Visuo-Tactile Transformers for Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learning representations in the joint domain of vision and touch can improve manipulation dexterity, robustness, and sample-complexity by exploiting mutual information and complementary cues. Here, we present Visuo-Tactile Transformers (VTTs), a novel multimodal representation learning approach suited for model-based reinforcement learning and planning. Our approach extends the Visual Transformer [1] to handle visuo-tactile feedback. Specifically, VTT uses tactile feedback together with self and cross-modal attention to build latent heatmap representations that focus attention on important task features in the visual domain. We demonstrate the efficacy of VTT for representation learning with a comparative evaluation against baselines on four simulated robot tasks and one real world block pushing task. We conduct an ablation study over the components of VTT to highlight the importance of cross-modality in representation learning for robotic manipulation.

Keywords: Multimodal Learning, Reinforcement Learning, Manipulation

1 Introduction

Deep reinforcement learning (RL) algorithms have solved numerous challenging tasks from raw observations (e.g., images) [2, 3, 4, 5]. Despite these successes, learning directly from such high-dimensional observation spaces can be numerically difficult/unstable, sensitive to hyper-parameters, and sample inefficient [6]. These challenges have limited the utility of these methods for robotics applications because data collection is time intensive, costly, and task variance is high.

To address this, recent research in representation learning aims to build compact "latent" representations of the underlying task [6, 7, 8, 9, 10, 11]. These compact representations are typically trained together with a policy using prediction and reconstruction losses along with task-dependant rewards. The policy then learns from the low-dimensional latent representation rather than the high-dimensional raw sensory observation. This approach offers stability and sample-efficiency for policy learning owing to the information-dense latent representations generated from rich prediction and reconstruction supervisory signals. As such, latent representations are a promising tool for real-world robotic planning and RL.

Most research in robot latent representation learning has focused on images and proprioception. The sense of touch, together with vision, plays an important role in robotic manipulation; however, visuo-tactile representation learning is still in its relative nascency. Recent works [11, 12, 13, 14, 15, 16, 17, 18] have proposed visuo-tactile methods; however, the latent representations used are typically vectors or clusters in $\mathbb{R}^n$ and do not exploit attention architectures [19]. These representations can smooth over fine details and struggle with locality in images. These issues are particularly troublesome for manipulation as the difference between in-contact and out-of-contact can be small in pixel space and the robot may need to move significantly in the scene to make contact with objects.

In this paper, we propose a novel latent representation learning method that addresses these challenges. Our approach extends the popular Visual Transformer (ViT) [1] to integrate multimodal feedback from vision and touch – the Visuo-Tactile Transformer (VTT). Our approach focuses

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

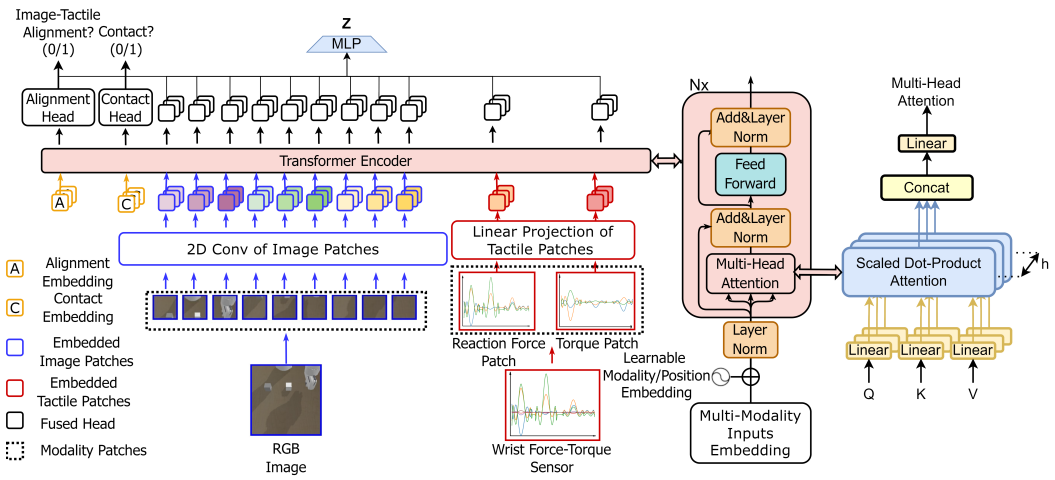

Figure 1: **Architecture:** Schematic of our proposed Visuo-Tactile Transformer architecture.

spatially-aware attention on important visual task features using tactile feedback. We show how this cross-modal attention creates a rich spatially distributed latent space that facilitates more efficient task execution. We evaluate the efficacy of our approach on four simulated and one real-world contact-rich interactions, benchmarking against state-of-the-art techniques.

2 Related Work

The learning community has made significant progress in latent space dynamics learning, demonstrating promise in relatively low-dimensional [20, 21, 22, 23] and high-dimensional [6, 7, 8, 9, 10] spaces. The key idea in these approaches is to learn a latent dynamics model that compactly represents the high-dimensional observation space that is then used for planning or model-based RL. Many of these approaches consider continuous control tasks that may be applicable to the robotics domain [24, 25, 26, 27, 28, 29]. Existing approaches focus on learning latent dynamics from visual feedback. Here, our work explicitly addresses multimodal learning in the visuo-tactile domain.

The robot learning community has also made progress in learning latent space dynamics. In particular, cross-modal learning has recently gained traction [11, 12, 13, 14, 15, 16, 17, 18] where the current trend is in visuo-tactile representations. Our work is most similar to [11, 12] and we use their encoders as our baselines. The key difference between these works and ours is the latent representation. Rather than using a vector in $\mathbb{R}^n$, VTT learns a multimodal latent heatmap where attention is distributed spatially. We hypothesize that VTT's explicitly spatially-aware cross-attention is more effective for representation learning in terms of sample efficiency and expected reward than either of these approaches. We evaluate this claim by benchmarking against existing approaches.

Our proposed approach is an extension of the Transformer [19] and ViT [1] to the visio-tactile domain. Recent research has demonstrated the efficacy of attention mechanisms for sequence-to-sequence and spatially-distributed inputs. A tighter integration of encoders developed by the computer vision community [30, 31, 32] with robotic applications can lead to significant advances in representation learning that enable efficient planning and RL.

3 Visuo-Tactile Transformers

At a high level, VTT produces compact latent representations by fusing visuo-tactile inputs with self and cross-modal attention mechanisms. This enables robotic agents to leverage the complementary nature of these sensing modalities for policy learning. Additionally, VTT supplements image and tactile input patches with learned contact, alignment, and position/modality embeddings to further improve multi-modal reasoning. The overall VTT architecture is shown in Fig. 1. We present the details of the architecture and its components in the following sections.

3.1 Modality Patches

Before sending inputs to the transformer encoder to compute self and cross-modal attention, both the image and tactile inputs need to be patched. RGB image inputs are patched and embedded with one 2-D convolution layer as in ViT [1]. Tactile inputs from a wrist-mounted force-torque sensor are patched into reaction forces and torques, then a linear projection is applied to embed them. Image and tactile modalities are notated I and T respectively. We denote the embedded modality patch as $X_M \in \mathbb{R}^{(P_I+P_T)\times d}$, where P is the number of patches in each modality and d is the number of features in each modality, which we assume to be identical. This embedded modality patch X_M can be written $[X_I; X_T]$ and is used as input to the first attention layer, as shown in Fig. 1.

3.2 Self and Cross-Modal Attention

We implement self and cross-modal attention mechanisms by adapting the multi-head attention method from Transformer architecture [19]. This adaptation is shown on the right side of Fig. 1. The transformer encoder is composed of a stack of N identical attention layers. Each attention layer has two sub-layers: a multi-head attention sub-layer and a feed-forward sub-layer. When $1 \leq n+1 \leq N$, the output X_n of the nth attention layer A_n will be the input for the $(n+1)^{\text{th}}$ attention layer in the transformer encoder. To show how self and cross-modal attention are computed through N attention fusion iterations, we first analyze $A_{n=1}$ and then generalize to $n > 1$ iterations.

To calculate the self and cross-modal attention at iteration $n = 1$, we first project the post layer normalization (LN) of embedded modality patch X_M into the Query ($Q^i_{n=1}$), Key ($K^i_{n=1}$), and Value ($V^i_{n=1}$) shown in Eq. 1, where h is the number of attention heads and $1 \leq i \leq h$.

$$Q^i_{n=1} = LN\begin{bmatrix} X_I \\ X_T \end{bmatrix} W^i_Q,\ \ K^i_{n=1} = LN\begin{bmatrix} X_I \\ X_T \end{bmatrix} W^i_K,\ \ V^i_{n=1} = LN\begin{bmatrix} X_I \\ X_T \end{bmatrix} W^i_V \tag{1}$$

In this formulation, $W^i_Q \in \mathbb{R}^{d\times d_K}$, $W^i_K \in \mathbb{R}^{d\times d_K}$, and $W^i_V \in \mathbb{R}^{d_K\times \frac{d}{h}}$ are weights for Query, Key and Value, where d is defined in Sec. 3.1 and d_K is the dimension of features in the Key. Following Eq. 1, we note that the Query, Key, and Value can be written as

$$Q^i_{n=1} = \begin{bmatrix} Q_I \\ Q_T \end{bmatrix}^i_{n=1},\ \ K^i_{n=1} = \begin{bmatrix} K_I \\ K_T \end{bmatrix}^i_{n=1},\ \ V^i_{n=1} = \begin{bmatrix} V_I \\ V_T \end{bmatrix}^i_{n=1} \tag{2}$$

After projecting the embedded modality patch X_M into $Q_{n=1}$, $K_{n=1}$, and $V_{n=1}$, we use Eq. 3 to calculate the attention at each head i using a scaled dot product with regularization factor $a = \sqrt{d}$.

$$A^i_{n=1} = Atten(X_{n-1} = X_M)^i = softmax(\frac{Q^i_{n=1}(K^i_{n=1})^T}{a})V^i_{n=1} \tag{3}$$

For clarity and brevity, we expand Eq. 3 in terms of Q, K, and V to isolate self and cross-modal attention.

$$\begin{aligned} A^i_{n=1} \sim QK^TV &= \begin{bmatrix} Q_I \\ Q_T \end{bmatrix} [K_I \quad K_T] \begin{bmatrix} V_I \\ V_T \end{bmatrix} = \underbrace{\begin{bmatrix} Q_IK_I & Q_IK_T \\ Q_TK_I & Q_TK_T \end{bmatrix}}_{Attention\,Heatmap} \begin{bmatrix} V_I \\ V_T \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} Q_IK_IV_I + Q_IK_TV_T \\ Q_TK_IV_I + Q_TK_TV_T \end{bmatrix}^i}_{Attention\,Output} = \underbrace{\begin{bmatrix} Q_IK_IV_I + \mathbf{0} \\ \mathbf{0} + Q_TK_TV_T \end{bmatrix}}_{A^{i,self}_{n=1}} + \underbrace{\begin{bmatrix} \mathbf{0} + Q_IK_TV_T \\ Q_TK_IV_I + \mathbf{0} \end{bmatrix}}_{A^{i,cross}_{n=1}} \end{aligned} \tag{4}$$

From Eq. 4 we can see that the attention heatmap is constructed with image and tactile self attentions Q_IK_I and Q_TK_T on the diagonal and cross-modal attentions Q_IK_T and Q_TK_I in the upper and lower triangles. The attention output of Eq. 4 is the sum of self and cross-modal attention. Finally, the multi-head attention $A_{n=1}$ is formed by concatenating each attention head as shown in Eq. 5. Multi-head attention encourages the model to explore the attention subspace.

$$A_{n=1} = [A^1_{n=1}, A^2_{n=1}, ..., A^h_{n=1}] \in \mathbb{R}^{(P_I+P_T)\times d} \tag{5}$$

To connect the sub-layers in each attention layer, we implement the residual connection method from [33]. Finally, the output $X_{n=1}$ of the attention layer $A_{n=1}$ can be written

$$X_{n=1} = f_{n=1}(A^{self}_{n=1}, A^{cross}_{n=1}, X_{n-1} = X_M), \tag{6}$$

where $f_{n=1}$ is a learned nonlinear function induced by nonlinear activation functions and operations.

To generalize to $n > 1$ attention fusion iterations, X_{n-1} is the input for attention layer A_n and is projected into Q_n, K_n and V_n using Eq. 1. Based on Eqs. 4 and 6, we can write A_n as the nonlinear combination of self attention and cross-modal attention shown in the Eq. 7. This proposed attention mixture allows the agent to iteratively leverage weights of self and cross-modal attention.

$$A_n = Atten(X_{n-1}) = Atten(f_{n-1}(A^{self}_{n-1}, A^{cross}_{n-1}, X_{n-2})) \tag{7}$$

3.3 Learned Embeddings

To improve feature learning for RL policies, we propose three learnable embeddings to bolster the modality patches and attention mechanisms we introduced in Sec. 3.1 and Sec. 3.2. We evaluate effects of these embeddings on model performance in the ablation study presented in Sec. 4.2.

Contact Embedding: Intuitively, contact embeddings are used to pull latent codes for alike contact states together while pushing latent codes for dissimilar contact states apart. After N iterations, the contact embedding $X_C \in \mathbb{R}^{1\times d}$ becomes the contact head $C_{head} \in \mathbb{R}^{1\times d}$ following the same method outlined in Eq. 7. We form a contact loss in Eq. 8 by using C_{head} to predict the contact state of the system then comparing that prediction to the ground truth C_{gt}.

Alignment Embedding: Although contact embeddings introduce binary contact recognition, temporal alignment of tactile signatures to images is non-trivial [11, 12]. Alignment embeddings shape the latent space to seek similarities in temporally aligned modalities and differences in temporally misaligned modalities [11, 34]. Therefore, alignment embedding $X_{Al} \in \mathbb{R}^{1\times d}$ is introduced and becomes the alignment head $Al_{head} \in \mathbb{R}^{1\times d}$ after N iterations. As in the previous section, we form a loss in Eq. 8 by using Al_{head} to predict whether the tactile and image data are aligned then comparing that prediction to the ground truth Al_{gt}.

The contact and alignment losses are defined by a binary cross entropy with logits loss $BCE_{logits}(\cdot)$.

$$\ell_{VTT} = BCE_{logits}(MLP(Al_{head})), Al_{gt}) + BCE_{logits}(MLP(C_{head}), C_{gt}) \tag{8}$$

Position/Modality Embedding: The final type of learnable embedding we implement is a position/modality embedding. We found that $[X_C; X_{Al}; X_M]$ do not contain the position or modality of information, so it is necessary to include position/modality embedding $X_P \in \mathbb{R}^{1\times(2+P_I+P_T)}$. Inspired by [35, 36], we add this embedding to our structure to form $[X_C; X_{Al}; X_M] + X_P$.

3.4 Discussion of Compressed Representation Head

The output X_N of the transformer encoder is a series of fused heads originating from the visuo-tactile embedded patches X_M, contact embeddings X_C, alignment embeddings X_{Al}, and position/modality embeddings X_P. In order to use our learned representation from VTT for RL, we need to think about the dimensionality of the latent vector $\mathbf{z}$. If we use $X_N \in \mathbb{R}^{(2+P_I+P_T)\times d}$ directly, $\mathbf{z}$ will be too high-dimensional. Consequently, we propose compressing all fused heads from $\mathbb{R}^{(2+P_I+P_T)\times d}$ to $\mathbb{R}^{(2+P_I+P_T)\times \frac{d}{c}}$ with constant $c > 4$ by using an MLP. This compression results in $\mathbf{z} \in \mathbb{R}^{1\times \frac{d}{c}(2+P_I+P_T)}$. Our proposed method will allow image and tactile attention heads from VTT to be used for latent dynamics learning while introducing minimum inductive bias.

3.5 Combining with Reinforcement Learning

Due to the frequency of visual occlusions and high levels of tactile perception uncertainty in our manipulation benchmarks, we formulate our manipulation tasks as partially observable Markov decision processes (POMDPs) [37]. There have been a number of advances in model-based RL for POMDPs [38, 6, 39, 40]. From these, we chose the Stochastic Latent Actor Critic (SLAC) [6] algorithm because of its stability and robustness. These characteristics accommodate high stochasticity and provide an effective baseline method to compare visuo-tactile fusion mechanisms.

SLAC is composed of a model-learning component and a policy-learning component. The model-learning piece is built with a factorized sequential variational autoencoder (VAE) f_θ [41]. f_θ uses a Gaussian distribution to form a prior model p and a posterior model q. The prior model $p(\mathbf{z}_t^d|\mathbf{z}_{t-1}^d, a_{t-1})$ propagates latent dynamics from an action a and the posterior model $q(\mathbf{z}_t^d|\mathbf{z}_{t-1}^d, \mathbf{z}_{t-1}, a_{t-1})$ integrates observations. In this notation, $\mathbf{z}_t^d \sim f_\theta(\mathbf{z}_{0:t}, \mathbf{z}_{0:t-1}^d, a_{0:t-1})$. To close the distribution gap between the prior and posterior models, KL divergence is introduced. Finally, $\mathbf{z}_t^d$ is decoded to reconstruct the visual input O and infer a reward r. VTT losses from Eq. 8 are composed with SLAC losses to form the total model-learning losses in Eq. 9.

$$\ell_{model} = \ell(O_t|\mathbf{z}_t^d, a_{t-1}) + \ell(r_t|\mathbf{z}_t^d, a_{t-1}) + \ell_{KL}(q||p) + \ell_{VTT} \tag{9}$$

The policy-learning component is implemented using the soft actor critic (SAC) algorithm [5]. SAC is an off-policy actor-critic deep RL algorithm that maximizes future return and entropy for exploration. For further details and derivation on the model and policy learning, see [5, 6]. It is important to note that the latent representation $\mathbf{z}$ from VTT is utilized by both components. Additionally, the critic's value function loss is backpropagated through VTT to the attention mechanism.

4 Experiments and Results

In this section, we benchmark VTT's performance against two popular multi-modality fusion techniques on four simulated robot manipulation tasks. Next, we present an ablation study that evaluates the importance of our contact loss and alignment loss on simulation tasks. We then evaluate VTT on a real-world pushing task. Finally, we visualize the attention maps produced by VTT on simulation and real-world tasks to illustrate the effects and intuition of cross-modal attention in the latent space.

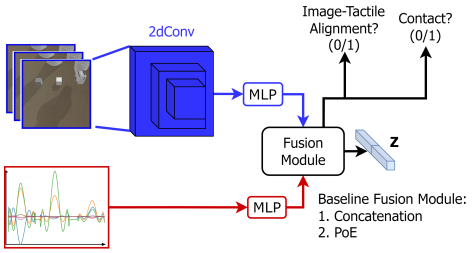

Figure 2: **Baseline Structure**: We compare VTT to concatenation and product of experts (PoE) baselines. These methods are inserted at the 'fusion module'.

4.1 Simulated Experiments

Baseline Structures: To evaluate the performance of VTT, we benchmark against two popular multi-modality fusion methods: concatenation and product of experts (PoE) [11, 12]. Fig. 2 shows the architecture of these baselines, where the fusion model is the key distinction between the methods. Both baselines follow the same training pipeline as VTT to allow for fair comparisons. We denote the inputs of the fusion module as $E_I \in \mathbb{R}^{1\times d_I}$ and $E_T \in \mathbb{R}^{1\times d_T}$.

In the concatenation fusion module, the latent vector $\mathbf{z}$ becomes $\mathbf{z} \in \mathbb{R}^{1\times(d_I+d_T)} = [E_I; E_T]$. However, the PoE fusion module treats E_I and E_T independently and applies a VAE-like structure to each latent code individually. E_I and E_T pass through separate MLPs to generate separate means (μ_{E_I}, μ_{E_T}) and variances $(\sigma_{E_I}, \sigma_{E_T})$ that are mapped to a normal distribution using KL divergence, then fused using PoE as shown in Eq. 10 for i modalities and j features. The latent space in this method is also shaped by contact and alignment classification. Further details about each of these baselines can be found in their published works.

$$\sigma_j^2 = \left(\sum_{i=1}^{2} \sigma_{ij}^2\right)^{-1}, \mu_j = \left(\sum_{i=1}^{2} \frac{\mu_{ij}}{\sigma_{ij}^2}\right)\left(\sum_{i=1}^{2} \sigma_{ij}^2\right)^{-1} \tag{10}$$

Simulation Experiment Setup: We conduct simulated experiments of four manipulation tasks in Pybullet. The selected tasks are Pushing, Door-Open, Picking, and Peg-Insertion. Based on the continuous control manipulation benchmark developed by ElementAI [42], we construct a dense reward setting for these tasks. The agent receives an $84 \times 84 \times 3$ RGB image and a 1×6 wrist reaction wrench. Task rewards and parameters are outlined below.

1. Pushing Push a white block to a grey target pose. The agent receives +25 reward for finishing the task. In this task, we vary the initial pose and mass of the block as well as the target pose.

2. Door-Open Open a door from $\theta_{door} = 0$ to θ_{goal}. When $\theta_{door} < \theta_{goal}$, we penalize the agent $-\lambda(\theta_{goal} - \theta_{door})$. When $\theta_{goal} = \theta_{goal}$, the agent receives +25 reward. We incorporate curriculum learning [43] into this task by randomly initializing $\theta_{door} > 0$ during training only. In this task, we vary the color of the door and frame.

3. Picking Pick up a white block. We force the gripper to touch the block with a location-based -$\lambda(|Loc_{gripper} - Loc_{block}|)$ penalty. The agent receives +1 reward for grasping when the wrist feels significant downward force. The agent receives +25 reward when the block is lifted to a certain height. In this task, we vary the initial pose, mass, and size of the block.

4. Peg-Insertion Insert a peg into a hole. We force peg-hole alignment with a location-based $-\lambda(|Loc_{peg} - Loc_{hole}|)$ penalty. To encourage the agent to lower the peg into the hole, the agent receives +$\lambda(|z_{max} - z_{peg})|)$ reward after alignment. The agent receives +25 reward when the peg in fully inserted. In this task, we vary the color of the peg and the color of the hole.

Training Details: We follow the SLAC training schedule to pretrain the dynamics model f_θ and fusion modules VTT, concatenation, and PoE. For the model we use $lr = 0.0001$ and batch size 32. For the policy we use $lr = 0.0003$ and batch size 256. We train using the Adam optimizer [44]. Further implementation details are included in the supplementary material.

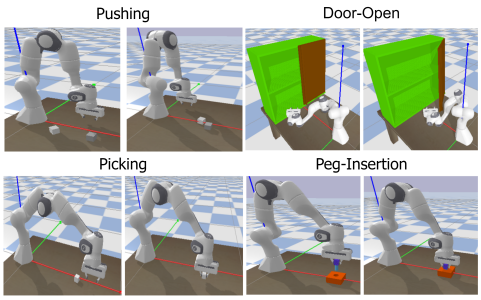

Figure 3: **Manipulation Tasks**: We evaluate VTT on four tasks in Pybullet. We vary visual and physical parameters in each task.

Baseline Comparison: When comparing VTT to the concatenation and PoE baselines, we selected the top five of eight results from each fusion module. The results for each task with each fusion module are shown in Fig. 4. Compared to the baselines, our proposed approach achieves higher sample efficiency on all tasks and higher success rate in all but the Peg-Insertion task. Interestingly, peg insertion was the primary task evaluated in the baseline papers [11, 12]. The notable distinction between this task and the others is that the robot is always grasping the peg and only has to make contact with the table and insert. The remaining tasks require the robot to first reach for and grasp/push an object, then perform the remainder of the task. This may suggest that peg insertion is an easier task that does not require as much spatial reasoning. Additionally, our approach shows lower variance w.r.t. to the baselines. This suggests that latent representation is numerically easier and more stable to learn.

4.2 Ablation Study

To show the importance of contact and alignment losses in VTT, we conduct an ablation study by sequentially removing the contact and alignment embeddings and losses and performing the Door-Open and Peg-Insertion tasks. Overall, our results in Fig. 5a indicate that each of these methods

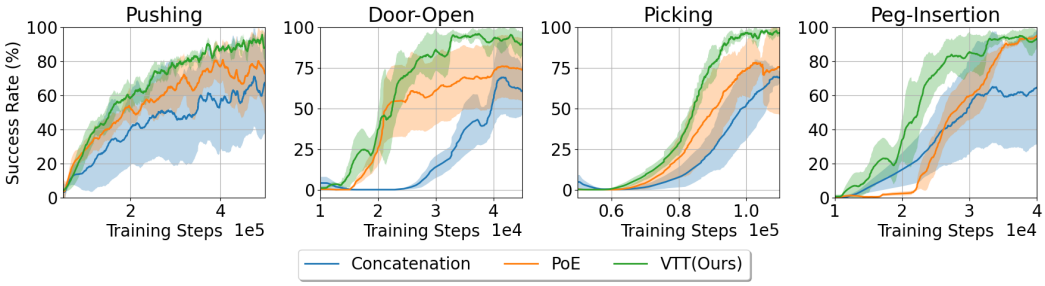

Figure 4: **Simulation Results:** We compare the performance of our method to baselines on the four simulated tasks and find that VTT outperforms the concatenation and PoE baselines in all but one task. The variance of VTT's learning curve is significantly smaller than either baseline, which indicates that representations are learned more easily by VTT.

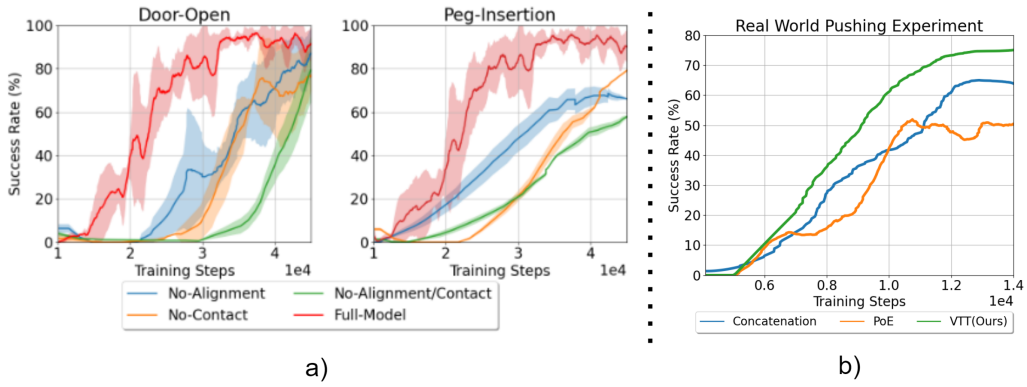

Figure 5: **a) Ablation Results:** We sequentially remove the contact and alignment losses from VTT to show performance in two simulated tasks without them. We find that the full method outperforms each partial method. **b) Real-World Results:** We find that VTT also outperforms the baselines in the real-world Pushing task in terms of sample efficiency and success rate.

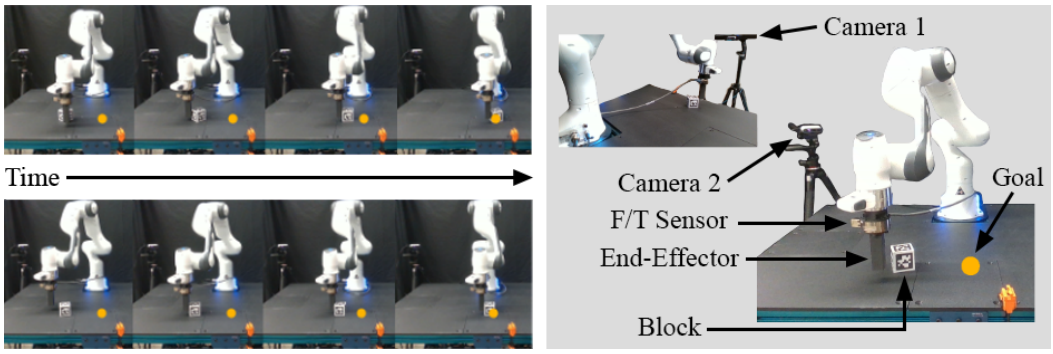

Figure 6: **Real-World Experimental Setup:** In the left pane we show two successful block pushing tasks. In the right pane we note the location of our cameras and F/T sensor as well as the task-specific end-effector, block, and goal.

is critical for achieving high sampling efficiency and success rate. The lack of contact and alignment losses can cause miscorrelation of images and tactile feedback. The representation induced by such miscorrelations may mislead the agent's state value estimation and decision making, leading to worse overall performance. Additionally, we notice that the agent performs worse without the contact loss than without the alignment loss. This behavior is likely due to the contact-rich tasks we use to evaluate our method. One potentially useful application of this misalignment may be as a metric for evaluating just how out of distribution a novel observation is.

4.3 Real-World Experiments

Real-World Experiment Setup: We conduct the Pushing task in the real world using a Franka Emika Panda robot arm with a wrist-mounted ATI Gamma F/T sensor and custom end-effector. The block is a 2.25-in steel cube with Apriltags on its visible sides. These tags are tracked by two Intel RealSense D435 cameras located in front of the robot (Camera 1) and to the back right of the robot (Camera 2). We use the RGB image from Camera 1 as input E_I and utilize Camera 2 to track the block in case of occlusion. We use the reaction force and torque from the F/T sensor as input E_T. This setup is shown in Fig. 6. The agent will receive +25 reward for finishing the task and $-10|Loc_{block} - Loc_{goal}|$ penalty when the task is not finished. Each trial has a 100 step maximum.

Real-World Baseline Comparison: Similarly to the simulated tasks, we evaluated VTT against the concatenation and PoE baselines. As in the simulation results, our approach outperforms the other fusion modules in the real world as shown in Fig. 5.

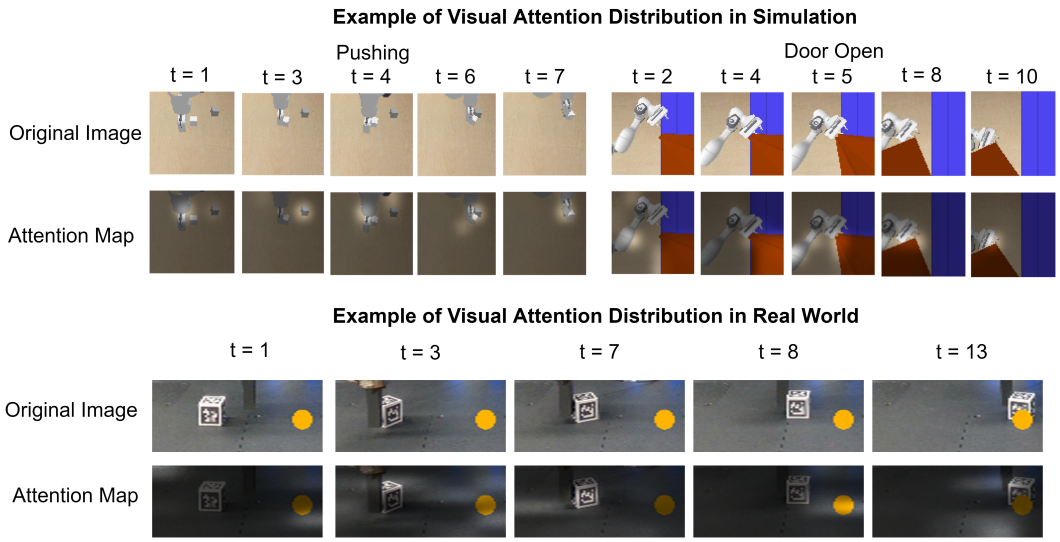

Figure 7: **Attention Heatmap:** We ovelay the attention heatmaps from Eq. 4 onto RGB inputs.

4.4 Attention Map Visualization

We illustrate the output of VTT's attention mechanism in Fig. 7. The upper half shows the attention heatmaps from single trials of simulated Pushing and Door-Open tasks, while the lower half shows the real-world Pushing task. These heatmaps are the average of the multi-head attention from Eq. 4. We empirically observe 2 distinct phases: contact-free and in-contact. In contact-free steps, we observe the attention heatmap highlight the robot and the object (Pushing $t = 1$, Door-Open $t = 2$). This is likely due to correlation between the goal and the change in the tactile signature that arises from contact between end-effector and object. Once contact is present, the attention focuses mostly on the contact interaction and the goal. Again, this is likely due to the correlation between the reward signal and the tactile signature. We note that the object and the goal can be far from the robot in the visual scene, yet our attention mechanism is able to correctly identify these key features.

5 Discussion and Limitations

In this work we present Visuo-Tactile Transformers, a novel multi-modal encoding method for model-based RL and planning. Our approach fuses visuo-tactile feedback using a self and cross-modal attention mechanism. This attention is spatially distributed, which allows our approach to attend to features of interest across an image that are correlated with tactile feedback. We expect that VTT will enable important downstream applications such as visuo-tactile curiosity. Curiosity [45] makes exploration more effective by maximizing a proxy reward. This reward is often formulated as a series of actions that increase representation entropy. Similarly to [46], VTT can measure potential mismatch in vision and touch and use this score to explore. VTT advantageously incorporates spatial attention to localize regions of interest in an image, which sets it apart from other multi-modal representation methods.

Despite its promise, the current iteration of VTT has a few limitations. Firstly, our implementation uses 6-dimensional reaction wrenches from F/T sensors as tactile feedback. This low-dimensional signal can often alias important contact events and is generally not as informative as collocated tactile sensing like [47, 48]. To address this, the tactile patching and embedding need to be adapted to handle high-dimensional tactile signatures. Secondly, we only evaluated VTT on rigid-body interactions. This choice was motivated by the availability of simulation platforms that incorporate both tactile feedback and deformables and that are fast enough for RL applications. It is not entirely clear how VTT will handle deformables; though we suspect that attention will be distributed over both the deformation and the contact region.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[3] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[6] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020.

[7] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[8] T. De Bruin, J. Kober, K. Tuyls, and R. Babuška. Integrating state representation learning into deep reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):1394–1401, 2018.

[9] D. Hafner, T. P. Lillicrap, I. Fischer, R. Villegas, L. H. Ha, David, and J. Davidson. Learning latent dynamics for planning from pixels. *CoRR, abs/1811.04551*, 2018.

[10] A. Stooke, K. Lee, P. Abbeel, and M. Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pages 9870–9879. PMLR, 2021.

[11] 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, 2019.

[12] 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*, 36(3):582–596, 2020.

[13] Y. Li, J.-Y. Zhu, R. Tedrake, and A. Torralba. Connecting touch and vision via cross-modal prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10609–10618, 2019.

[14] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez. See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. *Science Robotics*, 4 (26):eaav3123, 2019.

[15] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox. Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6444–6451. IEEE, 2021.

[16] W. Zheng, H. Liu, and F. Sun. Lifelong visual-tactile cross-modal learning for robotic material perception. *IEEE transactions on neural networks and learning systems*, 32(3):1192–1203, 2020.

[17] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint. Learning multi-object dynamics with compositional neural radiance fields. *arXiv preprint arXiv:2202.11855*, 2022.

[18] Y. Wi, P. Florence, A. Zeng, and N. Fazeli. Virdo: Visio-tactile implicit representations of deformable objects. *arXiv preprint arXiv:2202.00868*, 2022.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[20] M. Deisenroth and C. E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472. Citeseer, 2011.

[21] Y. Gal, R. McAllister, and C. E. Rasmussen. Improving pilco with bayesian neural network dynamics models. In *Data-Efficient Machine Learning workshop, ICML*, volume 4, page 25, 2016.

[22] M. Henaff, W. F. Whitney, and Y. LeCun. Model-based planning with discrete and continuous actions. *arXiv preprint arXiv:1705.07177*, 2017.

[23] B. Amos, L. Dinh, S. Cabi, T. Rothörl, S. G. Colmenarejo, A. Muldal, T. Erez, Y. Tassa, N. de Freitas, and M. Denil. Learning awareness models. *arXiv preprint arXiv:1804.06318*, 2018.

[24] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control*, pages 1154–1168. PMLR, 2021.

[25] O. Rybkin, C. Zhu, A. Nagabandi, K. Daniilidis, I. Mordatch, and S. Levine. Model-based reinforcement learning via latent-space collocation. In *International Conference on Machine Learning*, pages 9190–9201. PMLR, 2021.

[26] A. Allshire, R. Martín-Martín, C. Lin, S. Manuel, S. Savarese, and A. Garg. Laser: Learning a latent action space for efficient reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6650–6656. IEEE, 2021.

[27] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

[28] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

[29] M. Okada and T. Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4209–4215. IEEE, 2021.

[30] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy, 7 2019. Association for Computational Linguistics.

[31] X. Wei, T. Zhang, Y. Li, Y. Zhang, and F. Wu. Multi-modality cross attention network for image and sentence matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10938–10947, 2020. doi:10.1109/CVPR42600.2020.01095.

[32] P. Gao, H. You, Z. Zhang, X. Wang, and H. Li. Multi-modality latent interaction network for visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[33] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[34] T. Baltrušaitis, C. Ahuja, and L.-P. Morency. Multimodal machine learning: A survey and taxonomy, 2017.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[36] M. S. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova. Tokenlearner: What can 8 learned tokens do for images and videos?, 2021.

[37] H. Kurniawati. Partially observable markov decision processes (pomdps) and robotics, 2021.

[38] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for pomdps, 2018.

[39] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels, 2019.

[40] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination, 2020.

[41] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2014.

[42] S. Rajeswar, C. Ibrahim, N. Surya, F. Golemo, D. Vazquez, A. Courville, and P. O. Pinheiro. Touch-based curiosity for sparse-reward tasks, 2021.

[43] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone. Curriculum learning for reinforcement learning domains: A framework and survey, 2020.

[44] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[45] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[46] S. Rajeswar, C. Ibrahim, N. Surya, F. Golemo, D. Vazquez, A. Courville, and P. O. Pinheiro. Touch-based curiosity for sparse-reward tasks. *arXiv preprint arXiv:2104.00442*, 2021.

[47] I. Taylor, S. Dong, and A. Rodriguez. Gelslim3. 0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. *arXiv preprint arXiv:2103.12269*, 2021.

[48] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake. Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 597–604. IEEE, 2019.