Masked Imitation Learning: Discovering Environment-Invariant Modalities in Multimodal Demonstrations

Yilun Hao*, Ruinan Wang*, Zhangjie Cao1, Zihan Wang1, Yuchen Cui1, Dorsa Sadigh1

Abstract—Multimodal demonstrations provide robots with an abundance of information to make sense of the world. However, such abundance may not always lead to good performance when it comes to learning sensorimotor control policies from human demonstrations. Extraneous data modalities can lead to state over-specification, where the state contains modalities that are not only useless for decision-making but also can change data distribution across environments. State over-specification leads to issues such as the learned policy not generalizing outside of the training data distribution. In this work, we propose Masked Imitation Learning (MIL) to address state over-specification by selectively using informative modalities. Specifically, we design a masked policy network with a binary mask to block certain modalities. We develop a bi-level optimization algorithm that learns this mask to accurately filter over-specified modalities. We demonstrate empirically that MIL outperforms baseline algorithms in simulated domains and effectively recovers the environment-invariant modalities on a multimodal dataset collected on a real robot. Videos and supplemental details are at: https://tinyurl.com/masked-il

I. INTRODUCTION

Humans are born with the ability to perceive and integrate multiple sources of sensory information, including vision, touch, sound, and proprioception. Cognitive science research has demonstrated that humans form a coherent and robust perception of the world by efficiently integrating multiple sensory information [1, 2]. A robust perception is the foundation for sensorimotor control. To mimic human sensing, modern robotic systems are often equipped with multitudes of sensors. Such multimodal sensory information is important for solving robotics tasks [3–5].

Multimodal sensory data provides abundant information for decision making and much research effort has been dedicated to developing better robot sensing capabilities [6–8]. However, somewhat unintuitively, more data modalities do not necessarily promise higher performance for sensorimotor policies learned from such data. Multiple prior works have observed that when learning from multimodal sensory data, using a subset of input modalities can lead to better task performance than using all modalities [9–12]. This phenomenon is caused by state over-specification—the state contains extraneous data modalities that do not provide useful information for solving the task but can introduce a different data distribution across environments. Consider the example in Fig. 1, a robot is trained to perform a pick-and-place task in simulation from three different modalities (RGB image, depth image, and proprioception). To locate and pick up the object, depth and proprioception information are sufficient. The remaining modality of RGB image that changes from simulation to the real setting over-specifies the state for this particular task. When such extraneous information is used by learned policies to predict action, they are less likely to generalize at test time, especially when the testing environment changes, e.g., when training in simulation and testing the policy on the real robot (see Fig. 1). Prior work in learning from multimodal data only focuses on fusing various sensory modalities [5, 10, 13–16], but do not explicitly address the potential state over-specification problem. In this paper, we focus on addressing this problem to avoid overfitting to training data when learning from multimodal data. Going back to the example in Fig. 1, if we remove the RGB modality, the robot still has sufficient information for performing the task and now the testing input looks more aligned with training data such that policy trained in simulation may generalize to the real environment. Our key insight is that, for a particular task, we need to be selective about what modalities the policy relies on for decision-making to avoid state over-specification.

We propose Masked Imitation Learning (MIL), which learns a binary mask for each modality to decide whether the modality should be used for action prediction. MIL is a bi-level optimization algorithm: in the inner-level, MIL learns a policy embedded with a fixed mask on the training datasets using the training loss; in the outer-level, MIL updates the mask according to the validation loss evaluated on the validation dataset for the policy learned for each mask in the inner loop. The validation loss selects a policy that generalizes well to the validation dataset, which is more likely to be learned with no over-specified modalities, i.e. the corresponding mask removes over-specified modalities. We demonstrate the effectiveness of MIL empirically on several robotic tasks both in simulation including existing robotics datasets such as Robominic [11] and on multimodal data collected for a real robot manipulation task. We show learning from a selected set of modalities can improve the performance by 5.6 × than learning from all modalities in certain domain.

II. RELATED WORK

Our work addresses the state over-specification problem experienced by imitation learning from multimodal sensory data and therefore is closely related to prior work in learning from multimodal sensory data, imitation learning, and the broad area of invariant representation learning.

Learning from Multimodal Sensory Data. Motivated by the potential of leveraging information from multiple sensory modalities, prior works have explored using multimodal data for robot learning. These works mainly focus on what data
multimodal sensory input should be included for robot learning, where single-view images [3, 15, 17], multi-view images [13], haptic data [3, 17–19], range sensing [14, 20], audio [4, 21, 22], depth images [15] and 3D point clouds [5] are adopted to learn different manipulation and navigation tasks. These works demonstrate that, for a given task, one can leverage a comprehensive set of modalities to provide the necessary information for decision-making. Prior works have also investigated how to learn a robust state representation from multimodal sensory data using auxiliary objectives for better test-time generalization [15, 16, 23, 24].

Finally, a growing body of work also focuses on using multimodal data to learn end-to-end sensorimotor policies [1, 5, 10, 13–15, 21]. However, these works do not consider the state over-specification problem that can occur due to extraneous modalities. We show that overfitting — while being overlooked by most prior works — is a common problem when learning from multimodal data, and we address it by our masked imitation learning approach.

**Imitation Learning.** Imitation learning aims to learn a policy from demonstrations [25, 26]. Behavioral cloning [27] is the simplest form of imitation learning that treats the problem as supervised learning but often suffers from compounding errors at test time since the test data is not independent and identically distributed (i.i.d.). More advanced imitation learning techniques can address overfitting by rolling out the learned policies in testing environments and minimize the distributional shift [28–33]. However, rolling out agent policies is not only expensive but also often unsafe for real-world robotics applications. In contrast, our algorithm learns from offline data and does not need to iteratively interact with the environment, and at the same time can be easily adapted to learn the environment-invariant modality mask in an online manner.

**Invariant Representation Learning.** Deep neural networks are known to have the capacity to memorize noise or pick up spurious correlations [34, 35]. To reduce environment-specific overfitting, techniques including invariant risk minimization [36, 37], self-training [38, 39], dropout [40] and feature selection [41–43] are proposed to focus more on features with causal relationships to the outcome. However, all of these methods are only verified for non-robotics tasks. For robotics tasks, several works propose information bottleneck to learn the task-relevant representation, which is invariant across domains [44, 45], but these approaches require a well-defined reward and interactions with the environment. Invariant risk minimization games is a theoretical framework to learn an invariant policy in many different environments to reduce the effect of spurious features [46]. However, creating the set of environments that capture all the variations of spurious features is quite challenging especially in robotics domains. To address this, domain randomization approaches [47–49] try creating diverse environments by randomizing factors such as texture, lighting, etc., but a large number of variations of these factors, which often need to be done in simulation, might still not be able to capture all the spurious features. Inspired by the idea of invariant risk minimization that explicitly leverages the notion of environments, our proposed method learns to mask out extraneous modalities that lead to poor generalization error in the validation environment so that our learned policies do not suffer from overfitting.

## III. Problem Setting

We consider sequential decision-making problems modeled as Markov Decision Processes (MDPs). An MDP is defined by the tuple \( \langle S, A, T, R \rangle \), where: \( S \) and \( A \) are the state space and action space; \( T : S \times A \rightarrow S \) is a transition probability function; and \( R \) is a reward function. Here, we focus on deterministic MDPs but our method can be easily extended to the stochastic case. In this paper, we focus on the setting where the state space \( S \) consists of \( M \) modalities: \( S = \times_{i=1}^{M} S_{i} \), where each \( S_{i} \) indicates the state space of a modality. A trajectory \( \tau = \{(s_{0}, a_{0}), (s_{1}, a_{1}), ..., (s_{n}, a_{n})\} \) is a sequence of state-action pairs, where every state at time \( t \) has \( M \) modalities \( s_{t} = \{s_{t}^{1}, ..., s_{t}^{M}\} \). The return of a trajectory is the sum of rewards \( \sum_{t=0}^{n} |R(s_{t}^{M})| \). Let \( e \in E \) denote an environment, which we define as a subspace of states \( S_{e} \subseteq S \) that are reachable by the transition function \( T \) when initialized at \( s_{0} \in S_{e} \). A policy is a mapping from states...
to actions, $\pi : S \rightarrow A$. Similarly, an expert policy in an environment $e$ is $\pi^E : S_e \rightarrow A$ that maximizes the expected return. Expert demonstrations can be sampled from this policy to create a dataset $D_e = \{\tau_0, \tau_1, ..., \tau_k\}$ of size $|D_e|$. The goal of imitation learning is to learn a policy $\pi$ that generalizes across environments from multimodal demonstrations. Specifically, we have a training environment $e_{\text{train}}$, a validation environment $e_{\text{val}}$, and a testing environment $e_{\text{test}}$. Given a training dataset $D_{\text{train}}$ with $|D_{\text{train}}|$ demonstrations collected from $e_{\text{train}}$ and a validation dataset $D_{\text{val}}$ with $|D_{\text{val}}|$ demonstrations collected from $e_{\text{val}}$, our goal is to learn a policy that achieves high performance in $e_{\text{test}}$ using the data in $D_{\text{train}}$ and $D_{\text{val}}$ ($|D_{\text{train}}| \gg |D_{\text{val}}|$). We assume that, when providing demonstrations, the expert has access to the same raw state and data modalities as the learning agent does.

Imitation learning algorithms often learn a policy that minimizes the training loss $L$ on the training dataset. In practice, this loss function $L$ is usually the maximum likelihood loss or L2 distance for continuous action spaces or a cross-entropy loss for discrete action spaces. During training, the best model is selected by validation loss, which is of the same form as the training loss but is evaluated on the validation dataset $D_{\text{val}}$. Overfitting is a phenomenon that the model learned by the training loss on $D_{\text{train}}$ and selected by the validation loss on $D_{\text{val}}$ performs poorly on the test environment $e_{\text{test}}$.

In this work, we focus on addressing overfitting caused by state over-specification, which happens when the state observed by the imitating agent contains more modalities than what was used by the demonstrator to perform the task, and such modalities change the data distribution across environments. For example, to perform the task of cutting an apple into slices, a human demonstrator only needs the location, shape, and size of the apple while the texture of the cutting board or the color of the knife handle is useless and may change in different kitchens. Though both observing the full state information, a human demonstrator selects the useful and generalizable modalities in the state to make decisions, enabling humans to perform the task across environments, but an imitating agent may overfit to the over-specified modalities, e.g., if we have only observed knives with green handles in training, the agent at test time can only cut the apple when the knife’s handle is green.

**Problem Statement.** Let $s^M_{i=0} \in S$ denote the full state with $M$ modalities and $s^N_{i=0} \in S^*$ denote the modalities of size $N \leq M$, which the expert uses to make decisions ($S^*$ is the environment-invariant modalities for the demonstrated task). We let $\tilde{s}_{i=0}^{M-N} \in S$ of size $M-N$ denote the extraneous modalities that are not used by the expert to act. Our goal is to find a policy $\pi_0 : S \rightarrow A$ trained and validated on $D_{\text{train}}$ and $D_{\text{val}}$ respectively, which matches the performance of an expert demonstrator $\pi^E : S^* \rightarrow A$ in test environment $e_{\text{test}}$, while the agent observes the full state including the extraneous modalities $\tilde{s} \in \tilde{S}$. We define this problem as the state over-specification problem for imitation learning from multimodal data.

**IV. MASKED IMITATION LEARNING**

To address overfitting caused by state over-specification, our key insight is to be selective about what modalities the policy relies on for decision making, and remove the over-specified modalities and only preserve the modalities that are generalizable across environments. We develop a masked imitation learning (MIL) method to achieve this. In this section, we present the model architecture and discuss loss design.

**A. Policy Network Architecture**

Fig. 2 shows an overview of our proposed masked imitation learning (MIL) method. Our model consists of three parts: the feature encoder for each modality $G^i_{\theta_{i=1}^M}$, a learnable binary mask selector $\Psi \in \{0, 1\}^M$ with one bit mask for each modality, and an action predictor $F_\theta$. We use $\theta$ to denote all the parameters of the encoders $G^i_{\theta_{i=1}^M}$ and the action predictor $F$.

The feature encoder $G^i$ extracts a feature vector $G^i(s^i)$ from the $i$-th modality. Then the feature $G^i(s^i)$ is multiplied by the $i$-th dimension of the mask $\Psi[i]$ and all the masked features are then concatenated into a single feature vector: $[\Psi[1]G^1(s^1), ..., \Psi[M]G^M(s^M)]$, where $F$ uses this masked featurized state to predict the final action. In all, our policy can be represented as:

$$\pi(s; \Psi, \theta) = F_\theta([\Psi[1]G^1_{\theta}(s^1), ..., \Psi[M]G^M_{\theta}(s^M)]). \tag{1}$$

**B. Bi-Level Optimization**

The goal of MIL is to simultaneously optimize $G^i_{\theta_{i=1}^M}$ and $F_\theta$ and find a mask $\Psi$ that assigns zero weight to over-specified modalities. To achieve this, at the high level, we develop a bi-level optimization framework, where the inner-level takes a fixed mask and optimizes $G^i_{\theta_{i=1}^M}$ and $F_\theta$ with imitation training loss $L_{\text{in}}$ using standard gradient descent over $\theta$ and the outer loop optimizes the modality mask $\Psi$ with the validation loss $L_{\text{out}}$ using the coordinate descent algorithm [50]. The key idea of MIL is that the inner-level optimization process could find a model that minimizes the imitation learning loss on the training data for a specific mask, and the outer loop evaluates the generalizability of the learned model with a validation loss and decides whether the mask selects the robust modalities.

Specifically, following the coordinate descent algorithm that is widely used for learning binary variables, we start from a mask $\Psi$ with all the entries as one and iteratively update the mask from the first bit to the last bit cyclicly one bit at a time. At each iteration, we have a mask $\Psi$ and a bit index $j$ indicating which bit we want to update in this iteration. We create two masks $\Psi^0$ and $\Psi^1$ by setting the $j$-th bit of $\Psi$ as 0 and 1 respectively. We then execute the inner-level imitation learning process for $\Psi^0$ and $\Psi^1$ respectively with the inner-level imitation loss, where we take the widely-adopted L2 loss optimized on the training data $D_{\text{train}}$:

$$L_{\text{in}}(\theta) = E_{(s, a) \in D_{\text{train}}} \| \pi(s; \Psi, \theta) - a \|^2. \tag{2}$$

After convergence, we learn the parameters $\theta(\Psi)$ that optimize the imitation learning loss $L_{\text{in}}$ for the given masks $\Psi^0$ and $\Psi^1$. This method provides a generalization guarantee for the learned model.
The key elements of MIL are a bi-level optimization that learns a binary mask in the outer loop using coordinate.

\[ \mathcal{L}_{\text{out}}(\Psi) = \mathbb{E}_{(s,a) \in \mathcal{D}_{\text{val}}} \| \pi(s; \Psi, \theta(\Psi)) - a \|^2. \]  

(3)

The only difference between \( \mathcal{L}_{\text{in}} \) and \( \mathcal{L}_{\text{out}} \) is the dataset they are trained on, i.e., \( \mathcal{D}_{\text{train}} \) vs. \( \mathcal{D}_{\text{val}} \). The mask that includes extraneous modalities will overfit to the training data and introduce a high \( \mathcal{L}_{\text{out}} \) on \( \mathcal{D}_{\text{val}} \). Thus, at the \( j \)-th iteration (corresponding to the \( j \)-th bit of the mask), we select the mask \( (\Psi^0 \text{ or } \Psi^1) \) that minimizes the outer loss \( \mathcal{L}_{\text{out}} \) to update the \( j \)-th bit of mask. We then repeat this procedure to update the next bit in the mask.

**Algorithm 1 Masked Imitation Learning**

**Require:** \( \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}} \); convergence error \( \epsilon \)

1: (Optional) Train \( \mathcal{T} \) with \( \mathcal{D}_{\text{val}} \) and the loss in Eqn. (4)
2: Initialize mask \( \Psi \leftarrow \{0\}^M \); \( \mathcal{L}_{\text{current}} = \infty \)
3: repeat
4: \( \mathcal{L}_{\text{last}} = \mathcal{L}_{\text{current}} \)
5: for \( i = 0, \ldots, (M-1) \) do
6: \( \Psi^0 \leftarrow \Psi; \Psi^0[i] = 0; \Psi^1 \leftarrow \Psi; \Psi^1[i] = 1 \)
7: for \( \Psi^k = \Psi^0, \Psi^1 \) do
8: Initialize \( \pi(s; \Psi^k, \theta) \) with mask \( \Psi^k \)
9: while \( \mathcal{L}_{\text{in}} \) not converged do
10: \( \mathcal{L}_{\text{in}} \) with \( \mathcal{L}_{\text{out}} \) on \( \mathcal{D}_{\text{train}} \);
11: Compute \( \mathcal{L}_{\text{out}}(\Psi^k) \) according to Eqn. (3) or (6)
12: if \( \mathcal{L}_{\text{out}}(\Psi^0) < \mathcal{L}_{\text{out}}(\Psi^1) \) then
13: \( \Psi \leftarrow \Psi^0; \mathcal{L}_{\text{current}} \leftarrow \mathcal{L}_{\text{out}}(\Psi^0) \)
14: else
15: \( \Psi \leftarrow \Psi^1; \mathcal{L}_{\text{current}} \leftarrow \mathcal{L}_{\text{out}}(\Psi^1) \)
16: until \( |\mathcal{L}_{\text{current}} - \mathcal{L}_{\text{last}}| \leq \epsilon \)
17: return \( \pi_{\theta}, \Psi \)

**Remark.** Note that although we adopt similar training and validation loss as common imitation learning algorithms, the modality mask and the bi-level optimization process allow MIL to learn a more generalizable policy. In a common imitation learning setting, the policy uses all the modalities and cannot avoid overfitting to extraneous modalities, the validation loss can only select the most generalizable policy within a pool of overfitting policies. However, with MIL, the validation loss selects which mask to use instead of selecting \( \theta \), which allows it to remove extraneous modalities that cause overfitting leading to learning a more generalizable imitation policy.

**C. Validation Loss with a Forward Dynamics Model**

The current validation loss defined on state-action pairs in Eqn. (3) only evaluates the per-step error of actions. However, as demonstrated in prior works [51, 52], imitation learning suffers from large compounding errors across long sequences even though the error in each step can be small. We would like the validation loss to be lower for policies that can stay closer to the state distribution of the validation trajectories. Since our method is offline, we cannot generate rollouts by interacting with the environment. Instead, we learn a forward dynamics model \( \mathcal{T} \) to approximate the transition dynamics \( \mathcal{T} \) based on \( \mathcal{D}_{\text{val}} \). Let us define the forward dynamics loss as:

\[ \mathcal{L}((\mathcal{T}) = \mathbb{E}_{(s_t, a_t, s_{t+1}) \in \mathcal{D}_{\text{val}}} \| \mathcal{T}(s_t, a_t) - s_{t+1} \|^2. \]  

(4)

Now, we can create a new trajectory \( \tilde{\tau} \) by rolling out the policy \( \pi(s; \Psi, \theta) \) from an initial state \( s_0 \) of each trajectory \( \tau \) in \( \mathcal{D}_{\text{val}} \) with the forward dynamics model \( \mathcal{T} \) as follows:

\[ \tilde{s}_0 = s_0, \quad \tilde{s}_{t+1} = \mathcal{T}(\tilde{s}_t, \pi(\tilde{s}_t; \Psi, \theta(\Psi))). \quad (5) \]

By rolling out the policy from each initial state in \( \mathcal{D}_{\text{val}} \) for the same length as the corresponding trajectory in the validation set, we create a new dataset \( \mathcal{D}_{\text{val}} \), within which each state \( \tilde{s} \) has a corresponding state \( s \) in \( \mathcal{D}_{\text{val}} \). We then compute the outer loop validation loss based on the state differences between \( s \in \mathcal{D}_{\text{val}} \) and \( \tilde{s} \in \mathcal{D}_{\text{val}} \):

\[ \mathcal{L}_{\text{out}}(\Psi) = \mathbb{E}_{(\tilde{s}, s) \in (\mathcal{D}_{\text{val}}, \mathcal{D}_{\text{val}})} \| \tilde{s} - s \|^2. \]  

(6)

This new validation loss using the learned forward dynamics model is designed for tasks with long-horizon trajectories or tasks that consist of multiple stages, which can suffer from large compounding errors. Note that the forward dynamics model suffers less from compounding error because we only query data near the trajectories in the validation set. In practice, we train the forward dynamics model with the proprioception states to avoid having to learn visual dynamics models that can be much less accurate.

**Remark.** The key elements of MIL are a bi-level optimization that learns a binary mask in the outer loop using coordinate.
descent and a learned forward dynamics model for constructing validation loss of long-horizon tasks. Note that MIL learns to filter extraneous modalities that induce large generalization errors (high validation loss) but do not necessarily return the smallest number of modalities needed to learn a task. If there are redundant modalities that do not influence the performance of the learned policy (whether it was used by the expert or not), MIL may not learn to filter them.

V. EXPERIMENTS

We evaluate MIL for imitation learning in three simulated robotic control tasks and on a multimodal dataset collected on a Franka Panda arm. We compare MIL with several baselines and ablations including BC-NoMask (vanilla BC using all the modalities), MaskDropout (the policy is learned with random dropout on the mask), MaskAverage (average performance of policies learned with randomly selected but fixed masks), HandPickedMask (a manually selected mask by the authors), ContinuousMask (a continuous/non-binary mask is learned altogether with the policy net using SGD), and MIL with online evaluation (allowing interactions with the validation environment).

Reacher is a MuJoCo-based task where an agent with a two-link arm needs to reach a specified target location in a 2D space. We create two experiment settings in this environment. 1) In the first setting, as shown in Fig. 3a, we divided the 2D space into 2 different regions for sampling the target such that train and validation/testing environments have their own target distribution. The 5 input modalities are the angle between two links, the angle between the first link and the target, the distance between the target and the center, the angular velocities of the first and second links, and the Cartesian target position. \( \mathcal{D}_{train} \) consists of targets in the top half of the space while \( \mathcal{D}_{val} \) comes from the bottom half. Here, we expect that the model may overfit to the Cartesian target position, which has a distribution shift from the top half to the bottom half. 2) In the second setting, the 4 input modalities include the rendered RGB image and the low-dimensional states in Reacher (the angular velocities of the first and second links, the target position and the relative location of the fingertip and the target). \( \mathcal{D}_{train} \) and \( \mathcal{D}_{val} \) consist of targets in the same distribution. The results in this task domain are presented in Fig. 4, in which MIL outperforms baseline methods in both settings. MIL reaches a higher reward once it learned to mask out the extraneous modalities (the Cartesian target position for the first setting and the image for the second setting). In this task, we also verify that redundant modalities (modalities containing the exact same information as existing ones or that can be derived) do not influence the performance of the learned policy.

Robomimic-Can is a task in Robomimic dataset [11], where a Franka Panda robot is learning to pick up a can and put it inside a target bin (Fig. 3b). We adapted this task such that the coke can has two different colors: red or blue. \( \mathcal{D}_{train} \) and \( \mathcal{D}_{val} \) both consist of demonstrations from two different demonstrators and the can color is consistent within a single demonstrator. One demonstrator is better than the other one in this task and therefore learning with RGB image can bias the policy to perform poorly on one particular color of can. The 3 input modalities are proprioceptive end-effector state, RGB images from the side and hand cameras, and depth images from the side and hand cameras. MIL learns to mask out the over-specifying RGB modality in this case and achieves an average success rate of 56.0% while BC-NoMask results in unstable policy performance (see Tab. I).

Robomimic-Square is also a task from Robomimic dataset [11], where a Franka Panda robot is learning to pick up a square nut and put it through the square-shaped pole (Fig. 3c). The 6 input modalities are object state, end-effector position, end-effector orientation, end-effector angular velocity, end-effector linear velocity, and gripper state (open/close). An example run of MIL in this task is shown in Fig. 5: as MIL learns to mask out end-effector orientation and velocities, the success rate of the learned policy increases. In this task, MIL uses the state-based validation loss. This task is not only long-horizon but also involves precise insertion. Therefore, we further developed a validation loss that leverages a small amount of augmented (failed) trajectory data such that the loss is more selective of high-performing policies instead of policies that get close to the state distribution of validation states but cannot actually finish the task. This version of MIL (MIL-aug) end up finding a policy with the highest the success rate of 71.3%, achieving 5.6× performance gain over the vanilla behavioral cloning baseline (BC-NoMask) and is even higher than using the HandPickedMask (Tab. I). Note that ContinuousMask achieves the lowest performance, which demonstrates that the choice of binary mask better addresses the state over-specification problem.
We experiment in both simulated and real-world robotic tasks and demonstrate the effectiveness of our proposed method.

**Limitations.** MIL is computationally expensive compared to traditional imitation learning methods since it relies on bi-level optimization and the inner loop needs to run imitation learning for every mask update. In our experiments, we observe that early stopping of the inner loop imitation learning does not hinder the performance of mask learning. Such early stop strategies can be further explored in the future. In addition, MIL may not return the smallest number of modalities, as it may preserve redundant modalities when these modalities do not influence the performance. At the same time, the ordering of the modalities influences the final mask learned, which means we may not recover the global optimal mask in rare cases in which there exists strong dependency between modalities.

**Acknowledgement** This work is supported by Ford, NSF Awards 1953032, 2125511, 2218760, and DARPA YFA.

**References**

[1] M. O. Ernst and H. H. Bülthoff, “Merging the senses into a robust percept,” *Trends in cognitive sciences*, vol. 8, no. 4, pp. 162–169, 2004.

[2] B. E. Stein and M. A. Meredith, *The merging of the senses*. The MIT press, 1993.

[3] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[4] K. Zhang, M. Sharma, M. Veloso, and O. Kroemer, “Leveraging multimodal haptic sensory data for robust cutting,” in *2019 IEEE-RAS 19th Humanoids*. IEEE, 2019, pp. 409–416.

[5] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, “Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations,” *arXiv preprint arXiv:2107.14483*, 2021.

[6] T. Liu, H. Liu, Y.-F. Li, Z. Chen, Z. Zhang, and S. Liu, “Flexible ftir spectral imaging enhancement for industrial robot infrared vision sensing,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 544–554, 2019.

[7] A. C. Abad and A. Ranasinghe, “Visuotactile sensors with emphasis on gelisight sensor: A review,” *IEEE Sensors Journal*, vol. 20, no. 14, pp. 7628–7638, 2020.

[8] K. Dai, X. Wang, A. M. Rojas, E. Harber, Y. Tian, N. Paiva, J. Gnehm, E. Schindewolf, H. Choset, V. A. Webster-wood et al., “Design of a biomimetic tactile sensor for material classification,” *2022 IEEE ICRA*, 2022.

[9] M. Janner, J. Fu, M. Zhang, and S. Levine, “When to trust your model: Model-based policy optimization,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[10] Y. Xiao, F. Coddewilla, A. Gurram, O. Urfaigolu, and A. M. López, “Multimodal end-to-end autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[11] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang,
...