Detect, Reject, Correct: Crossmodal Compensation of Corrupted Sensors

Michelle A. Lee[1], Matthew Tan[1], Yuke Zhu[2], Jeannette Bohg[1]

***Abstract*— Using sensor data from multiple modalities presents an opportunity to encode redundant and complementary features that can be useful when one modality is corrupted or noisy. Humans do this everyday, relying on touch and proprioceptive feedback in visually-challenging environments. However, robots might not always know when their sensors are corrupted, as even broken sensors can return valid values. In this work, we introduce the Crossmodal Compensation Model (CCM), which can detect corrupted sensor modalities and compensate for them. CMM is a representation model learned with self-supervision that leverages unimodal reconstruction loss for corruption detection. CCM then discards the corrupted modality and compensates for it with information from the remaining sensors. We show that CCM learns rich state representations that can be used for contact-rich manipulation policies, even when input modalities are corrupted in ways not seen during training time.**

I. Introduction

Here is an experiment that you can try at home: take a water bottle and unscrew the cap. Now close your eyes and try to close the water bottle. Most humans, even without visual senses, can rely on proprioceptive and tactile sensing when performing manipulation tasks. To study the inverse relationship (in an experiment best not done at home), Johansson et al. [14] anesthetized the fingertips of human subjects which impacted their ability to light a match. The experiment video [15] shows that while the human subject at first struggled to manipulate the match without haptic feedback, they were still able to light it within about 20 seconds of trial and error. Neuroscience research provides more evidence that humans can take information from one sensor modality to compensate for missing information of another in a *crossmodal* manner. This has been shown for visual and tactile feedback in tasks, such as object size prediction [9] and object manipulation [13], as well as visual and auditory information [1, 2, 25].

We aim to endow a robot with the same capability of crossmodal compensation. This is an important for avoiding potentially dangerous outcomes when deploying a robot to the real world. There are many cases when a sensor can break, produce erroneous data, become occluded, or change with lighting conditions. In this work, we focus on the case in which one of our sensors experiences failure modes or noise unseen during train time. We want our robot to accomplish tasks robustly, even in the face of the corrupted sensor data.

[1]Stanford University, [2]The University of Texas at Austin

This work has been supported by NSF CNS-1955523 and JD.com American Technologies Corporation ("JD") under the SAIL-JD AI Research Initiative. This article solely reflects the opinions and conclusions of its authors and not JD or any entity associated with JD.com. We are grateful to Mike Salvato and Brent Yi for providing invaluable feedback, and to Peter Zachares for his help in developing the simulation environment.

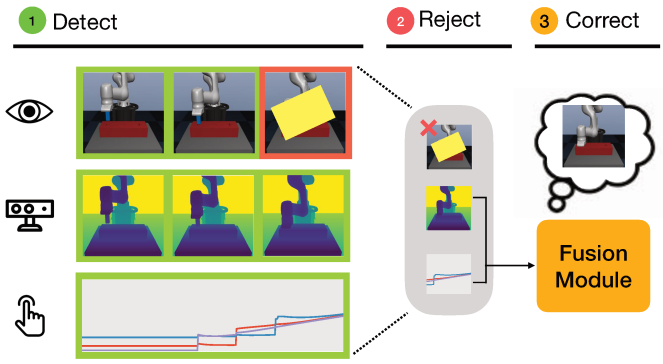


Fig. 1: We propose Crossmodal Compensation Model that can (1) **detect** when a sensor input is corrupted, which in this case is the image data, (2) **reject** the corrupted image as input to our policy, and (3) **correct** for the rejected image by compensating the missing image information with the remaining force and depth information in the fusion module.

There have been several recent works that can perform inference or complete downstream task in the presence of missing modalities [23, 24, 33–35, 37]. However, they need to know what modality is missing at inference time. In this work, we are interested in crossmodal compensation at inference time, when we lack knowledge on which sensor modality may be corrupted.

Other works, particularly in autonomous driving, actively detect when sensor inputs are *out-of-distribution* (OOD) with respect to the training data [16, 31] and compensate for it [10, 29]. However, these works focus on avoiding collision and require querying expert feedback, which limit the generalization of their methods to manipulation tasks.

Several works on robot manipulation have shown that Bayesian filtering approaches to sensor fusion can perform state estimation even when some sensor readings are corrupted [18, 26, 27, 39]. These methods require the user to explicitly define the estimated state, as well as the analytical forward and measurement models that may be hard to specify or intractable to compute online. While differentiable filters such as [20] can learn how to fuse multiple modalities, they were not demonstrated to recover from corrupted sensor inputs unseen during train time.

To detect corrupted sensor readings *and* compensate for them, we introduce the *Crossmodal Compensation Model* (CCM), a novel latent variable representation model that can be used to generate state feedback for a learned policy that compensates for corrupted sensor inputs. CCM performs crossmodal compensation for corrupted sensor inputs in three steps: CCM (1) **detects** which modality is corrupted through

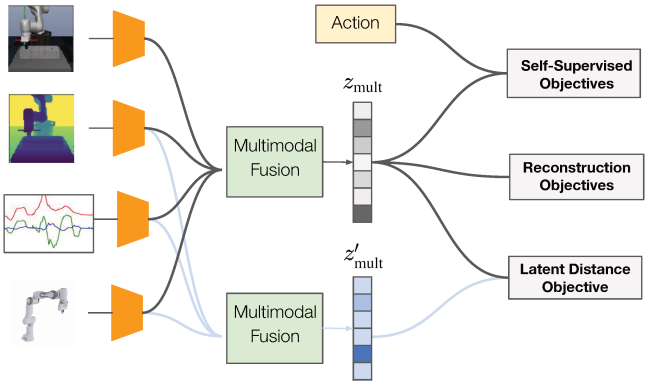


Fig. 2: CCM is a multimodal latent representation model that takes RGB image, depth, force, and robot proprioception data as inputs, and jointly trains to reconstruct the input modalities as well as to predict self-supervised objectives used in [19]. Additionally, we learn a latent distance objective: during training, we drop a random input modality and learn to reduce the L2 distance between our compensated latent representation z'_{mult} and the full modality representation z_{mult}.

out-of-distribution detection, (2) **rejects** and discards the corrupted sensor reading, and (3) **corrects** for the discarded modality with information from the remaining modalities. By learning to reconstruct each input modality, CCM compares unimodal reconstruction results with the sensor inputs to identify the corrupted sensor. CCM jointly learns this reconstruction objective with self-supervised objectives introduced in [19] that were shown to be crucial for learning a rich representation that lends itself to control. CCM learns to perform crossmodal compensation by minimizing the distance between a representation generated with dropped modalities and the representation generated with full modalities. CCM is robust to corrupted sensor readings, and, by correcting for them, generates rich state representations that can be used for contact-rich manipulation policies.

II. Related Work

Multimodal Representation Learning. The complementary nature of heterogeneous sensor modalities holds the promise of providing more informative feedback for solving perception and manipulation tasks than uni-modal approaches. Several works have combined visual and tactile information for object tracking [18, 27, 39], shape completion [36], grasp assessment [4, 7, 11, 32], and scene understanding [5, 6]. Many of these multimodal approaches are trained through a classification objective for inference tasks [4, 7, 11, 38]. In prior works [19, 21], we proposed a self-supervised approach to learn multimodal representations that were used as policy inputs, but do not explicitly use the multimodal information to compensate for corrupted sensor readings. In this work, we leverage similar architectures and self-supervised objectives from [21] to train our representations.

Missing Modality Representations. While many works have shown that using multimodal data sources can improve the robustness and accuracy of an inference task [3, 19, 30, 35, 37, 40], the multimodal data source can sometimes be noisy or incomplete. Motivated by this, previous works have tackled the problem of noisy or missing modalities by learning explicit crossmodal models or robust joint representations.

One approach to handle missing modalities is to explicitly predict how to transfer from one modality to another. [23, 33] use deep generative models to predict between raw vision and touch. Other works, such as [24, 34, 35, 37], drop modalities during train time to reconstruct the missing modality, perform inference, or both.

To account for the missing modalities during training, Tsai et al. [35] learns a new model for each missing modality. On the other hand, Wu et al. [37] and Tan et al. [34] use a variational product-of-experts (PoE) approach to recover the joint representation when inputs are missing, eliminating the need to parameterize new models. Our work uses the PoE approach similar to [37]. All recent work on compensating missing modalities assume to know which modality is missing, which is not always the case. In contrast, our approach both detects and compensates for missing modalities.

Out of Distribution Detection. Detecting OOD inputs in control tasks can prevent catastrophic failure in robots. While many works detect OOD data with deep neural network-based architectures [10, 16, 29, 31], none of these work look at using crossmodal compensation for OOD data. In this work, we follow [31] and train a variational autoencoder, using the reconstruction error at test time to detect when a single modality is OOD. There is a vast amount of literature relating to OOD detection, and we refer the reader to this survey paper [8] for more information.

III. Problem Statement

Our goal is to learn to perform a manipulation tasks even with corrupted sensor readings. Our algorithm learns the manipulation task with multiple modalities as input, and uses data from the un-corrupted sensor modalities to compensate for the corrupted ones. We model the manipulation task as a finite-horizon, discounted Markov Decision Process (MDP) $\mathcal{M}$, with a state space $\mathcal{S}$, an action space $\mathcal{A}$, state transition dynamics $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, an initial state distribution ρ_0, a reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow R$, horizon T, and discount factor $\gamma \in (0, 1]$. To determine an optimal policy π, we want to maximize the expected discounted reward $E_\pi \left[\sum_{t=0}^{T-1} \gamma^t r(s_t, \mathbf{a}_t)\right]$.

We represent the policy by a neural network with parameters θ_π that are learned as described in Sec. IV-D. $\mathcal{A}$ is defined over continuously-valued 3D displacements $\Delta\mathbf{x}$ in Cartesian space. $\mathcal{S}$ is defined by the low-dimensional latent representation $z_{mult} = f(o_1, o_2, ...o_n)$ inferred from multimodal sensory inputs o_i through an encoder f. This encoder is a neural network parameterized by ψ_s

In this work, we are interested in cases when the robot is receiving corrupted sensor readings during policy rollout at test time. We describe how our model compensates for corrupted sensor readings below.

IV. Method Overview

Our proposed model, CCM, attempts to detect and compensate for corrupted sensor readings at the representation

level. CCM is a multimodal latent variable model that encodes heterogeneous inputs into a multimodal representation z_{mult} using a variational PoE approach [37]. We jointly train modality reconstruction, self-supervised objectives, and a latent distance objective to learn useful representations.

To compensate for the corrupted sensor reading o_i', the representation model first detects which sensor is corrupted (see Sec. IV-B). Then, the representation model removes the corrupted input and performs crossmodal compensation, i.e. inferring a compensated latent representation z_t' that is close to z_{mult} (fully modality latent representation): $f(o_1, o_2, ...o_i', ...o_n) = \hat{z}'_{mult} \approx z_{mult}$(see Sec. IV-C).

A. Multimodal Representation Model

CCM is a multimodal latent variable model trained with self-supervision. Our model encodes 4 types of data: RGB images(o_{RGB}), depth images (o_{depth}) from a fixed RGB-D camera, haptic feedback from a wrist-mounted force-torque (F/T) sensor (o_{force}), and end-effector position, and linear velocity (o_{prop}). Information from each modality is then fused into a single multimodal latent representation z_{mult}.

We use the same modality specific encoders as our prior work [19] to capture domain-specific features, except for our force encoder, as we found that using a convolution-based architecture helps with force reconstruction. We take the last 32 readings from a F/T sensor as a one-channel ($1 \times 32 \times 6$) input into a 5-layer two-dimensional CNN. We add a single fully-connected layer to the end of each modality encoder to map into a $2 \times d$-dimensional variational parameter vector with $d = 128$ as in [19].

We assume that each modality is conditionally independent given the fused multimodal latent variable representation z_{mult}. Each modality encoder maps to a multivariate isotropic Gaussian parametrized by $z_m = \{\mu_m, \sigma_m\}$ which is then fused using a PoE approach [37]. The resulting multivariate Gaussian distribution of the multimodal latent space will have mean $\sigma_j^2 = (\sum_{i=1}^{n+1} \sigma_{ij}^2)^{-1}$ and variance $\mu_j = (\sum_{i=1}^{n+1} \mu_{ij}\sigma_{ij}^2)(\sum_{i=1}^{n+1} \sigma_{ij}^2)^{-1}$, where n is the number of modalities, μ_j and σ_j^2 are the variational parameters of the j-th dimension. We also add an isotropic multivariate Gaussian prior as an additional expert.

Following [19], we train our model using a variational objective by minimizing the Evidence of Lower BOund (ELBO) over our dataset $D = \{\{o_i, y_i, a_i\}|i = 1...n\}$ with observations, o, labels for self-supervised objectives, y, and actions, a: $\mathcal{L}_i(\theta_s, \phi_s) = E_{q_{\phi_s}(\mathbf{z}|\mathcal{D}_i)}[\log p_{\theta_s}(\mathcal{D}_i|\mathbf{z})] - \mathrm{KL}[q_{\phi_s}(\mathbf{z}|\mathcal{D}_i)||p(\mathbf{z})]$. We model the approximate posterior $q_{\phi_s}(z|o_i)$ as a neural network encoder parameterized by ϕ_s. We model the likelihood $p_{\theta_s}(o_i, y_i, a_i|z)$ with a decoder neural network, parameterized by θ_s.

Decoder Architectures In this work, our model jointly optimizes self-supervised objectives and reconstruction objectives. For the self-supervised objectives, our model learns to predict action-conditional optical flow, next-step end-effector pose, whether the end-effector will be in contact at the next time step, and whether the input modalities are paired. The decoder architectures for these four self-supervised objectives are described in [19]. We also learn to reconstruct our input modalities, commonly used for representation learning [22]. Models that learn to reconstruct their inputs can also be used for out-of-distribution detection [8], which we use for detecting when an input modality is corrupted during test time.

The force decoder uses a 4-layer deconvolutional decoder for reconstruction. We found it difficult to reconstruct the small and often noisy torques from the force/torque sensor, so our force decoder only reconstructs 3-dimensions forces.

For reconstructing the RGB and depth image from the the multimodal vector z_{mult}, we use a 5-layer deconvolutional decoder for each modality. To encourage our network to learn reconstruction of the robot interacting with the environment (instead of the static background), we take the boolean sum of the robot in image-space throughout the entire dataset to create a mask. At the end of each of our 5-layer depth and image decoders, one deconvolution layer reconstructs the entire image or depth data and another deconvolution layer reconstructs the image or depth data in the masked region. We find that learning to reconstruct both the masked and complete image/depth helps with learning speed and stability.

Missing Modality Training Objective CCM learns to perform modality compensation by dropping one of the 3 modalities $\{o_{RGB}, o_{Force}, o_{Depth}\}$ at each training step (we assume o_{prop} is always present). The latent representation with missing modality is referred to as as z'_{mult}.

We encourage z'_{mult} to be close to our full modality representation z_{mult}, so a policy trained on z_{mult} can take z'_{mult} as input. We encourage this by introducing a *latent distance loss* between them. Our full objective then becomes $ELBO(o_i, y_i, a_i) + ||z_{mult} - z'_{mult}||_2^2$.

B. Out-Of-Distribution Detection

Similar to [31], we use reconstruction error (L2 distance between input and its reconstruction) to detect input that is out-of-distribution and therefore deemed to come from a corrupted sensor. For some observed modality $o_m \in \{o_{RGB}, o_{Depth}, o_{Force}\}$, we assume that the multimodal model the reconstruction error will be large when the model reconstructs inputs that are OOD from training data. We can threshold the reconstruction error as a method of predicting OOD inputs. For each modality, we choose the thresholds with the best Area Under the Receiver Operating Characteristic Curve (AUROC) performance for detecting corruption in the validation dataset. For the reconstructed F/T data, we threshold the reconstruction error on each of the 3 dimensions and consider the modality an outlier when 2 or more of the dimensions are out-of-distribution. Since we are using a multimodal representation, corruption in one modality may affect reconstruction in another modality leading to the detection of more than one corrupted modality. We handle this ambiguity by selecting the modality with the largest standard deviation away from the mean reconstruction error for that modality in the training set. We found these hyperparameters to be robust in our experiments.

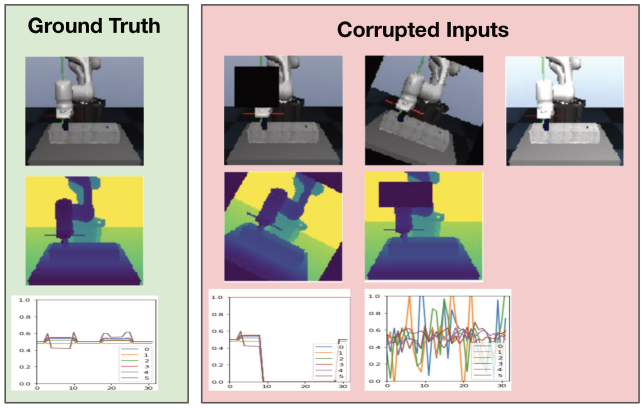


Fig. 3: Examples of the ground truth data and various kinds of corrupted inputs (from the top and left to right): RGB occlusion, RGB rotations, RGB lighting change, depth rotation, depth box occlusion, blackout force, and force noise.

C. Crossmodal Compensation

Following the detection of the corrupted input with OOD detection, we leverage the cross-modal relationships in our model to compensate for the corrupted input. Our approach does not make any assumptions about the nature of the modality corruption. We take the conservative approach and assume that the information of the corrupted modality is too out-of-distribution from our training data and cannot be used.

We then treat the crossmodal compensation problem as a missing modality problem. Since our representation model is trained with missing modalities, we can simply drop the corrupted modality and use the recalculated multimodal representation z'_{mult} as a proxy for z_{mult}, as seen in Fig. 1.

For the context of this study, we detect and compensate corrupted sensor readings from one of the three input modalities, namely RGB images, force, and depth.

D. Policy Learning

Our final goal is to equip a robot with a policy for performing contact-rich manipulation tasks that leverages our latent multimodal representation. We use our latent multimodal representation z_{mult} as state input to our policy. When there is a detected corrupted input, the policy takes the crossmodal compensated representation z'_{mult} as state input.

V. Experimental Design

The primary goal of our experiments is to examine the effectiveness of our pipeline in detecting and compensating for corrupted inputs while retaining useful features for policy learning. Our experiments attempt to answer two main questions: 1) How well can we learn a manipulation task using the learned representation as input? 2) Can a policy trained with CCM handle corrupted sensor input?

To analyze how our model detects and compensates for corrupted sensor modalities, we additionally design experiments to answer the following questions: 3) Can we use reconstruction error to reliably detect corrupted sensor readings? 4) How close to the latent full-modality representation z_{mult} is the crossmodal compensated representation z'_{mult}?

A. Experimental Setup.

Peg Insertion Task We use a contact-rich peg insertion task as an experimental test-bed for our algorithm. We perform our robot experiments in simulation using the RoboSuite [28] platform with the Franka Panda robot, a 7-DoF torque-controlled robot. Four sensor modalities are available in simulation, including proprioception, an RGB-D camera, and a force-torque sensor. The proprioceptive input is the end-effector pose as well as linear and angular velocity. RGB images and depth maps are recorded from a fixed camera pointed at the robot. Input images to our model are down-sampled to 128×128. We use a square peg with size (1.4 x 1.4 x 7.5)cm and 1mm clearance in all directions. We use the shaped reward for peg insertion from [21].

Dataset Collection and Pre-processing. We apply standard pre-processing techniques for our RGB image and depth inputs by normalizing the data with the min and max values. We clip and normalize the range of F/T sensor data to the 97th percentile and 3rd percentile of the data for each of the 6 dimensions; there is minimal difference between a robot applying 30N of force and 100N of force for our task. We also weight samples in the dataset to ensure that there are sufficient samples of contact data during training. Finally, for the model to learn pairing, we use unpaired examples where the robot end-effectors are at least 6 centimeters apart from each other to make sure that the inputs are sufficiently different from one other.

Reinforcement Learning Algorithm and Architecture. After representation training, we freeze the representation model and use our latent representation as state input to our RL policy. We use a state-of-the-art model-free off-policy RL algorithm, Soft Actor-Critic [12]. We use a 2-layer Tanh Gaussian policy that takes as input the 128-dimensional latent representation from our representation model, and produces 3D position displacement Δx of the robot end-effector. We adopt the shaped reward from [21] that encourages the peg to be close to the hole and insert. For simplicity, we refer to the policies trained with the CCM representation as CCM policies, and the representation model as simply CCM. We use the same format for our baselines.

Corrupted Sensor Inputs. We test our model in detecting and compensating for a variety of unimodal corrupted inputs after policy learning. To accomplish this, we design a wide range of sensor corruptions. For RGB images, we randomly insert black boxes around the robot, change the lighting, and perform in-plane image rotations. For depth inputs, we insert similar random occlusions, and transform the depth image with random in-plane rotations. For F/T inputs, we randomly set forces to 0 (which corresponds to the bottom third percentile of the representation training data) and random Gaussian noise with variances [0.5, 0.25, 0.1]. Examples of different kinds of corrupted inputs are shown in Figure 3.

Implementation Details. We train the representation model with the Adam optimizer [17] with a learning rate of 0.0001 and β values of (0.9, 0.999). The model is trained over 75 epochs with a batch size of 64. For policy learning, we train 3

Model Name	Normal Input	Corrupted Image		Corrupted Depth		Corrupted Force	
		Comp.	Not Comp.	Comp.	Not Comp.	Comp.	Not Comp.
CCM (Our Model)	96.7%	**80.7%**	29.3%	**82.0%**	0.7%	**78.0%**	81.3%
MFM	**100.0%**	8.7%	50.0%	0.7%	0.7%	57.3%	18.7%
Recon CCM	81.3%	71.3%	69.3%	67.3%	2.0%	69.3%	72.0%
SS CCM	43.3%	40.7%	6.0%	4.7%	2.0%	38.7%	30.0%
CCM No Dist	99.3%	0.7%	28.0%	3.3%	0.0%	30.0%	22.0%
CCM No Force	96.7%	78.7%	5.3%	44.7%	14.0%	n/a	n/a

TABLE I: The average success rates for our policies. We train 3 policies per model, and evaluate 50 trials per each policy with: normal inputs, corrupted but compensated inputs (Comp.), and corrupted but not compensated inputs (Not Comp.). We see that while MFM had the highest task success rates when given normal inputs, our proposed model, CCM, outperformed all other baselines when compensating for corrupted modality inputs. Not Comp. is given as comparison to see how policies perform when no compensation occurs.

random seeds per representation model, for 750,000 training steps. Each episode horizon is 200 steps.

Evaluation Metrics. We evaluate our algorithm's ability to compensate for corrupted inputs by reporting the success rate of our trained policies when given corrupted depth, image, and force readings. We also report the task success rate of our trained policies when given normal inputs.

B. Baselines

We choose to compare CCM with the Multimodal Factorized Model (MFM), another multimodal representation model that deals with missing modalities [35]. MFM uses Wasserstein Autoencoders (instead of a variational version) to learn a factorized multimodal representation, one for reconstruction objectives and one for self-supervised objectives, and explicitly parameterizes each missing modality model with neural networks. Although MFM was not introduced as a representation model for policy learning and does not perform OOD detection, we include it as part of our baseline model and implement the same corrupted sensor detection algorithm as CCM.

Our model architecture is similar to the Multimodal Variational Autoencoder (MVAE) [37], but with two key differences: we train with additional self-supervised and latent distance losses. Instead of considering MVAE as a baseline, we evaluate how the different components of CCM contribute to its success with an ablation study.

C. Ablation Study

A full CCM model learns a representation with self-supervised, reconstruction, and latent distance objectives. We propose the following ablation baselines:

1) **Recon CCM:** CCM trained with reconstruction and latent distance objective.
2) **SS CCM:** CCM trained with self-supervised and latent distance objectives. We use Recon CCM for corrupted sensor detection.
3) **CCM No Dist:** CCM trained with self-supervised and reconstruction objectives only.
4) **CCM No Force:** CCM trained with zeroed out forces during representation and policy learning.

VI. Experimental Results and Discussion

A. Policy Learning for Contact-rich Tasks

We report the learning curves of our policies for the peg insertion task in Fig. 4a. To evaluate the success rate of peg insertion given normal inputs, we evaluate every learned policy 50 times, and report for each model the average policy success rate in Table I (see Normal Input column).

The MFM policies are able to learn the task faster, and had a 100% success rate given normal inputs, compared to CCM policies' 96.7% success rate. By factorizing the self-supervised and reconstruction models and learning a new neural network model for each missing modality, MFM has a more complex model architecture than CCM and more parameters (7.7 million compared to 5.6 million), which might explain its success in learning the task.

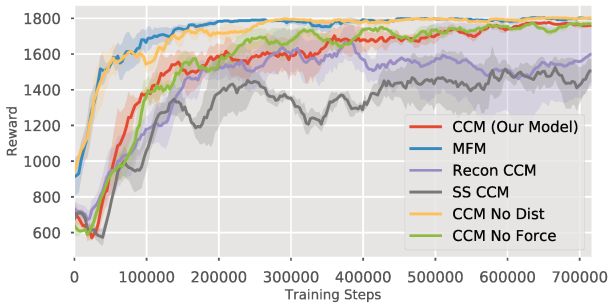
For our ablation study, we see that CCM No Dist policies also learned as fast as the MFM policies, and achieved a 99.3% success rate. In comparison, SS CCM and Recon CCM policies had high variances among its learned policies, and had low average task success rates of 43.3% and 81.3% respectively. CCM policies' better performance suggests that learning both self-supervised and reconstruction objectives helps to learn a more successful representation for policy learning. On the other hand, the success of CM No Dist policies suggests that learning the latent distance objective negatively affects the representation's efficacy as state representation for controls. One possible explanation is that when we drop a modality during training, the distance objective encourages the latent representation to only learn features that can be crossmodally compensated by the other modalities, which serves as a kind of regularization. However, we only observe a small 2.6% difference in task success between CCM and CCM No Dist.

B. Compensation of Corrupted Inputs

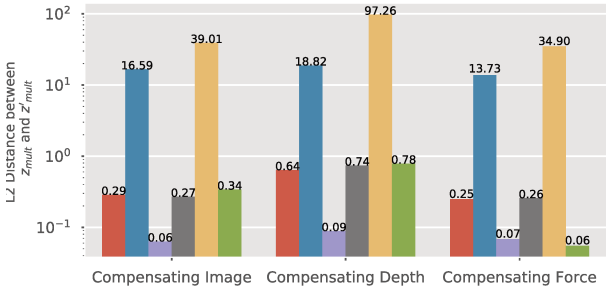
We evaluate each learned policy 50 times for corrupted force, image, and depth inputs. For each type of corrupted modality, we randomly choose a type of sensor corruption associated with the modality (described in Sec. V-A and Fig. 3) at every step of the policy rollout. The task success rates can be found in Table I. We also report the average success rate when sensor inputs are corrupted but not compensated for.

The MFM policies struggled with compensating corrupted inputs, especially for image and depth data, in which the success rates dropped from 100% to 8.7% and 0.7% respectively.

CCM No Dist policy success rates also dropped drastically when given corrupted inputs. This is expected, because CCM No Dist is not trained to map the compensated z'_{mult} to the full modality z_{mult}. Thus, the policies receive very different state feedback at test time with corrupted input compared to



(a) Policy training curve



(b) Latent L2 Distance

Fig. 4: (a) Training curves for reinforcement learning on the peg insertion task, 3 random seeds per model (b) We show in log-scale the latent L2 distance between full modality z_{mult} and compensated modality z'_{mult}, for different compensated modalities. CCM No Dist and MFM had relatively high latent L2 distances, providing an explanation why the two models struggled with compensation.

training with normal input. Both SS CCM and Recon CCM policies performed better than CCM No Dist when given corrupted inputs. However, the performances of SS CCM and Recon CCM in this experiment are upper-bounded by how well the policies trained with normal inputs.

Overall, CCM policies outperformed all others with corrupted sensor inputs. To better understand why CCM representations performed better than the others in crossmodal compensation, we analyze how well each model detects and compensates for corrupted inputs below.

C. Corrupted Reading Detection

To study detection of corrupted sensor data, we use corruptions shown in Fig. 3 and evaluate the AUROC for detecting them in the representation learning test set. As seen in Table II, all models were able to detect corrupted depth data with an AUROC score of 1.0. Both MFM and CCM No Dist have lower AUROC scores for detecting corrupted Image and Force data than other models, but not by much.

Because CCM No Dist and MFM both had high AUROC for corrupted depth data detection, but failed to compensate for corrupted depth data during policy rollout, this suggests that CCM No Dist and MFM struggled with compensation rather than detection of corrupted inputs.

D. Latent Representation Distance

We show the L2 distance between the full modality latent variable z_{mult} and the compensated latent variable z'_{mult} in Fig. 4b as evaluated on the test set for representation learning. While a small L2 distance does not guarantee successful

Models	Depth	Image	Force
CCM (Our Model)	**1.00**	**0.98**	0.97
MFM	**1.00**	0.97	0.89
Recon CCM	**1.00**	**0.98**	**0.98**
CCM No Dist	**1.00**	0.95	0.92
CCM No Force	**1.00**	0.97	n/a

TABLE II: AUROC for classifying if multimodal inputs had corrupted depth, image, or force data in our test data. We used a threshold reconstruction loss (comparing input data and reconstructed data) for each modality to perform corruption detection.

crossmodal compensation, it is a proxy measure for how well a model can compensate for corrupted modalities during policy rollout.

Notably, MFM and CCM No Dist have much higher latent distances than others. This explains their poor performance in compensating for corrupted inputs. Recon CCM had the lowest latent distances, explaining why Recon CCM policies had lower performance drop with corrupted inputs than CCM policies (on average, a 12% vs. a 16.5% drop). However, CCM policies had better normal input policy performance. In other words, the representations learned with CCM are able to balance good policy performance with normal inputs as well as crossmodal compensation with corrupted inputs.

E. Redundant Information among Modalities

Past works have demonstrated the ability to predict haptic information from vision and vice versa [23, 33], indicating that force and visual data share redundant information. In our results, we observe that CCM policies performed 3.3% better when using corrupted force input than when compensating for force, and Recon CCM performed 2.7% better when using corrupted force input than when compensating for it. This suggests the policies might be ignoring the force input.

Although these results show that the benefit of compensating for corrupted force information is limited when vision data is available, the results from our CCM No Force baseline show that force information helps compensate for corrupted depth and image inputs. CCM No Force has lower task success rates compared to CCM: 44.7% compared to 82% when compensating for corrupted depth, and 78.7% compared to 80.7% when compensating for corrupted images. It also results in higher latent distances when either visual input is missing.

VII. Conclusions

We introduced CCM, a self-supervised method for learning representations that crossmodally compensatse for corrupted inputs. By leveraging reconstruction losses, CCM can detect a variety of corrupted sensor inputs. Following detection, CCM rejects and discards the corrupted modality and use the remaining modalities to approximate the joint multimodal representation through crossmodal compensation. We showed that the policies learned with the CCM's representation is able perform a peg insertion task even when sensor inputs are corrupted. We compare CCM with other multimodal representation learning baselines, and perform a thorough analysis of how our model performs in detecting

corrupted sensor inputs and compensating for them. We find that our novel model outperforms all others for task completion with corrupted sensors.

REFERENCES

[1] D. Alais and D. Burr, "The ventriloquist effect results from near-optimal bimodal integration," *Current biology*, vol. 14, no. 3, pp. 257–262, 2004.

[2] M. Baart, B. C. Armstrong, C. D. Martin, R. Frost, and M. Carreiras, "Cross-modal noise compensation in audiovisual words," *Scientific reports*, vol. 7, p. 42 055, 2017.

[3] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[4] Y. Bekiroglu, R. Detry, and D. Kragic, "Learning tactile characterizations of object- and pose-specific grasps," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1554–1560.

[5] J. Bohg, "Multi-modal scene understanding for robotic grasping," Ph.D. dissertation, KTH Royal Institute of Technology, 2011.

[6] J. Bohg, M. Johnson-Roberson, M. Björkman, and D. Kragic, "Strategies for multi-modal scene exploration," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 4509–4515.

[7] 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.

[8] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[9] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 6870, pp. 429–433, 2002.

[10] A. Filos, P. Tigas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?" *arXiv preprint arXiv:2006.14911*, 2020.

[11] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 536–543.

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[13] P. Jenmalm and R. S. Johansson, "Visual and somatosensory information about object shape control manipulative fingertip forces," *Journal of Neuroscience*, vol. 17, no. 11, pp. 4486–4499, 1997.

[14] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 345–359, 2009.

[15] *Johansson coding*, `https://www.youtube.com/watch?v=0LfJ3M3Kn80`.

[16] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, "Uncertainty-aware reinforcement learning for collision avoidance," *arXiv preprint arXiv:1702.01182*, 2017.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] A. Lambert, M. Mukadam, B. Sundaralingam, N. Ratliff, B. Boots, and D. Fox, "Joint inference of kinematic and force trajectories with visuo-tactile sensing," *arXiv preprint arXiv:1903.03699*, 2019.

[19] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, pp. 1–15, 2020.

[20] M. A. Lee, B. Yi, R. Martín-Martín, S. Savarese, and J. Bohg, "Multimodal sensor fusion with differentiable filters," *arXiv preprint arXiv:2010.13021*, 2020.

[21] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and‘ touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[22] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, 2018.

[23] Y. Li, J.-Y. Zhu, R. Tedrake, and A. Torralba, "Connecting touch and vision via cross-modal prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 609–10 618.

[24] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor, "Learning end-to-end multimodal sensor policies for autonomous navigation," *arXiv preprint arXiv:1705.10422*, 2017.

[25] S. G. Lomber, M. A. Meredith, and A. Kral, "Cross-modal plasticity in specific auditory cortices underlies visual compensations in the deaf," *Nature neuroscience*, vol. 13, no. 11, pp. 1421–1427, 2010.

[26] R. Martín-Martín and O. Brock, "Building kinematic and dynamic models of articulated objects with multi-modal interactive perception," in *2017 AAAI Spring Symposium Series*, 2017.

[27] R. Martín-Martín and O. Brock, "Cross-modal interpretation of multi-modal sensor streams in interactive perception based on coupled recursion," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 3289–3295.

[28] R. Martín-Martín, M. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space. an action space for reinforcement learning in contact rich tasks," in *Proceedings of the International Conference of Intelligent Robots and Systems (IROS)*, 2019.

[29] R. McAllister, G. Kahn, J. Clune, and S. Levine, "Robustness to out-of-distribution inputs via task-aware generative uncertainty," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 2083–2089.

[30] H. Pham, P. P. Liang, T. Manzini, L.-P. Morency, and B. Póczos, "Found in translation: Learning robust joint representations by cyclic translations between modalities," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6892–6899.

[31] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," 2017.

[32] J. Sinapov, C. Schenck, and A. Stoytchev, "Learning relational object categories using behavioral exploration and multimodal perception," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 5691–5698.

[33] K. Takahashi and J. Tan, "Deep visuo-tactile learning: Estimation of tactile properties from images," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 8951–8957.

[34] Z.-X. Tan, H. Soh, and D. C. Ong, "Factorized inference in deep markov models for incomplete multimodal time series," *arXiv preprint arXiv:1905.13570*, 2019.

[35] Y.-H. H. Tsai, P. P. Liang, A. Zadeh, L.-P. Morency, and R. Salakhutdinov, "Learning factorized multimodal representations," *arXiv preprint arXiv:1806.06176*, 2018.

[36] S. Wang, J. Wu, X. Sun, W. Yuan, W. T. Freeman, J. B. Tenenbaum, and E. H. Adelson, "3d shape perception from monocular vision, touch, and shape priors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1606–1613.

[37] M. Wu and N. Goodman, "Multimodal generative models for scalable weakly-supervised learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 5575–5585.

[38] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo, "Deep multimodal representation learning from temporal data," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5066–5074.

[39] K.-T. Yu and A. Rodriguez, "Realtime state estimation with tactile and visual sensing for inserting a suction-held object," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1628–1635.

[40] M. Zambelli, A. Cully, and Y. Demiris, "Multimodal representation models for prediction and control from partial information," *Robotics and Autonomous Systems*, vol. 123, p. 103 312, 2020.