectories are assumed to be distributed around the reference, the variability of $\hat{\varsigma}$ can be locally captured by a linear projection:

$$\begin{aligned} \boldsymbol{\Sigma}_{\hat{\varsigma}}^{k\,-1} &= (G_{\hat{\boldsymbol{\mu}}}^k)^T \boldsymbol{\Sigma}_{\varsigma}^{k\,-1} G_{\hat{\boldsymbol{\mu}}}^k \\ \boldsymbol{\Sigma}_{\varsigma}^{k\,-1} &= \mathrm{diag}(\boldsymbol{\Lambda}_0, ..., \boldsymbol{\Lambda}_T) \end{aligned} \tag{4.15}$$

where $\boldsymbol{\Sigma}_{\varsigma}^{k\,-1}$ concatenates weight parameters $\{\boldsymbol{\Lambda}_t\}$ as a block-wise diagonal ma-

trix. $G^k_{\hat{\boldsymbol{\mu}}}$ is the Jacobian matrix evaluated at $\hat{\boldsymbol{\mu}}$ that locally embeds original state variability into the kinematics parameter space.

The advantage of having an ensemble of $\mathcal{J}^k(\hat{\varsigma}, \hat{\boldsymbol{\theta}})$ instead of $\mathcal{J}^k(\varsigma, \boldsymbol{\theta})$ is that one can learn handwriting motion with features that are both human-inspired and interpretable comparing with the position coordinates. Randomly sampling handwriting motion is efficient by evaluating (4.12) and (4.14) with a perturbed $\hat{\varsigma}$. The synthesis is also constrained by the incorporated kinematics structure so the variations are expected to be human-like, as will be shown in Section 4.7.2.

4.6 Mode Inference and Adaptation

This section discusses another extension to the proposed ensemble framework. Specifically, the latent variable $\boldsymbol{z}$ is proposed to be explicitly inferred for realizing adaptive behaviors. This is different from Equation (4.6), where a fixed and known mode variable z is assumed. If a potential mode change is expected, e.g., the human operator might change his/her intention during the execution, this variable should be dynamically inferred and conditioned. To see this, consider a toy task, where the robot end-effector is perturbed when writing a letter with a certain mode. The benefit of online mode adaptation is exemplified in Figure 4.4. Specifically, a spring-like local feedback control, which always rejects the perturbations, would undermine the legibility of the letter. On the other hand, if the deviation can instead be considered as an intention altering the task mode, the perturbation can be exploited to write the letter with another plausible style.

The Equation (4.6) is also an adaptive one by integrating out the mixture of z. This is applicable if z can be fully determined from the instantaneous state $\boldsymbol{x}_t$ because $\mathcal{J}(\boldsymbol{x}_t, z)$ does not consider the performance before $\boldsymbol{x}_t$. However, this is not usually the case. As exemplified in Figure 4.5, the perturbed pen-tip is supposed to adopt a correct adaptation based on the plausible motion modes. Unfortunately, the preferred mode is ambiguous if the inference is based on the instantaneous position [1]. In this case, the adaptation needs to consider the deposited trajectory, which complements the cost-to-go function $\mathcal{J}(\boldsymbol{x}_t, z)$.

1. One can of course argue to extend the state variable with velocity information to resolve the ambiguity. This effectively also considers the motion history, though a very short one. Section 4.7.3.2 demonstrates a task involving a long-term dependency.

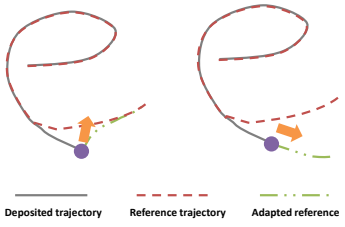


Figure 4.4: Accommodating perturbation through trajectory tracking or adjusting the reference to another mode. Local feedback control is inadequate while adapting the reference to a redundant style is desired to retain the letter legibility.

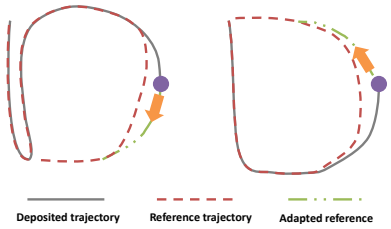


Figure 4.5: Accommodating perturbation considering the motion history. Looking at the instantaneous state (tip position) might not be sufficient to decide the motion direction and an undesired adaption might fail the writing task.

To this end, in additional to the dynamical mode inference based on $\mathcal{J}(\boldsymbol{x}_t, z)$, a prior is also introduced: the latent task mode passively evolves as a Markovian process. The goal of the prior is twofold. On one hand, it biases the estimation process to ensure a more robust inference, because in practice the state measurement inevitably suffers from sensory noises. On the other hand, the temporally propagated prior provides a compact way to accommodate global trajectory information, which is necessary if the mode is not fully determined by instantaneous state measurements.

The pipelines of mode estimation and control synthesis are schematically depicted in Figure 4.6. Here, the (unknown) state is denoted as $\boldsymbol{z} = [z^1, z^2, ..., z^M]$. Hence $\boldsymbol{z}$ is an M-dimensional vector representing the belief over all possible modes and the i-th entry is the likelihood of mode i. The evolution of the belief is modeled with a transition matrix $\boldsymbol{T}$, whose entry $\boldsymbol{T}_{ij}$ characterizes a prior possibility of switching from mode i to mode j. The learned cost-to-go functions provide evidence, evaluating the expected cost of all possible modes at the current state. Concretely, after observing $\boldsymbol{x}_{t+1}$, the mode belief $\boldsymbol{z}_{t+1}$, can be recursively inferred as:

$$\boldsymbol{z}_{t+1}(\boldsymbol{z}_t, \boldsymbol{x}_{t+1}) \propto (\boldsymbol{T}\boldsymbol{z}_t) \odot \begin{bmatrix} e^{-\mathcal{J}_1(\boldsymbol{x}_{t+1})} \\ \dots \\ e^{-\mathcal{J}_i(\boldsymbol{x}_{t+1})} \\ \dots \\ e^{-\mathcal{J}_M(\boldsymbol{x}_{t+1})} \end{bmatrix} \tag{4.16}$$

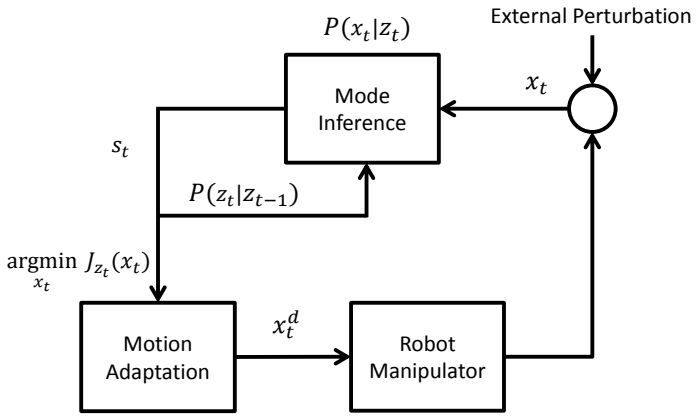


Figure 4.6: Pipelines of mode estimation and control synthesis based on learned cost-to-go functions ensemble.

where $\odot$ denotes an element-wise product.

It is easy to find that such a recursive inference works as Kalman filtering. From this perspective, the likelihood of each feasible demonstration mode is tracked as the latent state. The learned cost-to-go functions can be viewed as observational models, measuring the performance of each mode starting from the current state. Also, the latent dynamics $\boldsymbol{T}$ can be estimated by counting the occurrences of mode transition given the observation model and data. This shares similarities to learning an HMM-like model, though here the emission probability is separately learned and the distribution is nontrivial comparing with a categorical or a Gaussian one in HMM.

Note that here $\boldsymbol{T}$ is determined in an ad-hoc manner. The reason is that the latent state is understood as the trajectory mode, which is ideally invariant throughout each expert demonstration. This is conceptually different from most HMMs, whose latent state appears to be the label of a trajectory section. More importantly, the flexibility of designing $\boldsymbol{T}$ offers an intuitive way for users to shape the expected behavior, which requires a trade-off between the robustness against disturbances and responsiveness of mode adaptation. In fact, the proposed extension somehow blends Equation (4.6) and (4.10). Specifically, when $\boldsymbol{T}$ is selected as a uniform transition, the responsiveness to mode adaptation is maximized, while robustness might be compromised. The reason is that the sys-

tem will immediately adopt the new mode as long as its current state appears to be more likely with respect to the corresponding cost-to-go function. Moreover, this special case follows a multi-mode policy similar to Equation (4.10), which adapts by only considering the immediate state. On the other hand, a diagonally dominant $\boldsymbol{T}$ tends to assume an invariant the mode, unless the cost-to-go functions provide strong evidence that another mode is more plausible. In the extreme case where the diagonal entries are Dirac functions, the system, behaving like Equation (4.6), will reject any attempt of eliciting a mode adaptation, resulting in a maximized robustness.

4.7 Implementation and Results

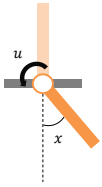
This section demonstrates the implementation of the proposed approach and extensions, as well as the obtained results. It starts with a simulated inverted pendulum task to analyze the influence of algorithm parameters and the performance in comparison to other approaches. The results of modeling latent behaviors are reported in the applications based on the two extensions. The effectiveness of incorporating human kinematics features in Section 4.5 will be demonstrated in synthesizing hardly distinguishable handwriting motions (Section 4.7.2), while the proposed motion adaptation mechanism in Section 4.6 will be examined in two robotic tasks involving human intervention (Section 4.7.3).

4.7.1 Inverted Pendulum: An Illustrative Example

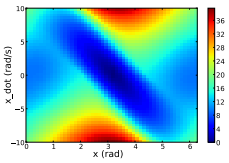
This task focuses on controlling an inverted pendulum, with the goal of applying torque u so as to let the pendulum stay upright (Figure 4.7a). The system has typical second-order dynamics, with one degree-of-freedom (DOF) and nonlinear passive dynamics. Thus the cost-to-go function is of a nontrivial form while simple enough for visualization.

The system parameters for the test are: pendulum mass $m = 1.0$kg; length $l = 0.5$m; joint damping $b = 0.1$N·m/(rad/s); gravity coefficient $g = 9.81$kg·m/s^2. The state comprises the angular position x and its derivative $\dot{x}$. A quadratic instantaneous cost function encoding the goal of control could be

$$C_{pend}(x) = \frac{1}{2}(x - \pi)^2 \tag{4.17}$$

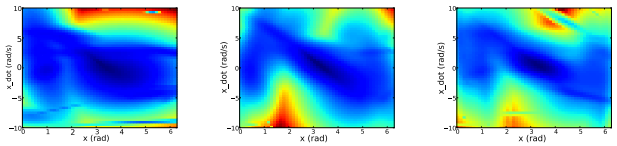


(a) Inverted Pendulum

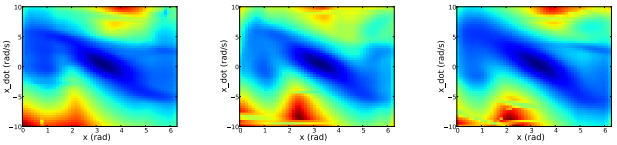


(b) Target cost-to-go

Figure 4.7: An illustrative example: inverted pendulum regulation and the optimal cost-to-go function



(a) M=5 (b) M=10 (c) M=20



(d) M=30 (e) M=50 (f) M=75

Figure 4.8: Cost evaluation of the learned ensemble models over the inverted pendulum state space: $M = \{5, 10, 20, 30, 50, 75\}$

where π denotes the target angular position in radians, indicating the upright configuration here. The optimal cost-to-go function can be derived through system discretization and standard value iteration. The control input saturates with a range imposed: $u \in$ [-5.0, 5.0]. The heat map of the underlying optimal cost-to-go is shown as Figure 4.7b.

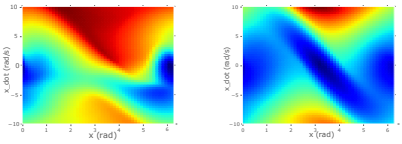
A total of 200 motion trajectories of 100 steps each, steered by the optimal cost-to-go function, are generated as demonstrations. Of these, 150 are used for sampling state-control pairs. The training dataset is corrupted by an additive noise with a standard deviation of 0.02 to simulate the sensory noise. The task for the proposed ensemble method is to determine the time invariant cost-to-go function from the demonstrations, assuming the passive dynamics $p_0(x'|x)$ are known. Also, the angular position is truncated to $[0, 2\pi]$ to ensure the Euclidean distance is properly defined, though such approximation does bias the outcome due to the bound effect. It is worth noting that the inverse problem is addressed in continuous state and control space without discretization, though the data is generated from the standard value iteration of the discretized system.

The result begins with examining the necessity of a model ensemble, whose size is controlled by the number of aggregated models. The learning results are

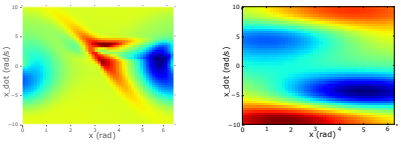
depicted throughout Figure 4.8a and 4.8f. Comparing with the target (Figure 4.7b), it can be observed that as more models are incorporated, the learning performance improves in terms of visual consistency. The observation demonstrates the anticipated advantage of model ensemble: each of the sub-models is limited due to its high sensitivity and dependence on the data partitioning (Figure 4.8a and 4.8b), while a prediction from the aggregated models leads to a better estimation than any individual model, with the overall variance significantly reduced.

For a comparison, other approaches (MaxEnt+Laplacian (126), GPIRL (127) and OptV (52)) are also applied. Two dimensions of performance, including the cost reconstruction error and training efficiency, are considered on the benchmark problem. All approaches use 64 demonstration trajectories and retrieve the estimated state value of $2,600$ test state samples. The reconstruction error is obtained as the sum of errors between the estimated value and the target cost-to-go. For algorithms that estimate a cost function (MaxEnt+Laplacian and GPIRL), the cost-to-go functions are computed based on the inferred cost function. The computation time for this additional step is not included for a fair comparison of the efficiency of original learning algorithms.

The estimated cost-to-go functions from these approaches are depicted in Figure 4.9a to 4.9d. Apparently, one of the MaxEnt setting (Figure 4.9b) shows the best qualitative results. This is expected because it learns a quadratic cost function which is consistent to the real goal. For more general cost parameterizations, such as RBFs (Figure 4.9a and 4.9d) and Gaussian process (4.9c), the recovered cost-to-go functions show some similar local geometry in certain regions but fail to capture the overall landscape comparing with Figure 4.9b and Figure 4.8f. Quantitatively, in Figure 4.10, one can observe a trend similar to the qualitative results: the reconstruction error of the ensemble method steadily decreases as more models are included. Regarding the training time, it is notable that the ensemble method is superior in terms of training speed thanks to the efficiency of learning naive local models. For the sake of comparison, the result also includes a MaxEnt version of the proposed method, which effectively works as a GMM over the demonstration state. It is not surprising to find a slight decrease in performance (in terms of sum-of-errors) since the



(a) MaxEnt+RBF (b) MaxEnt+QF



(c) GPIRL (d) OptV

Figure 4.9: Estimated cost-to-go functions from the MaxEnt (linear combination of RBF or quadratic functions), GPIRL and OptV results. An additional value iteration is performed for MaxEnt and GPIRL to visualize the cost-to-go function over the state space. OptV uses RBFs for the cost-to-go function approximation. 25 basis functions are used for all of the RBF-based approaches.

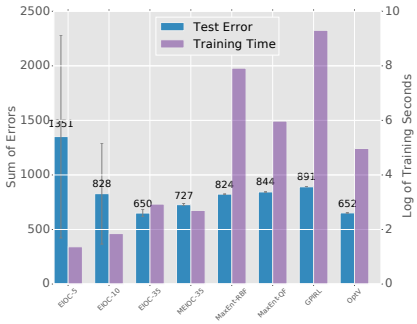


Figure 4.10: Cost-to-go function errors and training time of different approaches for the inverted pendulum problem. The proposed approach is tested by integrating different number of models in the ensemble. The MEIOC indicates the application of the approach without considering the passive dynamics (MaxEnt formulation). Note the training time is transformed to its logarithm for the visualization.

MEIOC is agnostic to the real passive dynamics model. The results for other algorithms are mixed because the visually best result (Figure 4.9b) does not lead to a smallest prediction error of the cost-to-go function values. This implies that the learning performance cannot be fully described by one metric and other dimensions need to be examined.

To have a more thorough conclusion, a policy perspective is taken in the following analysis, which examines whether the learned cost-to-go function indeed leads to behaviors that match the demonstrations. Two experiments are included with the first one focusing on the difference between the derived and demonstration trajectories, and the second one evaluating the trajectory performance under the real task cost function. Predicting the next state under the optimal policy requires a maximum posterior estimation in Equation (4.3). This boils down to a nonlinear optimization, for which the MaxEnt mean estimation is used as the prior guess to ensure the optimization performance and efficiency. The initial states of 10 test trajectories are exposed to the algorithms, seeding a

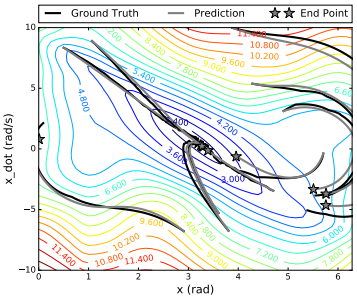


Figure 4.11: Prediction of the trajectory under the learned cost-to-go ensemble: the predicted trajectory is derived given the test initial state. The learned cost-to-go, which encodes the desirability of future state, is illustrated as the contour lines.

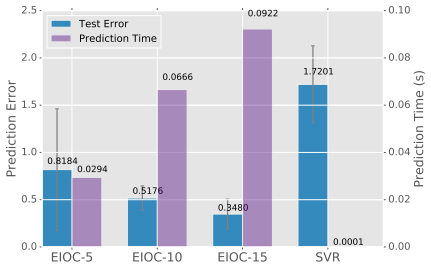


Figure 4.12: Comparing different settings with a SVR-based prediction. The regression of behavior cloning is fast for each iteration of the prediction but suffers from error cascading along the trajectory horizon.

recursive prediction of states or a trajectory optimization for the same number of steps to compare against the ground truth.

For the first experiment, the derived trajectories are visualized in Figure 4.11, where the stars denote the terminal states. It is clear that the predicted trajectories generally follow the demonstrated behavior. A quantitative result is given in Figure 4.12, where a support vector regressor (SVR) is trained as a baseline. The SVR-based prediction works as behavior cloning by predicting the next state given the current one so it is very efficient for the synthesis. Unfortunately, the accuracy of overall trajectory prediction is poor, due to the error cascading effect. The IOC-based prediction is more reliable, thanks to the bias about the future from the extracted cost-to-go. Again, the model aggregation improves the performance, while in exchange, it takes longer time to conduct the optimization when more models are integrated.

The result of the second numerical experiment is shown as Figure 4.13. Specifically, the accumulated trajectory costs are evaluated under the true cost function. The proposed ensemble approach outperforms all the other algorithms on this metric, except the MaxEnt approach with the true quadratic feature. Note that both of these two approaches achieve better performance comparing with the test trajectories themselves. This is because the test trajectories are obtained from a more limited action set due to the discretization, while the IOC

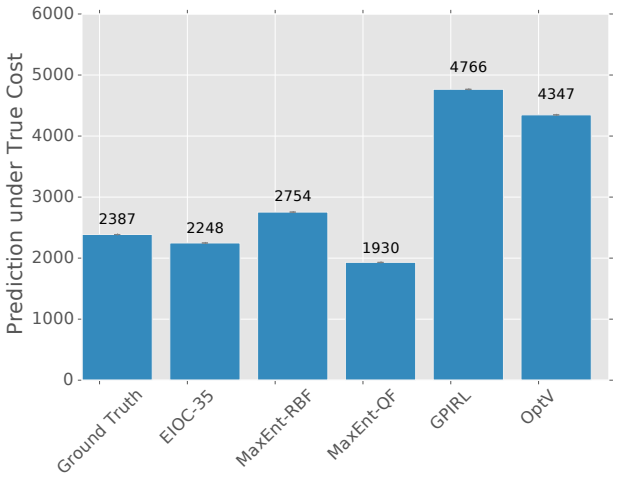


Figure 4.13: The performance of the predicted trajectories under the true cost function: comparing test trajectories and the results obtained from ensemble method, MaxEnt, GPIRL and OptV.

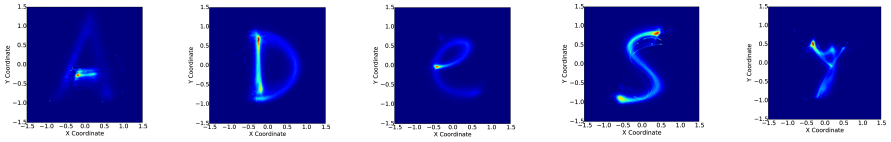
algorithms use continuous optimization to derive trajectories under the learned cost or cost-to-go functions.

4.7.2 Synthesis of Multi-mode Handwriting Motion

The success of learning latent behavior modes can be demonstrated in a synthesis task. Indeed, the quality and diversity of the generated samples depend on if the model ensemble correctly identifies and captures the demonstration modes. Here the synthesis is about dynamical handwriting motion, for which the cost parametrization based on the log-normal model is used.

4.7.2.1 Learning and Synthesizing Handwriting based on Human Data

The dataset used is the UJI Pen Characters repository (132). This repository contains online handwriting samples collected from 60 adult subjects, who could write in many different styles. Alphabetical instances with either single or multiple strokes are considered. Each stroke letter stroke is learned independently. Yet this is by no means true as the strokes are correlated temporally and is possible to be captured by introducing extra conditional models (118). The independence assumption is adopted here to focus on the ensemble method



(a) "A" (b) "D" (c) "e" (d) "S" (e) "y"

Figure 4.14: Illustration of the learned ensembles that encapsulate the patterns of character profile. This is demonstrated in the Cartesian space but not the log-normal feature space for the illustrative purpose. The statistics of the curvature-based features is captured by taking samples and convert them to the original planar Cartesian space. The heat value of a point in the Cartesian space is evaluated by folding the learned cost function along the time horizon and counting the occurrences of the coordinates in the trajectory samples.

itself, and such simplification turned out to work well in practice to synthesize reasonable motion trajectories. The results are depicted as Figure 4.14a to 4.14e. The most obvious observation is that the learned models successfully capture the legible shapes for either single or multiple-stroke characters. The variabilities of the heating magnitude can be explained by the inconsistency of forming the specific letter sections. For some strokes, human behavior tends to be comparatively consistent, such as the short straight strokes in Figure 4.14a, and 4.14b or the overall shape of "S" in 4.14d. The variability of this consistency implies multiple modes in writing a specific letter. The encodement of such diversity can be best illustrated as Figure 4.14e, which explicitly resembles the superimposition of two distinctive ways of forming a legible "y" in the Cartesian space. Note that the number of these patterns is not explicitly enforced beforehand but emerged from the ensemble of models which assign cost-to-go functions on random subsets of data.

The diversity of encoded motion patterns can be further demonstrated by synthesizing letter instances from the learned models. Shown here are a few typical sampling results, again for either single or multiple strokes, as Figure 4.15. The synthesis samples illustrate rich writing patterns that are diversified in the aspects of size, orientation, and most importantly, the style. For instance, the "d" that is constituted by a circle and a straight stroke, are successfully detected and encoded. Interestingly, the incorporation of log-normal features supports generating poorly written characters. Intuitively, a sample of

Figure 4.15: Synthesized motion samples from the learned cost ensemble models for different characters. The diverse modes and styles illustrate the multi-modal motion patterns encoded by the aggregation of simple cost functions

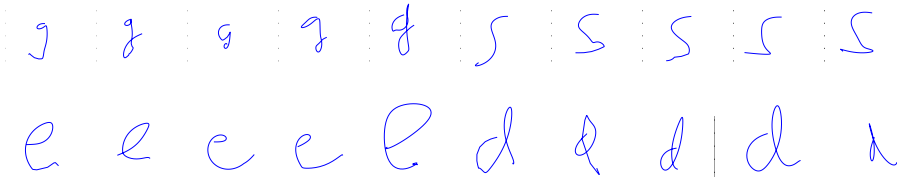


Figure 4.16: Synthesized motion of poor written samples by sampling from the learned model with random perturbations. The deformities can be intuitively controlled by modulating the local proportion, alignment and curvature of a specific component, as well as the continuity between the components.

$\hat{\boldsymbol{\theta}}$ that significantly deviates from $\hat{\boldsymbol{\mu}}$ would result in symbols that are different from demonstrations, while with the deformation constrained by the incorporated feature. This is realized by perturbing the distribution parameters with an increased noise. Figure 4.16 shows synthesized samples, which resembles various types of deformities such as inappropriate component proportion, misalignment or jerkiness in stroke transition. This demonstrates the potential of the framework to generate various good or poor handwriting motion. These results are applied in implementing human-robot interaction activities, where the children imitate and correct the letters generated by the algorithm and written by a robot. Refer to (31) for more details.

4.7.2.2 Evaluating the Human-likeness of the Synthesized Motion

One question remains to answer regarding the handwriting synthesis is that: how can one assess the quality of synthesized samples and as such be convinced that the behaviors are successfully modeled. This correlates to evaluating the similarity between samples from an unsupervised learning model and the training data. Qualitative results like Figure 4.15 are usually used as evidences because a unified metric is absent in general. In order to obtain a quantitative result, an online user study is run to examine how humans perceive the synthesized motion. Due to the obscurity of "human-likeness", the presented study

was performed in a form of Turing-like test, where the participants were presented with a mixture of human and artificial dynamic motion, without showing the physical body of both the robot or of the human. The participants were instructed to choose among these motion samples the one they believe was generated by the algorithm. Besides the rate of correct prediction, another interesting dimension that could be measured is the confidence of the humans on their decisions, serving as a fidelity measurement from the subjective perspective.

A – Study Hypothesis

H1. *By observing the dynamic motion of the characters, the participants cannot distinguish between the agent synthetic and human written character samples. The classification performance is close to a random guess.* It is expected the samples from learned ensemble models possess believable variabilities that are consistent with natural human handwriting. Thus most sampled motion parameters should result in characters which are hard to be identified from the mix up of synthesized and human samples. Quantitatively, this hypothesis implies an equivalence which can be numerically expressed as

$$\|\hat{c} - c\| \leq \delta \tag{4.18}$$

where $\hat{c}$ and c denote the classification performance from the experiment estimation and the random guess respectively. δ is a threshold quantifying the equivalence of the two tested values. The selection of δ will be presented in the results and analysis section.

H2. *Participants will not detain high confidence levels towards their choice.* This hypothesis checks the indistinguishability from a subjective perspective of the humans. It is expected to see the quantified confidence is lower than a certain level. It will also be interesting to examine the relation between the human confidence and concrete performance.

B – Study Procedure

The Turing-like test was carried out in the form of an online questionnaire. Concretely, the participants were instructed to evaluate 20 characters, containing

both synthesized and human handwritten ones, by accessing web pages anonymously. They were explicitly instructed that there was only one synthesized sample for each character question. They could neither skip character pages nor browse back to the past ones to modify the previous responses. Their evaluation was based on two questions for each character:

Q1. *Which letter do you believe is written by a robot*?

To answer to this question, participants were presented with five dynamic handwriting motion for the character. The animation could be intuitively resumed or stopped by moving the cursor on or off the images. The participants were allowed to replay the motion as many times as they wanted before they made the decisions.

Q2. *How confident are you about your choice*?

The second question could be answered in a five-point type-Likert scale ranging from 1 to 5: 1-very low; 2-low; 3-neutral; 4-high; 5-very high

The sequences of characters and answer options were randomized to counter balance ordering effects. Moreover, demographic information was also collected. Participants were notified not to respond the questionnaire for multiple times at the beginning of the web page. Each individual questionnaire took about 10 to 15 minutes to complete.

C – Study Analysis and Results

The participants were recruited through the mailing lists within a university. A total of 68 participants completed the online questionnaire. The sample ranges from 18 to 60 years old ($M = 28.7$; $SD = 8.7$).

In order to test **H1**, the threshold $c = 0.2$ is chosen as there were five options in each character question. δ is defined according to the deviation of random classification performance $\delta = \sigma \approx 0.089$, if the number of correct classification is subject to a Binomial distribution. The analysis shows that on average the participants achieve $\hat{c} = 0.226 \pm 0.086$ classification performance, which is close to the random guess $c = 0.2$. A further analysis show that the null hypotheses of **H1**, $\hat{c} > c+\delta$ and $\hat{c} < c-\delta$, are both rejected by the corresponding

one-sided t-test ($t_1(67) = 6.04$; $t_2(67) = 11.03$; $p < 0.01$). Therefore, the results show statistically significant equivalence between the performance from empirical data and a theoretical value from a random guess, thus **H1** is strongly supported which suggests that participants were not able to distinguish between the character motion (synthesized versus human handwritten), wherein their choices translate the same as the random guess.

For **H2**, the averaged confidence level is 2.71 ± 0.70. One sided t-test concludes that this value is significantly below the neutral confidence level [$t(67) = 3.38$; $p < 0.01$], which also supports **H2**. Note that there is indeed a small fraction of participants who exhibit high confidence levels, however, analysis shows that such high confidence is not necessarily related to a good classification performance. A selection of the performance and confidence for the most contrasting results regarding the selected characters are shown in Figure 4.17, where it is obvious that the confidence levels are relatively consistent across characters and are not complying the performance trend. Also examined is the confidence level associated to correct answers. The level turned out to be 2.71 ± 0.98, which is not significantly different from the overall confidence level (considering a threshold of 0.2; $t_1 = 4.63; t_2 = 3.79; p < 0.01$). A further analysis yields a rather weak Pearson's correlation ($\rho = 0.126$) between the performance and confidence level. Therefore the participants are indeed uncertain about their answers, even for the ones that happen to be correct.

To sum up, these results demonstrate the capability of the algorithm for generating hardly distinguishable handwriting motions, which in turn implies the success of apprehending rich data modes stemming from natural human handwriting with multiple styles.

4.7.3 Motion Adaptation based on Mode Inference

This section exploits the model to reason about the real-time sensory input, to estimate the desired task mode so as to realize adaption under execution uncertainties.

4.7.3.1 Handwriting Motion Adaptation

The goal of this task is to extend the result of encoding multiple handwriting styles with the adaption mechanism proposed in Section 4.6. The robot

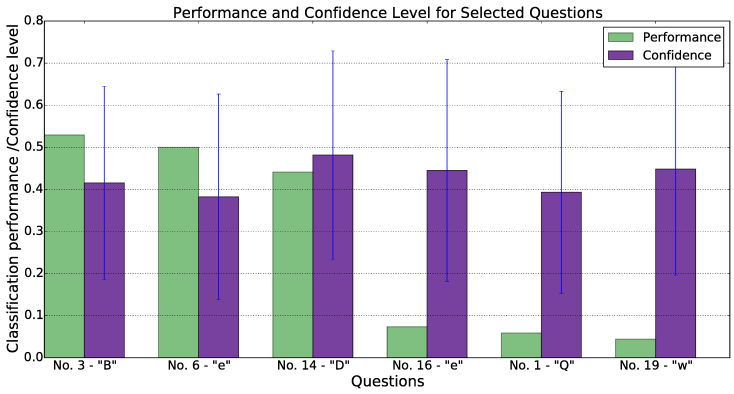


Figure 4.17: Classification performance and confidence levels for the selected characters on which the participants performed best and worst. The characters are sorted according to the performance, while the confidence levels are comparatively consistent. The overall performance 0.226 ± 0.086 is close to the random guess ($p < 0.01$).

acquires redundant ways of writing the target letter from the ensemble model. This knowledge is exploited to assess and modulate the task execution. As a consequence, the synthesized handwriting motion is implemented on a real robot and the writing style could be altered to accommodate disturbances, e.g., a human intervention.

The framework is exemplified on an ensemble model which learns a set of 120 planar trajectories of the letter "D", with two replications for each of the 60 people. The ensemble parameters were set to allow a maximum of 240 local models as we are not certain about how many styles are there in the demonstrations. The robot, a 7-DOFs KUKA IIWA manipulator, is used to follow the commanded trajectory, which is initially sampled from the learned model ensemble.

Figure 4.18 showcases the expected behavior. Specifically, the robot follows the initial mode that deposits a downward stroke at first, and plans to finish writing on the top of the canvas (Figure 4.18e). Then a human subject intervenes, making the compliant robot motion yield to moving upwards instead of following the planned direction. As a result, the perturbation elicits the alternation to other task modes, as depicted in the mixture of letter profiles in Figure 4.18f. These modes are regarded as more probable ones, which jointly

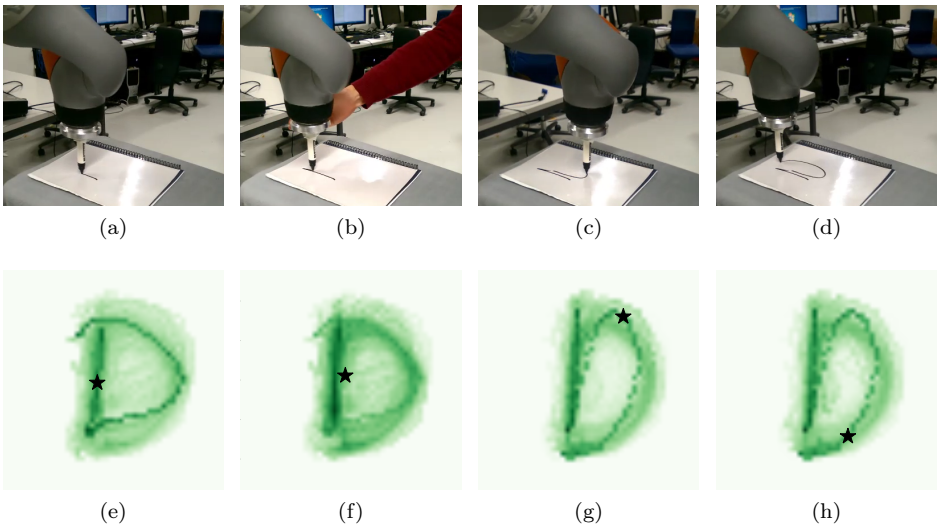


Figure 4.18: Adapting the motion of writing a "D" on a KUKA IIWA 7-DOFs manipulator. The lightness of the reference trajectories indicates the associated mode weights and the star marks the current regulating point. Under the human intervention, the task mode shifts to the alternative modes that are plausible w.r.t. the deposited trajectory and future cost. The online adapted writing motion yields a different letter profile comparing with the original intention.

consider the history (the downward stroke) and the probable future motion styles. The mode estimation proceeds with the shifted mode reinforced and finally resembles an adapted written letter, which retains the legibility under the perturbation (Figures 4.18c and 4.18d).

As a descriptive experiment, the above process shows the evolution of mode estimation serves as a compact dynamical encoding of the latent letter style, which may change subject to the human intervention. This is necessary as the position state itself is not sufficient to determine the motion, because the velocities might be conflicting at a same position for different writing styles. Here, the instantaneous position helps to decide which trajectory mode will cost less if the subsequent writing departs from the current state. Therefore, the learned cost-to-go representation enables the robot to evaluate, comply and, as such, exploit a perturbation when there exist potential modes that turn out to be suitable with the future steps taken into account.

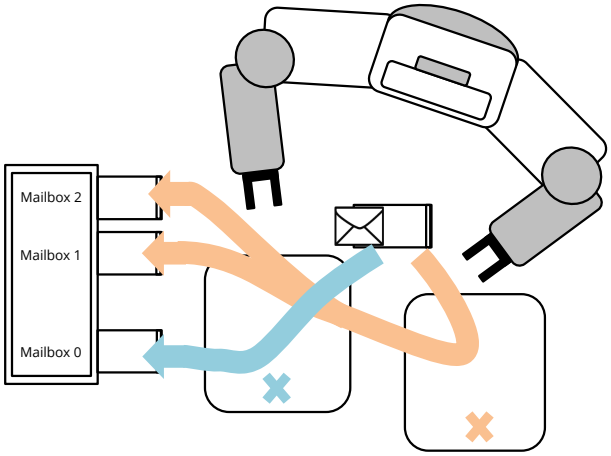


Figure 4.19: Assisting in a mail delivery task. The robot needs to learn multi-mode behavior that manipulates the mail to different target boxes. The validity of the targets depends on which path was taken in the intermediate step.

4.7.3.2 Assisting Mail Delivery

This section envisions the application of the framework in a more general scenario: a mail delivery task, where a robot assists in picking, transporting and delivering mail to different target mailboxes (Figure 4.19). In this task, the mail messages are supposed to go via specific locations in the workspace (marked by colored crosses in the figure), for a hypothetical intermediate processing—such as stamping or labeling mails with different priorities. The delivery target depends on the spots by which the mail has passed. Moreover, during the execution, humans may intervene through a physical interaction. The robot, on the other hand, should decide if it will adapt its motion to collaborate the human intervention, or insist on its current motion plan.

A – Experimental Setup

The task is carried out on a Baxter robot platform, with the setup illustrated in Figure 4.20. The AR trackers are used to label the reference frames that might be relevant to specific task modes. The poses of these frames are estimated through a camera. The locations of these interested frames are defined as the task configuration. 12 demonstrations are recorded through kinesthetic

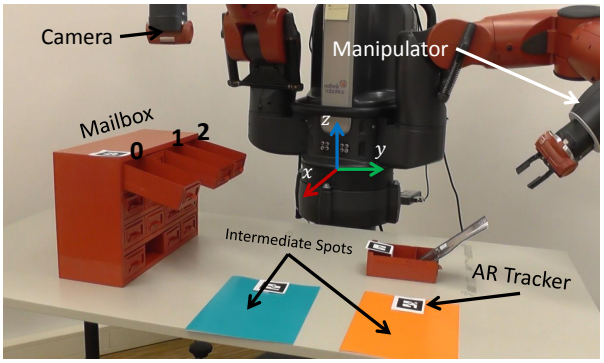


Figure 4.20: Setup for the mail delivery task: the candidate objects/frames (mailbox, cyan/orange regions and mail location) are labeled by AR trackers, which can in turn be detected by a mono-camera at the right wrist of Baxter. The left arm is used for manipulation.

teaching, with four replications for each mode. Three task modes correspond to motion trajectories via different landmarks:

- {mail location, cyan area, mailbox-0};
- {mail location, orange area, mailbox-1};
- {mail location, orange area, mailbox-2}.

Note that the constraints of the sequence modes, e.g., which area should pass and then which mailbox to deliver to, are unknown to the robot. Humans can only program them through demonstrations. For each demonstration, the locations of the scene objects are rearranged, but the aforementioned sequences are always followed. The recorded states have a dimensionality of 18, with the position in each reference frame and the time index included. The trajectories are clustered with a random embedding from $1,000$ ensemble trees. For each extracted mode, an ensemble of 10 models under a finite horizon formulation are trained, and the resulting models are used to infer the task mode and derive the command for the next step. Except the baseline methods, a latent transition dynamics

$$\boldsymbol{T} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 \\ 0.0 & 0.2 & 0.8 \end{bmatrix}$$

is used throughout all the experiment sessions. Such a latent dynamics represents a prior knowledge that the motion mode tends to keep constant, although there is a moderate possibility to switch between mode 1 and mode 2.

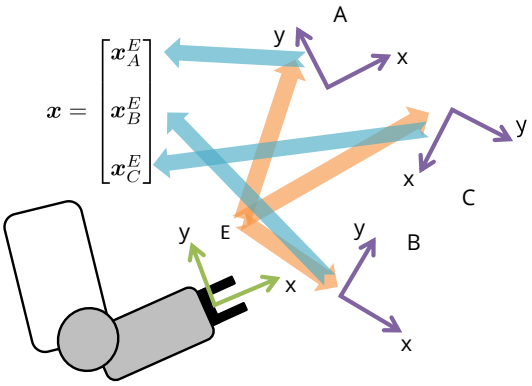


Figure 4.21: An illustration of the task-parameterized representation: the interested state, e.g., the pose of the robot end-effector E, is projected into different reference frames in the scene. The resulting state is an augmentation of all relative representations, yielding a high-dimensional state variable.

B – Task Goal and Task-parameterized Feature

The learning goal of this task is to encode constraints regarding both the static environment configuration and the process dynamics. On one hand, the robot needs to extract important task-relevant landmarks in order to adapt the synthesis for a general environment configuration (e.g., untrained locations of mailboxes and intermediate via-points). On the other hand, constraints about the task dynamics also need to be conveyed in the form of cost-to-go function learning. It is critical for the robot to exploit this knowledge to evaluate and react to the deviations, which can source from the motor noise or human intervention. In a nutshell, the robot should resist the deviation when it is due to the motion noise or a human intervention that violates the task constraints, while adapt to human intended motion when it is compatible to the task constraints. Notably, here the constraints stem from the trajectory history—namely, which via point has been passed through. This implies that the adaptation cannot be exercised based on static or time invariant observations.

In order to generalize to different static configurations, the quadratic cost-to-go function is generalized to incorporate a task-parameterized representation (29). The representation augments the interested state with representations in different reference frames of the task scenario. For instance, as illustrated

in Figure 4.21, the interested robot end-effector pose could be represented in different reference frames, such as A, B and C in the scene. The final state is the augmentation of these local descriptions thus is of a higher dimension than the original pose. A task-parameterized feature encapsulates the information relative to landmarks that are potentially important to the task execution, as such supports the generalization under an unseen arrangement of the landmark configuration. (29) uses this representation to obtain a task-parameterized Gaussian Mixture Model (TPGMM). Here the representation is used under the proposed IOC framework. Specifically, the model learns a varying quadratic cost-to-go function over this representation:

$$J(\boldsymbol{x}_t, \boldsymbol{\theta}_t) = \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Lambda}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t) \tag{4.19}$$

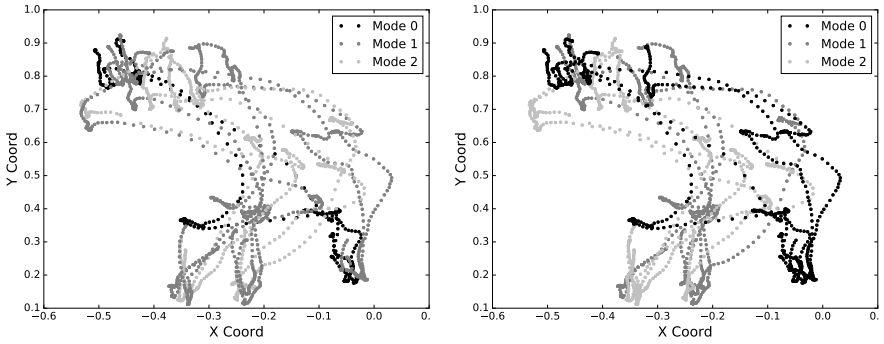
with $\boldsymbol{x}_t$ denoting the concatenate state similar to Figure 4.21. Note here $\boldsymbol{\Lambda}_t$ is block diagonal to factorize the cost with respect to landmark reference frames and impose a model sparsity to fit finite demonstrations.

The parameters vary because the importance of the via-points and destinations is not static. The inference of model parameters is compatible to TPGMM because the local models are also Gaussian. For the detailed Gaussian inference with a task-parameterized model, interested readers might refer to (23) and Section 2.3.3.

C – Challenges for Baseline Approaches

As discussed in Section 4.4 and 4.6, one might imagine that the task can be simply addressed by first grouping the trajectories with a simple clustering, e.g., K-means, and then following the closest reference trajectory given the current state. To illustrate the challenges involved in this scenario, this section shows this is not applicable in terms of both learning and exercising the task constraints.

First, for each demonstration sample, the locations of the starting point and the via-points are different. The invariant constraint of reaching correct via-point and destination is implicit and cannot be trivially revealed from an isotropic distance. Figure 4.22 shows that the K-means result is poor for assigning demonstrations to the correct behavior mode. As a comparison, the proposed approach obtains a better result because it assesses the similarity



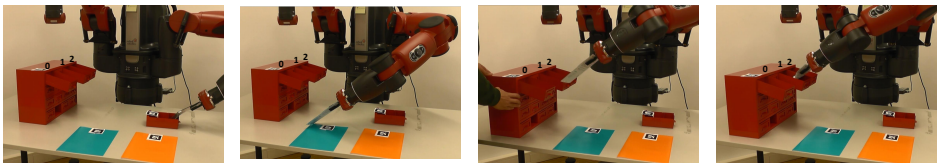
(a) KMeans (b) Random Embeddings

Figure 4.22: Clustering demonstration trajectories (dot lines) into three modes: The trajectories are transformed to the mailbox reference frame and projected into the XY surface for the clarity of comparison. The KMeans method takes the best result from 500 random initializations of the cluster centroids. An ideal clustering is supposed to group the demonstrations with a similar behavior mode: trajectories of a same color should reach a same destination.

with an aggregated nonlinear metric. Here the insight is that the importance of the state dimensions is non-uniform and implicitly correlated to the critical reference frames which depends on the task mode. The proposed approach identifies discriminating feature dimensions through a consideration over a group of naive selections, and as a result, a nontrivial metric emerges and captures the implicit static task constraints.

Secondly, even though a perfect demonstration clustering is given, it is insufficient to decide the mode straightforwardly based on the current observable state. To see this, a TPGMM is trained over the perfectly clustered data. A reproduction instance is then exercised by starting to follow mode 1: {mail location, orange area, mailbox-1} and adapted according to the likelihood of the observed state with respect to each mode.

Figure 4.23 illustrates a typical reproduction instance. Ideally, the execution should follow the initial mode in the absence of any perturbations. However, the robot actually deviates from the intended intermediate target by heading to the cyan area. This is due to the intrinsic motor noise and the mode ambiguity. Concretely, the robot motor noise will occasionally result in an end-effector position that is more close to one other mode than the current one. Even worse, this effect is aggravated in earlier stages of the execution, in which all modes



(a) (b) (c) (d)

Figure 4.23: Task reproduction with the baseline multi-mode behavior cloner. (a)(b): The robot starts with the intention to follow mode 1 (mail location-orange area-mailbox-1) but heads to the wrong intermediate area under its own motor noise. (c)(d): The location of mailbox is perturbed hence the mailbox-1 is again the most probable target given the current motion status. The robot delivers the mail to the mailbox-1 even the mail passed the cyan area.

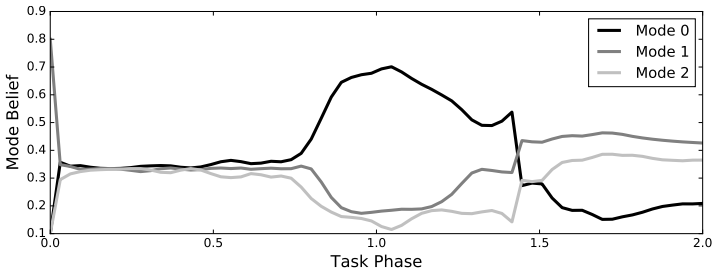


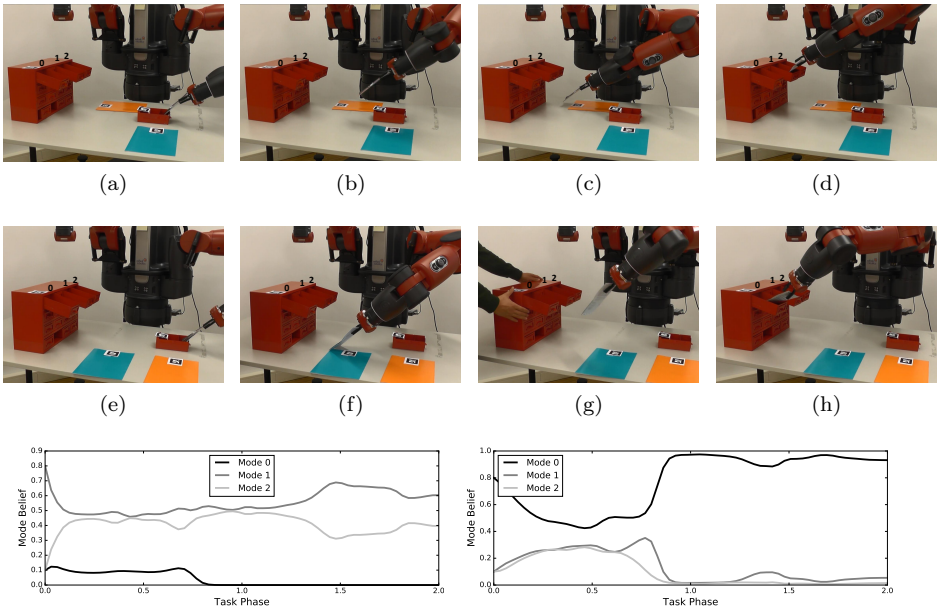
Figure 4.24: History of mode activation for a multi-mode behavior cloner: the robot agent always follows the most likely mode given its observation at each time step. This will result in undesired adaptations in certain cases.

are following similar trajectories to reach and collect the mail. Due to this ambiguity, the likelihood of all three modes is close and a change of mode will be triggered even under a small perturbation.

The figures illustrate yet another type of failure, which results from extrinsic disturbances. The robot, having passed via the cyan area, is moving towards mailbox-0. While it is approaching, the mailbox is relocated by humans. Therefore, the motion trajectory is heading to mailbox-1 in that instant. Given the likelihood of the current state, the mode 1 is regarded as a more likely one so an erroneous mode shift is triggered. The above analysis can be evidenced from the evolution of the mode belief, which is depicted in Figure 4.24. In brief, due to lack of robustness against both intrinsic and extrinsic disturbances, the baseline adaptation cannot reliably reproduce the intended behavior and conform to the demonstration constraints.

D – Results

In contrast to the above results from the baseline methods, Figure 4.25 illustrates successful reproductions, with the proposed latent dynamics enforced. In the first case (the snapshots in the upper row), the robot successfully follows the task mode 1 in a constant way. In second case (snapshots in the middle row), the robot correctly passes the cyan region and reaches the mailbox-0, even if the mailbox is moved on-the-fly. The difference from the baseline adaptation mechanism (Figure 4.24) is evidenced from the belief estimation (bottom row of Figure 4.25). Although the belief about the initial mode still decreases because of the ambiguity in the early parts of the trajectory, the prior biasing towards the current mode persists. As a result, the task reproduction is robust to the uncertainty about the robot intrinsic dynamics or a step disturbance such as pulling the mailbox away.



(i) Belief evolution under new configuration (upper reproduction)

(j) Belief evolution under perturbation on-the-fly (lower reproduction)

Figure 4.25: Task reproduction with the proposed framework under a novel task configuration. The robot adapts the intended motion (mail location-cyan area-mailbox-0) against the external perturbation of moving the mailbox away.

The baseline results are further compared by setting different configurations of the via-points. Here the metric is the success rate of the multi-mode con-

Table 4.1: Results of task reproduction under different targets and configurations: a reproduction is marked as a success if the robot follows the intended task mode and deliver the mail to the correct target. For each target mode, five trials are taken with the via-point layout randomly arranged.

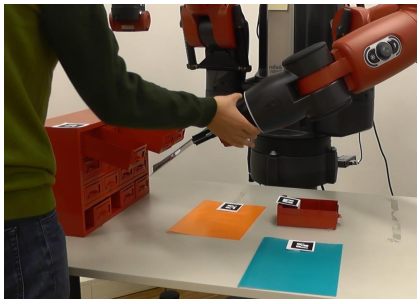
	Mode 0	Mode 1	Mode 2
Baseline	1/5	0/5	0/5
Proposed Approach	5/5	5/5	4/5

trollers for delivering mails to the correct targets under randomly arranged task configurations. The results are given in Table 4.1. The baseline multi-mode adaptation seldom succeeds. Especially when the intended targets are mailbox-1 or mailbox-2, the robot tends to lose the target while collecting the mail, as already exemplified in Figure 4.24. Thus it is quite frequent for the baseline method to fail in this task, even the task-parameterization is also used. On the contrary, the proposed method performs consistently better, reliably generalizing and executing the motion under various task configurations.

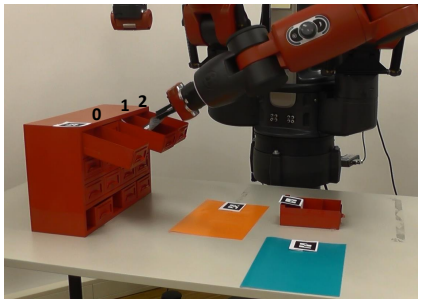
The robustness to external disturbance can also be seen from the point of view of collaboration, where the robot chooses to dominate the execution and reject the human guidance. This is shown in Figure 4.26. In this situation, the human intervenes with a manual guidance, aiming to redirect the delivery to mailbox-0. In light of the intervention, the "human preferred mode" is temporarily more likely w.r.t. the cost values of the current state, as seen in Figure 4.26d. However, since the robot has passed the orange intermediate area, a strong prior (that mode 0 is very unlikely) has been established. Thus the robot chooses to ignore the guidance so as to not violate the constraint imposed by the already executed trajectory.

On the other hand, the robot may also adapt and yield to the human intervention, when such intervention is in accordance with the learned constraints. Figure 4.27 demonstrates a similar execution but where the human intervention pushes the delivery towards mailbox-2. This example is different from one previously discussed, since the orange via-point is admissible for both modes. Therefore, there is a moderate possibility of switching modes and it does not require much effort from the human to enforce his/her intention and get the robot to collaborate accordingly.

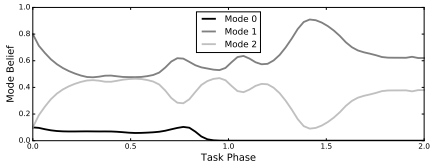
Table 4.2 gives more results about adaptation under different configurations.



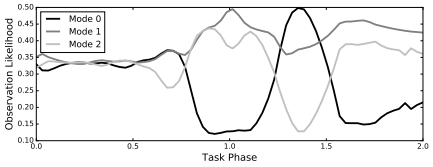
(a)



(b)

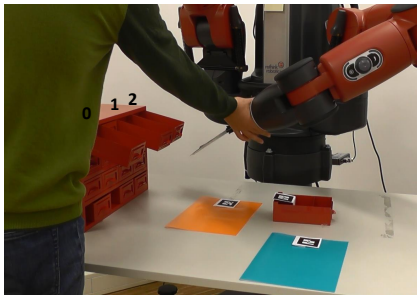


(c) Estimated mode belief

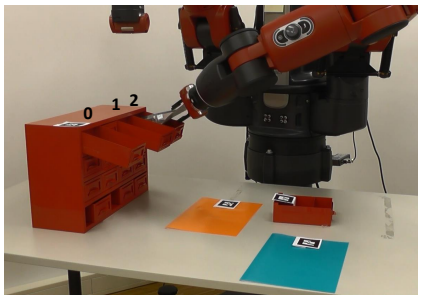


(d) Observation Likelihood

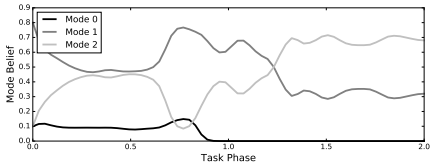
Figure 4.26: Reject to human intervention of guiding the delivery to an unlikely goal: the robot holds a low belief about the mode of reaching mailbox-0 since it has passed the orange area.



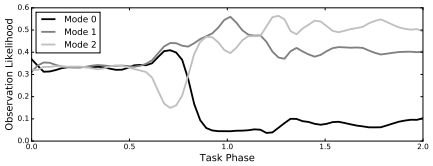
(a)



(b)



(c) Estimated mode belief



(d) Evolution of observation likelihood

Figure 4.27: Yielding to the external perturbation: the robot collaborates by adjusting the motion (mail location-orange area-mailbox 1) to an alternative target mailbox-2. The prior of mode 1 is not completely dominant against mode 2.

Table 4.2: Results of task adaptation under human intervention for different configurations: an adaptation is marked as a success if the robot (R) follows the human (H) intended task mode under the intervention and deliver the mail to the correct target. For each target mode, five trials are taken with the via-point layout randomly arranged.

R Mode	0	1	2
H Mode 0	5/5	5/5	5/5
H Mode 1	5/5	5/5	4/5
H Mode 2	4/5	5/5	4/5

Figure 4.28: Contour of the learned cost-to-go functions with the time and Z axes collapsed. The areas with dense contours indicate the demonstrations are locally consistent hence some of the regions will be discriminative for differentiating motion modes.

In this experiment, a human supervisor has his/her own intended task mode in mind, and intervenes by physically moving the robot motion if he/she thinks that the robot is not behaving correctly. All combinations of the robot initial mode (R Mode) and the human intention (H Mode) are tested. The metric is the success rate of the collaboration. A collaboration is considered as a success if: 1) the robot identifies the human intention and follows the guidance when the task constraint is fulfilled; 2) the robot follows its own intended motion when the human guidance violates the task constraints. The results demonstrate that the proposed framework allows the robot to understand the human intended target and adapt its motion accordingly throughout almost all of the test cases.

Some additional insights regarding the emerging behavior can be elicited from Figure 4.28. This figure overlaps the layout of the workspace and the corresponding cost evaluation, with the dimensions of mode z, time and vertical spatial axis collapsed. It is clear that the peaks of the cost coincide with the key objects in the scene. Moreover, steep cost gradient is visible due to the high consistency of the demonstration behaviors around these objects, especially the two intermediate spots. They are automatically identified as critical and discriminative frames. Passing either of them will lead to very strong constraints, preventing the follow-up motion to switch to the other modes, unless if such switching is compatible to the constraint (for example, switching between modes 1 and 2).

In all, this experiment showcases a task in which the proposed ensemble model helps to infer the intended task mode from the sensory feedback readings. With a prior upon the dynamical mode transition combined, a mixed behavior emerges: the robot can automatically decide when and where to collaborate with/reject human interventions based upon constraints extracted from the demonstrations.

4.8 Discussion

The ensemble technique discussed in this chapter overcomes the limitation in Chapter 3, which assumes the skill is composed as a single trajectory. Even though representing more sophisticated behaviors, the proposed model tackles the learning in an efficient way. Summarizing an answer to the questions raised in the beginning of the chapter:

- **Robotics**: A generative model for trajectories and latent motion modes can be learned. The model can be exploited in a mutual inference between the modes and trajectories, e.g., estimating the task mode for a real-time trajectory adaptation.
- **Machine learning**: Ensemble methods can be utilized to infer a probabilistic encoding of the trajectory modes. Efficient IOC models can be separately learned from demonstrations labeled as similar modes.

The efficiency of the inference are assured by the local LQR control and the discrete constraint on the mode variable (conjugate exponential family of probabilistic models). These formulations, though demand latent variables of a specific form, have showcased to be useful for the reported tasks which require real-time motion synthesis and adaptation.

The adopted ensemble principle is based on tree and bagging techniques. A bagging based ensemble alleviates overfitting by smoothing over multiple predictions. Hence, the approach is robust to noisy demonstrations. Moreoever, tree-based techniques generally scale well to a large dataset. Thus the model capacity is potential to learn a large skill repertoire. While on the other hand, one of the limitations is that it might face difficulties in selecting model parameters to learn from a limited number of demonstrations. In that case, the boosting scheme might be a better choice, since it focuses on predictive power

while the goal of bagging is variance reduction. Unlike the tree-based bagging, however, it is a bit vague that in what form the weak models can relate to a simple and meaningful IOC problem. Also, the standard boosting often aggregates the decisions through majority vote, which might be problematic for obtaining a continuous cost.

The framework demonstrates its capability of generalizing to untrained task configurations. This is enabled by the adopted task-parameterized feature. Generally, the generalization capability depends on the feature design. The framework leaves some room for incorporating the prior about the feature structure. For instance, the random subspace embedding is open to various types of the decision boundary and feature selection, capturing data structures beyond the axis-aligned grid used in this paper. The discussion about more general options can be found in the seminal tutorial about random forest, referring (43) for details.

As per locally grouped data, the adopted quadratic cost form demands a feature space in which an Euclidean distance serves as an effective norm. This actually does not impose much constrains on the original demonstration data, as long as one knows how to convert it to the task-relevant feature $\boldsymbol{x}$. For instance, forward kinematics can be used to project the raw joint positions to a task-relevant feature space, e.g., the robot end-effector or manipulated object pose. For the model synthesis, it is flexible to introduce features based on robot dynamics for adding more complexities, such as inverse dynamics control. Indeed, choosing a proper task-relevant feature entails a manual design. This is definitely one of the most phenomenal problems, not only in IOC, but also in general AI and machine learning. To put it in perspective, this framework is not straightforwardly applicable to extremely high-dimensional demonstrations (e.g., visual pixels) since the statistics are nontrivial and hard to be handcrafted. This problem will be discussed and addressed in the next chapter.

Another direction to explore is how the learned models can be used as priors to steer the posterior trajectory optimization. Since the model has the potential to encode a large amount of demonstration data, it would be interesting to explore how can it be applied to probabilistic trajectory planning with non-trial dynamical constraints or in a model-free settings like (28, 105). In light of that,

the consolidated skill knowledge can benefit the downstream control synthesis in terms of its exploration, refinement, generalization and ultimately, integration with learning from human demonstrations. The next chapter will also touch this topic in the domain of handwriting motion synthesis.

5

Linking Perception and Control

5.1 Introduction

Associating perception and control entails correlating variables of various sensory modalities, which are subject to different feature representations. In inverse optimal control, features determine the hypothesis space of a cost function. The previous chapters generally tackle a quadratic cost function with a well defined feature, such as the pose of an end-effector or kinematic model parameters. This chapter aims at automatically extracting the cost feature and learning perception and control represented by unstructured demonstrations.

From the robotics point of view, this topic is important because handcrafting features for some measurements, such as high-dimensional camera pixels, is impractical or requires substantial domain knowledge. A learning from demonstration (LfD) paradigm that automates the feature extraction is potential to reduce the feature engineering effort as well as free the restriction of sensor selection, thus substantially improving its empirical value. Also, jointly reasoning about multiple sensor modalities is interesting and feasible for nowadays robot

systems. After all, there is nothing preventing the task demonstrations being recorded through the lens of different types of sensors. An inspiring fact is that human beings are quite proficient in fusing the task knowledge or experience gathered from multiple sensing systems. For instance, humans can estimate the shape of an object through both vision and tactile sensations. This results in a redundant description since each channel provides a facet of the information of interest. Establishing an association between the redundant modalities is beneficial when only partial observation is presented in the task reproduction: in darkness, humans can still effortlessly perceive the object shape through a hand exploration. Therefore, by learning from multi-modal demonstrations, the robots are endowed with a more complete task description, a natural mechanism to estimate what is unknown from what is known, and as such, a capacity of robustly executing the task in face of uncertainty.

Motivating from the machine learning perspective, learning features together with the task further relaxes the common prerequisites about feature design in conventional IOC methods. Extracting non-trivial features from data is one of the main strengths of representation learning, which is gaining momentums amongst roboticists after its remarkable successes in general pattern recognition tasks. Hence it would be interesting to explore how the representation learning can be incorporated into the IOC framework. Also, a feature extraction demands constructing a mapping to project the raw data into a latent space. The manifold of the latent variable is often structured and well-behaved for a more simple and efficient model inference. From this perspective, this chapter also learns a latent variable in the same spirit of Chapter 4. However, the latent variable in Chapter 4 is discrete in order to secure a tractable posterior distribution. This assumption could be over restrictive for encoding unstructured demonstrations. Here the extension considers learning with a continuous latent variable. This will definitely enrich the model capacity with a more general assumption, while on the other hand, additional challenges arise as the posterior is no longer tractable.

Summarizing above discussion, the research questions from the robotics and machine learning perspectives can be identified as:

– **Robotics**: how can a robot learn from and reason about high-dimensional

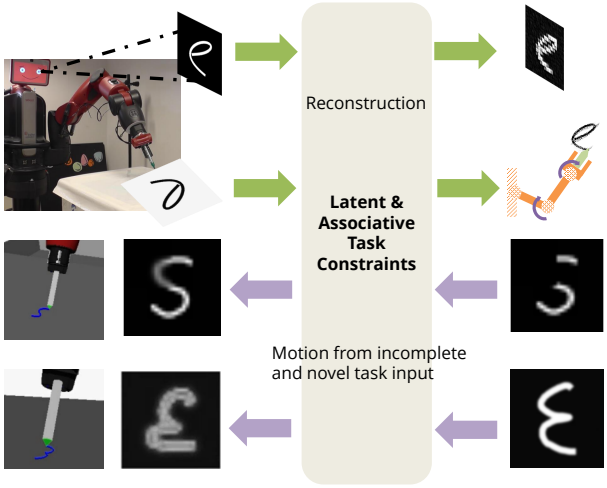


Figure 5.1: Learning representations for multiple sensory perceptions (vision and joint position) and associating them in the latent space for linking perception and control. The desired sensory/motor state, e.g., joint motion command, can be efficiently derived from incomplete or novel input e.g., symbol images.

data to associate perception and control modalities?

– **Machine Learning**: how can an IOC approach learn the data feature together with the cost function while assuming a general latent space?

To address the identified problems, this chapter draws the connection between a general form of IOC and the variational auto-encoders, a popular representation learning framework for generative models. The relation is discussed in Section 5.3 for the insights about incorporating representation learning in IOC approaches. An adapted variational auto-encoders is then developed in 5.4 as the main technical advancement for associating unstructured perception and control modalities. Also presented is an application of the cross-entropy optimization introduced in Chapter 3, which exploits the learned model to derive trajectories in more challenging settings (Section 5.5). To this end, the results in Section 5.6 can demonstrate succinct task manifolds and representations, which are then leveraged for an efficient motion derivation from the raw sensory input (Figure 5.1). The main contributions of this chapter are:

– An approach which enables an agent to learn from high dimensional raw demonstration data, with an adaptation from unsupervised representation learning.

– A KL-divergence-based metric that compactly associates the statistics of latent encodings of different demonstration modalities, resulting in efficient stochastic gradient descent training.
– An end-to-end system that enables the robot to generate arm joint writing motion from observed symbol images, with a robustness against image occlusion.

The main algorithm and results have been presented in (241). The chapter contains an extension about bootstrapping trajectory optimization with the learned model in Section 5.5. Additional results about the latent space and the extension are also included in Section 5.6.3 and 5.6.6.

5.2 Problem Statement

The central problem of this chapter is modeling multi-modal demonstrations with an IOC-based probabilistic model. Without a loss of generality, two modalities of raw sensor readings, such as vision pixels and joint positions, are considered here. The raw features are represented by variables with subscripts indicating the data modality, e.g., $\boldsymbol{x}_v$ for vision and $\boldsymbol{x}_m$ for joint motion. Under the MaxEnt assumption, the demonstration distribution can be parameterized by the cost function:

$$p(\boldsymbol{x}_v, \boldsymbol{x}_m) = \frac{e^{-\mathcal{J}(\boldsymbol{x}_v, \boldsymbol{x}_m, \boldsymbol{\theta})}}{\int e^{-\mathcal{J}(\boldsymbol{x}'_v, \boldsymbol{x}'_m, \boldsymbol{\theta})} d\boldsymbol{x}'_v d\boldsymbol{x}'_m} \tag{5.1}$$

in which the original feature or trajectory is now a concatenation of the involved modalities. Hence the IOC model eventually describes the multi-modal demonstrations as a joint data distribution.

Structures can be exploited for further induction of the general IOC formulation. Like the Chapter 4, a latent variable is assumed to factorize the joint distribution as:

$$p(\boldsymbol{x}_v, \boldsymbol{x}_m) = \int p(\boldsymbol{x}_v|\boldsymbol{z}) p(\boldsymbol{x}_m|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} \tag{5.2}$$

which means the raw readings are independent conditioned on the latent variable $\boldsymbol{z}$. The latent structure can be leveraged to obtain the factorization because using multiple modalities to describe a task could be redundant and the raw features are simply different views of perspective to the underlying task goal.

However, unlike the Chapter 4, the latent variable here is not restricted to be discrete because it relates to a quite general task-relevant feature, which does not have a clear interpretation like the demonstration style in the discrete case. Moreover, $\boldsymbol{z}$ in fact integrates the latent variables for both modalities with $\boldsymbol{z} = \{\boldsymbol{z}_v, \boldsymbol{z}_m\}$. The prior probability of $\boldsymbol{z}$ can thus be further factorized if the coupling $p(\boldsymbol{z}_v, \boldsymbol{z}_m)$ can assume more structures.

The main learning problem is to estimate the parameters of the above distribution. This is challenging because the latent variable is not of a simple discrete type so the marginal cannot be efficiently evaluated. Also, for an inference problem, one may also be interested in the posterior distributions $p(\boldsymbol{z}|\boldsymbol{x}_v)$ and $p(\boldsymbol{z}|\boldsymbol{x}_m)$. These in effect provide feature mappings to project the raw data into a more compact feature space for describing the task. In the low dimension space, one can seek a simpler cost-to-go function to describe the task manifold. Hence unstructured demonstrations can be captured by a cost-to-go function with a simple form and yet informative features. Lastly, the model should also allow for efficient and robust inference of $p(\boldsymbol{x}_v|\boldsymbol{x}_m)$ or $p(\boldsymbol{x}_v|\boldsymbol{x}_m)$. This is important to establish a link between perception and control modalities, for instance, inferring joint motion from a given visual cue.

5.3 Generative Representation Learning: PCA and Variational Auto-encoders from IOC perspective

This section discusses feature learning of IOC problems and motivates to address it by resorting to general generative representation learning techniques, such as PCA and Variational Auto-encoders (VAE). Revisiting the MaxEnt IOC form used in previous chapters:

$$p(\varsigma) = \frac{e^{-\mathcal{J}(\varsigma)}}{\int e^{-\mathcal{J}(\varsigma')} d\varsigma'} \qquad \mathcal{J}(\varsigma) = \sum_{t=0}^{T} \frac{1}{2} [(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{Q}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t) + \boldsymbol{u}_t^T \boldsymbol{R} \boldsymbol{u}_t] \quad (5.3)$$

where trajectory states in ς can be subject to a (locally) linear dynamics and one can also tie the cost parameters by omitting the index t. The learning is comparatively easy because the data is already represented with an informative representation, such as the pose in the operational space. In fact, if the raw feature, e.g., the joint positions $\boldsymbol{y}$, are used, the cost-to-go function is defined

as

$$\mathcal{J}(\varsigma^y) = \sum_{t=0}^{T} \frac{1}{2}[(\boldsymbol{\phi}(\boldsymbol{y}_t) - \boldsymbol{\mu}_t)^T \boldsymbol{Q}_t (\boldsymbol{\phi}(\boldsymbol{y}_t) - \boldsymbol{\mu}_t) + \boldsymbol{u}_t^T \boldsymbol{R} \boldsymbol{u}_t] \tag{5.4}$$

with a kinematic mapping $\boldsymbol{x}_t = \boldsymbol{\phi}(\boldsymbol{y}_t)$. The trajectory cost is no longer of an simple form in $\boldsymbol{y}_t$ because $\boldsymbol{\phi}(\cdot)$ is a nonlinear feature. Another more general example is the popular parameterization with linearly combined basis functions:

$$\mathcal{J}(\varsigma^y) = \sum_{t=0}^{T} \frac{1}{2}[\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{y}_t) + \boldsymbol{u}_t^T \boldsymbol{R} \boldsymbol{u}_t] \tag{5.5}$$

where $\boldsymbol{\phi}(\boldsymbol{y}_t)$ defines a nonlinear feature $\boldsymbol{x}_t = [\phi_1(\boldsymbol{y}_t), \phi_2(\boldsymbol{y}_t), ..., \phi_K(\boldsymbol{y}_t)]^T$, with $\phi_k(\cdot)$ commonly chosen as radial basis functions, e.g.,:

$$x_k = \phi_k(\boldsymbol{y}) = e^{-\gamma \|\boldsymbol{y} - \boldsymbol{\mu}_k\|} \quad \text{or} \quad x_k = \phi_k(\boldsymbol{y}) = \frac{e^{-\gamma \|\boldsymbol{y} - \boldsymbol{\mu}_k\|}}{\sum_{k'=1}^{K} e^{-\gamma \|\boldsymbol{y} - \boldsymbol{\mu}_{k'}\|}} \tag{5.6}$$

Again, if the feature parameters $\{\gamma, \boldsymbol{\mu}_k\}$ are defined, the learning cost is effectively hypothesized as a simple linear form in the feature space. When $\boldsymbol{\phi}(\cdot)$ is unknown or hard to craft, e.g., as the case of abstracting image pixels, the IOC approaches need to learn this feature mapping alongside the cost parameter $\boldsymbol{\theta}$. The complexity of such IOC problems depends on the choice of $\boldsymbol{\phi}(\cdot)$ since it also parameterizes the distribution for generating the data. In below, the notations are a bit abused to be consistent with general generative model, with $\boldsymbol{x}$ denoting the raw representation of the entire trajectory and $\boldsymbol{z}$ representing its projection in the latent space.

Let the feature mapping be assumed as a linear projection $\boldsymbol{z} = \boldsymbol{L}\boldsymbol{x}$, where the dimension of $\boldsymbol{z}$ is assumed to be much smaller than the original $\boldsymbol{x}$ ($d_{\boldsymbol{z}} \ll d_{\boldsymbol{x}}$). If a quadratic parameterization is used for the task feature $\boldsymbol{z}$, similar to the previous chapters, the MaxEnt model in the raw feature space $p(\boldsymbol{x})$ is also a Gaussian:

$$\begin{aligned} p(\boldsymbol{x}) &= |\boldsymbol{L}| p_{\boldsymbol{z}}(\boldsymbol{L}\boldsymbol{x}) = |\boldsymbol{L}| \frac{e^{-\frac{1}{2}(\boldsymbol{L}\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{L}\boldsymbol{x} - \boldsymbol{\mu})}}{\int e^{-\frac{1}{2}(\boldsymbol{z}' - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{z}' - \boldsymbol{\mu})} d\boldsymbol{z}'} \\ &= \frac{1}{\sqrt{(2\pi)^{d_{\boldsymbol{x}}} |\boldsymbol{L}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{L}^{\dagger}|}} e^{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{L}^{\dagger}\boldsymbol{\mu})^T (\boldsymbol{L}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{L}^{\dagger})^{-1} (\boldsymbol{x} - \boldsymbol{L}^{\dagger}\boldsymbol{\mu})} = \mathcal{N}(\boldsymbol{L}^{\dagger}\boldsymbol{\mu}, \boldsymbol{L}^{\dagger T} \boldsymbol{\Sigma} \boldsymbol{L}^{\dagger}) \end{aligned} \tag{5.7}$$

where $\boldsymbol{L}^{\dagger}$ denotes the pseudo-inverse of the feature mapping $\boldsymbol{L}$. Note the flexibility of $\boldsymbol{L}$ makes the estimation of cost parameters ill-posed. One can fix the

variance in the latent space as identity and reparameterize $\boldsymbol{L}^\dagger \boldsymbol{\mu}$ as $\bar{\boldsymbol{\mu}} = \boldsymbol{L}^\dagger \boldsymbol{\mu}$. In that sense, the new mean can be independently estimated and $\boldsymbol{L}^{\dagger T}\boldsymbol{L}^\dagger$ is a low-rank approximation to the data covariance because $\boldsymbol{L}$ is constrained by $d_{\boldsymbol{z}} \ll d_{\boldsymbol{x}}$. A best approximation, e.g., subject to a Frobenius norm, can be obtained through the singular value decomposition (SVD) (53). To this end, one can identify that solving this IOC effectively conducts a principle component analysis hence the PCA can be understood as learning a linear feature for a quadratic cost-to-go function defined in a low dimension space.

A linear feature, though efficient for learning, cannot parameterize an expressive model with a simple latent space. In order to express richer structures, the feature $\boldsymbol{\phi}(\cdot)$ entails nonlinearity. Variational Auto-encoders (VAE) is such a kind of generative model. Recall the derivation in the background chapter about using a parameterized distribution to approximate the true posterior:

$$\begin{aligned}\mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})] =& \mathbb{E}_{q_\phi}[\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) - \log p(\boldsymbol{z}|\boldsymbol{x})] \\ =& \mathbb{E}_{q_\phi}[\log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) - \log p(\boldsymbol{x}|\boldsymbol{z}) - \log p_0(\boldsymbol{z}) + \log p(\boldsymbol{x})]\end{aligned} \tag{5.8}$$

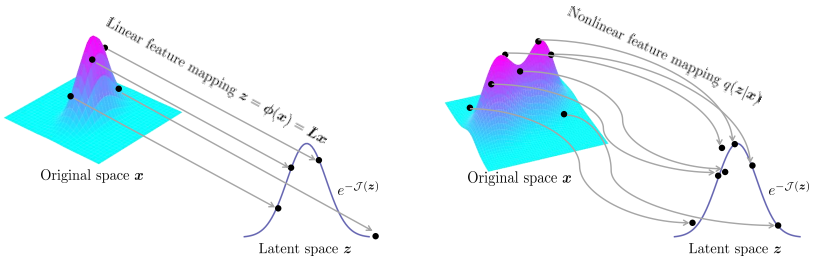
and the relation between the training objective and full data likelihood:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}) =& \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})] - \log p(\boldsymbol{x}) \\ =& \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p_0(\boldsymbol{z})] - \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p(\boldsymbol{x}|\boldsymbol{z})]\end{aligned} \tag{5.9}$$

Note that VAE assumes Gaussian probabilistic latent variable, prior and reconstructions: $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}^e(\boldsymbol{x}), \boldsymbol{\sigma}_1^2(\boldsymbol{x})\boldsymbol{I})$, $p_0(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}^d(\boldsymbol{z}), \boldsymbol{I})$. Apply the logarithm to recover the cost-to-go function from the likelihood:

$$\begin{aligned}\mathcal{J}(\boldsymbol{x}) &= -\log p(\boldsymbol{x}) + C = \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p_0(\boldsymbol{z})] - \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})] \\ &= \frac{1}{2}\|\boldsymbol{\mu}^{\boldsymbol{e}}(\boldsymbol{x})\|_{\boldsymbol{\sigma}_1^2(\boldsymbol{x})} + \frac{1}{2}\mathbb{E}_{q_{\boldsymbol{\phi}}}[\|\boldsymbol{x} - \boldsymbol{\mu}^d(\boldsymbol{z})\|_2] - \mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})] + C' \\ &\approx \frac{1}{2}\|\boldsymbol{\mu}^{\boldsymbol{e}}(\boldsymbol{x})\|_{\boldsymbol{\sigma}_1^2(\boldsymbol{x})} + C'\end{aligned} \tag{5.10}$$

where the approximation is accurate when 1) the reconstruction loss $\mathbb{E}_{q_{\boldsymbol{\phi}}}[\|\boldsymbol{x} - \boldsymbol{\mu}^d(\boldsymbol{z})\|_2]$ is small; 2) the latent mapping is approximated well so the divergence $\mathrm{KL}[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x})]$ is small. Both of the two can be realized with nonlinear encoder and decoder functions, which are deep neural networks in VAE. Removing these terms, an estimation of the cost $\mathcal{J}$ up to a constant term emerges.



(a) Linear feature (b) Nonlinear feature

Figure 5.2: Characterizing the data distribution with the cost defined in the latent feature space (a) a linear mapping for projecting Gaussian distributed trajectories, yielding a quadratic cost for a low dimension manifold; (b) a nonlinear feature mapping for projecting non-trivially distributed trajectories, approximately fitting a quadratic cost from the latent structure prior.

In contrast to the analysis of PCA, the function is a simple quadratic one in a nonlinear feature space.

It is worth noting that the quadratic form from the VAE derivation is so simple that there is no need to learn the original cost parameter $\boldsymbol{\theta}$. This is because a general nonlinear feature $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is powerful enough to transform and match arbitrary raw features to a fixed low dimension manifold. Informative features can shape the hypothesis space for an efficient model learning. Here the feature itself is sufficiently informative so no more model seeking is desired (Figure 5.2). One can of course further parameterize the prior p_0 to allow for more flexible cost-to-go functions in the feature space, such as parameterizing a dynamical system to cascade the priors in the latent space. Incorporating structured priors other than an isotropic Gaussian is an on-going research topic in general VAE and other types of generative models (86, 34).

5.4 Associative Variational Auto-encoders

From the above discussion, variational auto-encoder can be interpreted in the IOC framework as a way of learning compact cost-to-go features. This section presents the main contribution, an associative variational auto-encoder, which adapts the original framework to link interested modalities through the extracted latent space. It will show that the approach is also flexible for the application of synthesizing motion from a perceptual input, hence accommodating

the needs of the efficient inference upon the model.

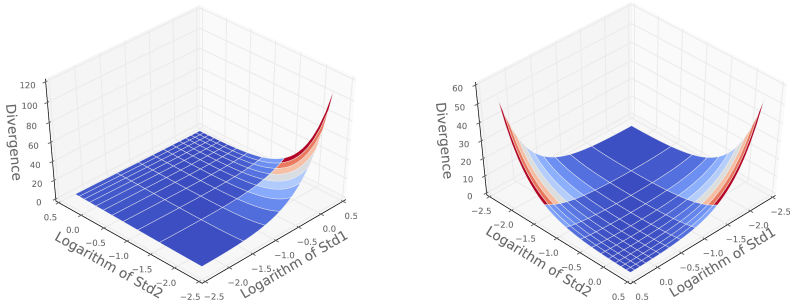
5.4.1 Associating Latent Representations

An associative variational auto-encoder consists of a collection of VAEs, each of which models one modality of the demonstration. The factored probabilistic model is correlated as stated in the Equation (5.2) if the raw feature $\boldsymbol{x}$ of each modality is considered a different perspective on the underlying task. So far $p(\boldsymbol{z}_v, \boldsymbol{z}_m)$ is a general joint distribution that captures this correlation. Specifically, a deterministic assumption is adopted here, implying the latent encodings are constrained by a metric, in the general form $h(\boldsymbol{z}_v, \boldsymbol{z}_m) = 0$. The constraint $h(\cdot)$ should not be very complicated because the features are already structured and the inference across modalities necessities a simple correlation. While there exist numerous assumptions about the form of this relation, it is reasonable to adopt an identity constraint. The intuition about the validity of this design is twofold:

- The latent variables actually correspond to features that are arbitrarily abstract for describing the task. A most direct description is to label the task behind the demonstration instance with the latent variable itself. In that sense, the latent variables obtained from multiple modalities should be identical because they are describing a same underlying task.
- The expressiveness of the nonlinear encoding and decoding features could be sufficient to support an abstraction of this level, while without compromising the model flexibility much.

Note that in VAE, the identity should be expressed as a match between the distributions of probabilistic latent encodings, namely $q_{\boldsymbol{\phi}_v(\boldsymbol{z}|\boldsymbol{x}_v^i)} = q_{\boldsymbol{\phi}_m(\boldsymbol{z}|\boldsymbol{x}_m^i)}$, $\forall \boldsymbol{z}$. The discrepancy between two probabilistic distributions can be captured in many ways, e.g., KL-divergence. A standard KL-divergence (Figure 5.3a), however, could be problematic here because it is not a metric which allows the exchangeability. The learning might be misled to yield a encoding with an infinitely large variance for the first modality, making the difference between $\boldsymbol{\mu}^e(\boldsymbol{x})$ irrelevant [1]. In light of this, a symmetrical KL-based metric is composed

1. The isotropic regularization in each modality might occasionally alleviate it but this is not guaranteed.



(a) Standard KL-divergence (b) Symmetrical KL-divergence

Figure 5.3: Standard and symmetrical KL-divergences between $\mathcal{N}(0, \sigma_1^2)$ and $\mathcal{N}(0, \sigma_2^2)$. The standard KL-divergence fails to capture the discrepancy for certain cases, e.g., $\sigma_1 = e^{-2}$ and $\sigma_2 = 1$, while the symmetrical one is invariant w.r.t. the commutation.

to quantify this relation:

$$\begin{aligned}\mathcal{L}_{assoc} =& \mathrm{KL}(q_{\boldsymbol{\phi}_v}(\boldsymbol{z}_v|\boldsymbol{x}_v^i) \| q_{\boldsymbol{\phi}_m}(\boldsymbol{z}_m|\boldsymbol{x}_m^i)) + \mathrm{KL}(q_{\boldsymbol{\phi}_m}(\boldsymbol{z}_m|\boldsymbol{x}_m^i) \| q_{\boldsymbol{\phi}_v}(\boldsymbol{z}_v|\boldsymbol{x}_v^i)) \\ =& \frac{1}{2}[\log \frac{|\Sigma_m(\boldsymbol{x}_m^i)|}{|\Sigma_v(\boldsymbol{x}_v^i)|} + \log \frac{|\Sigma_v(\boldsymbol{x}_v^i)|}{|\Sigma_m(\boldsymbol{x}_m^i)|} \\ &+ (\mu_m(\boldsymbol{x}_m^i) - \mu_v(\boldsymbol{x}_v^i))\Sigma_m^{-1}(\boldsymbol{x}_m^i)(\mu_m(\boldsymbol{x}_m^i) - \mu_v(\boldsymbol{x}_v^i)) \\ &+ (\mu_v(\boldsymbol{x}_v^i) - \mu_m(\boldsymbol{x}_m^i))\Sigma_v^{-1}(\boldsymbol{x}_v^i)(\mu_v(\boldsymbol{x}_v^i) - \mu_m(\boldsymbol{x}_m^i)) \\ &+ \mathrm{tr}(\Sigma_m^{-1}(\boldsymbol{x}_m^i)\Sigma_v(\boldsymbol{x}_v^i)) + \mathrm{tr}(\Sigma_v^{-1}(\boldsymbol{x}_v^i)\Sigma_m(\boldsymbol{x}_m^i))]\end{aligned} \tag{5.11}$$

which is still of a closed-form and differentiable with respect to the feature parameters $\boldsymbol{\phi}_v$ and $\boldsymbol{\phi}_m$, because of the Gaussianity of latent encodings. It can be shown that, as illustrated in 5.3b, this constraint implies an exchangeable modality sequence, as such, avoiding a directional dependency in $p(\boldsymbol{z}_v, \boldsymbol{z}_m)$. The final joint objective for the training can be obtained by putting together the proposed constraint and the applications of Equation (5.9) over the involved modalities $\boldsymbol{x}_v$ and $\boldsymbol{x}_m$, yielding:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_v, \boldsymbol{\theta}_m, \boldsymbol{\phi}_v, \boldsymbol{\phi}_m, \boldsymbol{x}_v^i, \boldsymbol{x}_m^i) &= \mathcal{L}_v + \mathcal{L}_m + \lambda \mathcal{L}_{assoc} \\ &= \mathrm{KL}[q_{\boldsymbol{\phi}_v}(\boldsymbol{z}_v|\boldsymbol{x}_v^i) \| p_0(\boldsymbol{z}_v)] - \mathbb{E}_{q_{\boldsymbol{\phi}_v}}[\log p(\boldsymbol{x}_v^i|\boldsymbol{z}_v)] \\ &+ \mathrm{KL}[q_{\boldsymbol{\phi}_m}(\boldsymbol{z}_m|\boldsymbol{x}_m^i) \| p_0(\boldsymbol{z}_m)] - \mathbb{E}_{q_{\boldsymbol{\phi}_m}}[\log p(\boldsymbol{x}_m^i|\boldsymbol{z}_m)] \\ &+ \lambda \mathrm{KL}[q_{\boldsymbol{\phi}_v}(\boldsymbol{z}_v|\boldsymbol{x}_v^i) \| q_{\boldsymbol{\phi}_m}(\boldsymbol{z}_m|\boldsymbol{x}_m^i)] + \lambda \mathrm{KL}[q_{\boldsymbol{\phi}_m}(\boldsymbol{z}_m|\boldsymbol{x}_m^i) \| q_{\boldsymbol{\phi}_v}(\boldsymbol{z}_v|\boldsymbol{x}_v^i)]\end{aligned} \tag{5.12}$$

with λ denoting the weight of the imposed constraint. It is worth noting that the introduced loss term of association adds no extra complexity, comparing

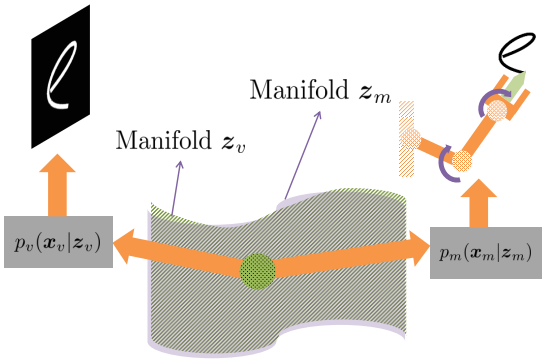


Figure 5.4: Learning overlapped task manifolds (surfaces with solid color and textures) for different demonstration modalities (vision and motion). Associative letter image and handwriting motion are retrieved by having an identical latent encoding go through corresponding decoders.

with a regular variational auto-encoder training. Also the standard stochastic gradient descent still applies for optimizing this adapted objective.

5.4.2 Efficient Inference on Perpetual Input

Learning and featuring associative demonstrations can be understood as extracting low dimensional task manifolds that are, in an ideal condition, fully overlapped (Figure 5.4) . The projections of different observation modalities are co-located on the manifolds. Exploiting this intuition, one can perform an inference for predicting one modality given the other one, for instance, deriving arm joint motion from a target letter image:

$$p(\boldsymbol{x}_m|\boldsymbol{x}_v) = \int p(\boldsymbol{x_m}|\boldsymbol{z}) q_{\boldsymbol{\phi}_v}(\boldsymbol{z}|\boldsymbol{x}_v) d\boldsymbol{z} \tag{5.13}$$

Such an inference is viable because the latent variable identity is implicitly used as an intermediate step to link the conditioned modality to the target one. Also, the integral can be efficiently evaluated by sampling from the shared low-dimensional manifold $\boldsymbol{z}$.

Moreover, a full probabilistic model provides additional inference options besides linking modalities in a basic manner. The low dimensional latent encodings can be leveraged to evaluate the marginal probability thus alleviating the intractability of inference within each modality space. This can be applied to a more practical and challenging scenario: while the input features are in-

complete or corrupted, the robot can still exploit what it learned to evaluate the imperfect perception, recover a more accurate estimation, and as such, derive the desired motion in a robust manner.

Concretely, the incomplete input feature, e.g., a letter image $\tilde{\boldsymbol{x}}_v$ with some parts occluded, is projected into the feature space to obtain a rough estimation of the latent encoding. With this as an initial guess, the manifold is explored to find a most likely latent variable whose reconstructed feature matches the observable part of $\tilde{\boldsymbol{x}}_v$ well. Quantitatively, it is proposed to solve:

$$\boldsymbol{z}_v^* = \underset{\boldsymbol{z}_v}{\operatorname{argmin}} - \log p_0(\boldsymbol{z}_v) + \eta \|\boldsymbol{x}_v^{(obs)}(\boldsymbol{z}_v) - \tilde{\boldsymbol{x}}_v^{(obs)}\| \tag{5.14}$$

where η weights the difference between the observable parts of the reconstructed and the target images. This objective literally seeks the latent encoding of an image which, on one hand matches the observable part of the target one, and on the other hand, is more probable w.r.t the learned cost function.

Note that the norm penalizing the difference of observable parts depends on the task modality [2]. Problems arise when the adopted norm is not differentiable. Thus, as a unified solution, $\boldsymbol{z}_v^*$ is proposed to be optimized through the cross entropy method used in previous chapters. The cross entropy method optimizes the target objective by alternating between taking samples from a proposal distribution and re-estimating it with the samples weighted under the target objective, hence removing the requirement of a differentiable norm. Again, since the samples are taken from the low dimension manifold, this method can secure an efficient inference.

5.5 Posterior Trajectory Optimization

It is a long-standing challenge for an agent to reuse the learned experience to bootstrap the solution in novel tasks. As for the running example, it desires the agent to develop the motion from the images of symbols that are different from the ones included in the training set. One of the viable solutions to this out-of-sample test, which realizes a transfer learning to some extent, is to fine-tune

2. In case of a perfect feature learning, the similarity could be surrogated in the extracted latent space by a simple norm, e.g., an Euclidean distance, and the optimization would be trivial. In practice, the distribution of corrupted data might be different from the training set, while the projections might be close if sufficient information is already provided by the observable part.

the result obtained from the source task model (208). In light of this, another application of the proposed approach is to seed the posterior policy search with the prior guess from a task relevant input. The intuition is that, the projection of the novel image encodes a similar learned letters, thus the associated initial motion approximation is expected to be close to the optimal one and in turn boosts the performance or efficiency for the posterior trajectory optimization.

5.6 Implementation and Results

This section presents the implementation and application of the proposed method in an illustrative task: associating handwriting arm motion and the letter image. Details about the experiment setup are given and the presented approach is also compared with other alternatives.

5.6.1 Data Augmentation

The dataset used for the implementation is UJI Char Pen 2 dataset, from which, for simplicity, only one-stroke-formed alphabetical letters and digits are considered. The data instances feature 2D trajectories, which are spatially and temporally aligned trough scaling and interpolation. The corresponding letter images are generated from the trajectories, yielding 28×28 grayscale thumbnails and a $\boldsymbol{x}_v$ of a length of 784. To emulate a less explicit motion representation, iterative LQR (220) is used to derive the optimal joint motion of a 7-DOFs Baxter robot arm. The arm joint motion is recovered to fit the 2D letter trajectories in the operation space with the joint torque efforts minimized. The joint trajectories are further parametrized by the function approximator $\boldsymbol{x}_t = \boldsymbol{w}^T \boldsymbol{\Phi}(t)$, which is used in Chapter 3. Thus the effective output for the motion modality is the coefficient of the function approximator. The motivation of introducing this representation is to incorporate a smoothness prior and reduce the complexity of the output dimension. For each DOF, 20 nonlinear basis functions plus a linear term are used, yielding a 147-dimension vector for the modality of $\boldsymbol{x}_m$.

Unfortunately, the UJI Char Pen 2 dataset is sparse and unbalanced for different letters and digits. The most number of samples for each type of character is 120. The difficulty is that representation learning methods are usually data-hungry and a primary test on the original dataset shows the model tends to

either overfit or fail in learning rare samples. This is proposed to be addressed by a data augmentation. Specifically, the dataset is augmented by exploiting the handwriting synthesis result in Chapter 4. The motion trajectories for each character are first learned with the ensemble probabilistic model, with the log-normal kinematics feature enforced. Then the characters for each category are re-sampled through the efficient multi-mode motion synthesis and obtain the corresponding images. Readers can revisit Section 4.7.2.1 for details of this procedure. Note that this is different from augmenting the size of dataset by simply adding white noise to the original coordinates and pixels. The randomness is constrained in the space of kinematics feature, which is borrowed from the research characterizing natural human movement. Also the quality of the synthetic samples is partly assured by the result of Turing-like test (Section 4.7.2.2. Eventually, more than 70000 pairs of images and arm motion are synthesized, with about 1000 samples per each character.

5.6.2 Model Implementation

Similar to the standard variational auto-encoder, neural network (NN) models are used as the data encoder $q(\boldsymbol{z}|\boldsymbol{x})$ and decoder $p(\boldsymbol{x}|\boldsymbol{z})$. Each of the NNs is comprised of two layers of rectified linear units (ReLU) as the nonlinear hidden features. Sigmoid functions are adopted as the output features of the vision modality, in order to obtain valid gray-scale values. The model architecture can be over-viewed as Figure 5.5. The training is carried out through the stochastic gradient descent with an adaptive moment estimation (ADAM) (101), a learning rate of 10^{-4} and a batch size of 64. The other hyper parameters, such as the length of the latent variable and the weight of association term, are selected according to the cross-validation of the reconstruction performance.

To illustrate the strength of the incorporated feature learning, Gaussian Mixture Models (GMM) on raw features are also trained as competing baselines. Training these models with full covariance matrices suffers from severe overfitting issues and is quite slow for a moderate number of components due to the high data dimensionality. To alleviate it, some variants are also explored. These encompass a GMM model with diagonal covariance matrices, a GMM model with a PCA dimension reduction and the combination of these two. For

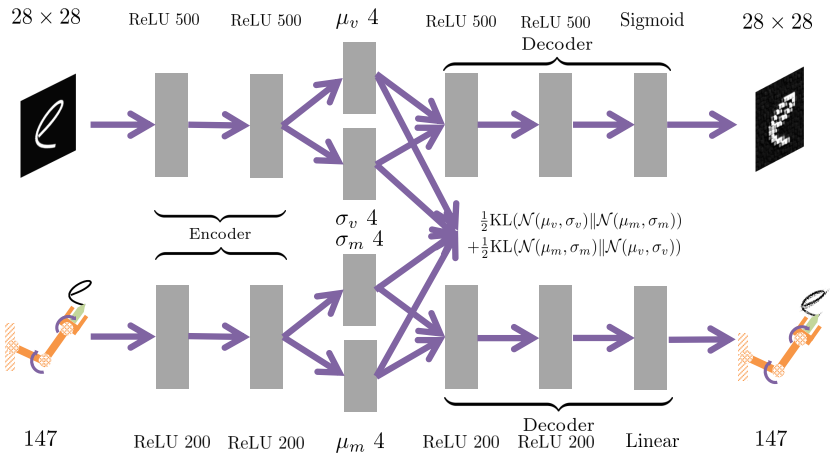
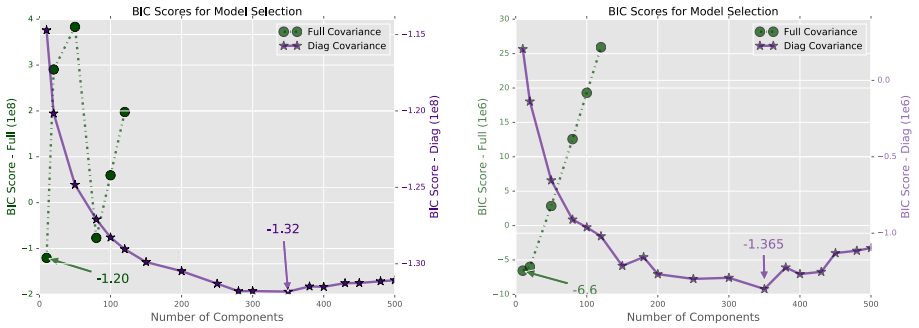


Figure 5.5: Model architecture of learning latent representations and association on different modalities of demonstrations. Latent layers of representation is annotated with feature type (Rectified Linear Unit) and size. The association is captured by a symmetrical KL-divergence.

the PCA preprocessing, the number of eigenvectors is selected to explain 99% data variance, yielding a reduced dimension of 240 for the image modality and 37 for the motion modality. The number of mixture components is determined based on the BIC criterion. We fit GMM models with a K-Means initialization and 15 random restart to find the best estimation. In our experiment, GMMs with 350 components and diagonal covariance matrices give the best BIC score (Figure 5.6a). Since a diagonal matrix cannot capture the correlation across feature dimensions, the best full covariance models with 10 components are also included in subsequent comparisons.

5.6.3 Wandering in the Latent Space

Figure 5.7 demonstrates the learned association by comparing the images and the arm motion decoded from identical latent variables. Here the two modalities are compressed in a 4-dimensional latent space. The latent encodings are selected by walking along the first two dimensions between the interval



(a) GMM (b) GMM-PCA

Figure 5.6: BIC scores for model selection of GMMs: (a) with the complete feature (b) with the feature subject to a PCA dimension reduction. Selected number of components: full - 10; diagonal - 350.

of $[-5, 5]$. The reconstructed images show plausible transition of morphologies with varying size or curvature of the loops or strokes. The corresponding motion, which is transformed as the end-effector trajectory in the operation space, resembles consistent profile throughout the wandering over the manifold. Also it is notable that the Cartesian trajectories always stay within the writing surface, with a deviation as small as 10^{-4}m, even though the model is agnostic about the arm forward kinematics . Therefore these observations conclude that the model indeed learns expressive representations and a global association on the manifold of the target task.

5.6.4 Deriving Joint Motion from Image Perception

A natural application of the learned encodings and association is to infer one data modality from the other one. In our handwriting context, this implies the model can be used to immediately derive the handwriting motion when a symbol image is presented, as such linking a feedforward control to a perceptual input.

Figure 5.8 depicts concrete samples of the predicted writing motion from symbol images. It is worth noting that the images here are not from the dataset itself but generated by a person with a different handwriting style. Specifically, the symbols are drawn by hand on a canvas or a user interface. The images are then retrieved and fed to the model to obtain the writing motion. For

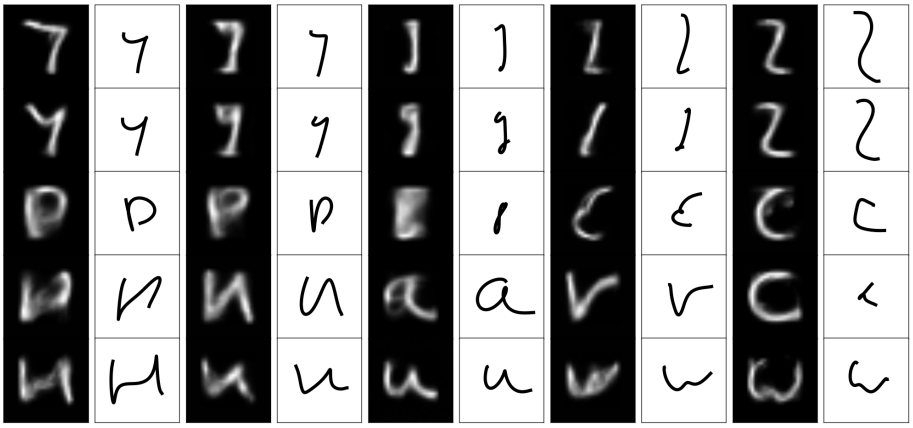


Figure 5.7: Decoded letter image (dark background) and arm motion in Cartesian space (light background) by walking along the first two main axes of the latent space ($\boldsymbol{z}^{(1)}$, $\boldsymbol{z}^{(2)} \in [-5, 5]$).

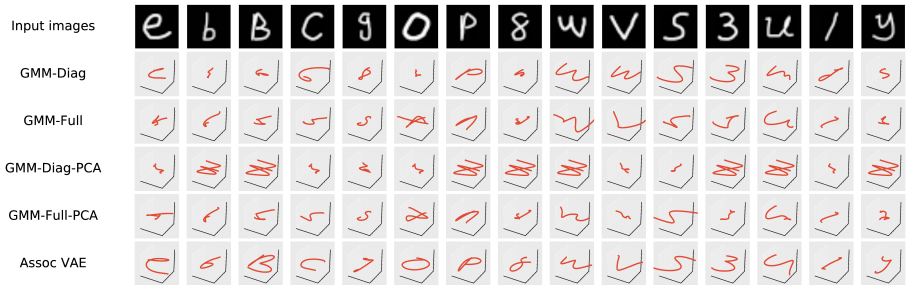


Figure 5.8: Deriving handwriting motion with different models and symbol images outside the test dataset: the resulted trajectories are transformed to the Cartesian space and shown in 3D plots. The input samples are generated by brushing with the mouse and are not cherry-picked.

the convenience of visualization, all of the joint motion is transformed into the Cartesian space and rendered as 3D plots.

As is clear from the figure, the proposed approach generates the most plausible arm joint motion for the drawn image samples. Because of the rich mode patterns of data, the model learned in the original feature space requires a large amount of local models to fully cover the data modes. Henceforth, among the alternative methods, GMM with diagonal covariance matrices, which admits a larger number of components, appears to have a comparatively better performance. However, due to the high dimensionality, such a shallow model still fails at times. Additionally, the PCA, aiming to reduce the data dimension, is not helpful in this task. In fact, the methods with PCA preprocessing perform

worse than the GMMs learned in the original feature space. This can be partially explained by the fact that the PCA inherently learns linear correlations as the features, which are not expressive in general cases. In our experiment, we observe that sometimes the generated movement forms an incomplete loop, like the cases of "g" and "8" in Figure 5.8. A possible cause is that, in the data augmentation, the samples are perturbed without an explicit constraint of maintaining the closeness of a loop thus the samples with a loop cut dominate the training data. Hence synthesized motion samples with an open loop dominates the augmented dataset, though similar samples might also emerge in cursive handwriting. One can expect an improvement by further constraining this in the synthesis of data augmentation.

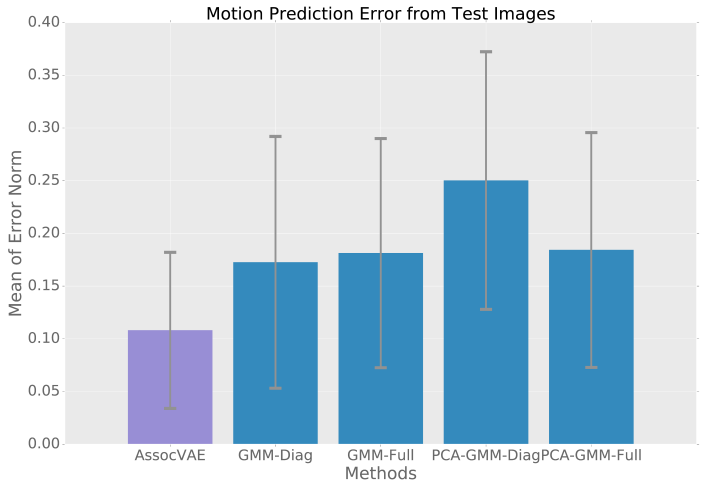


Figure 5.9: Error comparison of different models on predicting the arm joint motion from a symbol image of the test dataset. The error is measured as the Euclidean distance in the space of the coefficient of the trajectory function approximators.

The qualitative visual results are also in accordance with the numerical result. In this experiment, motion trajectories are predicted for the test dataset and the Euclidean distance between the prediction and ground-truth is measured in the function approximator basis space. As is clear from Figure 5.9, the presented associative VAE outperforms the competing methods by a significant margin. These results demonstrate the advantage of the proposed nonlinear feature learning in such a challenging task that involves high dimensional raw

sensory input.

5.6.5 Handling Imperfect Perception - Occluded Images

In this experiment, the letters are again written by a person whose handwriting is not included in the dataset. However, the model only receives a corrupted symbol image, with a random quartile covered. In order to guarantee the real-time performance, the number of iterations and samples of the cross-entropy optimization are both limited to 20. Figure 5.10 presents some instances of the experiment and clearly illustrates how the proposed inference proceeds. Initially, the algorithm attempts to make up the missed pixels with a plausible component. Then the recovered part is progressively refined and sharpened as the iteration continues. At last, the resultant latent encoding appears to be a good representation of the full underlying image, leading to correct writing motion (the last column). In practice, 20 iterations are often more than enough

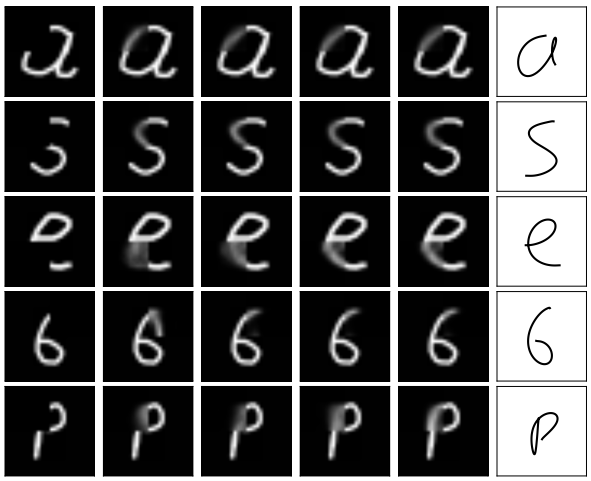


Figure 5.10: Inferring arm joint motion given occluded letter images: the latent encodings are explored to search complete images to match the observed parts before deriving the associated handwriting motion. The first column: input images; the second to the fifth columns: evolution of the recovered full images in iteration steps of 3, 8, 13, 18; the last column: Cartesian letter trajectories resulted from the inferred arm joint motion.

to reconstruct the image, thanks to the efficiency from the learned latent representation. With a projection from the observed pixels, the obtained initial guess is expected to be close to the ideal reconstruction on the manifold. In addition, the learned low dimension parameter space only desires a limited number of

samples to secure a stable exploration.

The GMM-based models are not compared here as it could be notoriously expensive to apply the cross-entropy method to sample pixels of hundreds of dimensions in the original space. Also, this experiment showcases a unique benefit of learning a generative model of demonstrations. Indeed, it provides a principled way to handle sensor uncertainties in the task execution. The robot systems can benefit from this in terms of skill generalization and robustness. Approaches in which sensory states are mapped directly to actions are unable to achieve this.

5.6.6 Bootstrapping Posterior Control for Novel Samples

In this experiment, the learned model is tested to examine if it could provide an informative prior for the posterior trajectory optimization. Ideally, the encapsulated knowledge should suggest a trajectory which is close enough to the optimal one, thus the trajectory optimization could potentially benefit from a more efficient exploration and avoiding trapping in local optima. From a broader point of view, this paradigm demonstrates how LfD can be utilized to adapt and transfer learned skills to completely novel tasks.

For the novel test tasks, images are retrieved from a free drawing, including "d" with a script font, symbol "square", "Δ", "moon", composed "7" and "6" and "Σ". This collection of symbols are selected with a qualitative and intuitive control about the task novelty. For instance, one can imagine that the motion for drawing a square can be relatively easy to search by adapting a prior for writing "O" and composing "Σ" is less straightforward due to its dissimilarity to learned letters.

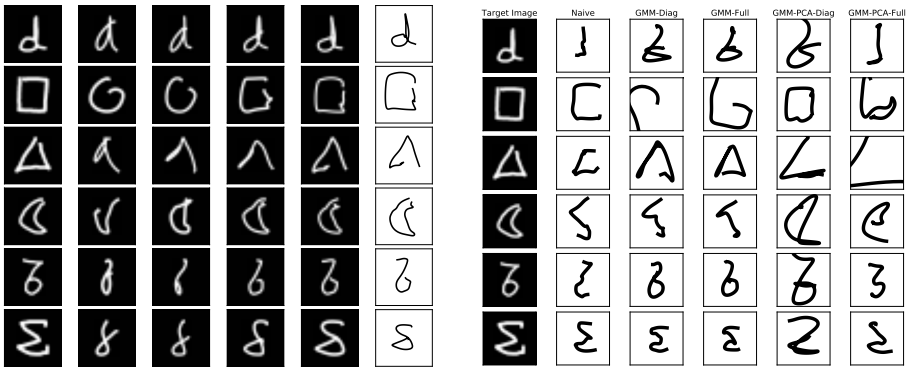
The above competing methods are used to generate initial trajectories for a comparison. Besides GMM-based models, a naive initialization, the motion of drawing a straight line, is also included. The posterior trajectory optimization is consistently performed with the cross entropy method. Another motivation of using the cross entropy method is because an explicit gradient for the process of generating images from motion is not available. The priors from the associative VAE are used in two ways. The first way is to apply the cross entropy optimization in the original (parameterized) trajectory space $\boldsymbol{x}_t = \boldsymbol{w}^T \boldsymbol{\Phi}(t)$ ("AssocVAE

full"). The second is defining the proposal distribution over the latent space, hence solving the task in a low dimension and constrained space ("AssocVAE latent"). For all the optimization initial guesses, the task performance is measured by the sum of pixel-wise quadratic errors realized in a fixed number of iteration steps. Unless explicitly stated, the cross entropy method parameters are set to be identical to assure a fair comparison.

As is shown as Figure 5.11a, the results of searching the trajectories with initial approximations from the proposed associative variational-autoencoder are presented. Indeed, given the novel symbol images, the learned model proposes plausible initial writing motion, such as generating the motion of writing a "G" for the "square" and "V" for the "moon". The script-style "d" is first approximated with the motion close of "a", which is not a perfect guess but close enough for the posterior trajectory optimization. At the end, visually believable results are obtained within 20 iteration steps. An interesting observation is the result of writing the symbol "Σ", which is shown in the last row of Figure 5.11a and expected to be a challenging one. The novel symbol is recognized to be close to the digit "8". Such an approximation, whose motion might not be that close to the target one, is nonetheless reasonable for the agent to perceive the input image with respect to what it has learned. Departing from such a motion prior, the trajectory optimization yields an "innovative" way of writing a "Σ", whose overall profile is visually well formed.

The performance of the proposed approach is compared with the naive and GMM-based initializations, whose numerical and visual results are respectively depicted in Figure 5.12 and 5.11b. For some of the symbols, these competing predictions are fine to yield reasonable writing results (e.g., "square" for GMM-PCA-Diag and "Δ" for "GMM-Full"). However, in general, the performance of searching in the original trajectory space with the proposed initial guess ("AssocVAE full") is more favorable. This is particularly phenomenal when an approximately correct prior is crucial for the trajectory optimization to escape from a poor local optima, such as the script-style "d" and symbol "moon".

In terms of the quantitative performance, the approaches based on associative variational-autoencoder is either comparable or superior across all the tasks. Specifically, searching in the latent space ("AssocVAE latent") is much



(a) (b)

Figure 5.11: Model-free trajectory optimization and refinement with the inferred arm motion as the initial guess. The searching is conducted in the original motion trajectory space. 5.11a The first column: the input image; the second to the fifth columns: symbol images resulted from the evolving motion trajectories in iteration steps of 3, 8, 13, 18; the last column: Cartesian trajectories. All testing cases except the first one are novel to the model. 5.11b The Cartesian motion result of trajectory optimization with initial guesses from competing approaches. All of the methods are using the same cross entropy method parameters and the figures are from the results after 20 iteration steps.

more rapid and stable for both the mean and covariance of trajectory costs, even when fewer sampling rollouts are used. This is similar to what has been observed in Section 5.6.5, where the searching space is constrained by a informative latent representation. While in this experiment, since the symbols are novel and not necessarily aligned with the learned manifold, such a constraint tends to result an approximation that is close to the projection of the target symbol on the manifold. Therefore, when there is an informative approximation to shape the searching direction, exploring in the full trajectory space ("Assoc-VAE full") offers more flexibility to yield a better performance in terms of visual consistency (Figure 5.11a).

5.7 Discussion

This chapter approaches a challenge arising in the practical LfD: learning and reasoning about high dimension data of multiple modalities. Perceptual and control modules can be linked by correlating multiple data modalities. The proposed algorithm learns feature mappings that are, on one hand effective

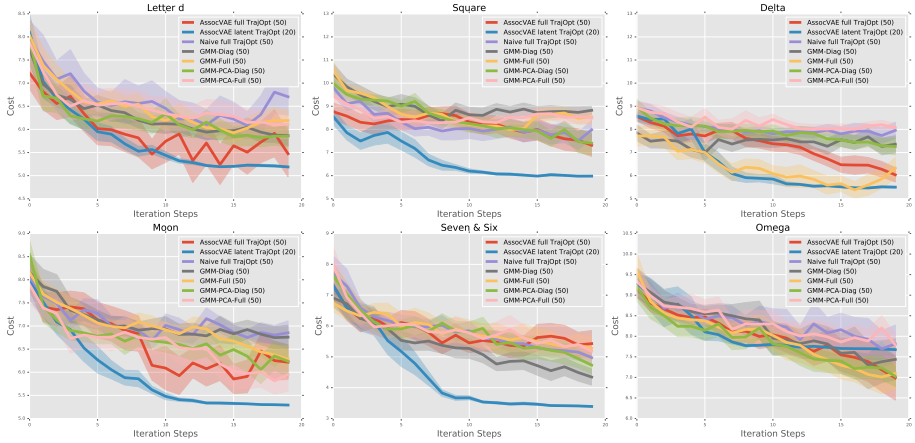


Figure 5.12: The evolution of cost values through cross entropy trajectory optimization with different initial approximations. Except the prior guesses, all methods and testing cases start with same parameter settings. The AssocVAE with latent representation explores in a low dimensional feature space and uses fewer samples (20) in iteration steps. Other methods use 50 trajectory samples. The curve and shaded area represent the mean and standard deviation of the cost of samples.

for compressing and reconstructing the raw data, and on the other hand, simple enough to afford an intuitive and efficient representation of the associativity. The underlying IOC problem thus assumes a nonlinear featured cost-to-go function, which much increases the model capacity to capture high dimension unstructured patterns. As a result, the answers of the raised problems in the beginning of the chapter can be summarized as:

- **Robotics**: the IOC framework can be extended to compress the high-dimensional data by extracting a succinct representation. Concurrently, the correspondence between perception and control modalities can be correlated as the joint distribution over the extracted representations.
- **Machine Learning**: representation learning can be adapted to parameterize a lower-bound of the demonstration likelihood with a continuous latent variable prior. The differentiability of the re-parameterized sampling allows to efficiently optimize this surrogate together with exploring a nonlinear transformation to the latent space.

The proposed approach is largely following one of the main venues in general machine learning: bridging expressive neural models and probabilistic inference. Placing the IOC framework in this perspective, many variants beyond

the vanilla VAE can be explored. It is worth noting that the distribution of the synthesized motion conditioned on the input image is still unimodal, even the VAE represents a sophisticated distribution over each sensory modality. This might cause problem because the correspondence of the data modalities is not necessarily a bijection. For instance, as the case in Chapter 4, a letter image could be generated from different handwriting motion. As a result, the motion prediction based on a unimodal latent distribution might miss the other viable modes. Intuitively, one can model $\boldsymbol{x}_m$ by adopting an advanced $p(\boldsymbol{x}|\boldsymbol{z})$, e.g., a GMM parametrized on $\boldsymbol{z}$. However, it might be risky to have an over-powered $p(\boldsymbol{x}|\boldsymbol{z})$. The VAE training might not learn a meaningful mapping $q(\boldsymbol{z}|\boldsymbol{x})$ by simply setting it to be the prior $p_0\boldsymbol{z}$, because the generator itself is sufficiently rich to represent the data-likelihood with $p(\boldsymbol{x}|\boldsymbol{z}) = p(\boldsymbol{x})$ (See discussions about the optimization challenge in (17, 35)). A promising solution is to assume a multi-modal posterior $q(\boldsymbol{z}|\boldsymbol{x})$. In a vanilla VAE, the posterior is parameterized as a diagonal Gaussian, which, from the variational point of view, resorts to a mean-field approximate inference. Latest works have proposed discrete latent variable (84) and stick-breaking-based probabilistic encoding whose length itself is stochastic (147). Also, noticing the posteriors entail efficient sampling and back-propagation of the parameter gradient, complex posteriors in (171, 102) are constructed from flows. Concretely, a flow-based posterior distribution is formalized by recursively applying an invertible transformation to an encoding $\boldsymbol{z}(\boldsymbol{x})$, which is initially with a simple distribution. From the IOC perspective, the autoregressive process in (102) could be used to factorize the posterior distribution conditioned on the entire trajectory $\boldsymbol{x} = \{\boldsymbol{x}_t\}$, e.g., $q(\boldsymbol{z}|\boldsymbol{x}) = \prod_t q(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})$. This effectively assumes and learns a dynamical system in the latent space. A prior about the agent dynamics, e.g., based on general physics laws (195), might be incorporated to model the sequence of high-dimension observations.

One of the most phenomenal challenges of applying representation learning in robotics, admittedly, is the importance of possessing massive high quality data. Unfortunately, the applicability of the introduced solution in this chapter is task dependent. Specifically, the data augmentation relies on the domain knowledge about the kinematics feature characterizing natural human movements. Apart from that, here the synthesis of the corresponding image modality is affordable

because it is cheap to simulate and convert trajectory coordinates to canvas pixels. For other types of data, e.g., the tactile of finger phalanges and the manipulated object pose, one might face challenges in rapidly generating the target pattern with the noise at a satisfying level. The robot might be exposed to substantial risks if it takes a large set of rollouts to collect the data. Physical simulation with a high-fidelity might alleviate this by safely synthesizing a large amount of control and perception pairs. Also, it is worth investigating how to reuse the knowledge from the experience of executing other related tasks, e.g., convolution filters from general image classification, to incorporate features that abstract many tasks and make the learning less demanding on the data volume. This might also help to learn new task skills with a few shots when an accurate model is not available to simulate real-world physics. This chapter takes some preliminary steps which seed the solution for a novel task with the prediction from the model learning a relevant task. It would be interesting to explore how a robot can incrementally aggregate the data collection from bootstrapped executions, and even orchestrate a sequence of subtasks (e.g., from writing simple strokes to composing complex calligraphies) to facilitate the mastery of motor skill.

6

Summary and Conclusions

In this final chapter, the thesis concludes with a summary of the main contributions. Also discussed are the important limitations. Detailed technical limitations have been covered at the end of each chapter. Here, the chapter focuses on high-level issues with a look ahead on future research directions.

6.1 A Recap of Contributions

One main contribution of this thesis is to offer an approach at using human demonstrations for identifying parameter of impedance control. The thesis does so by taking an IOC approach, which is not as explicit as programming the desired path of the tool-tip. Deriving an impedance controller from a learned cost function is not the only novel aspect of the work offered in Chapter 3. Importantly, the approach in Chapter 3 introduces task-relevant priors that shapes the general dynamics of the controller, while estimating the structured cost parameters according to the demonstration data. From the computing perspective, unlike the works following a standard IOC formulation, the algorithm in Chapter 3 treats both cost learning and control synthesis as probabilistic inference problems, so an importance-sampling-based technique can be uniformly used.

The model-free setting makes the algorithm less restrictive about the task dynamics and feature construction, implying the possibility of incorporating other type of priors.

In general, imitation learning approaches expect the demonstration data to cover all of the interested task dimensions. Chapter 4 and 5 take a different view towards this and argue that sometimes it could be advantageous to assume the data is incomplete. Explicitly considering incomplete demonstrations is rarely explored in general IOC research, with a notable exception of (154). At a first glance, introducing unobserved dimensions complicates the learning problem. However, as shown in these chapters, this added complexity has many advantages if the implicit variable is subject to an appropriate design. The general insight is that, comparing with the original demonstration features, the introduced latent variable could be cast as a more succinct description about the task. This could be greatly helpful for understanding the raw sensory data, and in turn benefit both learning and reasoning about the task. Specifically, Chapter 4 has shown that, once the estimation about a discrete latent variable is established, a general IOC learning can be decomposed to a set of less challenging problems. Each of sub problem resembles a form that has been somehow addressed in Chapter 3 and the extra computational cost for this transformation is modest. Chapter 5 introduces the latent variable with a more practice-oriented consideration. In this chapter, the latent variable is taken as a dimension-reduced equivalence to the original feature, which can be high-dimensional and unstructured. This is useful because one can alleviate the curse-of-dimensionality by reasoning about the sensory data in this low dimension space. Taking a further step from Chapter 4, the latent variable is continuous. Therefore, it represents a spectrum of variations and allows for an interpolation to capture a smooth transition among demonstrations.

Chapter 4 and 5 also close the loop, in which the above latent variables are employed to develop the control. Most LfD approaches cope with the link between perception and control by learning a coupled system. Though less straightforward, these two chapters adopt an architecture that decouples perception and control modules. The advantages of this choice is twofold. On one hand, a modular approach is flexible for incorporating priors in the intermediate

step to shape the task execution. In Section 4, it has been demonstrated that, for human robot collaboration, the adaptability and robustness of the robot can be modulated by enforcing different priors about the latent variable evolution. On the other hand, disentangled representations support a natural filtering and recovery of the perception from noisy measurements, thereby realizing a robust control that is less demanding about the data volume. As an example, Section 5.6.5 has shown synthesizing handwriting motion with an incomplete image input. Plausible feed-forward trajectories are obtained without needing to train on a dataset that includes corrupted character images.

To sum up, learning from demonstration is utilized and extended to obtain an internal model, which exploits human expertise for an improved representation, inference and synthesis of robot motion. The thesis considers a wide range of human expertise, which fuses task demonstration and established priors about perception and control.

6.2 An Outlook of Future Works

This section discusses potential directions along different dimensions. As shown in Equation (4.3), the representation of task skills comprises two parts: a goal-relevant cost function and task-independent passive dynamics. The first two subsections discuss the possibility of adopting dynamics and tasks of more general forms. The remained sections view in a larger picture, envisioning extensions from temporal and high-level perspectives.

6.2.1 Task Dynamics Beyond Discrete Motion

The cost-to-go function and linear-solvable system discussed in the thesis encode a stroke of discrete motion. Parameterizing a periodic state reference in the cost-to-go can be potential to learn rhythmic movements. More importantly, it would be interesting to learn with hybrid dynamics. The hybrid dynamics comprise continuous differential equations and discrete state transition to describe jumping events such as physical contacts. This is a more general form that describes multi-staged motions. In many contact-rich tasks, such as object in hand manipulation, the dynamics keeps switching among the modes of free and contact-constrained stages. Synthesizing such kind of dexterous motion

might entail an accurate hybrid dynamics model or efficient learning method. Furthermore, hybrid dynamics explicitly consider the contact force in the model. This could be useful when the contact contributes to the task performance. For instance, humans can restore the balance by pushing against the wall to exploit the environment reaction. As another example, impedance control might be insufficient if the task goal is not just accommodating the contact but exerting the force of a desired magnitude. In these cases, it would be more appropriate to regard the contact force as a task state instead of a disturbance. However, synthesizing motion under hybrid dynamics is hard due to the difficulty from contacts. Possible venues include model-based approaches which deal with limited types of contacts (214, 116, 55) and learning-based methods which avoid an explicit model (117). Learning from demonstrations could be useful to provide informative initialization or at least high-level plans such as ordered hybrid modes and dynamics switching surfaces.

6.2.2 Task-agnostic Learning

Learning from demonstrations generally targets solving a specific task. One of the substantial challenges is how a robot can generalize in the real-world and master a range of task skills. However, it would be rather tedious to require humans to exhaustively demonstrate all the task variations. It can be helpful to use the data, which targets addressing specific tasks, for learning other (related) tasks. The problem of lacking labeled data in target domains is also faced in general machine learning, where massive datasets of related but unlabeled are usually exploited (semi-supervised learning). In the robot learning practice, the question is that how the "unlabeled" demonstrations or experiences, which are not generated for the target task, can be leveraged for a domain adaptation. One of the viable ways could be collecting task-agnostic data through an exploration driven by general criteria like motion smoothness or curiosity. As a simple example, an off-line motor babbling could be used to estimate robot dynamics for learning different tasks. This relaxes the assumption about knowing robot passive dynamics in the thesis. Another extension could be capturing the variations of tasks rather than of demonstrations. Namely, the robot learns a spectrum of tasks by extracting some common features and builds a task-agnostic manifold.

It could be sample efficient to learn a relevant task by exploring on this manifold. As a result, the robot can rapidly adapt to learning a new task, realizing the generalization at the task level. The work in Chapter 5 touches this with learning to write a set of characters. It is worth exploring a similar idea in more general robotic tasks.

6.2.3 Interactive and Incremental Learning

The thesis focuses on learn from demonstration in a batch mode. However, in a few cases, learning in an interactive and incremental manner is desirable. For instance, it might be more efficient and user friendly for the robot to actively request human demonstrations when it is uncertain about how to act under the given task configurations. Also, as is shown in (114), the robot can replicate what it has learned and allow humans to adjust the robot skill through online correction. However, an eventual incremental learning of task variations or new skills requires consolidating the new data, instead of replacing what has been learned. Research efforts are still necessary to achieve this, because many models "forget" what they have learned after a training on the new task. This is identified as catastrophic forgetting problem (137). In machine learning, exploiting an external memory module is proposed as a potential to address this issue (65). In robotics, it would be appealing to realize an incremental learning of multiple tasks so the robot can progressively build up its skill repertoire, envisioning a life-time learning.

6.2.4 High-level Knowledge and Cues

Another observation about the data efficiency is that humans generally need much fewer demonstrations or trials to learn a new task skill. The priors established in learning other tasks, as discussed in Section 6.2.2, indeed play an important role. However, it is also worth noting that humans are proficient in reasoning about high-level salient cues. To bring up a concrete example, imagine a robot learning from a single demonstration of reaching an object on the table. Without showing the variation, e.g., reaching the object placed at different locations, it would be unclear if the robot should imitate the motion path or the reaching goal. In fact, the presence of an object itself is a strong

implication about the expected behavior. The similar importance of such contextual cues in resolving the imitation ambiguity has been observed in (13, 142), where children imitate the motor gesture or the goal of touching depending on the existence of target dots. In robot imitation tasks, such high-level knowledge can be used to identify important scene objects and understanding their properties, relations and potential functions, as such, biasing the model design for sparse demonstrations. The thesis employs the task-parametrized representation in Section 4 to extract the potential goals from a set of predefined objects. Moreover, the inference is based on variance so the expressed relation is limited and the demonstration variation is still required. Learning a general task in limited shots calls for a model prior beyond that. The capability of inferring the graspable parts from the object geometry, for instance, could let the robot bias its interpretation about the demonstration so as to imitate correctly in face of a novel object. To this end, it would be promising to have a framework that incorporates high-level common knowledge in certain forms, and thereby to yield an improved generalization performance.

Bibliography

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the International Conference on Machine Learning (ICML), pages 1–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015430. URL http://doi.acm.org/10.1145/1015330.1015430.

[2] Rachid Alami, Thierry Siméon, and Jean-Paul Laumond. A geometrical approach to planning manipulation tasks. The case of discrete placements and grasps. In Hirofumi Miura, editor, Proceedings of the International Symposium of Robotics Research (ISRR), pages 453–463. MIT Press, 1990. URL https://hal.archives-ouvertes.fr/hal-01309950.

[3] Alin Albu-Schäffer and Antonio Bicchi. Actuators for soft robotics. In Bruno Siciliano and Oussama Khatib, editors, Springer Handbook of Robotics, pages 499–530. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_21. URL http://dx.doi.org/10.1007/978-3-319-32552-1_21.

[4] V. M. Aleksandrov, V. I. Sysoyev, and V. V. Shemeneva. Stochastic optimization. Engineering Cybernetics, 5:11–16, 1968.

[5] R. McN. Alexander. A minimum energy cost hypothesis for human arm trajectories. Biological Cybernetics, 76(2):97–105, 1997. doi: 10.1007/s004220050324. URL http://dx.doi.org/10.1007/s004220050324.

[6] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew E. Taylor. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. In Proceedings of the National Conference on Artificial Intelligence (AAAI), AAAI'15, pages 2504–2510.

AAAI Press, 2015. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2886521.2886669.

[7] Olov Andersson, Wzorek Mariusz, and Patrick Doherty. Deep Learning Quadcopter Control via Risk-Aware Active Learning. In Proceedings of the National Conference on Artificial Intelligence (AAAI), 2017.

[8] Suguru Arimoto. Mathematical theory of learning with applications to robot control. In Kumpati S. Narendra, editor, Adaptive and Learning Systems: Theory and Applications, pages 379–388. Springer US, Boston, MA, 1986. ISBN 978-1-4757-1895-9. doi: 10.1007/978-1-4757-1895-9_27. URL http://dx.doi.org/10.1007/978-1-4757-1895-9_27.

[9] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. Journal of Robotic Systems, 1(2): 123–140, 1984. ISSN 1097-4563. doi: 10.1002/rob.4620010203. URL http://dx.doi.org/10.1002/rob.4620010203.

[10] H. Asada. Representation and learning of nonlinear compliance using neural nets. IEEE Transactions on Robotics and Automation, 9(6):863–867, Dec 1993. ISSN 1042-296X. doi: 10.1109/70.265932.

[11] H. Asada and H. Izumi. Automatic program generation from teaching data for the hybrid control of robots. IEEE Transactions on Robotics and Automation, 5(2):166–173, Apr 1989. ISSN 1042-296X. doi: 10.1109/70.88037.

[12] J. Andrew (Drew) Bagnell. An invitation to imitation. Technical Report CMU-RI-TR-15-08, Robotics Institute, Pittsburgh, PA, March 2015.

[13] H. Bekkering, A. Wohlschlager, and M. Gattis. Imitation of gestures in children is goal-directed. Quarterly Journal of Experimental Psychology, 53(1):153–164, Feb 2000.

[14] Richard Bellman. The theory of dynamic programming. Bull. Amer. Math. Soc., 60(6):503–515, 11 1954. URL http://projecteuclid.org/euclid.bams/1183519147.

[15] Aude Billard, Sylvan Calinon, and Rüdiger Dillman. Learning from humans. In Bruno Siciliano and Oussama Khatib, editors, Springer Handbook of Robotics, pages 1995–2014. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_21. URL http://dx.doi.org/10.1007/978-3-319-32552-1_21.

[16] A. Boularias, J. Kober, and J. Peters. Relative entropy inverse reinforcement learning. In JMLR Workshop and Conference Proceedings Volume 15: AISTATS 2011, pages 182–189, Cambridge, MA, USA, April 2011. MIT Press.

[17] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016, pages 10–21, 2016.

[18] Justin A. Boyan and Andrew W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 369–376. MIT Press, 1995. URL http://papers.nips.cc/paper/1018-generalization-in-reinforcement-learning-safely-approximating-the-value-functic pdf.

[19] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. Linear Matrix Inequalities in System and Control Theory, volume 15 of Studies in Applied Mathematics. SIAM, Philadelphia, PA, June 1994. ISBN 0-89871-334-X.

[20] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. van der Smagt, A. Albu-Schäffer, and S. Vijayakumar. Robots driven by compliant actuators: Optimal control under actuation constraints. IEEE Transactions on Robotics, 29(5):1085–1101, Oct 2013. ISSN 1552-3098. doi: 10.1109/TRO.2013.2271099.

[21] Leo Breiman. Bagging predictors. Machine Learning, 24(2):123–140, August 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350.

[22] Jonas Buchli, Freek Stulp, Evangelos Theodorou, and Stefan Schaal. Learning variable impedance control. The International Journal of Robotics Research, 30(7):820–833, 2011. doi: 10.1177/0278364911402527. URL http://dx.doi.org/10.1177/0278364911402527.

[23] S. Calinon. Robot learning with task-parameterized generative models. In Proceedings of the International Symposium of Robotics Research (ISRR), 2015.

[24] S. Calinon, F. Guenter, and A. Billard. On learning the statistical representation of a task and generalizing it to various contexts. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 2978–2983, May 2006. doi: 10.1109/ROBOT.2006.1642154.

[25] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37(2):286–298, April 2007. ISSN 1083-4419. doi: 10.1109/TSMCB.2006.886952.

[26] S. Calinon, I. Sardellitti, and D. G. Caldwell. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 249–254, Oct 2010. doi: 10.1109/IROS.2010.5648931.

[27] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell. Statistical dynamical systems for skills acquisition in humanoids. In Proceedings of IEEE International Conference on Humanoid Robots (Humanoids), pages 323–329, Osaka, Japan, 2012.

[28] S. Calinon, A. Pervez, and D. G. Caldwell. Multi-optima exploration with adaptive gaussian mixture model. In Proceedings of the International Conference on Development and Learning (ICDL-EpiRob), San Diego, USA, 2012.

[29] S. Calinon, D. Bruno, and D. G. Caldwell. A task-parameterized probabilistic model with minimal intervention control. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 3339–3344, May 2014. doi: 10.1109/ICRA.2014.6907339.

[30] S. Chandra, R. Paradeda, H. Yin, P. Dillenbourg, R. Prada, and A. Paiva. Do children perceive whether a robotic peer is learning or not. In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2018.

[31] Shruti Chandra, Raul Paradeda, Hang Yin, Pierre Dillenbourg, Rui Prada, and Ana Paiva. Affect of robot's competencies on children's perception. In Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), pages 1490–1492, May 2017.

[32] Yevgen Chebotar, Karol Hausman, Marvin Zhang, Gaurav Sukhatme, Stefan Schaal, and Sergey Levine. Combining model-based and model-free updates for trajectory-centric reinforcement learning. In Proceedings of the International Conference on Machine Learning (ICML), August 2017.

[33] N. Chen, J. Bayer, S. Urban, and P. van der Smagt. Efficient movement representation by embedding dynamic movement primitives in deep autoencoders. In Proceedings of IEEE International Conference on Humanoid Robots (Humanoids), pages 434–440, Nov 2015.

[34] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In D. D. Lee, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 2172–2180. Curran Associates, Inc., 2016.

[35] Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[36] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. IEEE Transactions on Robotics and Automation, 9(4):361–373, Aug 1993. ISSN 1042-296X. doi: 10.1109/70.246048.

[37] Jaedeug Choi and Kee-Eung Kim. Bayesian nonparametric feature construction for inverse reinforcement learning. In Proceedings of International Joint Conference on Artificial Intelligence, IJCAI '13, pages 1287–1293. AAAI Press, 2013. ISBN 978-1-57735-633-2. URL http://dl.acm.org/citation.cfm?id=2540128.2540314.

[38] J. Chung, K. Kastner, L. Dinh, K. Goel, A C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In Proceedings of Neural Information Processing Systems (NIPS), 2015.

[39] Adam Coates, Pieter Abbeel, and Andrew Y. Ng. Apprenticeship learning for helicopter control. Commun. ACM, 52(7):97–105, July 2009. ISSN 0001-0782. doi: 10.1145/1538788.1538812. URL http://doi.acm.org/10.1145/1538788.1538812.

[40] A. Colomé, G. Neumann, J. Peters, and C. Torras. Dimensionality reduction for probabilistic movement primitives. In Proceedings of IEEE International Conference on Humanoid Robots (Humanoids), pages 794–800, Nov 2014.

[41] John J. Craig. Adaptive control of manipulators through repeated trials. American Control Conference, 21:1566 – 1573, 1984. doi: 10.1109/ACC.1984.4171549.

[42] John J. Craig. Introduction to Robotics: Mechanics and Control. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1989. ISBN 0201095289.

[43] Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and Trends in Computer Graphics and Vision: Vol. 7: No 2-3, pp 81-227, 2012.

[44] Mayne David. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. International Journal of Control, 3(1):85–95, 1966. doi: 10.1080/00207176608921369. URL http://dx.doi.org/10.1080/00207176608921369.

[45] Peter Dayan and Geoffrey E. Hinton. Using expectation-maximization for reinforcement learning. Neural Computation, 9(2):271–278, 1997. URL citeseer.ist.psu.edu/dayan97using.html.

[46] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. Annals of Operations Research, 134(1):19–67, 2005.

[47] T.L. DeFazio, D.S. Seltzer, and D.E. Whitney. Instrumented remote center compliance. Industrial Robot, 11(4):238–242, 1984.

[48] M P. Deisenroth and C E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In Proceedings of the International Conference on Machine Learning (ICML), 2011.

[49] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, 39(1):1–38, 1977.

[50] Anca Dragan and Siddhartha Srinivasa. Formalizing assistive teleoperation. In Proceedings of Robotics: Science and Systems (RSS), July 2012.

[51] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In Proceedings of the International Conference on Machine Learning (ICML), ICML'16, pages 1329–1338. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045531.

[52] Krishnamurthy Dvijotham and Emanuel Todorov. Inverse optimal control with linearly-solvable mdps. In Proceedings of the International Conference on Machine Learning (ICML), pages 335–342, 2010.

[53] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. Psychometrika, 1(3):211–218, 1936. ISSN 1860-0980. doi: 10.1007/BF02288367. URL http://dx.doi.org/10.1007/BF02288367.

[54] M. Ewerton, G. Neumann, R. Lioutikov, H. Ben Amor, J. Peters, and G. Maeda. Learning multiple collaborative tasks with a mixture of interaction primitives. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 1535–1542, 2015. URL http://www.ausy.tu-darmstadt.de/uploads/Team/MarcoEwerton/marcoewerton_proceedings_icra_2015_.pdf.

[55] Siyuan Feng, Eric C. Whitman, X. Xinjilefu, and Christopher G. Atkeson. Optimization-based full body control for the DARPA robotics challenge. Journal of Field Robotics, 32(2):293–312, 2015. doi: 10.1002/rob.21559. URL https://doi.org/10.1002/rob.21559.

[56] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. In Proceedings of Neural Information Processing Systems (NIPS): Workshop on Adversarial Training, volume abs/1611.03852, 2016.

[57] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. Proceedings of the International Conference on Machine Learning (ICML), abs/1603.00448, 2016.

[58] Karl Friston. A free energy principle for biological systems. Entropy, 14 (11):2100–2121, 2012. ISSN 1099-4300. doi: 10.3390/e14112100. URL http://www.mdpi.com/1099-4300/14/11/2100.

[59] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4):193–202, 1980. ISSN 1432-0770. doi: 10.1007/BF00344251. URL http://dx.doi.org/10.1007/BF00344251.

[60] G. Ganesh, A. Albu-Schäffer, M. Haruno, M. Kawato, and E. Burdet. Biomimetic motor behavior for simultaneous adaptation of force,

impedance and trajectory in interaction tasks. In 2010 IEEE International Conference on Robotics and Automation, pages 2705–2711, May 2010. doi: 10.1109/ROBOT.2010.5509994.

[61] M. Geisert and N. Mansard. Trajectory generation for quadrotor based systems using numerical optimal control. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 2958–2964, May 2016. doi: 10.1109/ICRA.2016.7487460.

[62] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. Machine Learning, 63(1):3–42, April 2006. ISSN 0885-6125. doi: 10.1007/s10994-006-6226-1.

[63] A. Gijsberts and G. Metta. Incremental learning of robot dynamics using random features. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 951–956, May 2011.

[64] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 2672–2680. Curran Associates, Inc., 2014.

[65] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio G. Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià P. Badia, Karl M. Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. Nature, advance online publication, October 2016. ISSN 0028-0836. doi: 10.1038/nature20101. URL `http://dx.doi.org/10.1038/nature20101`.

[66] E. Gribovskaya, S.M. Khansari-Zadeh, and A. Billard. Learning non-linear multivariate dynamics of motion in robotic manipulators. The International Journal of Robotics Research, 30(1):80–117, 2011.

doi: 10.1177/0278364910376251. URL http://dx.doi.org/10.1177/0278364910376251.

[67] E. Gribovskaya, A. Kheddar, and A. Billard. Motion learning and adaptive impedance for robot control during physical interaction with humans. In 2011 IEEE International Conference on Robotics and Automation, pages 4326–4332, May 2011. doi: 10.1109/ICRA.2011.5980070.

[68] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6):1291–1307, Nov 2012. ISSN 1094-6977. doi: 10.1109/TSMCC.2012.2218595.

[69] Shixiang Gu, Ethan Holly, Timothy P. Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2017.

[70] Abhishek Gupta, Coline Devin, Yuxuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[71] Sami Haddadin, Michael Weis, Sebastian Wolf, and Alin Albu-Schäffer. Optimal control for maximizing link velocity of robotic variable stiffness joints. IFAC Proceedings Volumes, 44(1):6863 – 6871, 2011. ISSN 1474-6670. doi: http://dx.doi.org/10.3182/20110828-6-IT-1002.01686. URL http://www.sciencedirect.com/science/article/pii/S1474667016447087.

[72] Josiah Hanna and Peter Stone. Grounded action transformation for robot learning in simulation. In Proceedings of the National Conference on Artificial Intelligence (AAAI), February 2017.

[73] Nikolaus Hansen, Sibylle D. Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy

with covariance matrix adaptation (cma-es). Journal of Evolution Computation, 11(1):1–18, March 2003. ISSN 1063-6560. doi: 10.1162/106365603321828970. URL http://dx.doi.org/10.1162/106365603321828970.

[74] Christopher M Harris and Daniel M Wolpert. Signal-dependent noise determines motor planning. Nature, 394(6695):780–784, 1998. URL http://www.nature.com/nature/journal/v394/n6695/full/394780a0.html.

[75] Matthew Hausknecht and Peter Stone. Learning powerful kicks on the aibo ers-7: The quest for a striker. In Javier Ruiz del Solar, Eric Chown, and Paul G. Plöger, editors, RoboCup-2010: Robot Soccer World Cup XIV, volume 6556 of Lecture Notes in Artificial Intelligence, pages 254–65. Springer Verlag, Berlin, 2011.

[76] Neville Hogan. Impedance control: An approach to manipulation: Part i—theory. ASME Journal of Dynamical Systems, Measurements and Control, 107(1):1–7, 1985. doi: 10.1115/1.3140702.

[77] Deanna Hood, Severin Lemaignan, and Pierre Dillenbourg. When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), Portland, USA, 2015.

[78] M. Howard, D. J. Braun, and S. Vijayakumar. Transferring human impedance behavior to heterogeneous variable impedance actuators. Transactions on Robotics, 29(4):847–862, Aug 2013. ISSN 1552-3098. doi: 10.1109/TRO.2013.2256311.

[79] B. D. Huang, S. El-Khoury, M. Li, J. J. Bryson, and A. Billard. Learning a real time grasping strategy. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 593–600, May 2013. doi: 10.1109/ICRA.2013.6630634.

[80] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. Neurocomputing, 70(1–3):489 – 501, 2006. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.

2005.12.126. URL http://www.sciencedirect.com/science/article/pii/S0925231206000385. Neural NetworksSelected Papers from the 7th Brazilian Symposium on Neural Networks (SBRN '04)7th Brazilian Symposium on Neural Networks.

[81] Dongsung Huh and Terrence J. Sejnowski. Spectrum of power laws for curved hand movements. Proceedings of the National Academy of Sciences (PNAS), 112(29):E3950–E3958, 2015. doi: 10.1073/pnas.1510208112. URL http://www.pnas.org/content/112/29/E3950.abstract.

[82] K.J. Hunt, D. Sbarbaro, R. Żbikowski, and P.J. Gawthrop. Neural networks for control systems—a survey. Automatica, 28(6):1083 – 1112, 1992. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/0005-1098(92)90053-I. URL http://www.sciencedirect.com/science/article/pii/000510989290053I.

[83] A J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2002.

[84] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In Proceedings of the International Conference on Learning Representations (ICLR), volume abs/1611.01144, 2017.

[85] Ziviani Jenny and Wallen Margaret. The development of graphomotor skills. In Anne Henderson and Charlane Pehoski, editors, Hand function in the child: Foundations for remediation, 2nd Edition. Mosby, Inc, 2006.

[86] M J. Johnson, D. Duvenaud, A B. Wiltschko, S R. Datta, and R P. Adams. Composing graphical models with neural networks for structured representations and fast inference. In Proceedings of Neural Information Processing Systems (NIPS), 2016.

[87] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal. Learning objective functions for manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 1331–1336, May 2013. doi: 10.1109/ICRA.2013.6630743.

[88] Rudolf Kalman E. Contributions to the theory of optimal control. Boletin de la Sociedad Matemática Mexicana, 5:102–119, 1960.

[89] Rudolf Kalman E. When is a linear control system optimal. Journal of Basic Engineering, pages 51–60, Mar 1964.

[90] Hilbert J. Kappen and Wim Wiegerinck. A path integral approach to agent planning. In Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2007.

[91] Hilbert J. Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. Machine Learning, 87(2):159–182, 2012. ISSN 0885-6125. doi: 10.1007/s10994-012-5278-7.

[92] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[93] Tadashi Kashima and Yoshihisa Isurugi. Trajectory formation based on physiological characteristics of skeletal muscles. Biological Cybernetics, 78(6):413–422, 1998. doi: 10.1007/s004220050445. URL `http://dx.doi.org/10.1007/s004220050445`.

[94] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12(4):566–580, Aug 1996. ISSN 1042-296X. doi: 10.1109/70.508439.

[95] Homayoon Kazerooni. Exoskeletons for human performance augmentation. In Bruno Siciliano and Oussama Khatib, editors, Springer Handbook of Robotics, pages 773–793. Springer International Publishing, Cham, 2008.

[96] Mohammad Khansari, Klas Kronander, and Aude Billard. Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control. In Proceedings of Robotics: Science and Systems (RSS), 2014.

[97] S. M. Khansari-Zadeh and A. Billard. Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models. Transactions on Robotics, 2011.

[98] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), volume 2, pages 500–505, Mar 1985. doi: 10.1109/ROBOT.1985.1087247.

[99] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. IEEE Journal on Robotics and Automation, 3(1):43–53, February 1987. ISSN 0882-4967. doi: 10.1109/JRA.1987.1087068.

[100] D P. Kingma and M. Welling. Stochastic gradient vb and the variational auto-encoder. In Proceedings of the International Conference on Learning Representations (ICLR), 2014.

[101] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), volume abs/1412.6980, 2015.

[102] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 4743–4751. Curran Associates, Inc., 2016.

[103] Jens Kober and Jan Peters. Policy search for motor primitives. KI - Zeitschrift Künstliche Intelligenz, 23(3):38–40, August 2009.

[104] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. International Journal of Robotics Research, 32(11): 1238–1274, 2013. doi: 10.1177/0278364913495721. URL `http://dx.doi.org/10.1177/0278364913495721`.

[105] M. Kobilarov. Cross-entropy motion planning. International Journal of Robotics Research, 31(7):855–871, 2012.

[106] Daniel E. Koditschek. The robotics review. chapter Robot Planning and Control via Potential Functions, pages 349–367. MIT Press, Cambridge, MA, USA, 1989. ISBN 0-262-11135-7.

[107] Konrad P. Körding and Daniel M. Wolpert. Bayesian decision theory in sensorimotor control. Trends in Cognitive Sciences, 10(7):319–326, July 2006. ISSN 13646613. doi: 10.1016/j.tics.2006.05.003.

[108] Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. Robot motor skill coordination with em-based reinforcement learning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3232–3237, Oct 2010. doi: 10.1109/IROS.2010.5649089.

[109] Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. Advanced Robotics, 25(5):581–603, 2011. doi: 10.1163/016918611X558261. URL http://dx.doi.org/10.1163/016918611X558261.

[110] Sanjay Krishnan, Animesh Garg, Richard Liaw, Lauren Miller, Florian T. Pokorny, and Ken Goldberg. HIRL: hierarchical inverse reinforcement learning for long-horizon tasks with delayed rewards. CoRR, abs/1604.06508, 2016. URL http://arxiv.org/abs/1604.06508.

[111] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[112] K. Kronander. Control and Learning of Compliant Manipulation Skills. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2015.

[113] K. Kronander and A. Billard. Learning compliant manipulation through kinesthetic and tactile human-robot interaction. 2013.

[114] K. Kronander, S. M. Khansari Zadeh, and A. Billard. Incremental motion learning with locally modulated dynamical systems. Robotics and Autonomous Systems, 2015.

[115] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, and K. Mombaur. Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations. In Proceedings of IEEE International Conference on Humanoid Robots (Humanoids), pages 684–689, Nov 2015. doi: 10.1109/HUMANOIDS.2015.7363428.

[116] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. Autonomous Robots, 40(3): 429–455, 2015. ISSN 1573-7527. doi: 10.1007/s10514-015-9479-3.

[117] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 378–383, May 2016. doi: 10.1109/ICRA.2016.7487156.

[118] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015. ISSN 0036-8075. doi: 10.1126/science.aab3050.

[119] Steven M. LaValle and Jr. James J. Kuffner. Randomized kinodynamic planning. International Journal of Robotics Research, 20(5):378–400, 2001. doi: 10.1177/02783640122067453.

[120] Y. LeCun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Proceedings of neural information processing

systems (NIPS). chapter Handwritten Digit Recognition with a Back-propagation Network, pages 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-100-7. URL http://dl.acm.org/citation.cfm?id=109230.109279.

[121] Alex X. Lee, Henry Lu, Abhishek Gupta, Sergey Levine, and Pieter Abbeel. Learning force-based manipulation of deformable objects from multiple demonstrations. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 177–184, 2015. doi: 10.1109/ICRA.2015.7138997. URL http://dx.doi.org/10.1109/ICRA.2015.7138997.

[122] A. Lemme, K. Neumann, R.F. Reinhart, and J.J. Steil. Neural learning of vector fields for encoding stable dynamical systems. Neurocomputing, 141: 3 – 14, 2014. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2014.02.012. URL http://www.sciencedirect.com/science/article/pii/S0925231214003920.

[123] Ian Lenz and Ashutosh Saxena. Deepmpc: Learning deep latent features for model predictive control. In Proceedings of Robotics: Science and Systems (RSS), 2015.

[124] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. International Journal of Robotics Research, 34(4-5):705–724, 2015. doi: 10.1177/0278364914549607. URL http://dx.doi.org/10.1177/0278364914549607.

[125] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In Proceedings of Neural Information Processing Systems (NIPS), pages 1071–1079. 2014.

[126] Sergey Levine and Vladlen Koltun. Continuous inverse optimal control with locally optimal examples. In Proceedings of the International Conference on Machine Learning (ICML), 2012.

[127] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In J. Shawe-Taylor,

R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 19–27. Curran Associates, Inc., 2011.

[128] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. CoRR, abs/1504.00702, 2015.

[129] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. CoRR, abs/1603.02199, 2016. URL `http://arxiv.org/abs/1603.02199`.

[130] M. Li, H. Yin, K. Tahara, and A. Billard. Learning object-level impedance control for robust grasping and dexterous manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014.

[131] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[132] D Llorens, F Prat, A Marzal, JM Vilar, MJ Castro, JC Amengual, S Barrachina, A Castellanos, S España, JA Gómez, J Gorbe, A Gordo, V Palazón, G Peris, R Ramos-Garijo, and F Zamora. The ujipenchars database: a pen-based database of isolated handwritten characters. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, may 2008.

[133] Tomás Lozano-Pérez. Robot programming. Proceedings of the IEEE, 71 (7):821–841, July 1983. ISSN 0018-9219. doi: 10.1109/PROC.1983.12681.

[134] T. Lozano-Pérez and L. P. Kaelbling. A constraint-based method for solving sequential manipulation planning problems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3684–3691, Sept 2014. doi: 10.1109/IROS.2014.6943079.

[135] Matt T Mason. Compliance and force control for computer controlled manipulators. IEEE Transactions on Systems, Man, and Cybernetics, 11(6):418–432, June 1981. ISSN 0018-9472. doi: 10.1109/TSMC.1981.4308708.

[136] Zucker Matthew, Ratliff Nathan, Dragan Anca, Pivtoraiko Mihail, Klingensmith Matthew, Dellin Christopher, Bagnell J. Andrew (Drew), and Srinivasa Siddhartha. Chomp: Covariant hamiltonian optimization for motion planning. International Journal of Robotics Research, May 2013.

[137] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of Psychology of Learning and Motivation, pages 109 – 165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL http://www.sciencedirect.com/science/article/pii/S0079742108605368.

[138] J. R. Medina, D. Sieber, and S. Hirche. Risk-sensitive interaction control in uncertain manipulation tasks. In 2013 IEEE International Conference on Robotics and Automation, pages 502–507, May 2013. doi: 10.1109/ICRA.2013.6630621.

[139] Francisco S. Melo and M. Isabel Ribeiro. Q-Learning with Linear Function Approximation, pages 308–322. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-72927-3. doi: 10.1007/978-3-540-72927-3_23. URL http://dx.doi.org/10.1007/978-3-540-72927-3_23.

[140] Marvin Minsky. Steps toward artificial intelligence. Proceedings of the Institute of Radio Engineers, 49(1):8–30, Jan 1961. ISSN 0096-8390. doi: 10.1109/JRPROC.1961.287775.

[141] Djordje Mitrovic, Stefan Klanke, and Sethu Vijayakumar. Learning impedance control of antagonistic systems based on stochastic optimization principles. International Journal of Robotics Research, 30(5):556–573, 2011. doi: 10.1177/0278364910387653. URL http://dx.doi.org/10.1177/0278364910387653.

[142] T. Mizuguchi, R. Sugimura, and T. Deguchi. Children's imitations of movements are goal-directed and context-specific. Percept Mot Skills, 108 (2):513–523, Apr 2009.

[143] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In Proceedings of the International Conference on Machine Learning (ICML), 2014.

[144] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In NIPS Deep Learning Workshop. 2013.

[145] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 02 2015. URL `http://dx.doi.org/10.1038/nature14236`.

[146] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning (ICML), pages 1928–1937. 2016.

[147] Eric Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In Proceedings of the International Conference on Learning Representations (ICLR), volume abs/1611.01144, 2017.

[148] Kohl Nate and Stone Peter. Policy gradient reinforcement learning for fast quadrupedal locomotion. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), May 2004.

[149] Bojan Nemec, Nejc Likar, Andrej Gams, and Ales Ude. Bimanual human robot cooperation with adaptive stiffness control. In Proceedings of IEEE

International Conference on Humanoid Robots (Humanoids), pages 607–613, 2016. doi: 10.1109/HUMANOIDS.2016.7803337. URL http://dx.doi.org/10.1109/HUMANOIDS.2016.7803337.

[150] Gergely Neu and Csaba Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), UAI'07, pages 295–302, Arlington, Virginia, United States, 2007. AUAI Press. ISBN 0-9749039-3-0. URL http://dl.acm.org/citation.cfm?id=3020488.3020524.

[151] W.S. Newman. Stability and performance limits of interaction controllers. ASME Journal of Dynamical Systems, Measurements and Control, 114(4): 563–570, 1992.

[152] Andrew Y. Ng and Michael Jordan. Pegasus: A policy search method for large mdps and pomdps. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), UAI'00, pages 406–415, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-709-9. URL http://dl.acm.org/citation.cfm?id=2073946.2073994.

[153] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In Proceedings of the International Conference on Machine Learning (ICML), ICML '00, pages 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2. URL http://dl.acm.org/citation.cfm?id=645529.657801.

[154] Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 189–196, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-2883-8. doi: 10.1145/2696454.2696455.

[155] Christian O'Reilly and Réjean Plamondon. Development of a sigma-lognormal representation for on-line signatures. Pattern Recognition, 42

(12):3324 – 3337, 2009. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2008.10.017. New Frontiers in Handwriting Recognition.

[156] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic movement primitives. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 2616–2624. Curran Associates, Inc., 2013.

[157] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal. Online movement adaptation based on previous sensor experiences. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 365–371, Sept 2011. doi: 10.1109/IROS.2011.6095059.

[158] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In Proceedings of the International Conference on Machine Learning (ICML), pages 745–750, 2007. URL `http://www-clmc.usc.edu/publications//P/peters_ICML2007.pdf`.

[159] Jan Peters, Sethu. Vijayakumar, and Stefan Schaal. Reinforcement learning for humanoid robotics. In IEEE-RAS International Conference on Humanoid Robots (Humanoids2003), Karlsruhe, Germany, Sept.29-30, 2003. URL `http://www-clmc.usc.edu/publications/p/peters-ICHR2003.pdf`.

[160] Réjean Plamondon and Wacef Guerfali. The generation of handwriting with delta-lognormal synergies. Biological Cybernetics, 78(2):119–132, 1998. ISSN 0340-1200. doi: 10.1007/s004220050419.

[161] Dean A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. Neural Computing, 3(1):88–97, March 1991. ISSN 0899-7667.

[162] G. A. Pratt and M. M. Williamson. Series elastic actuators. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), volume 1, pages 399–406 vol.1, Aug 1995. doi: 10.1109/IROS.1995.525827.

[163] Martin L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.

[164] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[165] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, Advances in Neural Information Processing Systems 20, pages 1177–1184. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.pdf.

[166] Marc. H. Raibert and John. J. Craig. Hybrid position/force control of manipulators. ASME Journal of Dynamical Systems, Measurements and Control, 103(2):126–133, 1981. doi: 10.1115/1.3139652.

[167] Deepak Ramachandran and Eyal Amir. Bayesian Inverse Reinforcement Learning, 2007. URL http://www.aaai.org/Papers/IJCAI/2007/IJCAI07-416.pdf.

[168] Nathan Ratliff, J. Andrew (Drew) Bagnell, and Martin Zinkevich. Maximum margin planning. In Proceedings of the International Conference on Machine Learning (ICML), July 2006.

[169] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. Path integral control by reproducing kernel hilbert space embedding. In Proceedings of International Joint Conference on Artificial Intelligence, IJCAI '13, pages 1628–1634. AAAI Press, 2013. ISBN 978-1-57735-633-2. URL http://dl.acm.org/citation.cfm?id=2540128.2540362.

[170] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In Proceedings of the International Conference on Machine Learning (ICML), volume abs/1605.05396, 2016.

[171] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In David Blei and Francis Bach, editors, Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 1530–1538. JMLR Workshop and Conference Proceedings, 2015. URL http://jmlr.org/proceedings/papers/v37/rezende15.pdf.

[172] Martin Riedmiller. Neural Fitted Q Iteration – First Experiences with a Data Efficient Neural Reinforcement Learning Method, pages 317–328. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-31692-3. doi: 10.1007/11564096_32. URL http://dx.doi.org/10.1007/11564096_32.

[173] Cynthia A. Rohrbeck, Marika D. Ginsburg-Block, John W. Fantuzzo, and Traci R. Miller. Peer-assisted learning interventions with elementary school students: A meta-analytic review. Journal of Educational Psychology, 95(2):240–257, 2003. ISSN 0022-0663. doi: 10.1037/0022-0663.95.2.240.

[174] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras. Learning collaborative impedance-based robot behaviors. In Proceedings of the National Conference on Artificial Intelligence (AAAI), pages 1422–1428, Bellevue, WA, USA, 2013.

[175] L. Rozo, D. Bruno, S. Calinon, and D. G. Caldwell. Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1024–1030, Hamburg, Germany, Sept.-Oct. 2015.

[176] L. Rozo, S. Calinon, D. G. Caldwell, P. Jiménez, and C. Torras. Learning physical collaborative robot behaviors from human demonstrations. IEEE Transactions on Robotics, 32(3):513–527, June 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2540623.

[177] Reuven Y. Rubinstein. Some problems in Monte Carlo Optimization. PhD thesis, University of Riga, Riga, Latvia, 1969.

[178] Elmar A. Rückert, Gerhard Neumann, Marc Toussaint, and Wolfgang Maass. Learned graphical models for probabilistic planning provide a new class of movement primitives. Frontiers in Computational Neuroscience, 6(97), 2013. ISSN 1662-5188. doi: 10.3389/fncom.2012.00097.

[179] G. A. Rummery and M. Niranjan. On-line q-learning using connectionist systems. Technical report, 1994.

[180] Andrei A. Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. CoRR, abs/1610.04286, 2016. URL `http://arxiv.org/abs/1610.04286`.

[181] Tim Salimans, Jonathan Ho, Xi Chen, and Sutskever Ilya. Evolution strategies as a scalable alternative to reinforcement learning. CoRR, abs/1703.03864, 2017.

[182] J. K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, pages 95–100, Dec 1980. doi: 10.1109/CDC.1980.272026.

[183] Stefan Schaal. Dynamic movement primitives - a framework for motor control in humans and humanoid robots. In The International Symposium on Adaptive Motion of Animals and Machines, 2003.

[184] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. 2015.

[185] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. Proceedings of the International Conference on Machine Learning (ICML), 2015.

[186] A. Segre and G. DeJong. Explanation-based manipulator learning: Acquisition of planning ability through observation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), volume 2, pages 555–560, Mar 1985. doi: 10.1109/ROBOT.1985.1087311.

[187] A. P. Shon, K. Grochow, and R. P. N. Rao. Robotic imitation from human motion capture using gaussian processes. pages 129–134, Dec 2005. doi: 10.1109/ICHR.2005.1573557.

[188] Ashwini Shukla and Aude Billard. Augmented-svm: Automatic space partitioning for combining multiple non-linear dynamics. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 1025–1033. 2012.

[189] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. Chapter 8.4: Computed torque feedforward control. In Robotics: Modelling, Planning and Control. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 1846286417, 9781846286414.

[190] P. Sikka and B. J. McCarragher. Stiffness-based understanding and modeling of contact tasks by human demonstration. In Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on, volume 1, pages 464–470 vol.1, Sep 1997. doi: 10.1109/IROS.1997.649104.

[191] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587):484–489, January 2016. doi: 10.1038/nature16961.

[192] Jean-Jacques Slotine and Weiping Li. On the adaptive control of robot manipulators. International Journal of Robotics Research, 6(3):49–59, 1987.

[193] Jean-Jacques Slotine and Weiping Li. Chapter 9: Control of multi-input physical systems. In Applied Nonlinear Control, pages 392–433. Pearson, Upper Saddle River, NJ, 1991.

[194] L. Song, K. Fukumizu, and A. Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. IEEE Signal Processing Magazine, 30(4):98–111, July 2013. ISSN 1053-5888. doi: 10.1109/MSP.2013.2252713.

[195] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In Proceedings of the National Conference on Artificial Intelligence (AAAI), 2017.

[196] Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. Pac model-free reinforcement learning. In Proceedings of the 23rd International Conference on Machine Learning, Proceedings of the International Conference on Machine Learning (ICML), pages 881–888, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143955. URL http://doi.acm.org/10.1145/1143844.1143955.

[197] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. A. Theodorou, and S. Schaal. Model-free reinforcement learning of impedance control in stochastic environments. IEEE Transactions on Autonomous Mental Development, 4(4): 330–341, Dec 2012. ISSN 1943-0604. doi: 10.1109/TAMD.2012.2205924.

[198] Freek Stulp and Pierre-Yves Oudeyer. Adaptive exploration through covariance matrix adaptation enables developmental motor learning. Paladyn, 3(3):128–135, 2012.

[199] H. J. Sussmann and J. C. Willems. 300 years of optimal control: from the brachystochrone to the maximum principle. IEEE Control Systems, 17(3):32–44, Jun 1997. ISSN 1066-033X. doi: 10.1109/37.588098.

[200] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Proceedings of Neural Information Processing Systems (NIPS), NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969173.

[201] Richard S. Sutton. Learning to predict by the methods of temporal

differences. Machine Learning, 3(1):9–44, 1988. ISSN 1573-0565. doi: 10.1007/BF00115009. URL http://dx.doi.org/10.1007/BF00115009.

[202] Richard S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In Proceedings of Neural Information Processing Systems (NIPS), pages 1038–1044. MIT Press, 1996.

[203] Richard S. Sutton and Andrew G. Barto. Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

[204] Umar Syed, Michael Bowling, and Robert E. Schapire. Apprenticeship learning using linear programming. In Proceedings of the International Conference on Machine Learning (ICML), ICML '08, pages 1032–1039, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390286. URL http://doi.acm.org/10.1145/1390156.1390286.

[205] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 2154–2162. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6046-value-iteration-networks.pdf.

[206] Jie Tang and Pieter Abbeel. On a connection between importance sampling and the likelihood ratio policy gradient. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, Proceedings of Neural Information Processing Systems (NIPS), pages 1000–1008. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/3922-on-a-connection-between-importance-sampling-and-the-likelihood-ratio-policy-gra
pdf.

[207] A. K. Tanwani and S. Calinon. Learning robot manipulation tasks with task-parameterized semi-tied hidden semi-Markov model. IEEE Robotics

and Automation Letters (RA-L), 1(1):235–242, January 2016. doi: 10.1109/LRA.2016.2517825.

[208] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research, 10: 1633–1685, December 2009. ISSN 1532-4435.

[209] R. Tedrake, T. W. Zhang, and H. S. Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), volume 3, pages 2849–2854 vol.3, Sept 2004. doi: 10.1109/IROS.2004.1389841.

[210] Russ Tedrake. Chapter 12: Trajectory optimization. In Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). http://underactuated.mit.edu/, 2016.

[211] Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. Neural Comput., 6(2):215–219, March 1994. ISSN 0899-7667. doi: 10.1162/neco.1994.6.2.215. URL `http://dx.doi.org/10.1162/neco.1994.6.2.215`.

[212] E. Theodorou, Y. Tassa, and E. Todorov. Stochastic differential dynamic programming. In Proceedings of the 2010 American Control Conference, pages 1125–1132, June 2010. doi: 10.1109/ACC.2010.5530971.

[213] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. A generalized path integral control approach to reinforcement learning. Journal of Machine Learning Research, 11:3137–3181, December 2010. ISSN 1532-4435.

[214] E. Todorov. Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 6054–6061, May 2014. doi: 10.1109/ICRA.2014.6907751.

[215] Emanuel Todorov. Optimality principles in sensorimotor control. Nature Review Neuroscience, 7(9):907–915, 2004.

[216] Emanuel Todorov. Linearly-solvable markov decision problems. In Proceedings of Neural Information Processing Systems (NIPS), NIPS'06, pages 1369–1376, Cambridge, MA, USA, 2006. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2976456.2976628`.

[217] Emanuel Todorov. General duality between optimal control and estimation. In Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, pages 4286–4292, Dec 2008. doi: 10.1109/CDC.2008.4739438.

[218] Emanuel Todorov. Compositionality of optimal control laws. In Proceedings of Neural Information Processing Systems (NIPS), pages 1856–1864, USA, 2009. Curran Associates Inc. ISBN 978-1-61567-911-9.

[219] Emanuel Todorov and Michael Jordan. Optimal feedback control as a theory of motor coordination. Nature Neuroscience, January 2002.

[220] Emanuel Todorov and Weiwei Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In Proceedings of the 2005, American Control Conference, 2005., pages 300–306 vol. 1, June 2005.

[221] Marc Toussaint. Robot trajectory optimization using approximate inference. In Proceedings of the International Conference on Machine Learning (ICML), ICML '09, pages 1049–1056, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553508.

[222] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep probabilistic programming. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[223] T. Tsumugiwa, R. Yokogawa, and K. Hara. Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), volume 1, pages 644–650 vol.1, 2002. doi: 10.1109/ROBOT.2002.1013431.

[224] Aleš Ude, Christopher G. Atkeson, and Marcia Riley. Programming full-body movements for humanoid robots by observation. Robotics and Autonomous Systems, 47(2–3):93 – 108, 2004. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2004.03.004. URL http://www.sciencedirect.com/science/article/pii/S0921889004000405. Robot Learning from Demonstration.

[225] Ales Ude. Trajectory generation from noisy positions of object features for teaching robot paths. Robotics and Autonomous Systems, 1993.

[226] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. Biological Cybernetics, 61(2):89–101, 1989. ISSN 1432-0770. doi: 10.1007/BF00204593. URL http://dx.doi.org/10.1007/BF00204593.

[227] A.L.P. Ureche, K. Umezawa, Y. Nakamura, and A. Billard. Task parameterization using continuous constraints extracted from human demonstrations. Transactions on Robotics, 31(6):1458–1471, Dec 2015. ISSN 1552-3098. doi: 10.1109/TRO.2015.2495003.

[228] Jur van den Berg. Extended lqr: Locally-optimal feedback control for systems with non-linear dynamics and non-quadratic cost. In Proceedings of the International Symposium of Robotics Research (ISRR), pages 39–56, 2016. ISBN 978-3-319-28872-7. doi: 10.1007/978-3-319-28872-7_3.

[229] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. CoRR, abs/1609.03499, 2016. URL http://arxiv.org/abs/1609.03499.

[230] Mark J. Wagner and Maurice A. Smith. Shared internal models for feedforward and feedback control. Journal of Neuroscience, 28(42):10663–10673, 2008. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.5479-07.2008.

[231] Danwei Wang and C.C. Cheah. An iterative learning-control scheme for impedance control of robotic manipulators. The International Journal of Robotics Research, 17(10):1091–1104, 1998.

doi: 10.1177/027836499801701006. URL http://dx.doi.org/10.1177/027836499801701006.

[232] Christopher John Cornish Hellaby Watkins. Learning from Delayed Rewards. PhD thesis, King's College, Cambridge, UK, May 1989. URL http://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf.

[233] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In Proceedings of Neural Information Processing Systems (NIPS), pages 2728–2736. 2015.

[234] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3):229–256, 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL http://dx.doi.org/10.1007/BF00992696.

[235] Daniel M Wolpert, Jörn Diedrichsen, and J Randall Flanagan. Principles of sensorimotor learning. Nature reviews. Neuroscience, 12:739–51, 2011.

[236] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. CoRR, abs/1507.04888, 2015.

[237] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet. Human-like adaptation of force and impedance in stable and unstable interactions. Transactions on Robotics, 27(5):918–930, Oct 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.2158251.

[238] H. Yin, A. Paiva, and A Billard. Learning cost function and trajectory for robotic writing motion. In Proceedings of IEEE International Conference on Humanoid Robots (Humanoids), Madrid, Spain, 2014.

[239] H. Yin, A. Billard, and A. Paiva. Bidirectional learning of handwriting skill in human-robot interaction. In Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI): HRI Pioneer Workshop, 2015.

[240] H. Yin, P. Alves-Olivera, F. S. Melo, A. Billard, and A. Paiva. Synthesizing robotic handwriting motion by learning from human demonstrations. In

Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), New York, USA, 2016.

[241] H. Yin, F. S. Melo, A. Billard, and A. Paiva. Associate latent encodings in learning from demonstrations. In Proceedings of the National Conference on Artificial Intelligence (AAAI), San Francisco, USA, 2017.

[242] H. Yin, F. S. Melo, A. Billard, and A. Paiva. Boosting robot learning and control with domain constraints. In Proceedings of Robotics: Science and Systems (RSS), Pioneer Workshop, 2018.

[243] H. Yin, F. S. Melo, A. Paiva, and A. Billard. An ensemble inverse optimal control approach for robotic task learning and adaptation. Autonomous Robots, 2018.

[244] Fangyi Zhang, Jürgen Leitner, Michael Milford, Ben Upcroft, and Peter I. Corke. Towards vision-based deep reinforcement learning for robotic motion control. In Australasian Conference on Robotics and Automation (ACRA), 2015.

[245] Brian D. Ziebart. Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy. PhD thesis, Machine Learning Department, Carnegie Mellon University, Dec 2010.

[246] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In Proceedings of the National Conference on Artificial Intelligence (AAAI), pages 1433–1438, 2008.

[247] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Human behavior modeling with maximum entropy inverse optimal control. In AAAI Spring Symposium: Human Behavior Modeling, pages 92–. AAAI, 2009.

[248] Brian D. Ziebart, J. A. Bagnell, and Anind K. Dey. Modeling interaction via the principle of maximum causal entropy. In Johannes Fürnkranz and Thorsten Joachims, editors, Proceedings of the International Conference

on Machine Learning (ICML), pages 1255–1262. Omnipress, 2010. URL http://www.icml2010.org/papers/28.pdf.

A

Appendix

A.1 Proof for Proposition 1 in Chapter 4

Substituting the Gaussian passive dynamics and the quadratic cost-to-go function, we have:

$$P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \frac{e^{-\frac{1}{2}\|\boldsymbol{x}_{t+1}-f(\boldsymbol{x}_t))\|_{\Sigma_0^{-1}} - \frac{1}{2}\|\boldsymbol{x}_{t+1}-\boldsymbol{\mu}\|_{\boldsymbol{\Lambda}}}}{\int_{\boldsymbol{x}'_{t+1}} e^{-\frac{1}{2}\|\boldsymbol{x}'_{t+1}-f(\boldsymbol{x}_t)\|_{\Sigma_0^{-1}} - \frac{1}{2}\|\boldsymbol{x}'_{t+1}-\boldsymbol{\mu}\|_{\boldsymbol{\Lambda}}} d\boldsymbol{x}'_{t+1}} \tag{A.1}$$

The corresponding log-likelihood can be written as

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = & -\frac{1}{2}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t))^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t)) \\
& -\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}) \\
& -\log[\int \underbrace{e^{-\frac{1}{2}(\boldsymbol{x}'_{t+1} - f(\boldsymbol{x}_t))^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}'_{t+1} - f(\boldsymbol{x}_t))}}_{\leq 1 \text{ and positive}} \\
& e^{-\frac{1}{2}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})} d\boldsymbol{x}'_{t+1}] + \frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Sigma}_0| \\
\geq & \underbrace{-\frac{1}{2}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t))^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t))}_{\text{Independent of } \boldsymbol{\mu} \text{ and } \boldsymbol{\Lambda}} \\
& -\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}) + \frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Sigma}_0| \\
& \underbrace{-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Lambda}^{-1}|}_{-\log[\int e^{-\frac{1}{2}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})} d\boldsymbol{x}'_{t+1}]} \\
& = -\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}) - \frac{1}{2}\log|\boldsymbol{\Lambda}^{-1}| + \text{const} \\
& = \hat{\mathcal{L}}(\boldsymbol{\mu}, \boldsymbol{\Lambda})
\end{aligned}
\tag{A.2}
$$

where d denotes the state dimension. The exponential from the passive dynamics (the third line of the equation) can be considered as a positive coefficient that is always less than one. Replacing the coefficient with one results in a simple integral of Gaussian function (the exponential of negative cost-to-go function, line 7), which is always larger than or equal to the integral considering the passive dynamics.

$\hat{\mathcal{L}}(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ is thus a lower bound of the original likelihood by instead subtracting this simplified integral. Taking the derivatives $\frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\mu}} = 0$ and $\frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\Lambda}} = 0$, one can obtain:

$$
\begin{aligned}
\boldsymbol{\mu} &= \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{t+1}^i \\
\boldsymbol{\Lambda}^{-1} &= \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_{t+1}^i - \boldsymbol{\mu})(\boldsymbol{x}_{t+1}^i - \boldsymbol{\mu})^T
\end{aligned}
\tag{A.3}
$$

which happens to be the same as the MaxEnt estimation which assumes uniform passive dynamics:

$$
P_{MaxEnt}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \frac{e^{-\frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{\mu}\|_{\boldsymbol{\Lambda}}}}{\int_{\boldsymbol{x}'_{t+1}} e^{-\frac{1}{2}\|\boldsymbol{x}'_{t+1} - \boldsymbol{\mu}\|_{\boldsymbol{\Lambda}}} d\boldsymbol{x}'_{t+1}} \tag{A.4}
$$

Therefore the MaxEnt estimation is an approximate solution to the lower-bound of $\hat{\mathcal{L}}$. And the gap shrinks as noise magnitude $\|\mathbf{\Sigma}_0\| \to \infty$, with the original problem degenerating to the MaxEnt formulation.