VILD: Variational Imitation Learning with Diverse-quality Demonstrations

Voot Tangkaratt, Bo Han, Mohammad Emtiyaz Khan & Masashi Sugiyama
RIKEN Center for Advanced Intelligence Project
Tokyo, Japan
{voot.tangkaratt,bo.han,emtiyaz.khan,masashi.sugiyama}@riken.jp

Abstract

The goal of imitation learning (IL) is to learn a good policy from high-quality demonstrations. However, the quality of demonstrations in reality can be diverse, since it is easier and cheaper to collect demonstrations from a mix of experts and amateurs. IL in such situations can be challenging, especially when the level of demonstrators' expertise is unknown. We propose a new IL paradigm called Variational Imitation Learning with Diverse-quality demonstrations (VILD), where we explicitly model the level of demonstrators' expertise with a probabilistic graphical model and estimate it along with a reward function. We show that a naive estimation approach is not suitable to large state and action spaces, and fix this issue by using a variational approach that can be easily implemented using existing reinforcement-learning methods. Experiments on continuous-control benchmarks and real-world crowdsourced demonstrations denote that VILD outperforms state-of-the-art methods. Our work enables scalable and data-efficient IL under more realistic settings than before.

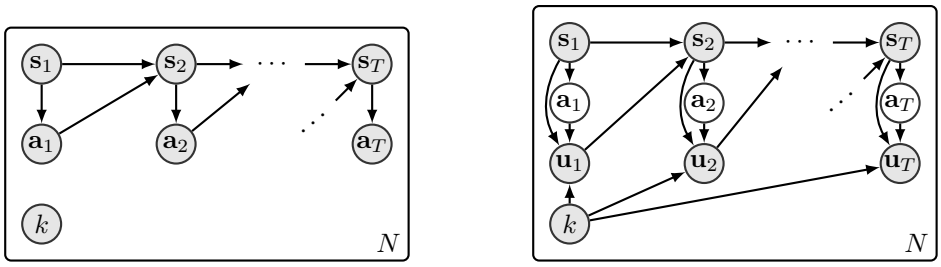
1 Introduction

The goal of sequential decision making is to learn a policy that makes good decisions [Puterman, 1994]. As an important branch of sequential decision making, Imitation learning (IL) [Russell, 1998, Schaal, 1999] aims to learn such a policy from demonstrations (i.e., sequences of decisions) collected from experts. However, high-quality demonstrations can be difficult to obtain in reality, since such experts may not always be available and sometimes are too costly [Osa et al., 2018]. This is especially true when the quality of decisions depends on specific domain-knowledge not typically available to amateurs; e.g., in applications such as the game of Go [Silver et al., 2016], robot control [Osa et al., 2018], and autonomous driving [Silver et al., 2012].

In practice, demonstrations are often diverse in quality, since it is cheaper to collect them from mixed demonstrators, containing both the experts and amateurs [Audiffren et al., 2015]. Unfortunately, IL in such settings tends to perform poorly since low-quality demonstrations often negatively affect the performance [Shiarlis et al., 2016, Lee et al., 2016]. For example, demonstrations for robotics can be cheaply collected via a robot simulation [Mandlekar et al., 2018], but demonstrations from amateurs may cause damages to the robot which is catastrophic in the real-world [Osa et al., 2018]. Similarly, demonstrations for autonomous driving can be collected from drivers in public roads [Fridman et al., 2017], but these demonstrations also include demonstrations that cause traffic accidents.

In this paper, we consider a realistic setting of IL where only diverse-quality demonstrations are available, while the level of demonstrator's expertise is absent. To tackle this challenging setting, we propose a new method called Variational Imitation Learning with Diverse-quality demonstrations (VILD). The central idea of VILD is to model the level of expertise via a probabilistic graphical model and learn it along with a reward function that represents an intention of experts' decision making. To

NeurIPS 2019 Workshop on Robot Learning: Control and Interaction in the Real World, Vancouver, Canada



(a) Expert demonstrations. (b) Diverse-quality demonstrations.

Figure 1: Graphical models describe expert demonstrations and diverse-quality demonstrations. Shaded and unshaded nodes indicate observed and unobserved random variables, respectively. Plate notations indicate that the sampling process is repeated for N times. $\mathbf{s}_t \in \mathcal{S}$ is a state with transition densities $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, $\mathbf{a}_t \in \mathcal{A}$ is an action with density $\pi^\star(\mathbf{a}_t|\mathbf{s}_t)$, $\mathbf{u}_t \in \mathcal{A}$ is a noisy action with density $p(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)$, and $k \in \{1, \ldots, K\}$ is an identification number with distribution $p(k)$.

scale up our model for large state and action spaces, we leverage the variational approach [Jordan et al., 1999], which can be implemented using reinforcement learning (RL) [Sutton and Barto, 1998]. To further improve data-efficiency when learning the reward function, we utilize importance sampling to re-weight a sampling distribution according to the estimated level of expertise. Experiments on continuous-control benchmarks and real-world demonstrations denote that VILD is robust against diverse-quality demonstrations and outperforms existing methods significantly. These results show that VILD is a scalable and data-efficient method for realistic settings of IL.

2 IL from diverse-quality demonstrations and its challenge

Imitation learning. One of limitations of RL is that it relies on a reward function which may be unavailable in practice. To address the above limitation, imitation learning (IL) was proposed [Schaal, 1999, Ng and Russell, 2000]. IL aims to learn an optimal policy from demonstrations that encode information about the optimal policy. A common assumption in IL is that, demonstrations are collected by $K \geqslant 1$ experts who execute actions $\mathbf{a}_t$ drawn from the optimal policy $\pi^\star(\mathbf{a}_t|\mathbf{s}_t)$ for every states $\mathbf{s}_t$. Under this assumption, we regard that these expert demonstrations are drawn independently from a probability density $p^\star(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}, k) = p(k)p_1(\mathbf{s}_1)\Pi_{t=1}^T p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi^\star(\mathbf{a}_t|\mathbf{s}_t)$, as shown in Figure 1(a). We note that k may be safely omitted, since all demonstrators are assumed to be experts.

IL has shown great successes in benchmark settings [Ho and Ermon, 2016, Fu et al., 2018]. However, practical applications of IL in the real-world is relatively few [Schroecker et al., 2019]. One of the main reasons is that most IL methods aim to learn with expert demonstrations. In practice, such demonstrations are often too costly to obtain due to a limited number of experts.

New setting: Diverse-quality demonstrations. To improve practicality, we consider a new problem called *IL with diverse-quality demonstrations*, where demonstrations are collected from demonstrators with different level of expertise. The graphical model in Figure 1(b) depicts the process of collecting such demonstrations from $K > 1$ demonstrators. Formally, we select the k-th demonstrator for demonstrations according to a probability distribution $p(k)$. After selecting k, for each time step t, the k-th demonstrator observes state $\mathbf{s}_t$ and samples action $\mathbf{a}_t$ using the optimal policy $\pi^\star(\mathbf{a}_t|\mathbf{s}_t)$. However, the demonstrator may not execute $\mathbf{a}_t$ in the MDP if this demonstrator is not expertised. Instead, he/she may sample an action $\mathbf{u}_t \in \mathcal{A}$ with another probability density $p(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)$ and execute it. Then, the next state $\mathbf{s}_{t+1}$ is observed with a probability density $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t)$ and the demonstrator continues making decision until time step T. We repeat this process for N times to collect a diverse-quality demonstration dataset $\mathcal{D}_\mathrm{d} = \{(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k)_n\}_{n=1}^N$. These demonstrations are regarded to be drawn independently from a probability density $p_\mathrm{d}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k) = p(k)p(\mathbf{s}_1)\prod_{t=1}^T p_1(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t)\int_\mathcal{A} \pi^\star(\mathbf{a}_t|\mathbf{s}_t)p(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)\mathrm{d}\mathbf{a}_t$. Since demonstrators may be amateurs, divere-quality demonstrations can be collected much more cheaply when compared to the expert demonstrations. The probability density $p(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)$ can be regarded as a noisy policy of the k-th demonstrator.

The deficiency of existing methods. We conjecture that existing IL methods are not suitable to learn with diverse-quality demonstrations according to p_d. Specifically, these methods always treat observed demonstrations as if they were drawn from $p^\star$. By comparing $p^\star$ and p_d, we can see that ex-

isting methods would learn $\pi(\mathbf{u}_t|\mathbf{s}_t)$ such that $\pi(\mathbf{u}_t|\mathbf{s}_t) \approx \Sigma_{k=1}^{K} p(k) \int_{\mathcal{A}} \pi^\star(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) \mathrm{d}\mathbf{a}_t$. In other words, they learn a policy that averages over decisions of all demonstrators. This would be problematic when amateurs are present, as averaged decisions of all demonstrators would be highly different from those of all experts. Worse yet, state distributions of amateurs and experts tend to be highly different, which often leads to the unstable learning: The learned policy oscillated between well-performed policy and poorly-performed policy. For these reasons, we believe that existing methods tend to learn a policy that achieves average performances, and are not suitable for handling the setting of diverse-quality demonstrations.

3 VILD: A robust method for diverse-quality demonstrations

Our model is based on a model of maximum entropy IRL (MaxEnt-IRL) [Ziebart et al., 2010]: $p_{\boldsymbol{\phi}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}) \propto p(\mathbf{s}_1) \Pi_{t=1}^{T} p_1(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) \exp(r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t))$. Based on this model, we propose to learn the reward function and the level of expertise by a model $p_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k) = p(k) p_1(\mathbf{s}_1) \Pi_{t=1}^{T} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t) \int_{\mathcal{A}} \exp(r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)) p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) \mathrm{d}\mathbf{a}_t / Z_{\boldsymbol{\phi},\boldsymbol{\omega}}$, where $\boldsymbol{\phi}$ and $\boldsymbol{\omega}$ are parameters and $Z_{\boldsymbol{\phi},\boldsymbol{\omega}}$ is the normalization term. To learn these parameters, we minimize the KL divergence from the data distribution to the model: $\mathrm{KL}(p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k) || p_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k))$. By rearranging and ignoring constant terms, minimizing this KL divergence is equivalent to solving an optimization problem $\max_{\boldsymbol{\phi},\boldsymbol{\omega}} f(\boldsymbol{\phi}, \boldsymbol{\omega}) - g(\boldsymbol{\phi}, \boldsymbol{\omega})$, where $f(\boldsymbol{\phi}, \boldsymbol{\omega}) = \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k)}[\Sigma_{t=1}^{T} \log(\int_{\mathcal{A}} \exp(r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)) p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) \mathrm{d}\mathbf{a}_t)]$ and $g(\boldsymbol{\phi}, \boldsymbol{\omega}) = \log Z_{\boldsymbol{\phi},\boldsymbol{\omega}}$. To solve this optimization, we need to compute the integrals over both state space $\mathcal{S}$ and action space $\mathcal{A}$. Computing these integrals is feasible for small state and action spaces, but is infeasible for large state and action spaces. To scale up our model to tasks with large state and action spaces, we leverage a variational approach which aims to lower-bound an integral by an expectation [Jordan et al., 1999].

Since we cannot lower-bound $f(\boldsymbol{\phi}, \boldsymbol{\omega}) - g(\boldsymbol{\phi}, \boldsymbol{\omega})$ directly, we propose to *separately* lower-bound f and g. Specifically, let $l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) = r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) + \log p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)$, by using a variational distribution $q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)$ with parameter $\boldsymbol{\psi}$, we obtain an inequality $f(\boldsymbol{\phi}, \boldsymbol{\omega}) \geqslant \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$, where $\mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}) = \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k)}[\Sigma_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}[l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)]]$. It is trivial to verify that the equality $f(\boldsymbol{\phi}, \boldsymbol{\omega}) = \max_{\boldsymbol{\psi}} \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$ holds [Murphy, 2013]. Meanwhile, by using a variational distribution $q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k)$ with parameter $\boldsymbol{\theta}$, we obtain an inequality $g(\boldsymbol{\phi}, \boldsymbol{\omega}) \geqslant \mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta})$, where $\mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta}) = \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)}[\Sigma_{t=1}^{T} l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k)]$. The lower-bound $\mathcal{G}$ resembles the maximum entropy RL (MaxEnt-RL) [Ziebart et al., 2010], and it can be shown that an equality $g(\boldsymbol{\phi}, \boldsymbol{\omega}) = \max_{\boldsymbol{\theta}} \mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta})$ holds. By using these lower-bounds, we have that $\max_{\boldsymbol{\phi},\boldsymbol{\omega}} f(\boldsymbol{\phi}, \boldsymbol{\omega}) - g(\boldsymbol{\phi}, \boldsymbol{\omega}) = \max_{\boldsymbol{\phi},\boldsymbol{\omega},\boldsymbol{\psi}} \min_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}) - \mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta})$. VILD learns the reward and expertise parameters by solving this max-min optimization problem.

We proceed by assuming that $q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k) = q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \boldsymbol{\Sigma})$ and $p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) = \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathbf{C}_{\boldsymbol{\omega}}(k))$. Under these assumptions, VILD solves $\max_{\boldsymbol{\phi},\boldsymbol{\omega},\boldsymbol{\psi}} \min_{\boldsymbol{\theta}} \mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta})$, where

$$\begin{aligned}\mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta}) = \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k)} \left[\textstyle\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)} \left[r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \frac{1}{2} \|\mathbf{u}_t - \mathbf{a}_t\|^2_{\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)} \right] + H(q_{\boldsymbol{\psi}}) \right] \\ - \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})} \left[\textstyle\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] + T \mathbb{E}_{p(k)} \left[\mathrm{trace}(\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k) \boldsymbol{\Sigma}) \right] / 2.\end{aligned}$$

Here, $\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}) = p_1(\mathbf{s}_1) \Pi_{t=1}^{T} \int_{\mathbb{R}} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t + \boldsymbol{\epsilon}_t) \mathcal{N}(\boldsymbol{\epsilon}_t|0, \boldsymbol{\Sigma}) \mathrm{d}\boldsymbol{\epsilon}_t q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)$ is a noisy trajectory density and $H(q_{\boldsymbol{\psi}}) = -\mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)} [\log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)]$ is the Shannon entropy. Minimizing $\mathcal{H}$ w.r.t. $\boldsymbol{\theta}$ resembles solving a MaxEnt-RL problem, except that trajectories are collected according to the noisy trajectory density. Therefore, this minimization problem can be implemented by existing RL methods, and $q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)$ can be regarded as an approximation of the optimal policy with reward function $r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)$. This policy imitates the optimal policy $\pi^\star$, which is the goal of IL. To improve the convergence rate of VILD when updating the reward parameter $\boldsymbol{\phi}$, we use importance sampling (IS). A pseudo-code and implementation details of VILD are provided in the appendix.

4 Experiments

We evaluate VILD against existing IL methods on Mujoco benchmark tasks and a robosuite reaching task [Fan et al., 2018]. For benchmarks, we use pre-trained RL agents as optimal policies and generate demonstrations by adding noises to actions. For robosuite, we use real-world crowdsourced demonstrations from Mandlekar et al. [2018]. More details of experiments are given in Appendix C.

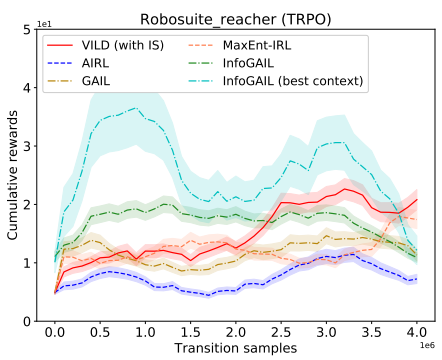


Figure 2: Performance of VILD with IS and baseline methods for the robosuite reaching task.

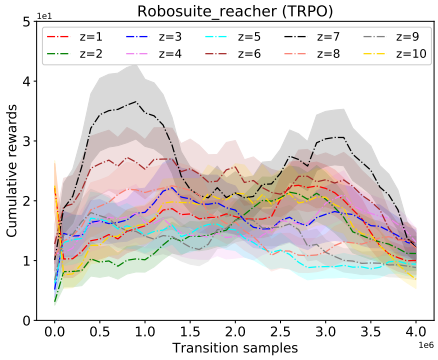


Figure 3: Performance of InfoGAIL with difffer-ent contexts z for the robosuite reaching task.

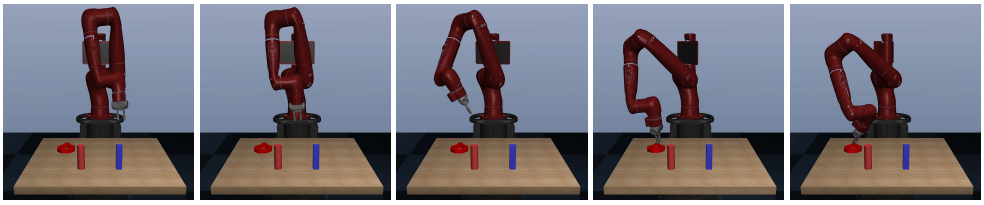


Figure 4: An example of trajectory generated by VILD in the robosuite reaching task. The goal is to control the robot's end-effector to reach the red object. The value of reward function (for performance evaluation) is inverse proportion to the distance between the end-effector and the red object.

Results on Mujoco benchmarks. Figure 6 in Appendix D shows the performance of VILD on Mujoco benchmark tasks. Clearly, VILD with IS significantly outperforms existing methods and quickly learns the optimal policies. On the other hand, existing methods perform very poorly and learn policy with averaged performance. This result supports our claim that low-quality demonstrations negatively affect the performance of existing IL methods.

Results on robosuite reacher. Figure 2 shows the performance of VILD on the robosuite reaching task. It can be seen that VILD performs better than GAIL, AIRL, and MaxEnt-IRL. VILD also performs better than InfoGAIL in terms of the final performance; InfoGAIL learns faster in the early stage of learning, but its performance saturates and VILD eventually outperforms InfoGAIL. These experimental results show that VILD is more robust against real-world demonstrations with diverse-quality when compared to existing state-of-the-art methods. An example of trajectory generated by VILD's policy is shown in Figure 4.

Figure 3 shows the performance of InfoGAIL with different context variables z [Li et al., 2017]. We can see that InfoGAIL performs well when the policy is conditioned on specific contexts, e.g., $z = 7$. Indeed, the best context during testing can improve the performance of InfoGAIL. The effectiveness of such an approach is demonstrated in Figure 2, where InfoGAIL (best context) performs very well. However, InfoGAIL (best context) is less practical than VILD, since choosing the best context requires an expert to evaluate the performance of all contexts. In contrast, the performance of VILD does not depend on contexts, since VILD does not learn a context-dependent policy. Moreover, the performance of InfoGAIL (best context) is quite unstable, and it is still outperformed by VILD in terms of the final performance.

5 Conclusion

In this paper, we explored a practical setting of IL where demonstrations have diverse-quality. We proposed a robust method called VILD, which learns both the reward function and the level of demonstrator's expertise by using the variational approach. Empirical results demonstrated that our work enables scalable and data-efficient IL under this practical setting.

References

Julien Audiffren, Michal Valko, Alessandro Lazaric, and Mohammad Ghavamzadeh. Maximum entropy semi-supervised inverse reinforcement learning. In *IJCAI*, pages 3315–3321. AAAI Press, 2015.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *CoRR*, abs/1606.01540, 2016.

Linxi Fan, Yuke Zhu, Jiren Zhu, Zihua Liu, Orien Zeng, Anchit Gupta, Joan Creus-Costa, Silvio Savarese, and Li Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In *Conference on Robot Learning*, 2018.

William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian J. Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. In *6th International Conference on Learning Representations, ICLR*, 2018.

Lex Fridman, Daniel E. Brown, Michael Glazer, William Angell, Spencer Dodd, Benedikt Jenik, Jack Terwilliger, Julia Kindelsberger, Li Ding, Sean Seaman, Hillary Abraham, Alea Mehler, Andrew Sipperley, Anthony Pettinato, Bobbie Seppelt, Linda Angell, Bruce Mehler, and Bryan Reimer. MIT autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation. *CoRR*, abs/1711.06976, 2017.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. 2018.

Shixiang (Shane) Gu, Timothy Lillicrap, Richard E Turner, Zoubin Ghahramani, Bernhard Schölkopf, and Sergey Levine. Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3846–3855. Curran Associates, Inc., 2017.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30*, pages 5769–5779, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems 29*, pages 4565–4573, 2016.

Matthew D. Hoffman and David M. Blei. Stochastic structured variational inference. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2015.

Edward L Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, November 1999. ISSN 0885-6125.

Kyungjae Lee, Sungjoon Choi, and Songhwai Oh. Inverse reinforcement learning with leveraged gaussian processes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 3907–3912, 2016.

Yunzhu Li, Jiaming Song, and Stefano Ermon. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations. In *Advances in Neural Information Processing Systems 30*, pages 3815–3825, 2017.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. ROBOTURK: A crowdsourcing platform for robotic skill learning through imitation. In *CoRL*, volume 87 of *Proceedings of Machine Learning Research*, pages 879–893. PMLR, 2018.

Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013.

Andrew Y. Ng and Stuart J. Russell. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670, 2000.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2): 1–179, 2018.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0-471-61977-9.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. 2005.

Stuart Russell. Learning agents for uncertain environments (extended abstract). In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 101–103. ACM, 1998. ISBN 1-58113-057-0.

Stefan Schaal. Is imitation learning the route to humanoid robots? 3(6):233–242, 1999.

Yannick Schroecker, Mel Vecerik, and Jon Scholz. Generative predecessor models for sample-efficient imitation learning. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SkeVsiAcYm`.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning, July 6-11, 2015, Lille, France*, 2015.

Kyriacos Shiarlis, João V. Messias, and Shimon Whiteson. Inverse Reinforcement Learning from Failure. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1060–1068, 2016.

David Silver, J. Andrew Bagnell, and Anthony Stentz. Learning autonomous driving styles and maneuvers from expert demonstration. In *Experimental Robotics - The 13th International Symposium on Experimental Robotics, ISER*, pages 371–386, 2012.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. 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, 2016.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning - an Introduction*. Adaptive computation and machine learning. MIT Press, 1998.

G. E. Uhlenbeck and L. S. Ornstein. On the theory of the brownian motion. *Physical Revview*, 1930.

Robert J. van Beers, Patrick Haggard, and Daniel M. Wolpert. The role of execution noise in movement variability. *Journal of Neurophysiology*, 91(2):1050–1063, 2004. doi: 10.1152/jn.00652.2003. URL `https://doi.org/10.1152/jn.00652.2003`. PMID: 14561687.

Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modeling Interaction via the Principle of Maximum Causal Entropy. In *Proceedings of the 27th International Conference on Machine Learning, June 21-24, 2010, Haifa, Israel*, 2010.

A Derivations

This section derives the lower-bounds of $f(\boldsymbol{\phi}, \boldsymbol{\omega})$ and $g(\boldsymbol{\phi}, \boldsymbol{\omega})$ presented in the paper. We also derive the objective function $\mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta})$ of VILD.

A.1 Lower-bound of f

Let $l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) = r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) + \log p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)$, we have that $f(\boldsymbol{\phi}, \boldsymbol{\omega}) = \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}\left[\sum_{t=1}^{T} f_t(\boldsymbol{\phi}, \boldsymbol{\omega})\right]$, where $f_t(\boldsymbol{\phi}, \boldsymbol{\omega}) = \log \int_{\mathcal{A}} \exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right) \mathrm{d}\mathbf{a}_t$. By using a variational distribution $q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)$ with parameter $\boldsymbol{\psi}$, we can bound $f_t(\boldsymbol{\phi}, \boldsymbol{\omega})$ from below by using the Jensen inequality as follows:

$$
\begin{aligned}
f_t(\boldsymbol{\phi}, \boldsymbol{\omega}) &= \log\left(\int_{\mathcal{A}} \exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right) \frac{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)} \mathrm{d}\mathbf{a}_t\right) \\
&\geqslant \int_{\mathcal{A}} q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k) \log\left(\exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right) \frac{1}{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\right) \mathrm{d}\mathbf{a}_t \\
&= \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\left[l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\right] \\
&= \mathcal{F}_t(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}). \qquad (1)
\end{aligned}
$$

Then, by using the linearity of expectation, we obtain the lower-bound of $f(\boldsymbol{\phi}, \boldsymbol{\omega})$ as follows:

$$
\begin{aligned}
f(\boldsymbol{\phi}, \boldsymbol{\omega}) &\geqslant \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}\left[\sum_{t=1}^{T} \mathcal{F}_t(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})\right] \\
&= \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}\left[\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\left[l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\right]\right] \\
&= \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}). \qquad (2)
\end{aligned}
$$

To verify that $f(\boldsymbol{\phi}, \boldsymbol{\omega}) = \max_{\boldsymbol{\psi}} \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$, we maximize $\mathcal{F}_t(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$ w.r.t. $q_{\boldsymbol{\psi}}$ under the constraint that $q_{\boldsymbol{\psi}}$ is a valid probability density, i.e., $q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k) > 0$ and $\int_{\mathcal{A}} q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\mathrm{d}\mathbf{a}_t = 1$. By setting the derivative of $\mathcal{F}_t(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$ w.r.t. $q_{\boldsymbol{\psi}}$ to zero, we obtain

$$
\begin{aligned}
q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k) &= \exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - 1\right) \\
&= \frac{\exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right)}{\int_{\mathcal{A}} \exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right) \mathrm{d}\mathbf{a}_t},
\end{aligned}
$$

where the last line follows from the constraint $\int_{\mathcal{A}} q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\mathrm{d}\mathbf{a}_t = 1$. To show that this is indeed the maximizer, we substitute $q_{\boldsymbol{\psi}^\star}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k) = \frac{\exp(l(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k))}{\int_{\mathcal{A}} \exp(l(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k))\mathrm{d}\mathbf{a}_t}$ into $\mathcal{F}_t(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$:

$$
\begin{aligned}
\mathcal{F}_t(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}^\star) &= \mathbb{E}_{q^\star_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\left[l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\psi}^\star}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\right] \\
&= \log\left(\int_{\mathcal{A}} \exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right) \mathrm{d}\mathbf{a}_t\right).
\end{aligned}
$$

This equality verifies that $f_t(\boldsymbol{\phi}, \boldsymbol{\omega}) = \max_{\boldsymbol{\psi}} \mathcal{F}_t(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$. Finally, by using the linearity of expectation, we have that $f(\boldsymbol{\phi}, \boldsymbol{\omega}) = \max_{\boldsymbol{\psi}} \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi})$.

A.2 Lower-bound of g

Next, we derive the lower-bound of $g(\boldsymbol{\phi}, \boldsymbol{\omega})$ presented in the paper. Recall that the function $g(\boldsymbol{\phi}, \boldsymbol{\omega}) = \log Z_{\boldsymbol{\phi},\boldsymbol{\omega}}$ is

$$
g(\boldsymbol{\phi}, \boldsymbol{\omega}) = \log\left(\sum_{k=1}^{K} p(k) \int \cdots \int_{(\mathcal{S}\times\mathcal{A}\times\mathcal{A})^T} p_1(\mathbf{s}_1) \prod_{t=1}^{T} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t) \exp\left(l(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right) \mathrm{d}\mathbf{s}_{1:T} \mathrm{d}\mathbf{u}_{1:T} \mathrm{d}\mathbf{a}_{1:T}\right).
$$

We use the structure variational approach [Hoffman and Blei, 2015], where the key idea is to predefine conditional dependency to ease computation. Specifically, we use a variational distribution

$q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k)$ with parameter $\boldsymbol{\theta}$ and define dependencies between states according to the transition probability of an MDP. With this variational distribution, we lower-bound g as follows:

$$
\begin{aligned}
g(\boldsymbol{\phi}, \boldsymbol{\omega}) &= \log \left(\sum_{k=1}^{K} p(k) \int \cdots \int_{(\mathcal{S}\times\mathcal{A}\times\mathcal{A})^T} p_1(\mathbf{s}_1) \prod_{t=1}^{T} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t) \exp\left(l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)\right) \right. \\
&\qquad \left. \times \frac{q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k)}{q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k)} \mathrm{d}\mathbf{s}_{1:T} \mathrm{d}\mathbf{u}_{1:T} \mathrm{d}\mathbf{a}_{1:T} \right) \\
&\geqslant \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k) \right] \\
&= \mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta}),
\end{aligned} \tag{3}
$$

where $\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k) = p(k)p_1(\mathbf{s}_1)\Pi_{t=1}^{T} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t) q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k)$. The optimal variational distribution $q_{\boldsymbol{\theta}^\star}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k)$ can be founded by maximizing $\mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta})$ w.r.t. $q_{\boldsymbol{\theta}}$. Solving this maximization problem is identical to solving a maximum entropy RL (MaxEnt-RL) problem [Ziebart et al., 2010] for an MDP defined by a tuple $\mathcal{M} = (\mathcal{S} \times \mathbb{N}_+, \mathcal{A} \times \mathcal{A}, p(\mathbf{s}', |\mathbf{s}, \mathbf{u})\mathbb{I}_{k=k'}, p_1(\mathbf{s}_1)p(k_1), l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}, \mathbf{a}, \mathbf{u}, k))$. Specifically, this MDP is defined with a state variable $(\mathbf{s}_t, k_t) \in \mathcal{S} \times \mathbb{N}$, an action variable $(\mathbf{a}_t, \mathbf{u}_t) \in \mathcal{A} \times \mathcal{A}$, a transition probability density $p(\mathbf{s}_{t+1}, |\mathbf{s}_t, \mathbf{u}_t)\mathbb{I}_{k_t=k_{t+1}}$, an initial state density $p_1(\mathbf{s}_1)p(k_1)$, and a reward function $l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k)$. Here, $\mathbb{I}_{a=b}$ is the indicator function which equals to 1 if $a = b$ and 0 otherwise. By adopting the optimality results of MaxEnt-RL [Ziebart et al., 2010, Haarnoja et al., 2018], we have $g(\boldsymbol{\phi}, \boldsymbol{\omega}) = \max_{\boldsymbol{\theta}} \mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta})$, where the optimal variational distribution is

$$
q_{\boldsymbol{\theta}^\star}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k) = \exp(Q(\mathbf{s}_t, k, \mathbf{a}_t, \mathbf{u}_t) - V(\mathbf{s}_t, k)). \tag{4}
$$

The functions Q and V are soft-value functions defined as

$$
Q(\mathbf{s}_t, k, \mathbf{a}_t, \mathbf{u}_t) = l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) + \mathbb{E}_{p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t)} \left[V(\mathbf{s}_{t+1}, k) \right], \tag{5}
$$

$$
V(\mathbf{s}_t, k) = \log \iint_{\mathcal{A}\times\mathcal{A}} \exp\left(Q(\mathbf{s}_t, k, \mathbf{a}_t, \mathbf{u}_t)\right) \mathrm{d}\mathbf{a}_t \mathrm{d}\mathbf{u}_t. \tag{6}
$$

A.3 Objective function $\mathcal{H}$ of VILD

This section derives the objective function $\mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta})$ from $\mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}) - \mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta})$. Specifically, we substitute the models $p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) = \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathbf{C}_{\boldsymbol{\omega}}(k))$ and $q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k) = q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \boldsymbol{\Sigma})$. First, we substitute $q_{\boldsymbol{\theta}}(\mathbf{a}_t, \mathbf{u}_t|\mathbf{s}_t, k) = q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \boldsymbol{\Sigma})$ into $\mathcal{G}$:

$$
\begin{aligned}
\mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta}) &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \boldsymbol{\Sigma}) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] \\
&= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) + \frac{1}{2}\|\mathbf{u}_t - \mathbf{a}_t\|^2_{\boldsymbol{\Sigma}^{-1}} - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] + c_1,
\end{aligned}
$$

where c_1 is a constant corresponding to the log-normalization term of the Gaussian distribution. Next, by using the re-parameterization trick, we rewrite $\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)$ as

$$
\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k) = p(k)p_1(\mathbf{s}_1) \prod_{t=1}^{T} p_1(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}_t)\mathcal{N}(\boldsymbol{\epsilon}_t|0, \boldsymbol{I}) q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t),
$$

where we use $\mathbf{u}_t = \mathbf{a}_t + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}_t$ with $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{\epsilon}_t|0, \boldsymbol{I})$. With this, the expectation of $\Sigma_{t=1}^{T}\|\mathbf{u}_t - \mathbf{a}_t\|^2_{\boldsymbol{\Sigma}^{-1}}$ over $\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)$ can be written as

$$
\begin{aligned}
\mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} \|\mathbf{u}_t - \mathbf{a}_t\|^2_{\boldsymbol{\Sigma}^{-1}} \right] &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} \|\mathbf{a}_t + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}_t - \mathbf{a}_t\|^2_{\boldsymbol{\Sigma}^{-1}} \right] \\
&= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}_t\|^2_{\boldsymbol{\Sigma}^{-1}} \right] \\
&= T d_{\mathbf{a}},
\end{aligned}
$$

which is a constant. Then, the quantity $\mathcal{G}$ can be expressed as

$$\mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta}) = \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] + c_1 + T d_{\mathbf{a}}.$$

By ignoring the constant, the optimization problem $\max_{\boldsymbol{\phi},\boldsymbol{\omega},\boldsymbol{\psi}} \min_{\boldsymbol{\theta}} \mathcal{F}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}) - \mathcal{G}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\theta})$ is equivalent to

$$\begin{aligned}\max_{\boldsymbol{\phi},\boldsymbol{\omega},\boldsymbol{\psi}} \min_{\boldsymbol{\theta}} \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)} \left[\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)} \left[l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k) \right] \right] \\ - \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right]. \end{aligned} \tag{7}$$

Our next step is to substitute $p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)$ by our choice of model. First, let us consider a Gaussian distribution $p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) = \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathbf{C}_{\boldsymbol{\omega}}(\mathbf{s}_t, k))$, where the covariance depends on state. With this model, the second term in Eq. (7) is given by

$$\begin{aligned}&\mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] \\ &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) + \log \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathbf{C}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] \\ &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \frac{1}{2} \|\mathbf{u}_t - \mathbf{a}_t\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)} - \frac{1}{2} \log |\mathbf{C}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)| - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] + c_2,\end{aligned}$$

where $c_2 = -\frac{d_{\mathbf{a}}}{2} \log 2\pi$ is a constant. By using the reparameterization trick, we write the expectation of $\Sigma_{t=1}^{T} \|\mathbf{u}_t - \mathbf{a}_t\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)}$ as follows:

$$\begin{aligned}\mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} \|\mathbf{u}_t - \mathbf{a}_t\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)} \right] &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} \|\mathbf{a}_t + \boldsymbol{\Sigma}^{1/2} \boldsymbol{\epsilon}_t - \mathbf{a}_t\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)} \right] \\ &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} \|\boldsymbol{\Sigma}^{1/2} \boldsymbol{\epsilon}_t\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)} \right].\end{aligned}$$

Using this equality, the second term in Eq. (7) is given by

$$\mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) - \frac{1}{2} \left(\|\boldsymbol{\Sigma}^{1/2} \boldsymbol{\epsilon}_t\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)} + \log |\mathbf{C}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)| \right) \right]. \tag{8}$$

Maximizing this quantity w.r.t. $\boldsymbol{\theta}$ has an implication as follows: $q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)$ maximizes the expected cumulative reward while avoiding states that are difficult for demonstrators. Specifically, a large value of $\mathbb{E}_{p(k)} \left[\log |\mathbf{C}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)| \right]$ indicates that demonstrators have a low level of expertise for state $\mathbf{s}_t$ on average, given by our estimated covariance. In other words, this state is difficult to accurately execute optimal actions for all demonstrators on averages. Since the policy $q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)$ should minimize $\mathbb{E}_{p(k)} \left[\log |\mathbf{C}_{\boldsymbol{\omega}}(\mathbf{s}_t, k)| \right]$, the policy should avoid states that are difficult for demonstrators. We expect that this property may improve exploration-exploitation trade-off in IL. Nonetheless, we leave an investigation of this property for future work, since this is not in the scope of the paper.

In this paper, we specify that the covariance does not depend on state: $\mathbf{C}_{\boldsymbol{\omega}}(\mathbf{s}_t, k) = \mathbf{C}_{\boldsymbol{\omega}}(k)$. This model specification enables us to simplify Eq. (8) as follows:

$$\begin{aligned}&\mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) - \frac{1}{2} \left(\|\boldsymbol{\Sigma}^{1/2} \boldsymbol{\epsilon}_t\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(k)} + \log |\mathbf{C}_{\boldsymbol{\omega}}(k)| \right) \right] \\ &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, \mathbf{a}_{1:T}, k)} \left[\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] - \frac{T}{2} \mathbb{E}_{p(k)\mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{I})} \left[\|\boldsymbol{\Sigma}^{1/2} \boldsymbol{\epsilon}\|^2_{\mathbf{C}^{-1}_{\boldsymbol{\omega}}(k)} + \log |\mathbf{C}_{\boldsymbol{\omega}}(k)| \right] \\ &= \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})} \left[\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) \right] - \frac{T}{2} \mathbb{E}_{p(k)} \left[\mathrm{Tr}(\mathbf{C}^{-1}_{\boldsymbol{\omega}}(k) \boldsymbol{\Sigma}) + \log |\mathbf{C}_{\boldsymbol{\omega}}(k)| \right],\end{aligned}$$

where $\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T}) = p_1(\mathbf{s}_1) \prod_{t=1}^{T} \int_{\mathcal{A}} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{u}_t) \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \boldsymbol{\Sigma}) \mathrm{d}\mathbf{u}_t q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)$. The last line follows from the quadratic form identity: $\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_t|0,\boldsymbol{I})}\left[\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}_t\|^2_{\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)}\right] = \mathrm{Tr}(\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)\boldsymbol{\Sigma})$. Next, we substitute $p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) = \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathbf{C}_{\boldsymbol{\omega}}(k))$ into the first term of Eq. (7).

$$
\begin{aligned}
&\mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}\left[\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\left[l_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{u}_t, k) - \log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\right]\right] \\
&= \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}\Bigg[\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\Big[r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \frac{1}{2}\|\mathbf{u}_t - \mathbf{a}_t\|^2_{\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)} - \frac{1}{2}\log|\mathbf{C}_{\boldsymbol{\omega}}(k)| \\
&\quad - \log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\Big]\Bigg] - T d_{\mathbf{a}} \log 2\pi/2. \qquad (9)
\end{aligned}
$$

Lastly, by ignoring constants, Eq. (7) is equivalent to $\max_{\boldsymbol{\phi},\boldsymbol{\omega},\boldsymbol{\psi}} \min_{\boldsymbol{\theta}} \mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta})$, where

$$
\begin{aligned}
\mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta}) = \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}&\left[\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\left[r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \frac{1}{2}\|\mathbf{u}_t - \mathbf{a}_t\|^2_{\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)} - \log q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)\right]\right] \\
&- \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})}\left[\sum_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t) - \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\right] + \frac{T}{2}\mathbb{E}_{p(k)}\left[\mathrm{Tr}(\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)\boldsymbol{\Sigma})\right].
\end{aligned}
$$

This concludes the derivation of VILD.

A.4 Importance sampling for reward learning

To improve the convergence rate of VILD when updating $\boldsymbol{\phi}$, we use importance sampling (IS). Specifically, by analyzing the gradient $\nabla_{\boldsymbol{\phi}}\mathcal{H} = \nabla_{\boldsymbol{\phi}}\{\mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}[\Sigma_{t=1}^{T}\mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}[r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)]] - \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})}[\Sigma_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)]\}$, we can see that the reward function is updated to maximize the expected cumulative reward obtained by demonstrators and $q_{\boldsymbol{\psi}}$, while minimizing the expected cumulative reward obtained by $q_{\boldsymbol{\theta}}$. However, low-quality demonstrations often yield low reward values. For this reason, stochastic gradients estimated by these demonstrations tend to be uninformative, which leads to slow convergence and poor data-efficiency.

To avoid estimating such uninformative gradients, we use IS to estimate gradients using high-quality demonstrations which are sampled with high probability. Briefly, IS is a technique for estimating an expectation over a distribution by using samples from a different distribution [Robert and Casella, 2005]. For VILD, we propose to sample k from a distribution $\tilde{p}(k) \propto \|\mathrm{vec}(\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k))\|_1$. This distribution assigns high probabilities to demonstrators with high estimated level of expertise (i.e., demonstrators with a small $\mathbf{C}_{\boldsymbol{\omega}}(k)$). With this distribution, the estimated gradients tend to be more informative which leads to a faster convergence. To reduce a sampling bias, we use a truncated importance weight: $w(k) = \min(p(k)/\tilde{p}(k), 1)$ [Ionides, 2008], which leads to an IS gradient: $\nabla_{\boldsymbol{\phi}}\mathcal{H}_{\mathrm{IS}} = \nabla_{\boldsymbol{\phi}}\{\mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)\tilde{p}(k)}[w(k)\Sigma_{t=1}^{T}\mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}[r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)]] - \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})}[\Sigma_{t=1}^{T} r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)]\}$. Computing $w(k)$ requires $p(k)$, which can be estimated accurately since k is a discrete random variable. For simplicity, we assume that $p(k)$ is a uniform distribution.

B Implementation details

We implement VILD using the PyTorch deep learning framework. For all function approximators, we use neural networks with 2 hidden-layers of 100 $\tanh$ units, except for the Humanoid task and the robosuite reaching task where we use neural networks with 2 hidden-layers of 100 relu units. We optimize parameters $\boldsymbol{\phi}$, $\boldsymbol{\omega}$, and $\boldsymbol{\psi}$ by Adam with step-size 3×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and mini-batch size 256. To optimize the policy parameter $\boldsymbol{\theta}$, we use trust region policy optimization (TRPO) [Schulman et al., 2015] with batch size 1000, except on the Humanoid task where we use soft actor-critic (SAC) [Haarnoja et al., 2018] with mini-batch size 256. Note that TRPO is an on-policy RL method that uses only trajectories collected by the current policy, while SAC is an off-policy RL method that use trajectories collected by previous policies. On-policy methods are generally more stable than off-policy methods, while off-policy methods are generally more data-efficient [Gu et al., 2017]. We use SAC for Humanoid mainly due to its high data-efficiency. When SAC is used, we

Algorithm 1 VILD: Variational Imitation Learning with Diverse-quality demonstrations

1: **Input:** Diverse-quality demonstrations $\mathcal{D}_{\mathrm{d}} = \{(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}, k)_n\}_{n=1}^N$ and a replay buffer $\mathcal{B} = \varnothing$.
2: **while** Not converge **do**
3: **while** $|\mathcal{B}| < B$ with batch size B **do** ▷ Collect samples from $\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})$
4: Sample $\mathbf{a}_t \sim q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)$ and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{\epsilon}_t|\mathbf{0}, \boldsymbol{\Sigma})$.
5: Execute $\mathbf{a}_t + \boldsymbol{\epsilon}_t$, observe $\mathbf{s}'_t \sim p(\mathbf{s}'_t|\mathbf{s}_t, \mathbf{a}_t + \boldsymbol{\epsilon}_t)$, and include $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)$ into $\mathcal{B}$
6: **end while**
7: Update $q_{\boldsymbol{\psi}}$ by an estimate of $\nabla_{\boldsymbol{\psi}} \mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta})$.
8: Update $p_{\boldsymbol{\omega}}$ by an estimate of $\nabla_{\boldsymbol{\omega}} \mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta}) + \nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega})$.
9: Update $r_{\boldsymbol{\phi}}$ by an estimate of $\nabla_{\boldsymbol{\phi}} \mathcal{H}_{\mathrm{IS}}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta})$.
10: Update $q_{\boldsymbol{\theta}}$ by an RL method (e.g., TRPO or SAC) with reward function $r_{\boldsymbol{\phi}}$.
11: **end while**

also use trajectories collected by previous policies to approximate the expectation over the trajectory density $\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})$.

For the distribution $p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k) = \mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathbf{C}_{\boldsymbol{\omega}}(k))$, we use diagonal covariances $\mathbf{C}_{\boldsymbol{\omega}}(k) = \mathrm{diag}(\mathbf{c}_k)$, where $\boldsymbol{\omega} = \{\mathbf{c}_k\}_{k=1}^K$ and $\mathbf{c}_k \in \mathbb{R}_+^{d_{\mathbf{a}}}$ are parameter vectors to be learned. For the distribution $q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)$, we use a Gaussian distribution with diagonal covariance, where the mean and logarithm of the standard deviation are the outputs of neural networks. Since k is a discrete variable, we represent $q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)$ by neural networks that have K output heads and take input vectors $(\mathbf{s}_t, \mathbf{u}_t)$; The k-th output head corresponds to (the mean and log-standard-deviation of) $q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)$. We also pre-train the mean function of $q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)$, by performing least-squares regression for 1000 gradient steps with target value $\mathbf{u}_t$. This pre-training is done to obtain reasonable initial predictions. For the policy $q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)$, we use a Gaussian policy with diagonal covariance, where the mean and logarithm of the standard deviation are outputs of neural networks. We use $\boldsymbol{\Sigma} = 10^{-8}\boldsymbol{I}$ in experiments.

To control exploration-exploitation trade-off, we use an entropy coefficient $\alpha = 0.0001$ in TRPO. In SAC, the value of α is optimized so that the policy has a certain value of entropy, as described by Haarnoja et al. [2018]. Note that including α in VILD is equivalent to rescaling quantities in the model by α, i.e., $\exp(r_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)/\alpha)$ and $(p_{\boldsymbol{\omega}}(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k))^{\frac{1}{\alpha}}$. A discount factor $0 < \gamma < 1$ may be included similarly, and we use $\gamma = 0.99$ in experiments.

For all methods, we regularize the reward/discriminator function by the gradient penalty [Gulrajani et al., 2017] with coefficient 10, since it was previously shown to improve performance of generative adversarial learning methods. For methods that learn a reward function, namely VILD, AIRL, and MaxEnt-IRL, we apply a sigmoid function to the output of a reward network to bound reward values. We found that without the bounds, reward values of the agent can be highly negative in the early stage of learning, which makes RL methods prematurely converge to poor policies. An explanation of this phenomenon is that, in MDPs with large state and action spaces, distribution of demonstrations and distribution of agent's trajectories are not overlapped in the early stage of learning. In such a scenario, it is trivial to learn a reward function which tends to positive-infinity values for demonstrations and negative-infinity values for agent's trajectories. While the gradient penalty regularizer slightly remedies this issue, we found that the regularizer alone is insufficient to prevent this scenario. Moreover, for VILD, it is beneficial to bound the reward function to control a trade-off between the immediate reward and the squared error when optimizing $\boldsymbol{\psi}$.

A pseudo-code of VILD with IS is given in Algorithm 1, where the reward parameter is updated by IS gradient in line 8. For VILD without IS, the reward parameter is instead updated by an estimate of $\nabla_{\boldsymbol{\phi}} \mathcal{H}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta})$. The regularizer $L(\boldsymbol{\omega}) = T\mathbb{E}_{p(k)}[\log|\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)|]/2$ penalizes large value of $\mathbf{C}_{\boldsymbol{\omega}}(k)$. A source-code of our implementation will be publicly available.

C Experiment Details

In this section, we describe experimental settings and data generation.

Table 1: Performance of a random policy π_0, the optimal policy $\pi^\star$, and demonstrators with the Gaussian noisy policy.

σ_k	Cheetah	Ant	Walker	Humanoid
(π_0)	-0.58	995	131	222
$(\pi^\star)$	4624	4349	4963	5093
0.01	4311	3985	4434	4315
0.05	3978	3861	3486	5140
0.01	4019	3514	4651	5189
0.25	1853	536	4362	3628
0.40	1090	227	467	5220
0.6	567	-73	523	2593
0.7	267	-208	332	1744
0.8	-45	-979	283	735
0.9	-399	-328	255	538
1.0	-177	-203	249	361

Table 2: Performance of a random policy π_0, the optimal policy $\pi^\star$, and demonstrators with the TSD noisy policy.

σ_k	Cheetah	Ant	Walker	Humanoid
(π_0)	-0.58	995	131	222
$(\pi^\star)$	4624	4349	4963	5093
0.01	4362	3758	4695	5130
0.05	4015	3623	4528	5099
0.01	3741	3368	2362	5195
0.25	1301	873	644	1675
0.40	-203	231	302	610
0.6	-230	-51	29	249
0.7	-249	-37	24	221
0.8	-416	-567	14	191
0.9	-389	-751	7	178
1.0	-424	-269	4	169

C.1 Experimental setting and data generation for benchmark tasks

For the benchmark experiment, we evaluate VILD on four continuous-control benchmark tasks from OpenAI gym platform [Brockman et al., 2016] with the Mujoco physics simulator: HalfCheetah, Ant, Walker2d, and Humanoid. To obtain the optimal policy for generating demonstrations, we use the ground-truth reward function of each task to pre-train $\pi^\star$ with TRPO. We generate diverse-quality demonstrations by using $K = 10$ demonstrators according to the graphical model in Figure 1(b). We consider two types of the noisy policy $p(\mathbf{u}_t|\mathbf{s}_t, \mathbf{a}_t, k)$: a Gaussian noisy policy and a time-signal-dependent (TSD) noisy policy.

Gaussian noisy policy. We use a Gaussian noisy policy $\mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \sigma_k^2 \boldsymbol{I})$ with a constant covariance. The value of σ_k for each of the 10 demonstrators is $0.01, 0.05, 0.1, 0.25, 0.4, 0.6, 0.7, 0.8, 0.9$ and 1.0, respectively. Note that our model assumption on $p_{\boldsymbol{\omega}}$ corresponds to this Gaussian noisy policy. Table 1 shows the performance of demonstrators (in terms of cumulative ground-truth rewards) with this Gaussian noisy policy. A random policy π_0 is an initial policy neural network for learning; The network weights are initialized such that the magnitude of actions is small. Note that this initialization scheme is a common practice in deep RL [Gu et al., 2017].

TSD noisy policy. To make learning more challenging, we generate demonstrations according to a noise characteristic of human motor control, where a magnitude of noises is proportion to a magnitude of actions and increases with execution time [van Beers et al., 2004]. Specifically, we generate demonstrations using a Gaussian distribution $\mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathrm{diag}(\mathbf{b}_k(t) \times \|\mathbf{a}_t\|_1/d_{\mathbf{a}}))$, where the covariance is proportion to the magnitude of action and depends on time steps and $\times$ denotes an element-wise product. We call this policy time-signal-dependent (TSD) noisy policy. Here, $\mathbf{b}_k(t)$ is a sample of a noise process whose noise variance increases over time. We obtain this noise process for the k-th demonstrator by reversing Ornstein–Uhlenbeck (OU) processes with parameters $\theta = 0.15$ and $\sigma = \sigma_k$ [Uhlenbeck and Ornstein, 1930][1]. The value of σ_k for each demonstrator is $0.01, 0.05, 0.1, 0.25, 0.4, 0.6, 0.7, 0.8, 0.9$, and 1.0, respectively. Table 2 shows the performance of demonstrators with this TSD noisy policy. Learning from demonstrations generated by TSD is challenging; The Gaussian model of $p_{\boldsymbol{\omega}}$ cannot perfectly model the TSD noisy policy, since the ground-truth variance is a function of actions and time steps.

C.2 Experimental setting for robosuite reaching task

For real-world data, we use a robot control task from the robosuite environment Fan et al. [2018] and a crowdsourced demonstration dataset from Mandlekar et al. [2018][2]. These demonstrations are collected for object-manipulation tasks such as assembly tasks. These object-manipulation tasks require the agent to perform three subtasks: reaching, picking, and placing. In our preliminary

[1] OU process is commonly used to generate time-correlated noises where the noise variance decays towards zero. We reserve this process along the time axis, so that the noise variance grows over time.

[2] We use the publicly available dataset: `http://roboturk.stanford.edu/dataset.html`

experiments, none of IL methods successfully learns object-manipulation policies, since the agent often fails at picking the object. We expect that a hierarchical policy is necessary to perform these manipulation tasks, due to the hierarchical structure (i.e., subtasks) of these tasks. Since hierarchical IL is not in the scope of this paper, we consider the subtask of reaching where non-hierarchical policies suffice. We leave an extension of VILD to hierarchical policy for future work.

In this experiment, we consider the subtask of reaching, which is still challenging for IL due to diverse quality of crowdsourced demonstrations. To obtain reaching demonstrations from the original object-manipulation demonstrations (we use the *SawyerNutAssemblyRound* dataset), we terminate demonstrations after the robot's end-effector contacts the target object. After applying such a termination procedure, the dataset used in this experiment consists of 10 randomly chosen demonstrations ($N = 10$) whose length T is approximately 500 time steps. The number of state-action pairs in this demonstration dataset is approximately 5000. Since we do not know the actual number of demonstrators that collected these $N = 10$ demonstrations, we set $K = N$ and $k = n$. We use true states of the robot and do not use visual observations. Since the reaching task does not require picking the object, we disable the gripper control command of the robot. The state space of this task is $\mathcal{S} \subseteq \mathbb{R}^{44}$, and the action space of this task is $\mathcal{A} \subseteq \mathbb{R}^7$. Figure 5 shows three examples of demonstrations used in this experiment. We can notice the differences in qualities of demonstrations, e.g., demonstration 2 is better than demonstration 1 since the robot reaches the object faster.

The performance of learned policies are evaluated using a reward function whose values are inverse proportion to the distance between the object and the end-effector (i.e., small distance yields high reward). We repeat the experiment for 5 trials using the same dataset and report the average performance (undiscounted cumulative rewards). For each trial, we generate 100 test trajectories for evaluating the performance. Note that the number of test trajectories in this experiment is larger than that in the benchmark experiments. This is because the initial states of this reaching task is much more varied than those in benchmark tasks. We do not evaluate VILD without IS and VAIL, since in benchmarks VILD with IS performs better than VILD without IS and VAIL is comparable to GAIL.

For all methods, we use neural networks with 2 hidden-layers of 100 relu units. We update policy parameters by TRPO with the same hyper-parameters as the benchmark experiments. We pre-train the mean of Gaussian policies for all methods by behavior cloning (i.e., we apply 1000 gradient descent steps of least-squares regression). To pre-train InfoGAIL which learns a context-dependent policy, we use the variable k as context for pre-training. For VILD, we apply the log-sigmoid function to the reward function. Specifically, we parameterize the reward function as $r_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a}) = \log D_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a})$ where $D_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a}) = \frac{\exp(d_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a}))}{\exp(d_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a})) + 1}$ and $d_{\boldsymbol{\phi}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. We also apply a substitution $-\log D_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a}) \to \log(1 - D_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a}))$, which is a common practice in GAN literature [Fedus et al., 2018]. By doing so, we obtain an objective of VILD that closely resembles the objective of GAIL:

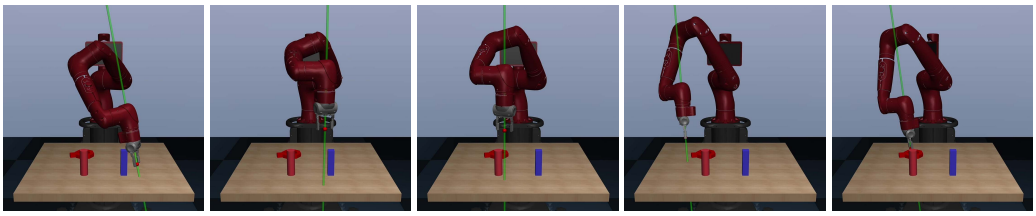
$$\mathcal{H}_{\log}(\boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\psi}, \boldsymbol{\theta}) = \mathbb{E}_{p_{\mathrm{d}}(\mathbf{s}_{1:T}, \mathbf{u}_{1:T}|k)p(k)}\Big[\sum_{t=1}^{T} \mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{a}_t|\mathbf{s}_t, \mathbf{u}_t, k)}\Big[\log D_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a}) - \tfrac{1}{2}\|\mathbf{u}_t - \mathbf{a}_t\|^2_{\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)}\Big] + H_t(q_{\boldsymbol{\psi}})\Big]$$
$$+ \mathbb{E}_{\widetilde{q}_{\boldsymbol{\theta}}(\mathbf{s}_{1:T}, \mathbf{a}_{1:T})}\Big[\sum_{t=1}^{T} \log(1 - D_{\boldsymbol{\phi}}(\mathbf{s}, \mathbf{a})) + \log q_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t)\Big] + \frac{T}{2}\mathbb{E}_{p(k)}\big[\mathrm{Tr}(\mathbf{C}_{\boldsymbol{\omega}}^{-1}(k)\boldsymbol{\Sigma})\big].$$

We use this variant of VILD in this experiment since it performs better than VILD with the standard reward function. Although we omit the IS distribution in this equation, we use IS in this experiment.

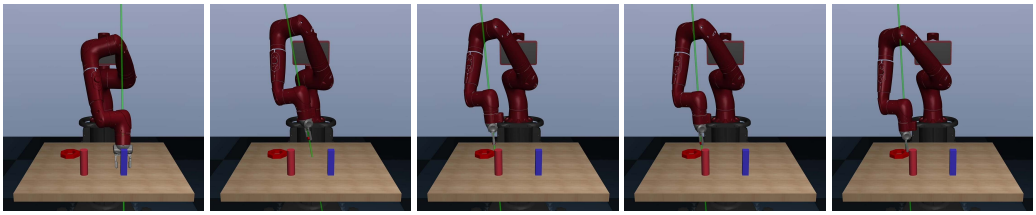
D Additional experimental results

Results against online IL methods. Figure 6 shows the learning curves of VILD and existing online IL methods against the number of transition samples. It can be seen that for both types of noisy policy, VILD with and without IS outperform existing methods overall, except on the Humanoid tasks where most methods achieve comparable performance.

Accuracy of estimated expertise parameter. Figure 7 shows the estimated parameters $\boldsymbol{\omega} = \{\mathbf{c}_k\}_{k=1}^K$ of $\mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \mathrm{diag}(\mathbf{c}_k))$ and the ground-truth variance $\{\sigma_k^2\}_{k=1}^K$ of the Gaussian noisy policy $\mathcal{N}(\mathbf{u}_t|\mathbf{a}_t, \sigma_k^2 \boldsymbol{I})$. VILD learns an accurate ranking of the variance compared to the ground-truth. The values of these parameters are also quite accurate compared to the ground truth, except for demonstrators with low-levels of expertise. A possible reason for this phenomena is that low-quality demonstrations are highly dissimilar, which makes learning the expertise more challenging.

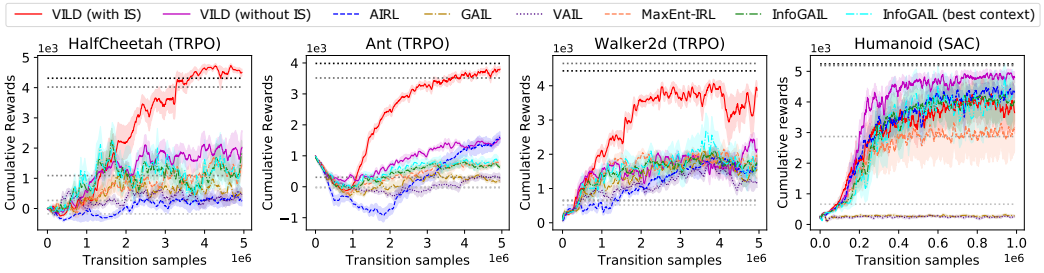


(a) Demonstration number 1.

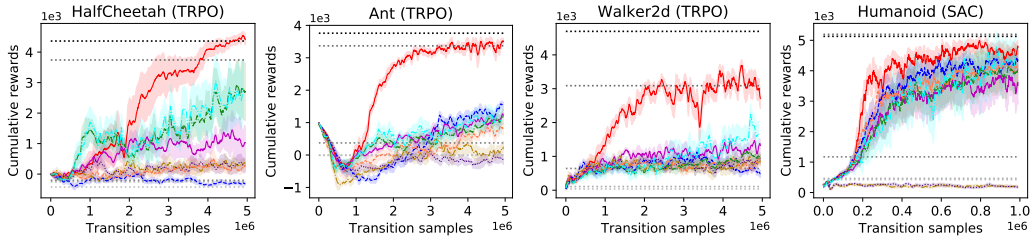


(b) Demonstration number 2.

Figure 5: Two examples of crowdsourced demonstrations in the robosuite reaching experiment.



(a) Performance of online IL methods when demonstrations are generated using Gaussian noisy policy.



(b) Performance of online IL methods when demonstrations are generated using TSD noisy policy.

Figure 6: Performance averaged over 5 trials of online IL methods against the number of transition samples. Horizontal dotted lines indicate performance of $k = 1, 3, 5, 7, 10$ demonstrators.

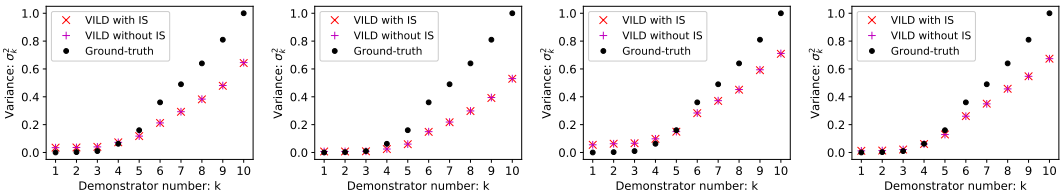


Figure 7: Expertise parameters $\boldsymbol{\omega} = \{\mathbf{c}_k\}_{k=1}^K$ learned by VILD and the ground-truth $\{\sigma_k^2\}_{k=1}^K$ for the Gaussian noisy policy. For VILD, we report the value of $\|\mathbf{c}_k\|_1/d_{\mathbf{a}}$.