

Inference-time Policy Steering via Vision and Touch

Yilin Wu

Carnegie Mellon Univeristy
yilinwu@andrew.cmu.edu

Zilin Si

Carnegie Mellon Univeristy
zsi@andrew.cmu.edu

Zeynep Temel

Carnegie Mellon Univeristy
ztemel@andrew.cmu.edu

Oliver Kroemer

Carnegie Mellon Univeristy
okroemer@andrew.cmu.edu

Andrea Bajcsy

Carnegie Mellon Univeristy
abajcsy@andrew.cmu.edu

Abstract: Inference-time steering adapts pre-trained generative robot policies during deployment by verifying candidate actions before execution. While prior methods typically perform this verification only with visual observations, vision alone is often insufficient for contact-rich manipulation, where success depends on both global task progress and subtle local interactions such as contact force. We introduce **ViTaL**, a visuo-tactile inference-time steering framework that formulates multimodal guidance as a bi-level optimization problem. At the high level, visual sampling-and-verification performs long-horizon mode selection, deciding *what* behavior the robot should execute. At the low level, tactile-guided diffusion editing refines the selected action sequence over a shorter horizon to satisfy local contact requirements. To support outcome-based steering, **ViTaL** learns a visuo-tactile latent world model and employs semantically aligned visual and tactile verifiers, including a novel text-conditioned tactile reward that scores predicted tactile futures directly in latent space. Across three real-world contact-rich manipulation tasks, **ViTaL** improves overall success by **51%** over the base policy, outperforms unimodal steering by at least **33%**, and exceeds naive multimodal fusion by at least **20%**. Website: corl2026vital.github.io.

Keywords: Inference-time Policy Steering, Visuo-Tactile Policy for Manipulation

1 Introduction

Inference-time steering [1, 2, 3, 4, 5, 6, 7, 8] offers a promising way to adapt pre-trained generative robot policies [9, 10] at deployment. Instead of retraining, inference-time steering treats a stochastic policy, e.g., a diffusion policy, as an action proposal generator and uses a verifier or reward to refine candidate actions before execution. Recent work have largely focused on visual verification: candidate actions are rolled out into predicted visual futures, such as images or latents, and then scored with semantic task progress [1, 2, 3, 8]. This is effective when task success is fully observable by visual progress in the scene, such as reaching the correct object or placing it in the desired region.

For contact-rich manipulation, however, steering must guide not only the visual progress, but also physical interactions through which the behavior is executed. This motivates *multimodal guidance* that incorporates both visual and tactile guidance during steering. Consider the pipetting task in Figure 1: the robot must choose the right high-level visual mode, such as moving toward the yellow cup, while also using touch to maintain a stable grasp on the pipette without slipping or applying excessive force. These requirements naturally induce two steering decisions: *which* behavior to pursue, and *how* to realize that behavior in contact. The former often requires long-horizon visual reasoning, since candidate trajectories must be rolled out far enough to reveal distinct semantic outcomes. The latter requires contact-sensitive guidance around interactions, since critical contact events and force transitions are local, transient and may be obscured by aggregate long-horizon visual verifica-

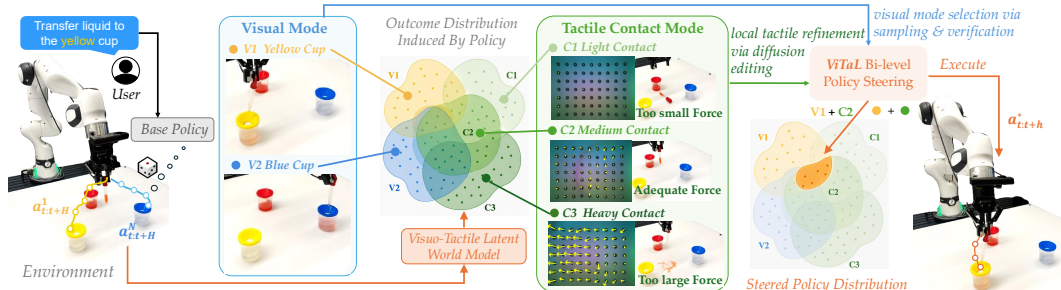


Figure 1: **Multi-Modal Policy Steering.** ViTaL selects visual modes and refines local contact, steering the base policy toward actions satisfying both task goals and contact constraints.

tion. Thus, effective visuo-tactile steering faces two challenges: an *observability and temporal-scale mismatch* between visual mode selection and tactile contact refinement, and a *multimodal reward specification* challenge for expressing phase-dependent visual and tactile objectives.

To address these challenges, our key insight is a bi-level visuo-tactile steering framework that decomposes multimodal guidance into visual mode selection followed by tactile contact refinement. In the visual stage, we perform sampling-and-verification: candidate actions are rolled out with a visuo-tactile latent world model, and a visual verifier selects the sequence that best satisfies the visual objective. In the tactile stage, we treat this selected sequence as an action anchor and apply tactile-guided diffusion editing [11] over a short horizon to improve contact execution while remaining close to the visual plan. This decomposition matches each modality to its role: vision guides the policy toward the intended behavioral mode, while touch refines the physical execution of that mode.

To specify rewards for this bi-level structure, we ground the high-level language instruction into phase-level visual and tactile objectives and evaluate them with semantically-aligned verifiers. A visual verifier scores decoded visual predictions for semantic progress, while our text-conditioned tactile verifier scores predicted tactile latents against textual contact objectives. This enables contact-sensitive steering that is *discriminative, flexible, and efficient*: the verifier can distinguish contacts, adapt to different task phases through language, and guide actions directly in latent space without hand-designing task-specific tactile rewards or decoding tactile predictions. Together, these components define **ViTaL**, a visuo-tactile inference-time steering framework for contact-rich manipulation.

Statement of Contributions. Our contributions are threefold: (i) we propose a bi-level multimodal guidance framework that uses vision for long-horizon semantic mode selection and touch for targeted contact refinement; (ii) we introduce, to our knowledge, the first language-conditioned tactile reward in the latent space, and integrate it with language-conditioned visual rewards and a visuo-tactile latent world model for outcome-based policy steering without task-specific reward learning; and (iii) we conduct extensive real-world experiments across three contact-rich manipulation tasks, where **ViTaL** improves overall success by **51%** over the base policy, exceeds unimodal steering by at least **33%**, and outperforms naive multimodal fusion by at least **20%**.

2 Related Work

Inference-time Steering of Robot Policies. Recent work on inference-time adaption for robot policies typically follow three paradigms: sample-then-verify [1, 4, 6, 8], noise-space optimization [7], and classifier-based guidance [3, 2, 5]. While effective, they all rely on visual guidance signals that only provide global task context but often miss fine-grained contact details in manipulation [12, 13, 14, 15]. A few tactile steering works instead use touch to provide direct contact information [16, 12] but do not capture global progress. They may also need costly demonstrations to learn action-space verifiers without explicit dynamics modeling [12]. By contrast, our work combines visual and tactile guidance at different temporal scales with a learned latent world model.

Tactile Sensing for Robot Decision Making. We use vision-based tactile sensors, specifically GelSight Mini [17] due to its dense measurements of contact geometry, deformation, and force-related signals. The existing datasets with GelSight [17, 18, 19] have been used to pretrain encoders for tactile images [20, 21, 22, 23, 24], providing semantically meaningful latent spaces for tactile reasoning. Tactile sensing has also been widely used for contact-rich manipulation [25, 26, 27, 28, 29], and recent work incorporates tactile feedback into learned policies for fine-grained reactive control [30, 31, 13, 32], primarily during policy training or low-level control. In contrast, we use tactile as an inference-time guidance signal: candidate actions are scored by aligning predicted tactile embeddings with textual contact goals, enabling scalable tactile verification without demonstrations.

World Modeling in Robot Decision Making. Dynamics models have been studied in planning, control, and policy learning for robotics [33, 34, 35]. Recent visual world models focus on learning latent dynamics [36, 37, 38], adapting diffusion or flow-based video models [39, 40, 41] or training with model-based reinforcement learning [42, 43, 44, 45]. Meanwhile, recent visuo-tactile world models incorporate touch into dynamics modeling [14, 16, 46, 47]. In this work, we train a visuo-tactile world model to evaluate actions in the latent space. We use learned latent dynamics to predict future visuo-tactile latents from pretrained encoders. Therefore, actions can be scored directly in the latent space without decoding images, enabling efficient text-conditioned tactile guidance.

3 Problem Formulation: Visuo-Tactile Inference-Time Steering

Notation & Setup. At time t , the robot observes *multimodal* data in the form of RGB camera images $o_t^v \in \mathcal{O}^v$ from C^v cameras (e.g., a third person and a wrist camera), vision-based tactile images $o_t^\tau \in \mathcal{O}^\tau$ from C^τ sensors (e.g., one GelSight Mini [17] attached to the gripper fingertip), and proprioceptive states (e.g. end-effector pose and gripper width), $q_t \in \mathcal{Q}$. Let the concatenated vision, tactile, and proprioceptive observation be denoted by $o_t := [o_t^v, o_t^\tau, q_t]$. We denote h -step *future observation sequences* from time t by $\mathbf{o}_t := o_{t:t+h}$ with vision, tactile, and proprioceptive sequences denoted by \mathbf{o}_t^v , \mathbf{o}_t^τ , and \mathbf{q}_t respectively. Let the robot’s low-level action space be $a \in \mathcal{A}$ (e.g., end-effector cartesian pose and gripper control). A h -step action sequence from time t is denoted by $\mathbf{a}_t := a_{t:t+h}$. The robot generates low-level “action chunks” via a pre-trained¹ diffusion policy, $\mathbf{a}_t \sim \pi_\theta(\cdot | o_t)$

Problem Formulation. Let \mathcal{L} denote a textual description for the high-level goal of the task (e.g. “transfer liquid to the blue cup”). Inference-time policy steering can be formalized as a model predictive control problem: the base policy π_θ generates candidate actions, each sample is forward simulated into future observations which capture *outcomes* of the action sequence, and these outcomes are scored by a verifier for its alignment with task description. When the robot uses multimodal observations, future outcomes require a *visuo-tactile world model* which yields a distribution over vision, tactile, and proprioceptive sequences induced by any action sample: $\mathcal{P}(\mathbf{o}_t^v, \mathbf{o}_t^\tau, \mathbf{q}_t | o_t^v, o_t^\tau, q_t, \mathbf{a}_t)$. Multimodal inference-time steering requires selecting the action sample $\mathbf{a}_t \sim \pi_\theta(\cdot | o_t)$ whose future observations maximize a task-conditioned reward $R(\mathbf{o}_t^v, \mathbf{o}_t^\tau; \mathcal{L})$. The reward R scores how well the predicted visual and tactile futures satisfy the task instruction \mathcal{L} .

Open Challenges. Solving the general multimodal steering problem introduces two key challenges.

Modality-dependent observability across temporal scales. Vision and touch provide complementary but asymmetric observations of task state. Vision captures global scene structure and semantic progress, but the relevant visual evidence may only emerge over a sufficiently long horizon: for example, in the pipetting task, candidate trajectories must be rolled out far enough to determine whether the target cup is approached. In contrast, touch provides direct evidence about physical interaction, such as grasp stability or wiping pressure. These contact-critical events are often local and transient, so they can be obscured from an aggregate long-horizon visual reward. Thus, a single

¹While our method is agnostic to the base policy, we focus on vision-only policies for their prevalence and strong off-the-shelf performance, with the visuo-tactile variant in App C.2.

shared prediction and verification horizon is suboptimal: the horizon needed to distinguish visual behavioral modes can differ from the horizon needed to assess contact quality from touch.

Multi-modal reward modeling. Even with access to visual and tactile observations, defining rewards for contact-rich manipulation is difficult because success depends on both semantic task progress and local physical interaction. A task instruction may specify the final goal, such as transferring liquid, but it often leaves intermediate visual subgoals and contact requirements implicit. Moreover, tactile rewards must distinguish subtle contact states, such as gentle grasping versus dropping, or sufficient versus excessive force. This makes it challenging to construct a unified multi-modal reward that is both informative about global task progress and sensitive to contact quality.

4 ViTaL: Visuo-Tactile Latent Steering

ViTaL addresses the challenges in Sec. 3 by decomposing multimodal steering into long-horizon visual mode selection and short-horizon tactile refinement. The visual stage selects an action mode based on predicted semantic progress, while the tactile stage refines it using predicted contact quality, matching the different temporal roles of vision and touch. We formulate this bi-level objective in Sec. 4.1, introduce a visuo-tactile latent world model for outcome prediction in Sec. 4.2, define visual and tactile verifiers in Sec. 4.3, and describe the actual implementation for steering in Sec. 4.4.

4.1 Bi-level Visuo-Tactile Steering Formulation

Sec. 3 defines the general objective as directly selecting the action chunk whose predicted visual and tactile futures jointly maximize a task-conditioned reward. However, as discussed in open challenges, specifying a single reward that faithfully captures both long-horizon semantic progress and short-horizon contact quality is intractable in practice. We therefore reformulate the original multimodal steering objective as a bi-level optimization problem. The outer optimization performs an H -step visual lookahead using the latent visuo-tactile world model and selects a globally promising action mode $\bar{\mathbf{a}}_{t:t+H}$ with visual task reward R^v . The inner optimization then refines the first h steps of this sequence to get the final action chunk $\mathbf{a}^*_{t:t+h}$ using tactile reward R^τ , while remaining close to the visual plan $\bar{\mathbf{a}}_{t:t+h}$. This decomposition allows vision to guide global mode selection, while tactile locally adjusts the action sequence to improve contact-rich execution.

$$\begin{aligned}
 \mathbf{a}^*_{t:t+h} &= \arg \max_{\tilde{\mathbf{a}}_{t:t+h}} \overbrace{\log p_\theta(\tilde{\mathbf{a}}_{t:t+h} \mid \bar{\mathbf{a}}_{t:t+h}, o_t)}^{\text{visual prior around selected mode}} + \beta \cdot \overbrace{R^\tau(\hat{\mathbf{z}}_{t:t+h}^\tau(\tilde{\mathbf{a}}); \mathcal{L})}^{\text{local tactile contact reward}} \\
 \text{s.t. } \bar{\mathbf{a}}_{t:t+H} &= \arg \max_{\mathbf{a}_{t:t+H} \sim \pi_\theta(\cdot \mid o_t)} \underbrace{R^v(\hat{\mathbf{z}}_{t:t+H}^v(\mathbf{a}); \mathcal{L})}_{\text{selected action mode from visual reward}}, \\
 \hat{\mathbf{z}}_{t:t+h'}^m(\mathbf{a}) &\sim p_\phi(\cdot, \cdot \mid z_t^v, z_t^\tau, \mathbf{a}_{t:t+h'}), \quad h' \in \{h, H\}, \quad z_t^m = \mathcal{E}_\psi^m(o_t^m), \quad m \in \{v, \tau\}.
 \end{aligned} \tag{1}$$

Here, p_θ is a visual action prior centered at the sequence $\bar{\mathbf{a}}$ which is selected by visual lookahead: candidate sequences $\mathbf{a}_{t:t+H} \sim \pi_\theta(\cdot \mid o_t)$ are rolled out with the latent visuo-tactile dynamics model p_ϕ , and the one with the highest long-horizon visual reward $R^v(\hat{\mathbf{z}}_{t:t+H}^v; \mathcal{L})$ is chosen to encourage global task progress. The inner optimization then locally refines this visually selected mode by reweighting short-horizon action chunks according to the tactile reward $R^\tau(\hat{\mathbf{z}}_{t:t+h}^\tau; \mathcal{L})$, which measures predicted h -step contact quality, with β controlling the strength of tactile refinement relative to the visual prior. Thus, our formulation uses vision for global mode selection and touch for local contact refinement, rather than optimizing a single joint visual-tactile reward over a shared horizon.

4.2 Visuo-Tactile Latent World Model for Multi-Modal Latent Outcome Prediction

Eq. 1 requires a latent world model p_ϕ that predicts the visual and tactile futures induced by candidate action sequences. We encode the current visual and tactile observations separately as $z_t^v = \mathcal{E}_\psi^v(o_t^v)$, $z_t^\tau = \mathcal{E}_\psi^\tau(o_t^\tau)$, $z_t = (z_t^v, z_t^\tau)$, where \mathcal{E}_ψ^v is pretrained DINOv3 encoder [48] and

\mathcal{E}_ψ^τ is pretrained AnyTouch2 encoder [21] with structured latent spaces for scene understanding and tactile reasoning. On top of the frozen encoders, we train an action-conditioned latent dynamics model inspired by [49]. Starting from $\hat{z}_t = z_t$, the model recursively predicts future joint visuo-tactile latents over action chunks: $\hat{z}_{t+h} = p_\phi(\hat{z}_t, a_{t:t+h})$. We train the model with a multi-step latent prediction objective following [14] with details in App B.2. We additionally train decoders $\hat{\delta}^v = \mathcal{D}_\varphi^v(z^v)$, $\hat{\delta}^\tau = \mathcal{D}_\varphi^\tau(z^\tau)$ for visualization. This model enables long-horizon visual lookahead for global mode selection and short-horizon tactile prediction for contact-sensitive refinement.

4.3 Multi-modal Reward Modeling for Verification

The latent world model enables multimodal prediction for candidate action sequences. The remaining challenge is to define rewards that capture both global task progress and local contact quality, which may vary across task phases. We therefore use language-conditioned visual and tactile verifiers, with phase-dependent textual objectives specifying the relevant criteria for each modality.

Phase-Dependent Verification Objectives. A single task instruction is often too coarse for long-horizon manipulation: e.g., “transfer liquid to the blue cup and return” does not explicitly specify intermediate visual objectives, such as moving back to the red cup, or contact requirements, such as squeezing only during dispensing. We therefore use a VLM offline to decompose the task into phase-level textual objectives, $\{\ell_p\}_{p=1}^P = g^{\text{VLM}}(\mathcal{L})$, where each phase provides a visual objective ℓ_p^v and a tactile objective ℓ_p^τ . The visual verifier determines when to switch to the next phase.

Visual Verifier. The visual verifier scores whether a predicted visual future makes progress toward the visual objective ℓ_p^v . Since off-the-shelf visual reward models operate on images, we decode predicted visual latents and adapt ROBOMETER [50] for online visual reward evaluation. $R^v(\hat{z}_{t:t+H}^v; \ell_p^v) = \mathcal{R}_{\text{RM}}^v(\hat{\delta}_{t+H}^v; \ell_p^v)$, $\hat{\delta}_{t+H}^v = \mathcal{D}_\varphi^v(\hat{z}_{t+H}^v)$. This reward is used for long-horizon visual mode selection, where the goal is to distinguish globally meaningful task progress, such as moving toward the correct cup, target mark, or insertion hole.

Tactile Verifier. Unlike visual progress, tactile success depends on fine-grained, often visually ambiguous interactions, such as stable grasping versus incipient slip. Without an off-the-shelf tactile reward model, we therefore design a tactile verifier that is *discriminative* over subtle contact outcomes, *efficient* for inference-time evaluation, and *flexible* to phase-specific textual contact goals.

To this end, we use a pretrained tactile encoder [21] with a semantically-aligned latent space and define the tactile reward as $R^\tau(\hat{z}_{t+h}^\tau, \ell_p^\tau) = \cos(\hat{z}_{t+h}^\tau, \mathcal{E}_\psi^\ell(\ell_p^\tau))$ where $\mathcal{E}_\psi^\ell(\cdot)$ is the corresponding text encoder. Computing rewards directly in this latent space makes the verifier discriminative over subtle contact states, efficient for steering without tactile image decoding, and flexible to language-specified contact goals such as “grasp lightly” or “grasp heavily.” To our knowledge, this is the first general language-conditioned tactile reward formulation for flexible contact-aware policy steering.

4.4 Inference-time Optimization for Bi-level Visuo-Tactile Steering Objective

At inference time, we solve the bi-level steering objective in Sec. 4.1 by combining sampling-based visual mode selection with tactile-guided diffusion editing. The details are in Algo. 1 in App. B.3.

Sampling & Verification as Visual Steering. As shown in the outer optimization in Eq. 1, the visual stage selects a global behavior mode by evaluating long-horizon candidate action sequences. We sample $\mathbb{A}_N = \{\mathbf{a}_{t:t+H}^{(i)} \mid \mathbf{a}_{t:t+H}^{(i)} \sim \pi_\theta(\cdot \mid o_t)\}_{i=1}^N$ by recurrently using world model imagination as policy inputs, and choose the sequence with the highest visual reward: $\bar{\mathbf{a}}_{t:t+H} = \arg \max_{\mathbf{a}_{t:t+H} \in \mathbb{A}_N} R^v(\hat{z}_{t:t+H}^v; \ell_p^v)$. We use sampling-based verification for the visual stage because R^v is computed by a large VLM over predicted visual futures. Although effective for assessing task progress, direct classifier-based guidance would require costly and potentially unstable gradients through the VLM, latent world model, and recursive policy generation. Thus, we use the visual verifier to rank candidate futures rather than guide denoising directly. The selected sequence $\bar{a}_{t:t+H}$ serves as the visual anchor, and its first h steps are passed to tactile refinement.

Diffusion Editing as Tactile Steering. Conditioned on the visual anchor $\bar{\mathbf{a}}_t := \bar{\mathbf{a}}_{t:t+h}$, the goal of tactile steering is to locally refine the action chunk to satisfy contact requirement while preserving the action mode selected by the visual steering. We evaluate tactile refinement using the short-horizon tactile reward in Sec. 4.3. The short horizon h is important because tactile signal is most informative for immediate contact correction, such as adjusting force or stabilizing a grasp.

Following Eq. 1 and prior diffusion editing method [11], we define a reference-conditioned prior around visual anchor $\bar{\mathbf{a}}_t$: $p_\theta(\mathbf{a}_t | \bar{\mathbf{a}}_t, o_t) := \int \rho_\theta(\mathbf{a}_t | \mathbf{x}_K, o_t) \nu_K(\mathbf{x}_K | \bar{\mathbf{a}}_t) d\mathbf{x}_K$, where ν_K partially noises $\bar{\mathbf{a}}_t$ to diffusion level K and ρ_θ denotes it under the base policy. This prior captures plausible local edits of the visual anchor, with smaller K enforcing closer refinements and larger K allowing larger deviation. We then define the tactile-refined policy p^{refine} by reweighting the policy distribution with predicted tactile reward, $\log p^{\text{refine}} = \log p_\theta(\mathbf{a}_t | \bar{\mathbf{a}}_t, o_t) + \beta R^\tau(\hat{z}_{t+h}^\tau, \ell_p^\tau)$, and approximate sampling from p^{refine} via tactile-guided reverse diffusion: we partially noise $\bar{\mathbf{a}}_t$ and bias denoising toward actions with higher tactile reward. Thus, the visual stage selects *what* global behavior to execute, while the tactile stage refines *how* to execute it with touch. Details are in App. B.3- B.4.

5 Experiments

We evaluate **ViTaL** on three real-world contact-rich manipulation tasks. Our experiments are designed to answer three questions: whether multimodal visuo-tactile steering improves policy over unimodal guidance (Sec. 5.1); whether learned latent world model and semantically aligned verifiers provide reliable rewards for predicted outcomes (Sec. 5.2); and whether the proposed bi-level optimization balances performance and efficiency compared to naive visual-tactile fusion (Sec. 5.3).

Robot Setup, Task and Data Collections. We use a Franka Emika robot with RGB and tactile sensors to evaluate **ViTaL** on three real-world contact-rich manipulation tasks, including *wiping*, *insertion*, and *pipette transfer*. These tasks require both global visual reasoning, such as selecting the correct mark, hole, or target cup, and local tactile feedback for maintaining contact, resolving alignment, or stabilizing grasp and dispensing behavior. For each task, we collect 50 expert demonstrations to train the base diffusion policy π_θ and with additional 250 policy rollouts from intermediate policy checkpoints to train the visuo-tactile world model. Details are in App A.

Base Policy. We train a diffusion policy that predicts a 16-step action chunk and executes the first $h = 8$ actions at each step. For long-horizon lookahead $H = 16$, we recursively roll out the visuo-tactile world model and condition the policy on predicted future observations. Our main experiments use a vision-conditioned policy $\pi_\theta(a | o_t^v)$; App C.2 evaluates a visuo-tactile policy $\pi_\theta(a | o_t^v, o_t^\tau)$.

World Model and Verifiers. We train separate visual and tactile decoders, and a transformer-based latent dynamics model for visuo-tactile prediction. For steering, we modify ROBOMETER [50] with KV caching for online visual reward estimation, and compute tactile rewards by comparing pretrained tactile embeddings [21] with CLIP text embeddings [51]. Details are in App B.2 - B.4.

5.1 Multi-Modal Visuo-Tactile Steering Outperforms Single-Modality Steering

We compare our multi-modal guidance **ViTaL** against five methods in Fig. 2: 1) **Base Policy**; 2) visual steering, including **Visual Lookahead (8 steps) vs (16 steps)**, which selects the best of $N = 10$ with visual verifier over different prediction horizons; 3) tactile steering, including **Tactile Sampling**, which selects the best of $N = 10$ samples with tactile verifier; **Tactile Guidance**, classifier-based tactile guidance; We omit classifier-based visual guidance due to the reasons discussed in Sec. 4.4. We report **Overall**, **Visual**, and **Contact** success, measuring full task completion, correct visual goal following, and appropriate local physical interaction, respectively.

Results. In Fig. 2, visual-only steering improves visual success but provides limited gains in overall task completion, while tactile-only steering improves contact behavior without reliably satisfying the task goal. In contrast, **ViTaL** jointly improves visual and contact success, leading to a 51% increase in overall task success over base policy and at least 33% gain over unimodal baselines.

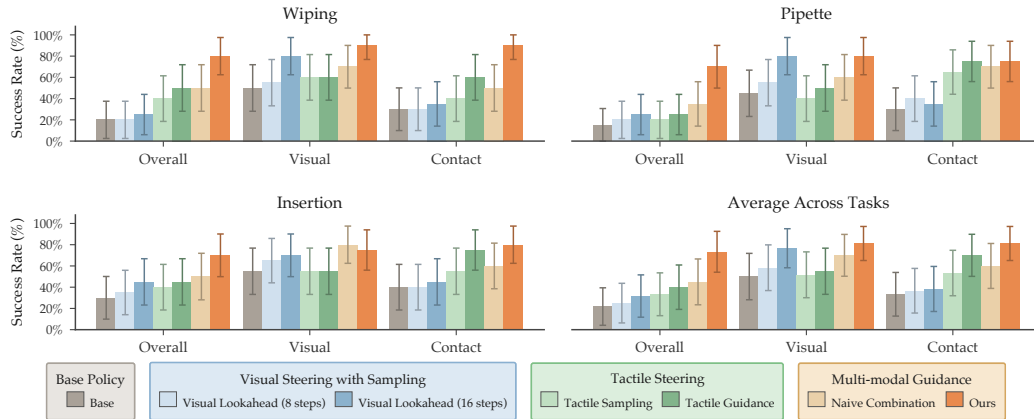


Figure 2: **Policy Steering Performance.** ViTaL outperforms vision-only, tactile-only, and naive multimodal baselines across three tasks. Error bars show binomial standard errors over 20 trials.

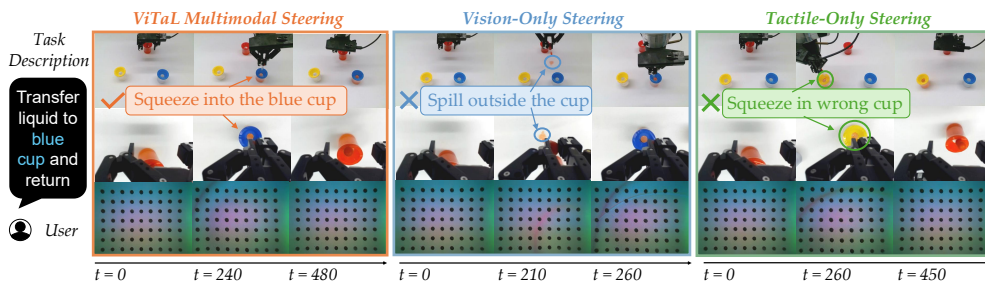


Figure 3: **Policy Steering with Different Guidance Modalities.** Vision-only Steering (middle) fail due to local contact errors, while tactile-only steering (right) can fail due to global task misalignment. ViTaL (left) with multimodal steering addresses both failure modes.

Fig. 3 illustrates this multimodal synergy on the pipette task: vision-only steering can choose globally plausible but imprecise actions (e.g. spilling), while tactile-only steering can improve local contact with adequate grasp force yet miss the intended visual goal. Our multimodal method succeeds by satisfying both constraints. More qualitative results for other tasks are provided in App D.1.

5.2 Semantically-Aligned Visual and Tactile Verifiers Enable Global and Local Adaptation

We evaluate whether the visual and tactile verifiers provide reliable semantic rewards for steering predicted futures. Effective steering requires both accurate visuo-tactile prediction and rewards that remain meaningful on predicted observations. We first compare our joint visuo-tactile world model against separately trained unimodal ones with standard visual metrics and tactile-related metrics; world models' results are in App C.1. Fig. 4 visualizes visual and tactile predictions in pipette task.

Metrics. We further evaluate reward quality on both ground-truth observations (**GT**) and world-model predictions (**Pred**) with preference-order accuracy [52] shown in App. D.2. For each task, we select rollout pairs with different human preferences, rank each pair by the sum of predicted reward divided by rollout length, and measure how often this predicted order matches the human order.

Results. Table 1 shows both adapted ROBOMETER visual reward and our tactile reward achieve at least 70% preference-order accuracy across tasks on both **GT** and **Pred** observations. In some cases, **Pred** rewards are slightly more accurate, likely because predicted latents smooth observation noise (Fig. 4 right) and are easier to rank due to lower visual/tactile variations than real rollouts.

Fig. 4 further illustrates the semantic alignment of both verifiers on the pipette task. The visual verifier assigns higher reward when the robot dispenses into the cup that matches the target cup specified in textual description (e.g. yellow cup), and phase-level textual objective focus on the alignment to the relevant subgoal (e.g. transfer to the target cup or move back to the red cup). The tactile verifier captures the transition from gentle to firm grasp, consistent with marker-tracking force

Table 1: **Reward Accuracy.** We evaluate reward with preference-order accuracy. GT versus Pred uses ground-truth versus predicted futures. Error bars show binomial standard errors over 40 trials.

Src.	Visual Reward (%)				Tactile Reward (%)			
	Wiping	Insertion	Pipette	Avg.	Wiping	Insertion	Pipette	Avg.
GT	80.0±6.3	70.0±7.2	100.0±0.0	83.3±5.9	85.0±5.6	70.0±7.2	80.0±6.3	78.3±6.5
Pred	82.5±6.0	72.5±7.1	100.0±0.0	85.0±5.6	90.0±4.7	77.5±6.6	77.5±6.6	81.7±6.1

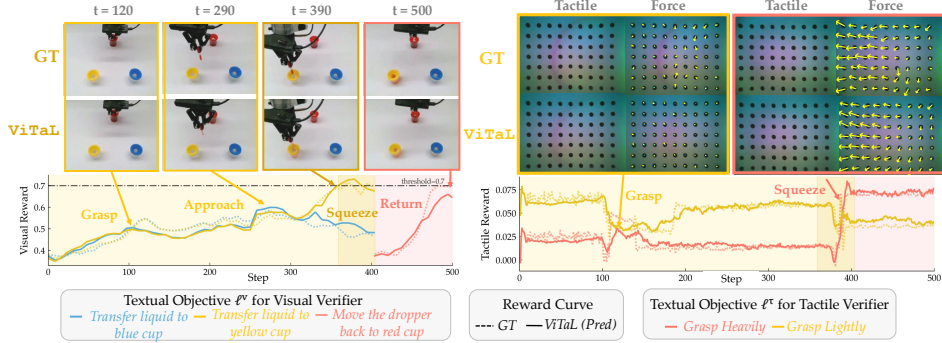


Figure 4: **Visual and Tactile Verification.** Our visual verifier (left) identifies whether the target cup is selected and the phase transition, while the tactile verifier (right) captures changes in grasp force.

estimates. Together, these results show that visual rewards support global goal adaptation, while our proposed tactile rewards provide local contact-sensitive adjustment.

5.3 Bi-Level Steering Achieves Balance Between Performance and Inference Speed

We evaluate whether **ViTaL** achieves a better trade-off between task success and inference speed than naive multimodal fusion. We compare **Base Policy**; **Naive Combination**, which normalizes visual and tactile rewards separately across $N = 10$ samples and sums them for action selection; and **ViTaL**, which first selects actions using long-horizon visual lookahead and then refines it with local tactile guidance. We report **Overall**, **Visual**, and **Contact** success as in Sec. 5.1, together with inference-time overhead, to assess whether **ViTaL** improves steering without sacrificing efficiency.

Results. As shown in Fig. 2, **ViTaL** outperforms **Naive Combination** by at least 20% across all three tasks. Naive reward fusion often suffers from reward imbalance: the visual objective can dominate selection, leading to actions that appear globally plausible but fail to establish proper contact. Fig. 5 illustrates these failures, where naive fusion can leave orange marks partially wiped, misalign the peg with the hole, or spill liquid. In contrast, **ViTaL** uses vision to efficiently select globally appropriate behaviors and touch to make targeted local corrections. As detailed in App. D.3, **ViTaL** adds 0.05 second per policy inference step (8 actions) over naive fusion while yielding substantially higher success. These results show that bi-level steering provides a stronger success–speed trade-off.

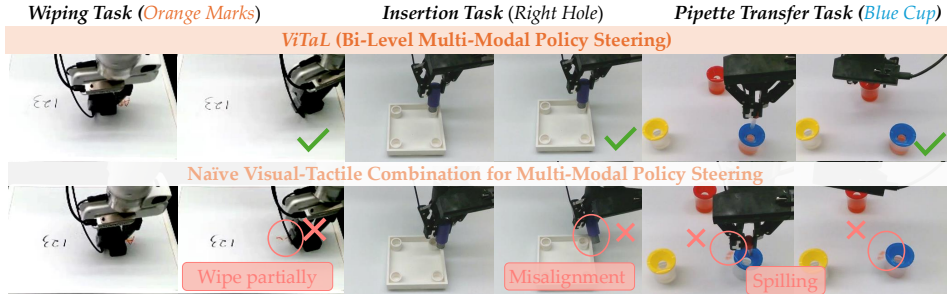


Figure 5: **Multi-Modal Policy Steering.** **ViTaL** (top) successfully complete three contact-rich tasks while naive combination (bottom) of both modalities during steering always fails.

6 Conclusion & Limitation

We presented **ViTaL**, a bi-level visuo-tactile inference-time steering framework for contact-rich manipulation. By combining long-horizon visual mode selection with short-horizon tactile refinement through a latent visuo-tactile world model, **ViTaL** improves performance across three real-world manipulation tasks. Despite the improvement of **ViTaL** over the base policy, several limitations remain. **ViTaL** depends on the fidelity of the latent world model, where compounding prediction errors can affect verification, especially for subtle contact events. Its tactile verifier is also limited by tactile encoders pretrained on much smaller datasets than modern VLMs, suggesting that larger-scale tactile pretraining could enable richer contact reasoning. Furthermore, future work can also extend multimodal policy steering to more complex embodiments, such as dexterous hands.

7 Acknowledgment

The authors were partially supported by the National Science Foundation (NSF) award [#2246447], NSF CAREER award [#2441014], and Samsung Research through the LEAP-U program. The views expressed are those of the authors and do not necessarily reflect those of NSF or Samsung.

References

- [1] Y. Wu, R. Tian, G. Swamy, and A. Bajcsy. From foresight to forethought: Vlm-in-the-loop policy steering via latent alignment. *Robotics: Science and Systems*, 2025.
- [2] M. Du and S. Song. Dynaguide: Steering diffusion policies with active dynamic guidance. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS)*, 2025.
- [3] Z. Sun and S. Song. Latent policy barrier: Learning robust visuomotor policies by staying in-distribution. *Advances in Neural Information Processing Systems*, 38:174280–174305, 2026.
- [4] J. Kwok, C. Agia, R. Sinha, M. Foutter, S. Li, I. Stoica, A. Mirhoseini, and M. Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models. *arXiv preprint arXiv:2506.17811*, 2025.
- [5] Y. Wang, L. Wang, Y. Du, B. Sundaralingam, X. Yang, Y.-W. Chao, C. Perez-D’Arpino, D. Fox, and J. Shah. Inference-time policy steering through human interactions. *arXiv preprint arXiv:2411.16627*, 2024.
- [6] H. Qi, H. Yin, A. Zhu, Y. Du, and H. Yang. Inference-time enhancement of generative robot policies via predictive world modeling. *IEEE Robotics and Automation Letters*, 11(5):5534–5541, 2026. doi:10.1109/LRA.2026.3673995.
- [7] A. Wagenmaker, M. Nakamoto, Y. Zhang, S. Park, W. Yagoub, A. Nagabandi, A. Gupta, and S. Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.
- [8] J. Yuan, Y. Wu, and A. Bajcsy. When to act, ask, or learn: Uncertainty-aware policy steering. *Robotic Science and System (RSS) 2026*, 2026.
- [9] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.
- [10] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [11] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.
- [12] Z. Zhang, J. Ma, X. Yang, X. Wen, Y. Zhang, B. Li, Y. Qin, J. Liu, C. Zhao, L. Kang, et al. Touchguide: Inference-time steering of visuomotor policies via touch guidance. *arXiv preprint arXiv:2601.20239*, 2026.
- [13] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.
- [14] C. Higuera, S. Arnaud, B. Boots, M. Mukadam, F. R. Hogan, and F. Meier. Visuo-tactile world models. *arXiv preprint arXiv:2602.06001*, 2026.
- [15] L. Cui, Z. Zhao, S. Xie, W. Zhang, Z. Han, and Y. Zhu. Vi-TacMan: Articulated object manipulation via vision and touch. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.
- [16] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine. Manipulation by feel: Touch-based control with deep predictive models. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 818–824. IEEE, 2019.

- [17] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors (Switzerland)*, 17(12):2762, 2017.
- [18] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 304–311. IEEE, 2015.
- [19] E. Helmut, L. Dziarski, N. Funk, B. Belousov, and J. Peters. Learning force distribution estimation for the gelsight mini optical tactile sensor based on finite element analysis. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8553–8560. IEEE, 2025.
- [20] R. Feng, J. Hu, W. Xia, A. Shen, Y. Sun, B. Fang, D. Hu, et al. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [21] R. Feng, Y. Zhou, S. Mei, D. Zhou, P. Wang, S. Cui, B. Fang, G. Yao, and D. Hu. Anytouch 2: General optical tactile representation learning for dynamic tactile perception. In *The Fourteenth International Conference on Learning Representations*, 2026.
- [22] C. Higuera, A. Sharma, C. K. Bodduluri, T. Fan, P. Lancaster, M. Kalakrishnan, M. Kaess, B. Boots, M. Lambeta, T. Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *8th Annual Conference on Robot Learning*, 2024.
- [23] C. Higuera, A. Sharma, T. Fan, C. K. Bodduluri, B. Boots, M. Kaess, M. Lambeta, T. Wu, Z. Liu, F. R. Hogan, and M. Mukadam. Tactile beyond pixels: Multisensory touch representations for robot manipulation. In *9th Annual Conference on Robot Learning*, 2025. URL <https://openreview.net/forum?id=sMs4pJYhWi>.
- [24] Y. Xie, M. Li, S. Li, X. Li, G. Chen, F. Ma, F. Yu, and W. Ding. Universal visuo-tactile video understanding for embodied interaction. *Advances in Neural Information Processing Systems*, 38:127864–127883, 2026.
- [25] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 314–323. PMLR, 13–15 Nov 2017. URL <https://proceedings.mlr.press/v78/calandra17a.html>.
- [26] 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*, 3(4):3300–3307, 2018.
- [27] 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 International conference on robotics and automation (ICRA)*, pages 8943–8950. IEEE, 2019.
- [28] Y. Chen, M. Van der Merwe, A. Sipos, and N. Fazeli. Visuo-tactile transformers for manipulation. In *Conference on Robot Learning*, pages 2026–2040. PMLR, 2023.
- [29] X. Zhu, B. Huang, and Y. Li. Touch in the wild: Learning fine-grained manipulation with a portable visuo-tactile gripper. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=WabVVQKTUF>.
- [30] B. Huang, Y. Wang, X. Yang, Y. Luo, and Y. Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing. In *Conference on Robot Learning*, pages 2557–2578. PMLR, 2025.

- [31] L. Heng, H. Geng, K. Zhang, P. Abbeel, and J. Malik. Vitacformer: Learning cross-modal representation for visuo-tactile dexterous manipulation, 2025. URL <https://arxiv.org/abs/2506.15953>.
- [32] H. Chen, J. Xu, H. Chen, K. Hong, B. Huang, C. Liu, J. Mao, Y. Li, Y. Du, and K. Driggs-Campbell. Multi-modal manipulation via multi-modal policy consensus. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. to appear, arXiv:2509.23468.
- [33] R. S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.
- [34] 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, 2011.
- [35] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.
- [36] G. Zhou, H. Pan, Y. Lecun, and L. Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, pages 79115–79135. PMLR, 2025.
- [37] Y. Huang. Vjepa: Variational joint embedding predictive architectures as probabilistic world models. *arXiv preprint arXiv:2601.14354*, 2026.
- [38] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.
- [39] Y. Guo, L. X. Shi, J. Chen, and C. Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.
- [40] H. A. Alhajja, J. Alvarez, M. Bala, T. Cai, T. Cao, L. Cha, J. Chen, M. Chen, F. Ferroni, S. Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.
- [41] S. Gao, W. Liang, K. Zheng, A. Malik, S. Ye, S. Yu, W.-C. Tseng, Y. Dong, K. Mo, C.-H. Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.
- [42] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.
- [43] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *Nature*, 2023.
- [44] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.
- [45] D. Hafner, W. Yan, and T. Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.
- [46] Y. Zheng, S. Gu, W. Li, Y. Zheng, Y. Zang, S. Tian, X. Li, C. Hao, C. Gao, S. Liu, et al. Omnivta: Visuo-tactile world modeling for contact-rich robotic manipulation. *arXiv preprint arXiv:2603.19201*, 2026.
- [47] H. Yuan, W. Yi, Z. Zhang, W. Chen, Y. Mo, J. Yin, X. Li, X. Zeng, C. Wen, C. Lu, et al. Vtam: Video-tactile-action models for complex physical interaction beyond vlas. *arXiv preprint arXiv:2603.23481*, 2026.

- [48] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.
- [49] G. Zhou, H. Pan, Y. LeCun, and L. Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025.
- [50] A. Liang, Y. Korkmaz, J. Zhang, M. Hwang, A. Anwar, S. Kaushik, A. Shah, A. S. Huang, L. Zettlemoyer, D. Fox, Y. Xiang, A. Li, A. Bobu, A. Gupta, S. Tu, E. Biyik, and J. Zhang. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. In *Robotics: Science and Systems 2026*, 2026.
- [51] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [52] R. Tian, Y. Wu, and A. Bacjysy. Position: Good embodied reward models need bad behavior data, 2026.
- [53] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.
- [54] B. Zheng, N. Ma, S. Tong, and S. Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

Appendix

A Real Robot Experiment Setup	15
A.1 Wiping Task	15
A.2 Insertion Task	15
A.3 Pipette Task	16
B Implementation Details	17
B.1 Base Policy	17
B.2 Visuo-Tactile World Model	17
B.3 Visual Steering as Sampling & Verification	21
B.4 Tactile Steering as Diffusion Editing	22
C Ablations	23
C.1 Visuo-Tactile World Model	23
C.2 Multi-Modal Guidance for Visuo-Tactile Policies	25
D Additional Results	27
D.1 Multi-modal Visuo-Tactile Steering Outperforms Steering with Single Modality . .	27
D.2 Semantically-Aligned Vision and Tactile Verifier Enables Global and Local Adaptation	29
D.3 Bi-level Optimization Balances Success and Steering Speed Better	33

A Real Robot Experiment Setup

A.1 Wiping Task

Setup. We use a 7-DoF Franka Emika robot equipped with 3D-printed soft compliant fingers attached to the end-effector, following [9]. We design a custom mount on one fingertip to attach a GelSight Mini tactile sensor [17]; the mount design is available on the project website. We use two RealSense D435 cameras: one mounted on the end-effector as a wrist camera and one placed in front of the robot as an external camera, as shown in Fig. 6. The action space of the robot is 8 dimension including the delta cartesian pose of the end-effector (3-dimensional position and 4-dimensional quaternion) and 1-dimensional binary gripper control. The low-level controller is implemented using Deoxys: https://github.com/UT-Austin-RPL/deoxys_control with OSC POSE control mode.

The wiping task requires the robot to grasp a black eraser and wipe marked regions on a whiteboard. The board contains black “123” marks on the left side and orange “ABC” marks on the right side. We evaluate two language instructions: “wipe the orange marks” and “wipe the black marks.”

Base Policy. We collect 25 demonstrations for wiping the orange marks and 25 demonstrations for wiping the black marks using the Meta Oculus Quest 3 as the teleoperation device. The demonstrations have mixed quality: some fully erase the marks, while others only partially erase them. For the visual policy, at each timestep, the policy receives two RGB observations, corresponding to the wrist and external views, together with the robot proprioceptive state, and predicts a 16-step action sequence. For the visuo-tactile policy, the policy receives an additional tactile image as input. All images are resized to 256×256 . During execution, we apply only the first 8 steps of the predicted action sequence.

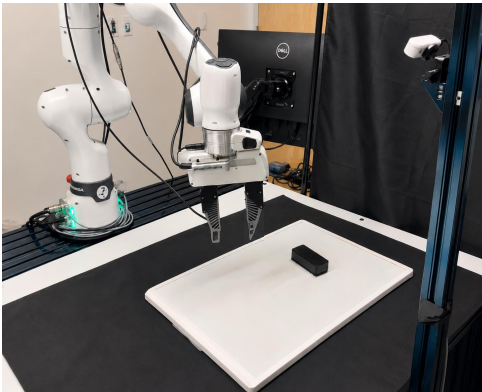


Figure 6: Wiping Task Robot Setup

Visuo-Tactile World Model. We collect 280 policy rollouts using different policy checkpoints. Together with the 50 demonstrations, this yields 330 trajectories, which we split into a training set of 300 trajectories and a validation set of 30 trajectories.

Visual & Tactile Verifiers. For the wiping task, the task goal is straightforward and consists of a single phase. We use either “wipe the orange marks” or “wipe the black marks” as the task description ℓ_1^v for the visual verifier, and we use “wipe the board” as the language input ℓ_1^t for the tactile verifier.

Evaluation. During training and evaluation, we slightly randomize the eraser position within the region directly under the gripper. For each new trial, we redraw the orange and black marks and randomize their positions within the right and left regions of the board, respectively. To evaluate success rate, we run 10 trials with the instruction “wipe the orange marks” and 10 trials with the instruction “wipe the black marks.” For each trial, we count visual success if the robot moves to the correct region of the board. We count contact success if the robot completely wipes the black or the orange marks. A trial is counted as an overall task success only when both visual and contact success are achieved.

A.2 Insertion Task

Setup. The insertion task uses the same overall hardware and data collection pipeline as the wiping task in Appendix A.1, with several task-specific modifications. We use a 7-DoF Franka Emika robot equipped with a Robotiq 2F-85 gripper. A GelSight Mini tactile sensor [17] is mounted to one of

the fingertip of the Robotiq gripper and the custom mount is also available on the website. For visual observations, we use a ZED Mini camera mounted on the end-effector as the wrist camera, together with an external RealSense D435 camera placed in front of the robot as shown in Fig. 7. We use a 8 dimension action to control the robot similar to the wiping task but the gripper control is a continuous value from $[0, 1]$ instead of the binary control as wiping task.

The task involves inserting a purple cylindrical peg into a white tabletop fixture that is fixed to the table. The fixture contains four holes, one at each corner. We consider two target goals: inserting the peg into the top-left hole or the top-right hole. The clearance between the peg and each hole is approximately 1 mm, requiring precise contact-rich manipulation.

Base Policy. We collect 50 demonstrations using the same teleoperation device, with 25 demonstrations targeting the top-left hole and 25 targeting the top-right hole. As in the wiping task, the demonstrations have mixed quality, including both successful insertions and partial or imperfect attempts. The policy observations, action prediction horizon, and execution strategy follow the wiping-task setup in Appendix A.1, except that the wrist RGB observation is captured using the ZED Mini camera and the tactile observation comes from the GelSight Mini mounted on the Robotiq gripper.

Visuo-Tactile World Model. We collect the world model data using the same procedure as in Appendix A.1. Policy rollouts collected from different checkpoints are combined with the demonstration data to form the training and validation sets for learning future latent imaginations.

Visual & Tactile Verifiers. The insertion task consists of a single phase. For the visual verifier, we use the task description ℓ_1^v corresponding to the commanded target hole: either “the robot inserts the purple peg into the top left corner of the tabletop” or “the robot inserts the purple peg into the top right corner of the tabletop.” For the tactile verifier, we use “insert the peg” as the language input ℓ_1^t .

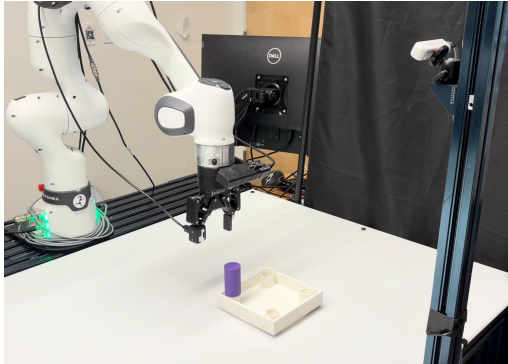


Figure 7: Insertion Task Robot Setup

Evaluation. For training and evaluation, we randomize the peg position within a $15\text{cm} \times 15\text{cm}$ region between the robot base and the gripper for each trial. During evaluation, we choose 10 trials with the left goal and 10 trials with the right goal and assess whether the robot selects the correct target side and whether it completes the insertion. We count visual success if the robot moves toward the correct side of the tabletop, corresponding to the commanded top-left or top-right target. We count contact success if the peg is inserted into one of the holes. A trial is counted as an overall task success only when both visual and contact success are achieved.

A.3 Pipette Task

Setup. The pipette transfer task uses the same hardware setup as the insertion task in Appendix A.2, including the Franka Emika robot, Robotiq 2F-85 gripper, GelSight Mini tactile sensor with the task-specific custom mount, ZED Mini wrist camera, and external RealSense D435 camera. The action space is the same as the insertion task.

Base Policy. We use the same policy architecture and demonstration collection procedure as in Appendix A.1. The policy takes wrist and external RGB observations, proprioceptive state, and, for the visuo-tactile policy, an additional GelSight tactile image as input.

Visuo-Tactile World Model. We collect policy rollouts for the visuo-tactile world model following the same procedure as the insertion task in Appendix A.2. Demonstrations and rollouts from

different policy checkpoints are combined to train the model to generate future latent imaginations conditioned on memory, history, and candidate action sequences.

Visual & Tactile Verifiers. Unlike the wiping and insertion tasks, the pipette transfer task consists of two phases. The high-level task description is either “transfer the liquid to the yellow cup and return” or “transfer the liquid to the blue cup and return.” In the first phase, the visual verifier uses the goal ℓ_1^v corresponding to the commanded target cup: either “transfer the liquid to the yellow cup” or “transfer the liquid to the blue cup.” The tactile verifier initially uses “grasp lightly” as the tactile language input ℓ_1^t . Once the visual reward exceeds 0.7, we switch the tactile description to “grasp heavily” and maintain this tactile goal for 40 steps. After this transition period, we switch the visual verifier to the second-phase goal, “put the dropper back to the red cup.”

Evaluation. During both training and evaluation, we randomize the red cup position and the liquid volume in the dropper. We decompose success into visual and contact components. Visual success evaluates whether the robot first moves toward the correct target cup specified by the language instruction and then returns the dropper to the red cup. Contact success evaluates whether the robot transfers liquid into one of the cups without spilling. A trial is counted as an overall task success only when both visual and contact success are achieved.

B Implementation Details

B.1 Base Policy

Architecture Overview. The base policy is a *Diffusion UNet* backbone [9] that conditions on multi-camera RGB images and robot proprioception to produce a chunk of future actions via DDIM denoising. We use the same Conditional 1D UNet from prior work [9]. Details of the architecture are shown in Table. 2.

Observation Encoders. Each camera image (224×224) is processed independently by a ResNet-18 [53] backbone (ImageNet pre-trained) with all batch normalization layers replaced by Group-Norm (groups = $C/16$, following the Diffusion Policy convention), with the global-average-pool and classification head removed. The resulting $512 \times 7 \times 7$ feature map is passed through a *Spatial Softmax* pooling layer (64 keypoints, temperature 1.0) and flattened to $64 \times 2 = 128$ values, which are then projected to a 64-dimensional embedding by a linear layer. For visual policy, two cameras (wrist, front-view) are each encoded by a *separate* copy of this network, so the per-timestep visual embedding is $64 \times 2 = 128$ dim. For visuo-tactile policy, three cameras (wrist, front-view, tactile) are encoded so the per-timestep visual embedding is $64 \times 3 = 192$ dim.

Observation encoding. Two consecutive image frames per camera (`img_chunk= 2`) are stacked, giving a total visual embedding of $64 \times 3 \times 2 = 384$ dim. This is concatenated with the 8-dimensional proprioceptive state (end-effector 3D position + quaternion + gripper width), yielding a $384 + 8 = 392$ dim global conditioning vector s_t . No shared MLP is applied; a dropout layer (rate 0.1) is the only non-linearity.

Training. Images are augmented on-GPU with random-crop shifts (ImageNet normalization applied after). diffusion timestep $k \sim \text{Uniform}[0, 100)$ is sampled, the action chunk is noised to k steps and the U-Net predicts the added noise. The details of the training parameters are in Table. 3.

B.2 Visuo-Tactile World Model

Observation Encoders. We use frozen pre-trained encoders to extract patch-level embeddings from each observation stream. RGB images from the wrist camera and front camera are encoded using DINOv3 [48] with a ViT-B/16 backbone, producing a 14×14 grid of 196 patch tokens each of dimension 768. Tactile images are encoded with AnyTouch2 [21], a TactileVideoMAE model that processes a sliding window of the last 4 consecutive tactile frames and outputs 512-dimensional patch embeddings; the spatial grid is standardized to 196 patches by truncation or zero-padding. Images are resized to 224×224 before encoding; tactile frames additionally undergo background

Table 2: **Diffusion Policy Architecture Hyperparameters.**

Hyperparameter	Value
<i>Visual encoder</i>	
Backbone	ResNet-18 (ImageNet pre-trained)
Normalization	GroupNorm ($C/16$ groups)
Pooling	Spatial Softmax (64 keypoints)
Per-camera embed dim	64
<i>Observation encoding</i>	
Cameras	3 (wrist, front, tactile)
Image frames per camera (img_chunk)	2
Visual embed dim	384 ($64 \times 3 \times 2$)
Proprioception dim	8
Total conditioning dim	392
<i>Diffusion U-Net</i>	
Action chunk (ac_chunk)	16
Action dim (ac_dim)	7
U-Net channel dims	[256, 512, 1024]
Diffusion step embed dim	256
FiLM conditioning dim	648 (step emb. + obs.)
Kernel size	3
GroupNorm groups	8
Activation	Mish
<i>DDIM noise schedule</i>	
Beta schedule	Squaredcos cap v2
$\beta_{\min} / \beta_{\max}$	$10^{-4} / 0.02$
Training diffusion steps	100
Inference diffusion steps	16 (DDIM)
Prediction type	Epsilon ($\hat{\epsilon}$)
Clip sample	Yes ($[-1, 1]$)

Table 3: **Diffusion Policy Training Hyperparameters.**

Hyperparameter	Value
Optimizer	AdamW
Learning rate	1×10^{-4}
Betas	(0.95, 0.999)
Weight decay	1×10^{-6}
Epsilon	1×10^{-8}
LR schedule	Cosine decay
Warmup steps	500
Training iterations	80,000
Batch size	100
Image augmentation	random crop (shift) + ImageNet norm
Dropout	0.1

subtraction (offset 0.130) and CLIP normalization. All encoder weights are *frozen* during world model training. The specifications of the encoders are in Table. 4.

Latent Dynamics Model Architecture. The world model is a causal patch-level transformer that maps one history frame of multi-modal observations and the corresponding collapsed action to a predicted next-frame embedding for every camera.

Each camera’s patch embeddings are projected from their native dimension to a common dimension of 384 via a linear layer followed by LayerNorm and ReLU. Actions are represented as a block of $F = 8$ consecutive raw actions per frame, each of dimension 7 (6-DoF end-effector delta +

Table 4: **Observation Encoder Specifications.**

Camera	Model	Emb. dim	Patches
wrist-view RGB	DINOv3 ViT-B/16	768	196 (14×14)
front-view RGB	DINOv3 ViT-B/16	768	196 (14×14)
tactile image	AnyTouch2 TactileVideoMAE	512	196 (14×14)

gripper), giving a per-frame action vector of dimension 56. This is encoded by a two-layer MLP to a 64-dimensional action embedding. The end-effector state (3D position, quaternion, gripper; dimension 8) is used directly. The projected camera tokens, action embedding, and state embedding are concatenated along the feature axis for each patch position, yielding tokens of total dimension $D_{\text{total}} = 3 \times 384 + 64 + 8 = 1224$. Learned spatial positional embeddings of the same dimension are added.

The transformer backbone consists of 6 identical blocks, each comprising multi-head self-attention (16 heads, 64 dimensions per head) with a causal mask that prevents cross-frame look-ahead, followed by a two-layer feed-forward network (hidden dimension 2048) with GELU activation and residual connections; dropout of 0.1 is applied throughout. Only the last context frame’s output tokens are passed to the prediction heads. Separate two-layer MLP heads decode the transformer output to each camera’s native embedding dimension; a mean-pooling state head predicts the next robot state. Details of the architecture of the latent dynamics model are in Table 5.

Table 5: **Latent Dynamics Model Architecture Hyperparameters.**

Hyperparameter	Value
Camera projection (common) dim	384
Action encoding dim	64
State dim	8
Frame skip F	8
Action input dim (per slot)	56 ($F \times 7$)
Total token dim D_{total}	1224
Transformer depth	6
Attention heads	16
Dim per head	64
MLP hidden dim	2048
Dropout	0.1
History frames (num_hist)	1
Prediction frames (num_pred)	1

Latent Dynamics Model Training. Actions are normalized per-dimension using dataset mean and standard deviation before collapsing. We use the *multi-step rollout loss*: at each training iteration a rollout horizon $k \sim \text{Uniform}\{3, 4, 5\}$ is sampled; the model is unrolled autoregressively for k steps by feeding each predicted embedding back as the next observation, and MSE loss is accumulated at every step over all cameras and the robot state. To improve robustness to noised actions during inference time for classifier-based guidance, we follow prior work [2] to additionally apply DDIM-schedule noise to the input action tokens: a per-sample diffusion timestep is drawn from $\text{Geometric}(p=0.05)$ and a linearly-ramped masking probability (reaching 0.5 at the final epoch) randomly replaces clean actions with their noised version.

Observations are subsampled at every $F=8$ raw steps following prior work [3], so each subsampled frame represents 8 environment steps. The train/test split holds out 30 trajectories; the remaining trajectories are used for training. Training uses mixed-precision (FP16 AMP) with torch compile (reduce-overhead mode). The details of the training are in Table 6.

Image Decoder Training. We train a convolutional image decoder to reconstruct RGB/tactile images from predicted patch embeddings, used only for visualization and evaluation. The decoder

Table 6: Latent Dynamics Model Training Hyperparameters.

Hyperparameter	Value
Optimizer	AdamW
Transformer LR	5×10^{-5}
Action encoder & positional emb. LR	5×10^{-4}
Projection & prediction head LR	5×10^{-5}
Training iterations	100,000
Batch size	16
Test split trajectories	50
Loss	Multi-step MSE
Rollout horizon k	Uniform{3, 4, 5}
Action noise scheduler	DDIM
Action noise p (geometric)	0.05
DDIM timesteps	100
Frame skip F	8
Temporal sampling	uniform subsampling
Mixed precision	FP16 (AMP)
Random seed	42

is a VQVAE-style network with a stride-4 convolutional decoder: a 3×3 input projection, 4 residual blocks (channel width 384, residual channels 128), followed by two transposed convolutions (stride 2) that upsample the 14×14 patch grid to 56×56 and then 224×224 . Separate decoders are trained for the vision cameras (wrist-view and front-view share one decoder with $d_{\text{emb}}=768$) and the tactile camera uses $d_{\text{emb}}=512$.

Training uses the RAE loss [54]: $\mathcal{L} = \mathcal{L}_{L1} + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}} + \lambda_{\text{GAN}} \mathcal{L}_{\text{GAN}}$. An adaptive GAN weight (gradient norm balancing between reconstruction and adversarial gradients, clamped to $[0.01, 100]$) is applied per camera using per-camera PatchGAN discriminators (ndf=64, 3 layers) with hinge loss; the discriminator is warmed up for 200 iterations before the adversarial loss enters the generator objective. We use RAE loss instead of L1 loss or L2 loss because empirically we find RAE loss is good at reconstructing the fine-grained details such as the marks on the board compared to L1 loss or L2 loss because marks only exist in a few pixels in the front-view image and therefore has very low penalty if the prediction is wrong. Details of the Decoder Architecture and Training are in Table. 7.

Table 7: Image Decoder Architecture and Training Hyperparameters.

Hyperparameter	Value
Architecture	VQVAE decoder
Vision decoder emb. dim	768
Tactile decoder emb. dim	512
Channel width	384
Residual blocks	4
Residual channels	128
Spatial stride	4 (two $\times 2$ upsample stages)
Output resolution	224×224
Loss	RAE (L1 + LPIPS + Adversarial)
λ_{L1}	1.0
λ_{LPIPS}	1.0
λ_{GAN}	0.75 (adaptive per camera)
Discriminator warm-up	600 iters (disc. train) / 800 iters (gen. adv.)
Discriminator	PatchGAN (ndf=64, 3 layers, hinge loss)
Optimizer (decoder)	AdamW, lr= 3×10^{-4}
Optimizer (discriminator)	AdamW, lr= 3×10^{-4} , $\beta = (0.5, 0.9)$
Batch size	64
Training iterations	5,000

Algorithm 1 ViTaL: Bi-level Visuo-Tactile Policy Steering

Require: Policy π_θ , latent world model p_ϕ , encoders $\mathcal{E}_\psi^v, \mathcal{E}_\psi^\tau$, decoders $\mathcal{D}_\phi^v, \mathcal{D}_\phi^\tau$, verifiers R^v, R^τ , instruction \mathcal{L}
Require: Visual samples N , visual horizon H , execution horizon h , edit level K , guidance scale λ

- 1: Decompose \mathcal{L} into phase objectives $\{(\ell_p^v, \ell_p^\tau)\}_{p=1}^P \leftarrow g_{\text{VLM}}(\mathcal{L})$
- 2: Initialize phase $p \leftarrow 1$, number of visual chunks $M \leftarrow H/h$
- 3: **while** task not finished **do**
- 4: Observe $o_t = (o_t^v, o_t^\tau, q_t)$ and encode $z_t = (\mathcal{E}_\psi^v(o_t^v), \mathcal{E}_\psi^\tau(o_t^\tau))$
- 5: **Outer level: batched visual mode selection**
- 6: Initialize N imagined rollouts: $\hat{o}_t^{(i)} \leftarrow o_t, \hat{z}_t^{(i)} \leftarrow z_t, A^{(i)} \leftarrow []$ for all $i \in \{1, \dots, N\}$
- 7: **for** $j = 0, \dots, M - 1$ **do**
- 8: Sample a batch of action chunks: $a_j^{(1:N)} \sim \pi_\theta(\cdot | \hat{o}_{t+jh}^{(1:N)})$
- 9: Roll out each chunk with the latent world model: $\hat{z}_{t+(j+1)h}^{(1:N)} \sim p_\phi(\cdot | \hat{z}_{t+jh}^{(1:N)}, a_j^{(1:N)})$
- 10: Decode the terminal imagined observation for the next policy call:
- 11: $\hat{o}_{t+(j+1)h}^{(i)} \leftarrow (\mathcal{D}_\phi^v(\hat{z}_{t+(j+1)h}^{v,(i)}), \mathcal{D}_\phi^\tau(\hat{z}_{t+(j+1)h}^{\tau,(i)}), \mathcal{D}_\phi^g(\hat{z}_{t+(j+1)h}^{g,(i)}))$
- 12: Append chunk to candidate sequence: $A^{(i)} \leftarrow A^{(i)} \| a_j^{(i)}$ for all i
- 13: **end for**
- 14: Score each candidate using the decoded final visual frame: $s_i^v \leftarrow R^v(\hat{o}_{t+H}^{v,(i)}, \ell_p^v)$ for all i
- 15: Select visual anchor: $\bar{a}_{t:t+H} \leftarrow A^{(i^*)}$, where $i^* = \arg \max_i s_i^v$
- 16: **Inner level: tactile contact refinement by diffusion editing**
- 17: Let $\bar{a} \leftarrow \bar{a}_{t:t+h}$ be the first h actions of the visual anchor
- 18: Partially noise the visual anchor: $x_K \leftarrow \sqrt{\bar{\alpha}_K} \bar{a} + \sqrt{1 - \bar{\alpha}_K} \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$
- 19: **for** $k = K, \dots, 1$ **do**
- 20: Predict diffusion noise: $\epsilon_\theta^k \leftarrow \epsilon_\theta(x_k, k, o_t)$
- 21: Estimate clean action: $\hat{a}_0^k \leftarrow (x_k - \sqrt{1 - \bar{\alpha}_k} \epsilon_\theta^k) / \sqrt{\bar{\alpha}_k}$
- 22: Predict tactile future under the current clean-action estimate: $\hat{z}_{t:t+h}^\tau \sim p_\phi^\tau(\cdot | z_t, \hat{a}_0^k)$
- 23: Compute phase-conditioned tactile reward: $r_k^\tau \leftarrow R^\tau(\hat{z}_{t:t+h}^\tau; \ell_p^\tau)$
- 24: Compute normalized tactile guidance: $g_k \leftarrow \nabla_{\hat{a}_0^k} r_k^\tau / (\|\nabla_{\hat{a}_0^k} r_k^\tau\|_2 + \epsilon)$
- 25: Bias the reverse diffusion direction: $\tilde{\epsilon}_k \leftarrow \epsilon_\theta^k - \lambda \sqrt{1 - \bar{\alpha}_k} g_k$
- 26: DDIM update: $x_{k-1} \leftarrow \sqrt{\bar{\alpha}_{k-1}} \hat{a}_0^k + \sqrt{1 - \bar{\alpha}_{k-1}} \tilde{\epsilon}_k$
- 27: **end for**
- 28: Execute refined action chunk $a_{t:t+h}^* \leftarrow x_0$
- 29: Update phase p when the visual verifier satisfies ℓ_p^v
- 30: **end while**

B.3 Visual Steering as Sampling & Verification

Algo. 1 specifies the bi-level policy steering algorithm. We begin by first talk about the outer level of visual steering.

At each policy query step, we sample N candidate long-horizon action sequences from the base diffusion policy,

$$\mathbb{A}_N = \left\{ a_{t:t+H}^{(i)} \mid a_{t:t+H}^{(i)} \sim \pi_\theta(\cdot | o_t) \right\}_{i=1}^N. \quad (2)$$

The candidates are generated in parallel using DDIM denoising. Each candidate is then evaluated by rolling it out with the latent visuo-tactile world model. Specifically, given the current visual and tactile embeddings $z_t^v = \mathcal{E}_\psi^v(o_t^v)$ and $z_t^\tau = \mathcal{E}_\psi^\tau(o_t^\tau)$, the world model predicts the future visual and tactile latent trajectories:

$$z_{t:t+H}^{v,(i)}, z_{t:t+H}^{\tau,(i)} \sim p_\phi(\cdot, \cdot | z_t^v, z_t^\tau, a_{t:t+H}^{(i)}). \quad (3)$$

The visual reward model then scores each predicted visual future against the phase-specific visual language goal ℓ_p^v . We select the candidate with the highest visual reward:

$$\bar{a}_{t:t+H} = \arg \max_{a_{t:t+H}^{(i)} \in \mathbb{A}_N} R^v(z_{t:t+H}^{v,(i)}; \ell_p^v). \quad (4)$$

This selected sequence defines the visual anchor. Its first h steps, $\bar{a}_{t:t+h}$, are passed to the tactile diffusion editing stage in Appendix B.4.

In implementation, we use recurrent world-model imagination and policy inference to obtain longer-horizon predictions. The policy predicts the first action chunk from the current observations and the world model consumes the current embedding history together with the first action chunk and predicts future latent embeddings. We decoded the future latent embeddings into the observations

and query the policy with the decoded observations to get the next action chunk. We retain the final slot prediction, shift the context window forward by replacing the oldest context frame with the predicted embedding, and query the world model again with the next action chunk. This recurrent rollout produces the long-horizon latent future used by the visual verifier.

For visual scoring, we decode the predicted camera-view latent corresponding to the final imagined step into a 224×224 RGB image using the frozen VQ-VAE decoder. The decoded image is sent to a ROBOMETER [50], which scores each candidate image against the task description using a 4B vision-language model. Since direct gradient-based guidance through the VLM, decoder, recurrent world model, and policy sampling process would be expensive and potentially unstable, we use the VLM only as a verifier to rank sampled futures.

We made the modifications to the original ROBOMETER code for online evaluation of the visual reward. To reduce per-step latency during online evaluation, we introduce some optimization to the visual reward inference. We introduce a sliding-window context eviction policy: rather than accumulating the full observation history in the model’s context, only the most recent K frames are retained, bounding per-step inference latency to a constant independent of trajectory length.

We also support a two-phase task structure. The robot begins in phase 1 with the visual language goal ℓ_1^v , such as “transfer the liquid to the yellow cup.” Once the maximum visual reward across candidates exceeds a threshold, the task switches to phase 2 with a new visual language goal ℓ_2^v , such as “put the dropper back to the red cup.” A new ROBOMETER session is created for the second phase. Details of the hyperparameters used in visual steering is shown in Table 8.

Table 8: **Visual Steering Hyperparameters.**

Hyperparameter	Value
Candidates sampled (N)	10
Action horizon per chunk (h)	8
Number of recurrent chunks	2
Total visual lookahead horizon (H)	16
WM reward camera	front-view camera (RGB)
Visual reward model	ROBOMETER-4B
Phase-switch visual reward threshold	0.70

B.4 Tactile Steering as Diffusion Editing

After the visual stage selects the best long-horizon candidate $\bar{a}_{t:t+H}$ from Eq. 1, we begin the inner level optimization in Algo. 1 and take its first h steps, denoted as the visual anchor $\bar{a}_{t:t+h}$, and refine it using tactile-guided diffusion editing. The goal of this refinement is to improve short-horizon contact quality while preserving the global behavior mode selected by visual steering.

Following Eq. 1 and prior diffusion editing methods [11], we define a reference-conditioned action prior around the visual anchor:

$$p_{\theta}(a_{t:t+h} \mid \bar{a}_{t:t+h}, o_t) := \int \rho_{\theta}(a_{t:t+h} \mid \mathbf{x}_K, o_t) \nu_K(\mathbf{x}_K \mid \bar{a}_{t:t+h}) d\mathbf{x}_K, \quad (5)$$

where ν_K partially noises the visual anchor to diffusion level K , and ρ_{θ} denotes the reverse denoising process induced by the base diffusion policy. The diffusion level K controls the edit strength: smaller K keeps the sample closer to the visual anchor, while larger K permits larger local deviations.

Partial re-noise. We first map the visual anchor $\bar{a}_{t:t+h}$, represented in the policy’s normalized $[-1, 1]$ action space, to a mildly noised action latent at the K -th-from-last DDIM timestep K :

$$\mathbf{x}_K = \sqrt{\bar{\alpha}_K} \bar{a}_{t:t+h} + \sqrt{1 - \bar{\alpha}_K} \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \quad (6)$$

This step initializes the reverse process near the visually selected action mode rather than sampling from pure noise.

Tactile-guided denoising. At each of the K guided DDIM steps, we reconstruct the clean action estimate from the current noisy action \mathbf{x}_K using the stop-gradient U-Net noise prediction $\hat{\varepsilon}_\theta$:

$$\hat{a}_0^{(K)} = \frac{\mathbf{x}_K - \sqrt{1 - \bar{\alpha}_K} \hat{\varepsilon}_\theta(\mathbf{x}_K, K, o_t)}{\sqrt{\bar{\alpha}_K}}. \quad (7)$$

We then pass the estimated clean action chunk through the tactile branch of the latent world model, conditioned on the current visual and tactile embeddings, to predict the short-horizon tactile future:

$$\tilde{z}_{t:t+h}^\tau \sim p_\phi \left(\cdot \mid z_t^v, z_t^\tau, \hat{a}_0^{(n)} \right), \quad z_t^v = \mathcal{E}_\psi^v(o_t^v), \quad z_t^\tau = \mathcal{E}_\psi^\tau(o_t^\tau). \quad (8)$$

The predicted tactile sequence is scored by the tactile reward model using the phase-specific tactile language input ℓ_p^τ :

$$r^\tau = R^\tau(\tilde{z}_{t:t+h}^\tau; \ell_p^\tau). \quad (9)$$

Equivalently, this implements the refinement objective

$$\log p^{\text{refine}}(a_{t:t+h}) = \log p_\theta(a_{t:t+h} \mid \bar{a}_{t:t+h}, o_t) + \beta R^\tau(\tilde{z}_{t:t+h}^\tau; \ell_p^\tau), \quad (10)$$

where the first term keeps the edited action close to the visual anchor and the second term biases the sample toward higher predicted tactile reward.

To approximately sample from this refined distribution, we compute the gradient of the tactile reward with respect to the clean action estimate, $\nabla_{\hat{a}_0^{(K)}} r^\tau$, normalize it to unit norm, and use it to modify the diffusion noise prediction:

$$\hat{\varepsilon}_{\text{guided}} = \hat{\varepsilon}_\theta - \lambda \sqrt{1 - \bar{\alpha}_K} \frac{\nabla_{\hat{a}_0^{(K)}} r^\tau}{\left\| \nabla_{\hat{a}_0^{(K)}} r^\tau \right\|}. \quad (11)$$

The scale factor $\sqrt{1 - \bar{\alpha}_n}$ follows LPB guidance [3] and bounds the guidance magnitude across noise levels, avoiding the high-noise gradient explosion that can arise in DynaGuide-style guidance [2].

We take one DDIM step using $\hat{\varepsilon}_{\text{guided}}$ and repeat Eqs. (7)–(11) for all K reverse steps. The final denoised action $a_{t:t+h}^*$ is unnormalized and executed on the robot. In this way, the visual stage determines *what* global behavior mode to execute, while tactile diffusion editing refines *how* to execute the first action chunk with better contact. The detailed hyperparameters used in this process are shown in Table 9.

Table 9: **Tactile Guidance Hyperparameters.**

Hyperparameter	Value
Re-noise / guided denoising steps (K)	4
Stochastic-sampling repetitions per step (ss)	1
Guidance scale (λ)	10
Gradient normalization	Unit norm (FGSM)
Gradient scale	$\sqrt{1 - \bar{\alpha}_t}$

C Ablations

C.1 Visuo-Tactile World Model

Methods. We compare three world model variants in Table 10, 11, 12: a tactile-only model, an vision-only model, and an visuo-tactile model. All unimodal baselines are trained with the same multi-step prediction loss as **ViTaL**. The multi-step loss supervises not only one-step dynamics prediction, but also autoregressive predictions over several future steps, encouraging the model to remain accurate under rollout. This is crucial for visual prediction since we need to predict longer horizon beyond a single action chunk.

For the quantitative and qualitative evaluations in this section, predicted videos from the visuo-tactile world model are generated with sliding-window prediction rather than full-trajectory autoregressive rollout. At each time step, we condition the model on the current visual observation, the previous four tactile observations, and the next eight actions to predict the observation at the eighth future frame. We then advance one step and repeat the procedure with the next observation-action window. This protocol matches how the world model is used in **ViTaL**: after each steered eight-step action chunk is executed, the model receives a new real observation and predicts the next future outcome from that updated state. Thus, we evaluate the model under the same receding-horizon setting used for steering, rather than under a fully autoregressive full-trajectory generation setting.

Visual Prediction Metrics. . We evaluate predicted camera observations using six complementary metrics. Mean Squared Error (**MSE**) and Mean Absolute Error (**MAE**) measure pixel-level reconstruction accuracy, with MAE being less sensitive to large outliers. Peak Signal-to-Noise Ratio (**PSNR**, in dB) expresses MSE on a logarithmic scale, where higher values indicate better fidelity. Structural Similarity Index (**SSIM**) captures perceptual quality by jointly assessing luminance, contrast, and local structure, with values closer to 1 denoting greater similarity. Learned Perceptual Image Patch Similarity (**LPIPS**) computes perceptual distance using deep VGG features, correlating more closely with human judgement than pixel-level metrics; lower scores indicate greater perceptual similarity. Fréchet Inception Distance (**FID**) evaluates distributional alignment between predicted and ground-truth frames using Inception-V3 embeddings, where lower values indicate the predicted frames are statistically closer to the real distribution. For **MSE**, **MAE**, **LPIPS**, and **FID**, lower is better; for **PSNR** and **SSIM**, higher is better.

Tactile Prediction Metrics. . We evaluate predicted GelSight tactile observations using five metrics that target distinct aspects of visuotactile fidelity. Surface deformation patterns are assessed via optical flow: mean End-Point Error (**Flow EPE**) measures the average Euclidean distance between predicted and ground-truth flow vectors, and flow angular error in degrees (**Flow AE**) measures the mean directional deviation of the predicted deformation field; both are lower-is-better. Gradient cosine similarity (**Grad. Cos.**) computes the cosine similarity between the spatial gradient fields of predicted and ground-truth tactile images, reflecting the accuracy of contact edges and surface texture patterns, where higher values (maximum 1) are better. Finally, height map MAE (**Height MAE**) and MSE (**Height RMSE**) measure the pixel-wise reconstruction error of the 3D surface depth maps recovered via photometric stereo from the tactile images, with lower values indicating more accurate tactile shape prediction.

Results. Across all three tasks, the visuo-tactile world model achieves comparable or better prediction performance than the unimodal baselines, indicating that joint visual-tactile modeling does not degrade either modality and often improves both. For RGB prediction, **ViTaL** consistently improves the distributional and perceptual metrics over the RGB-only world model. In particular, **ViTaL** obtains lower FID and LPIPS on both front and wrist cameras across all tasks, suggesting that incorporating tactile observations helps the model generate visually more realistic future predictions, even when pixel-level metrics such as MSE or PSNR are similar. This is especially important for visual steering, where candidate actions are ranked by a VLM over decoded future images rather than by pixel reconstruction alone.

For tactile prediction, the benefit of joint modeling is more pronounced. Compared with the tactile-only world model, **ViTaL** improves most tactile metrics across all tasks, including flow endpoint error, flow angular error, and height reconstruction error. The gains are particularly clear in the insertion and pipette transfer tasks, where precise contact dynamics are central to task success. In the insertion task, **ViTaL** substantially reduces both flow error and height-map error, indicating that visual context helps predict how the peg will make contact with the tight-clearance hole. In the pipette transfer task, **ViTaL** also improves all tactile metrics, suggesting that visual information about the dropper, cups, and liquid-transfer configuration provides useful context for forecasting contact and grasp behavior.

The wiping task shows smaller but still consistent improvements for most tactile and visual metrics. This is expected because wiping involves broader surface contact and less geometrically constrained

Table 10: **World Model Prediction Evaluation For Wiping Task.** We evaluate each method across 30 evaluation videos and best values are in bold. RGB metrics are lower-is-better except PSNR and SSIM; flow and height metrics are lower-is-better, while gradient cosine similarity is higher-is-better.

RGB Prediction								
Camera	Method	Loss Type	FID↓	LPIPS↓	MAE↓	MSE↓	PSNR↑	SSIM↑
Front Camera	RGB-only WM	Multi-step	9.0932	0.0995	0.0162	0.0023	26.9366	0.9293
	ViTaL (RGB+Tactile WM)	Multi-step	8.1015	0.0986	0.0161	0.0023	26.9702	0.9300
Wrist Camera	RGB-only WM	Multi-step	24.9114	0.1741	0.0202	0.0029	26.0685	0.9058
	ViTaL (RGB+Tactile WM)	Multi-step	22.1616	0.1719	0.0201	0.0030	25.8676	0.9045

Tactile Prediction						
Method	Loss Type	Flow AE↓	Flow EPE↓	Grad. Cos.↑	Height MAE↓	Height RMSE↓
Tactile-only WM	Multi-step	6.6576	0.1396	0.5907	0.2380	0.2965
ViTaL (RGB+Tactile WM)	Multi-step	6.4938	0.1364	0.5829	0.2352	0.2959

contact than insertion, making the tactile future somewhat easier to predict from tactile history alone. Even in this case, the visuo-tactile model improves flow and height prediction while maintaining strong RGB prediction quality.

For qualitative evaluation, we further show that the world model can accurately imagine both successful and failed outcomes. For the wiping task, we visualize success and failure cases for wiping the orange marks in Fig. 15 and Fig. 16, respectively. We also show success and failure cases for wiping the black marks in Fig. 17 and Fig. 18, as well as a failure mode where the robot begins wiping but becomes stuck midway in Fig. 19. For the insertion task, we visualize the success and failure cases for inserting in the right hole in Fig. 20 and Fig. 22 respectively. We also show success and failure cases for inserting into the left hole in Fig. 21 and Fig. 23, as well as a failure mode where the robot fails to grasp the purple peg in Fig. 24. For pipette transfer task, we visualize the success and failure cases (spilling on the table or failing to put the dropper back) for transferring the liquid to the yellow cup in Fig. 25, Fig. 27 and Fig. 26 respectively. We also show the success and failures (spilling on the table or failing to put the dropper back) for transferring the liquid to the blue cup in Fig. 28, Fig. 30 and Fig. 29. We also introduce one additional failure mode of failing to grasp the dropper in Fig. 31.

Overall, these results support the design choice of using a shared visuo-tactile world model for both visual and tactile steering. The model provides reliable long-horizon visual predictions for selecting global action modes, while also producing accurate short-horizon tactile predictions for contact refinement. This is consistent with our bi-level formulation: vision benefits from tactile context when imagining future task progress, and tactile prediction benefits from visual context when estimating upcoming contact interactions.

C.2 Multi-Modal Guidance for Visuo-Tactile Policies

We evaluate whether our multi-modal guidance framework can improve both visual policies and visuo-tactile policies. For each task, we compare four variants: a base visual policy, a base visuo-tactile policy, our method applied to the visual policy, and our method applied to the visuo-tactile policy.

Base Policy. The visual policy receives wrist and external RGB observations together with proprioceptive state, while the visuo-tactile policy additionally receives the GelSight tactile observation. Both policies use the same action prediction horizon and execution protocol described in Appendix A.1.

Table 11: **World Model Prediction Evaluation For Insertion Task.** We evaluate each method across 30 evaluation videos and best values are in bold. RGB metrics are lower-is-better except PSNR and SSIM; flow and height metrics are lower-is-better, while gradient cosine similarity is higher-is-better.

RGB Prediction								
Camera	Method	Loss Type	FID↓	LPIPS↓	MAE↓	MSE↓	PSNR↑	SSIM↑
Front Camera	RGB-only WM	Multi-step	8.1047	0.1576	0.0122	0.0012	29.5526	0.9367
	ViTaL (RGB+Tactile WM)	Multi-step	8.0094	0.1566	0.0121	0.0012	29.6457	0.9372
Wrist Camera	RGB-only WM	Multi-step	9.3695	0.1395	0.0195	0.0023	26.8440	0.9129
	ViTaL (RGB+Tactile WM)	Multi-step	8.9580	0.1382	0.0191	0.0022	27.1264	0.9156

Tactile Prediction						
Method	Loss Type	Flow AE↓	Flow EPE↓	Grad. Cos.↑	Height MAE↓	Height RMSE↓
Tactile-only WM	Multi-step	8.8939	0.2209	0.4977	0.6896	0.8166
ViTaL (RGB+Tactile WM)	Multi-step	7.5189	0.1742	0.5014	0.6507	0.7656

Table 12: **World Model Prediction Evaluation For Pipette Transfer Task.** We evaluate each method across a fixed evaluation set of 30 videos and best values are in bold. RGB metrics are lower-is-better except PSNR and SSIM; flow and height metrics are lower-is-better, while gradient cosine similarity is higher-is-better.

RGB Prediction								
Camera	Method	Loss Type	FID↓	LPIPS↓	MAE↓	MSE↓	PSNR↑	SSIM↑
Front Camera	RGB-only WM	Multi-step	12.4612	0.1700	0.0162	0.0017	27.8010	0.9245
	ViTaL (RGB+Tactile WM)	Multi-step	11.5609	0.1699	0.0163	0.0019	27.5515	0.9222
Wrist Camera	RGB-only WM	Multi-step	11.3101	0.1481	0.0173	0.0017	27.9782	0.9273
	ViTaL (RGB+Tactile WM)	Multi-step	10.5830	0.1480	0.0173	0.0018	27.7871	0.9266

Tactile Prediction						
Method	Loss Type	Flow AE↓	Flow EPE↓	Grad. Cos.↑	Height MAE↓	Height RMSE↓
Tactile-only WM	Multi-step	2.2372	0.0451	0.5391	0.6648	0.7642
ViTaL (RGB+Tactile WM)	Multi-step	1.9514	0.0381	0.5415	0.6061	0.7082

Our method is applied at inference time without changing the underlying policy weights. For each policy query, visual steering first samples candidate action sequences and selects the one with the highest predicted visual reward. Tactile steering then locally refines the selected action chunk using tactile-guided diffusion editing. This allows us to test whether guidance improves policies that do not directly observe tactile input, as well as policies that already use tactile observations during action generation.

Results. Fig. 8 reports visual success, contact success, and overall task success across the wiping, pipette transfer, and insertion tasks. Across all tasks, our method substantially improves both the visual and visuo-tactile base policies. For the visual policy, our method improves overall success from 30% to 90% on wiping, from 30% to 75% on pipette transfer, and from 40% to 80% on insertion. These gains indicate that multi-modal guidance can compensate for limitations of the base policy by using predicted visual futures for global goal selection and predicted tactile futures for contact refinement.

The base visuo-tactile policy generally improves contact success over the base visual policy, especially for pipette transfer and insertion, where tactile feedback is important for grasp stability and precise contact. However, adding tactile observations alone does not fully solve the tasks: overall

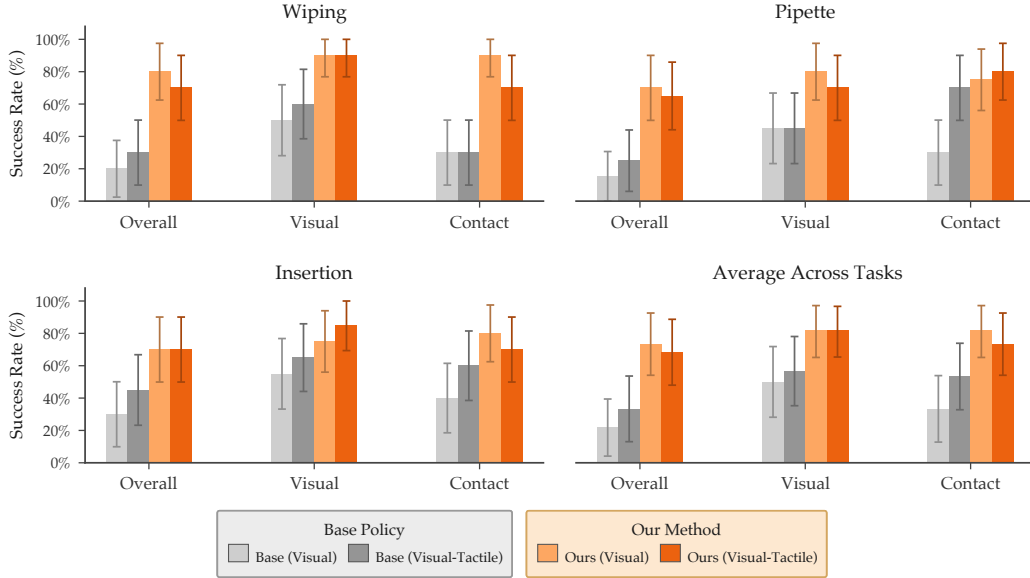


Figure 8: **Policy Steering Performance with Different Base Policy.** We ablate different base policies including policy conditioned on visual observations and policy conditioned on visuo-tactile observations. Our policy steering method **ViTaL** obtains a substantial performance gain no matter what the base policy is.

success remains 30%, 70%, and 60% for wiping, pipette transfer, and insertion, respectively. With our guidance, the visuo-tactile policy improves to 70%, 80%, and 70% overall success. This shows that explicit inference-time steering remains beneficial even when the policy already receives tactile observations.

Comparing our method on visual versus visuo-tactile policies, we observe that the stronger base policy does not always lead to higher final success after guidance. For wiping and insertion, our method with the visual policy achieves the highest overall success, while for pipette transfer, our method with the visuo-tactile policy performs best. This suggests that the benefit of tactile observations in the base policy depends on the task and on how well the policy uses tactile signals during action generation. Sometimes adding the tactile observations in the policy input leads to the overfitting of the policy and the base policy distribution is much narrower compared to the visual policy, resulting in worse steering performance. In summary, our guidance framework consistently improves performance by explicitly decomposing the problem into visual mode selection and tactile contact refinement.

D Additional Results

D.1 Multi-modal Visuo-Tactile Steering Outperforms Steering with Single Modality

In this section, we provide additional qualitative results for multi-modal visuo-tactile steering across the wiping, insertion, and pipette transfer tasks. These examples complement the quantitative results in Sec. 5.1 and illustrate how visual and tactile guidance contribute to different aspects of task success.

Success of Multi-Modal Steering. Across all tasks, **ViTaL** succeeds by using visual guidance to select the correct global behavior mode and tactile guidance to refine local contact-rich execution. In the wiping task, visual steering selects the correct marked region according to the language instruction, while tactile steering improves contact pressure between the eraser and the board, allowing the robot to remove the target marks more completely as shown in Fig. 12. In the insertion task, vi-

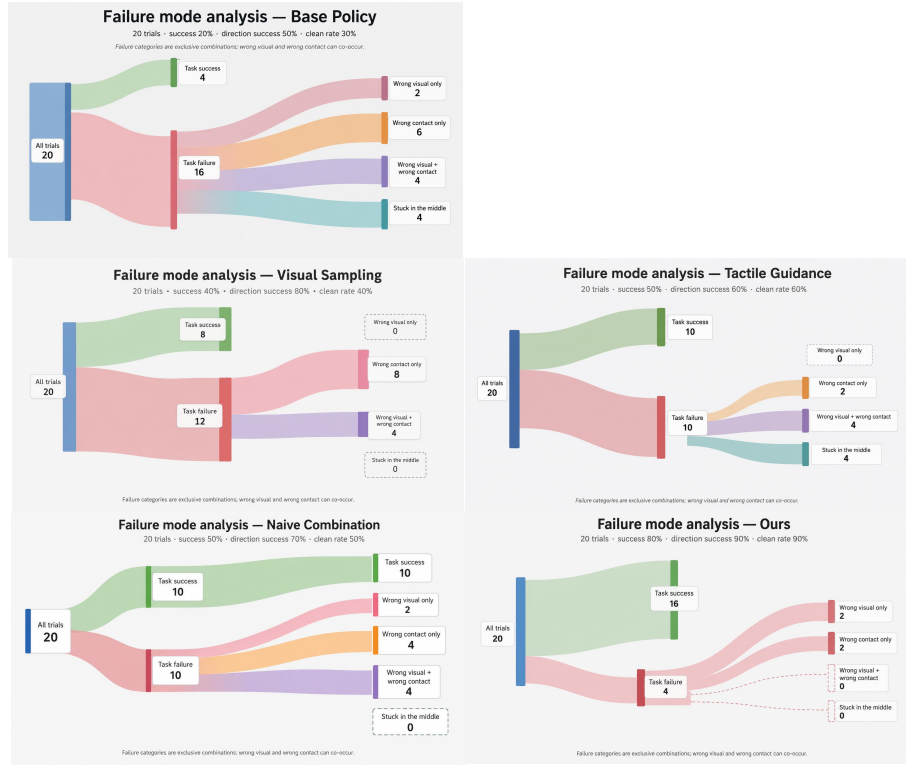


Figure 9: **Failure Mode Analysis For Wiping Task.** We demonstrate the different failure modes of ViTaL and baselines for the wiping task.

visual steering selects the correct target hole, while tactile steering helps adjust the peg motion during contact, improving alignment and insertion under the tight 1 mm tolerance as shown in Fig. 13. In the pipette transfer task, visual steering guides the robot toward the specified blue or yellow cup and then back to the red cup, while tactile steering improves grasp stability and contact force during liquid transfer and return as shown in Fig. 14.

Failure Modes of Visual-Only Steering. Visual-only steering primarily fails when visually promising futures do not correspond to physically reliable execution. For example, in the wiping task, visual lookahead may select an action sequence that reaches the correct mark but fails to apply sufficient pressure, resulting in incomplete erasing in Fig. 12. In the insertion task, the robot may move toward the correct hole but fail to insert the peg because small contact misalignments are difficult to resolve from visual predictions alone as shown in Fig. 13. In the pipette transfer task, the robot may approach the correct target cup but grasp the dropper with too large force, causing spilling in Fig. 14. These failures show that visual success alone is insufficient for contact-rich tasks where local force, alignment, and grasp stability determine whether the task is completed. Detailed failure mode summary of three tasks are in Fig. 9, Fig. 10 and Fig. 11.

Failure Modes of Tactile-Only Steering. Tactile-only steering exhibits the complementary limitation. Since tactile rewards emphasize local contact quality, tactile-only methods can improve grasping, pressing, or alignment behavior, but they do not reliably encode the global semantic target. In the wiping task, tactile-only guidance may produce correct contact between eraser and board but wipe the wrong region in Fig. 12. In the insertion task, it may encourage contact between the peg and tabletop but fail to select the instructed top-left or top-right hole as shown in Fig. 13. In the pipette transfer task, it may improve grasp force on the dropper but move toward the wrong cup or fail to complete the return phase. These examples indicate that tactile feedback is most effective for

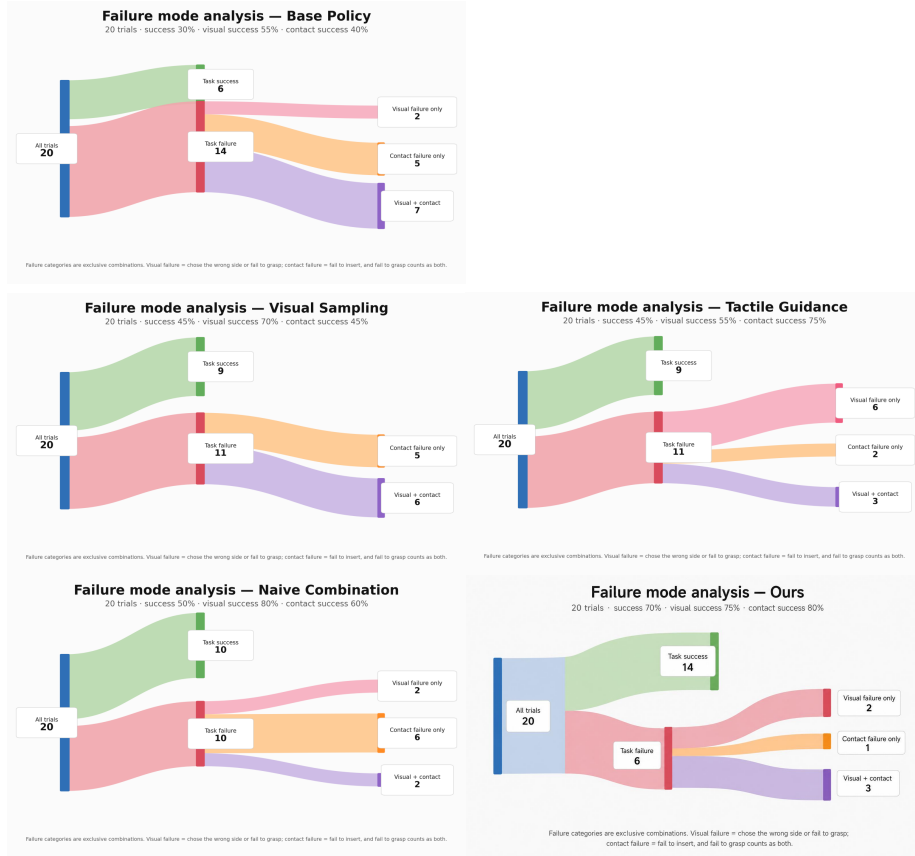


Figure 10: **Failure Mode Analysis For Insertion Task.** We demonstrate the different failure modes of **ViTaL** and baselines for the insertion task.

short-horizon local refinement, but is insufficient for long-horizon semantic task selection in Fig. 14. Detailed failure mode summary of three tasks are in Fig. 9, Fig. 10 and Fig. 11.

Summary. The qualitative results reinforce the quantitative trends in Fig. 2. Visual-only steering is effective for choosing where to act, but not necessarily how to interact. Tactile-only steering improves how the robot interacts locally, but cannot reliably determine the correct semantic target. **ViTaL** combines the strengths of both modalities by using visual lookahead for global task progress and tactile diffusion editing for local contact refinement, leading to more reliable execution across diverse contact-rich manipulation tasks.

D.2 Semantically-Aligned Vision and Tactile Verifier Enables Global and Local Adaptation

This section provides additional analysis for Sec. 5.2. Effective inference-time steering requires two properties: the world model must preserve reward-relevant information in its predicted futures, and the verifier must assign rewards that are semantically aligned with the task description. We therefore evaluate both the visual and tactile verifiers on ground-truth rollouts and world-model-predicted rollouts.

Preference-Order Accuracy. We evaluate reward quality using preference-order accuracy [52]. For each task and each modality, we construct pairs of rollouts with different human preference labels, such as successful versus failed executions or trajectories that better satisfy the language goal. Given a rollout i , we compute its normalized reward score by averaging the verifier reward

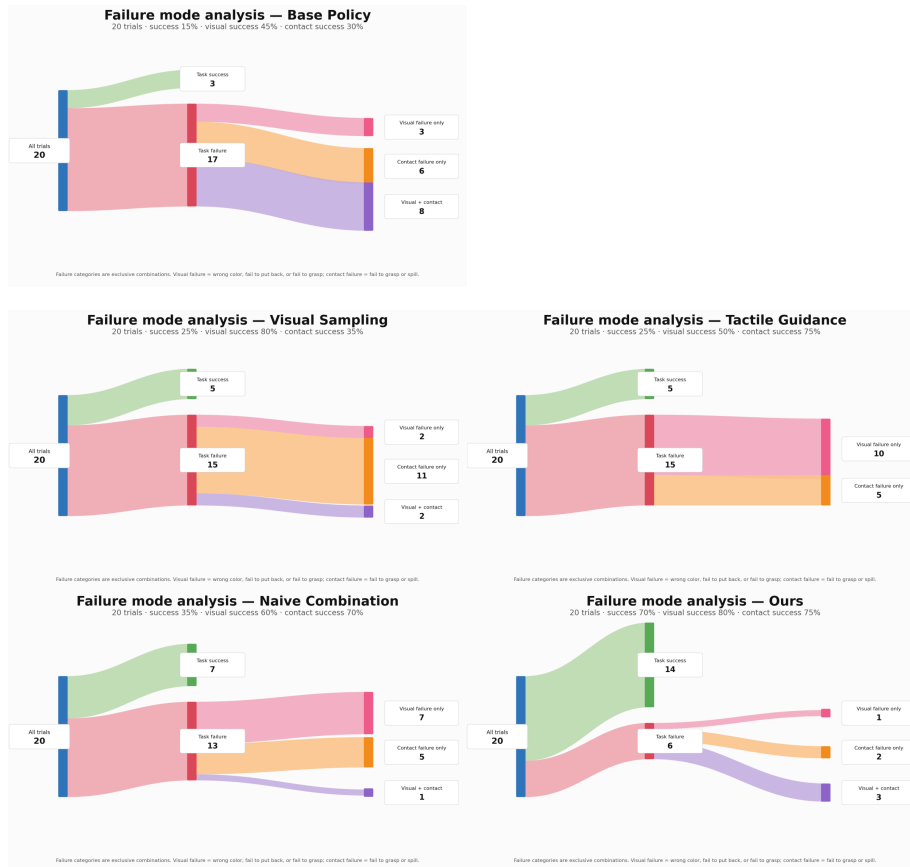


Figure 11: **Failure Mode Analysis For Pipette Transfer Task.** We demonstrate the different failure modes of ViTaL and baselines for the pipette transfer task.

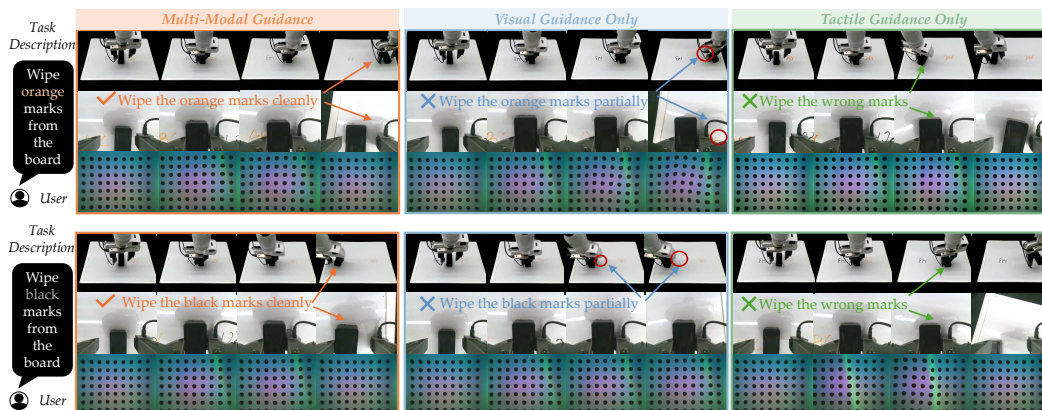


Figure 12: **Policy Steering with Different Guidance Modalities For Wiping Task.** For both task descriptions of "wiping the orange marks" and "wiping the black marks", vision-only guidance (middle) fail due to local contact errors, while tactile-only guidance (right) can fail due to global task misalignment. ViTaL (left) with multi-modal guidance addresses both failure modes.

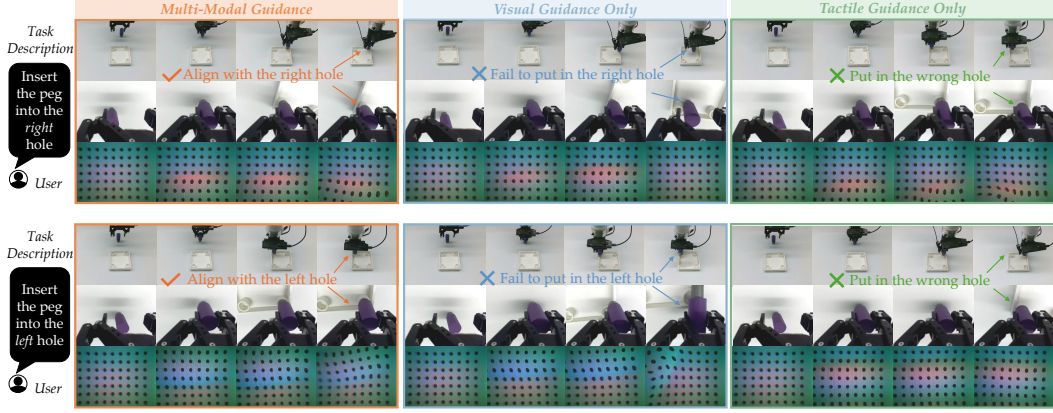


Figure 13: **Policy Steering with Different Guidance Modalities For Insertion Task.** For both task descriptions of ”inserting into the left hole” and ”inserting into the right hole”, vision-only guidance (middle) fail due to local contact errors, while tactile-only guidance (right) can fail due to global task misalignment. **ViTaL** (left) with multi-modal guidance addresses both failure modes.

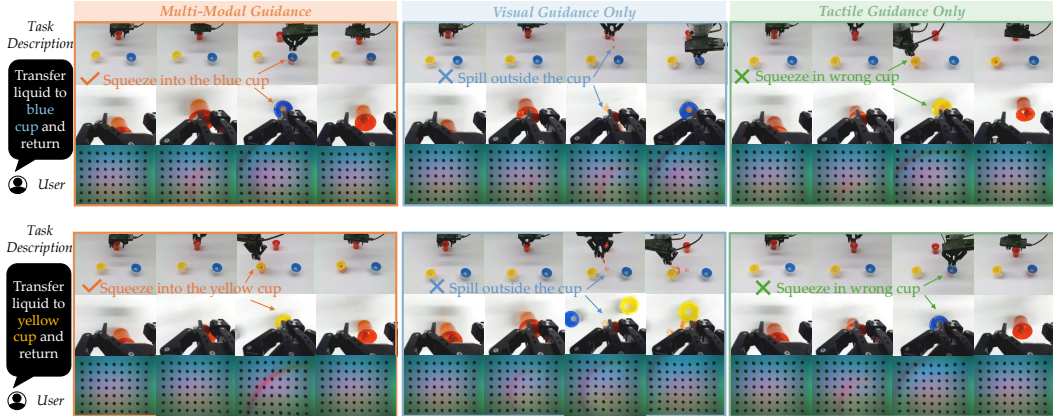


Figure 14: **Policy Steering with Different Guidance Modalities For Pipette Transfer Task.** For both task descriptions of ”transfer liquid to blue cup” and ”transfer liquid to right cup”, vision-only guidance (middle) fail due to local contact errors, while tactile-only guidance (right) can fail due to global task misalignment. **ViTaL** (left) with multi-modal guidance addresses both failure modes.

over the rollout:

$$\hat{y}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} R(\hat{z}_t^{(i)}; \ell), \quad (12)$$

where $\hat{z}_t^{(i)}$ denotes either the ground-truth encoded observation or the world-model-predicted latent at timestep t , and ℓ is the corresponding visual or tactile language description. For each pair (i, j) with different human preference labels $y_i \neq y_j$, we compare the human preference ordering with the reward-induced ordering:

$$s_{ij}^H = \text{sign}(y_i - y_j), \quad s_{ij}^R = \text{sign}(\hat{y}_i - \hat{y}_j). \quad (13)$$

The preference-order accuracy is then:

$$A = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \mathbb{1}[s_{ij}^H = s_{ij}^R], \quad (14)$$

where \mathcal{P} is the set of evaluated rollout pairs. Higher accuracy indicates that the verifier reward induces an ordering that better matches human judgments. We report results on both ground-truth

observations (**GT**) and predicted futures (**Pred**) to measure whether the verifier remains reliable when applied to world-model imagination.

Visual Verifier Performance. Table 1 shows that the adapted ROBOMETER visual verifier achieves strong preference-order accuracy across all tasks. On ground-truth observations, the visual reward obtains 80.0% on wiping, 70.0% on insertion, and 100.0% on pipette transfer, with an average of 83.3%. On world-model predictions, the visual reward remains similarly reliable as 82.5%, 72.5%, and 100.0%, respectively, with an average of 85.0%.

These results indicate that the predicted visual futures preserve the semantic information needed by the VLM-based verifier. In particular, the high pipette accuracy suggests that the visual verifier can robustly distinguish whether the robot is moving toward or dispensing into the language-specified target cup. The insertion task is more challenging, likely because the ROBOMETER reward model still struggles to understand the spatial relationship of left or right compared to recognizing the color. Even in this case, the visual verifier achieves above-chance and task-useful preference accuracy, supporting its use for global mode selection rather than precise contact evaluation.

Tactile Verifier Performance. The tactile verifier also achieves strong preference-order accuracy across tasks. On ground-truth tactile observations, it obtains 85.0% on wiping, 70.0% on insertion, and 80.0% on pipette transfer, with an average of 78.3%. On predicted tactile futures, the tactile reward obtains 90.0%, 77.5%, and 77.5%, respectively, improving the average to 81.7%.

These results show that the tactile verifier provides a meaningful contact-sensitive reward for both real and imagined tactile observations. The wiping task has the highest tactile accuracy, suggesting that the tactile embeddings reliably capture contact and pressure patterns. The insertion task achieves 77.5% accuracy on predicted futures, indicating that the world model preserves contact cues relevant to peg-hole interaction. The pipette task remains similar accuracy between ground-truth (80.0%) and predicted futures (77.5%).

Ground-Truth vs. Predicted Futures. The results also demonstrate verifier accuracy on predicted futures is comparable to, and sometimes higher than, accuracy on ground-truth observations. This suggests that the visuo-tactile world model does not merely reconstruct observations at the pixel level, but preserves the semantic and contact-relevant structure required for reward evaluation. Predicted futures may also smooth high-frequency sensor noise and remove irrelevant variations in camera or tactile observations, making preference ordering easier for the verifier in some cases. This effect is visible in the improved average visual accuracy from 83.3% on **GT** to 85.0% on **Pred**, and tactile accuracy from 78.3% on **GT** to 81.7% on **Pred**.

Qualitative Reward Behavior. Fig. 4 further illustrates the semantic alignment of the verifiers on the pipette transfer task. For visual verification, the reward depends on the language-conditioned task phase. Given the same trajectory where the robot approaches and dispenses toward the yellow cup, the verifier assigns high reward when the task description is “transfer the liquid to the yellow cup,” and assign lower reward when the actual task is to transfer to the blue cup. After phase switching, the visual objective changes to returning the dropper to the red cup, and the reward correspondingly shifts to favor predicted futures that move back toward the red cup.

For tactile verification, the reward captures local contact transitions. Early in the pipette task, the tactile description “grasp lightly” favors gentle contact suitable for approaching and positioning the dropper. Once the visual reward indicates sufficient task progress, the tactile description switches to “grasp heavily,” encouraging stronger contact for stable dispensing and return. The tactile reward trace aligns with marker-tracking force estimates, indicating that the language-conditioned tactile embedding captures meaningful changes in grasp force.

Implications for Steering. These results justify our use of semantically aligned verifiers in the bi-level steering framework. The visual verifier is reliable for long-horizon semantic evaluation, such as selecting the correct target mark, hole, or cup. The tactile verifier is reliable for short-horizon contact evaluation, such as determining whether the robot is applying sufficient pressure, maintaining stable grasp, or producing contact patterns consistent with insertion. Since both verifiers remain accurate

on world-model predictions, **ViTaL** can evaluate candidate futures before execution and steer the policy toward actions that satisfy both global task goals and local contact requirements.

D.3 Bi-level Optimization Balances Success and Steering Speed Better

This section provides additional qualitative and runtime analysis for Sec. 5.3. The main experiments compare **ViTaL** to a naive multimodal combination baseline that sums normalized visual and tactile rewards across sampled candidates. Here, we further analyze why the bi-level structure is important.

Why Naive Reward Fusion Fails. A straightforward way to combine visual and tactile rewards is to normalize both rewards across sampled candidates and select the candidate with the highest summed score. However, this approach assumes that the two rewards are comparable over the same horizon and should be optimized simultaneously. In practice, this assumption is problematic. Visual rewards are most informative over longer horizons, where the robot’s future state reveals whether it reaches the correct target. Tactile rewards are most informative over shorter horizons, where local contact changes indicate grasp force, insertion alignment, or wiping pressure.

Because the two modalities operate on different temporal and semantic scales, naive reward fusion can select candidates that are suboptimal for full task execution. A candidate with high visual reward may dominate the combined score even if it has poor contact quality. Conversely, a candidate with strong tactile contact may be selected even if it does not make progress toward the correct visual goal. This reward imbalance leads to failures such as incomplete wiping, peg misalignment, or spilling in the pipette task, as shown in Fig. 5.

Benefits of Bi-Level Steering. Our bi-level formulation avoids this issue by assigning different roles to the two modalities. The inner visual stage performs long-horizon sampling and verification to select the global action mode:

$$\bar{a}_{t:t+H} = \arg \max_{a_{t:t+H}} R^v(z_{t:t+H}^v; \ell_p^v).$$

This stage determines the task-relevant target, such as which mark to wipe, which hole to insert into, or which cup to transfer liquid to. The outer tactile stage then refines the first h steps of the selected sequence:

$$a_{t:t+h}^* = \arg \max_{a_{t:t+h}} \log p_\theta(\bar{a}_{t:t+h} | \bar{a}_{t:t+h}, o_t) + \beta R^t(\bar{z}_{t:t+h}^t; \ell_p^t).$$

This keeps the final action close to the visually selected mode while allowing local edits that improve contact quality. As a result, **ViTaL** avoids optimizing a single mixed reward over a mismatched horizon.

Qualitative Analysis. Fig. 5 shows representative examples where naive fusion fails but **ViTaL** succeeds. In wiping, naive fusion may move toward the correct mark but fail to maintain sufficient eraser-board contact, leaving marks partially visible. In insertion, naive fusion may choose the correct side but fail to align the peg with the hole during contact. In pipette transfer, naive fusion may approach the correct cup but spill liquid or fail to return the dropper. These failures arise because naive fusion does not explicitly separate global goal selection from local contact refinement.

By contrast, **ViTaL** first selects the correct global behavior using visual lookahead and then uses tactile diffusion editing to improve local physical execution. This produces actions that are both semantically correct and physically feasible.

Inference-Time Trade-Off. The bi-level design also provides a favorable efficiency trade-off as shown in Table 13. Visual lookahead is performed through sampling and verification because direct gradient-based guidance through a VLM, decoder, recurrent world model, and diffusion policy would be expensive and unstable. Tactile refinement, however, is applied locally over the first h actions, where classifier-based diffusion guidance is efficient and contact information is most relevant. As reported in Sec. 5.3, **ViTaL** adds only a small inference-time overhead relative to naive fusion while achieving substantially higher success. This suggests that the performance improvement comes from the structure of the optimization, rather than simply from spending more computation.

Table 13: **Inference Time Analysis.** Runtime values are reported in milliseconds as mean \pm standard deviation. The main bottleneck is visual reward inference from VLM-based ROBOMETER.

Method	Configuration		Runtime (ms)				
	Batch Size N	Horizon H	Sampling	WM + Policy Interleaving	Visual Verification	Tactile Verification	Total
Ours	10	16	69 \pm 3	121 \pm 3	216 \pm 10	65 \pm 3	471 \pm 12
Naive combination	10	16	69 \pm 3	121 \pm 3	216 \pm 10	24 \pm 1	429 \pm 13
Vision-only sampling (8-step)	10	8	69 \pm 3	11 \pm 1	216 \pm 10	–	295 \pm 11
Vision-only sampling (16-step)	10	16	69 \pm 3	121 \pm 3	216 \pm 10	–	405 \pm 12
Tactile-only sampling	10	8	69 \pm 3	11 \pm 1	–	24 \pm 1	103 \pm 4
Tactile-only guidance	1	8	–	–	–	94 \pm 4	94 \pm 4
Base policy	1	–	3 \pm 1	–	–	–	3 \pm 1

Summary. The additional results show that bi-level steering is important for both effectiveness and efficiency. Rather than combining visual and tactile rewards in a single flat objective, **ViTAL** uses each modality at the temporal scale where it is most reliable: long-horizon vision for global task selection and short-horizon touch for local contact refinement.

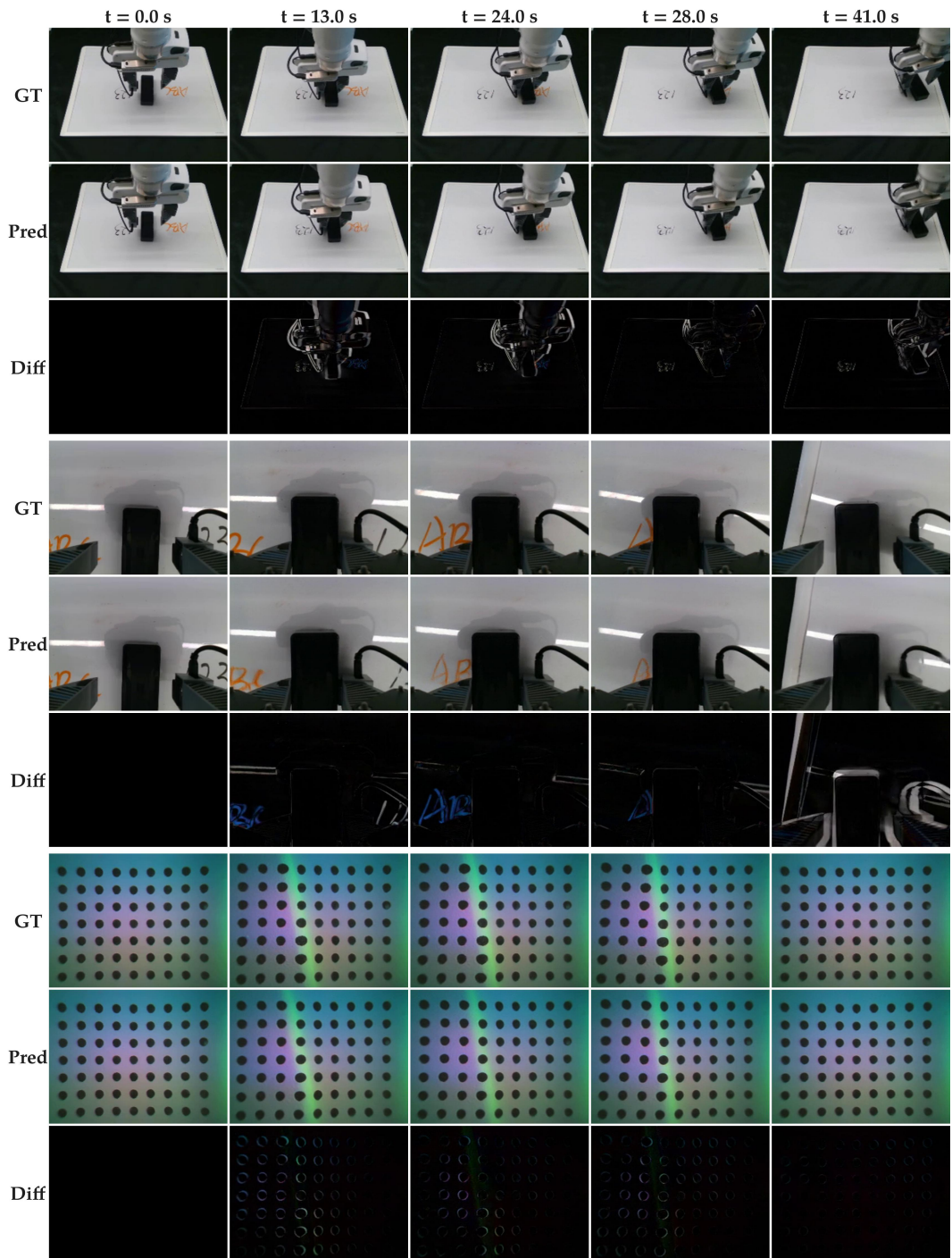


Figure 15: ViTaL Visuo-Tactile World Model Rollouts For Successfully Wiping the Orange Marks From the Board. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

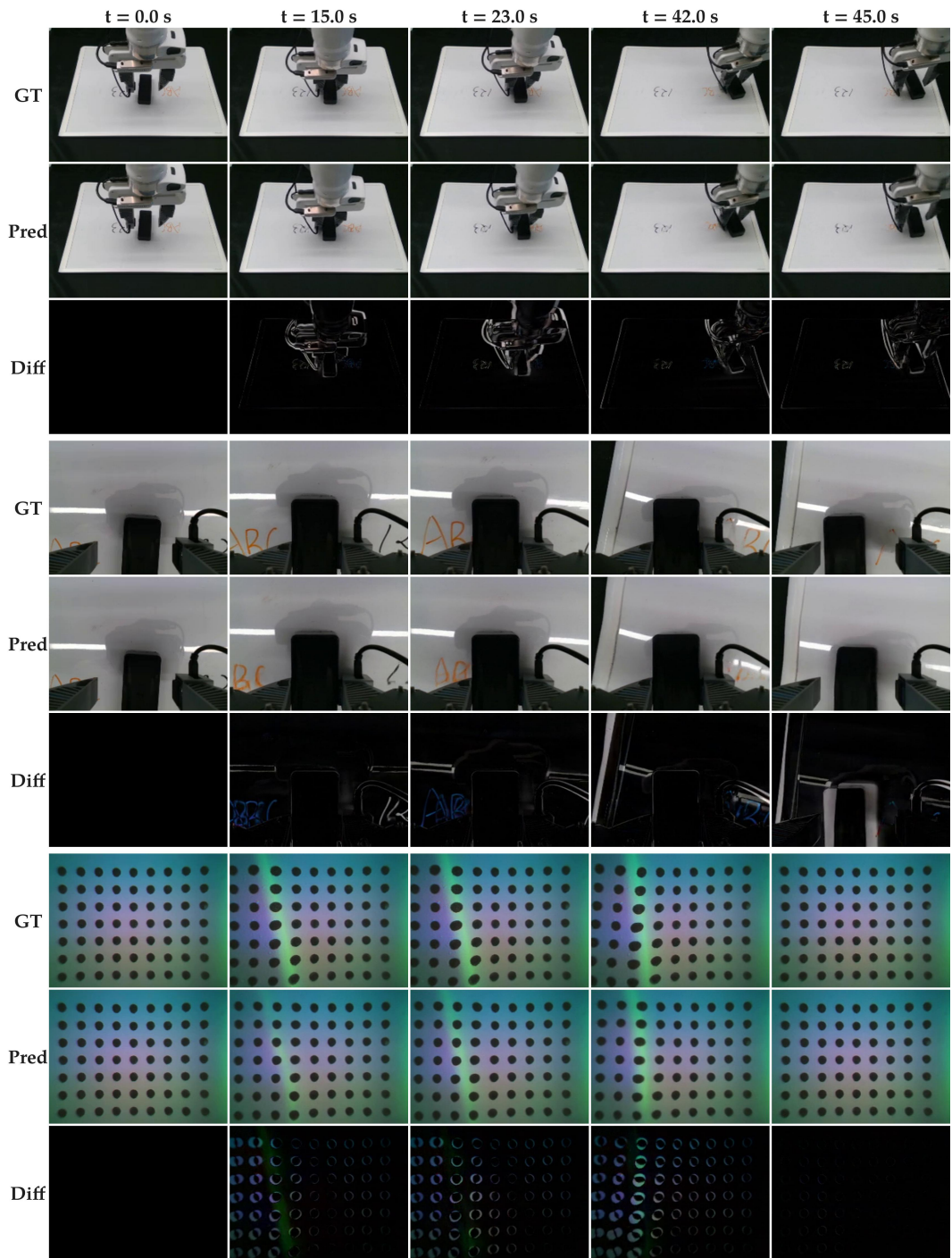


Figure 16: ViTaL Visuo-Tactile World Model Rollouts For Failing to Completely Wipe the Orange Marks From the Board. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

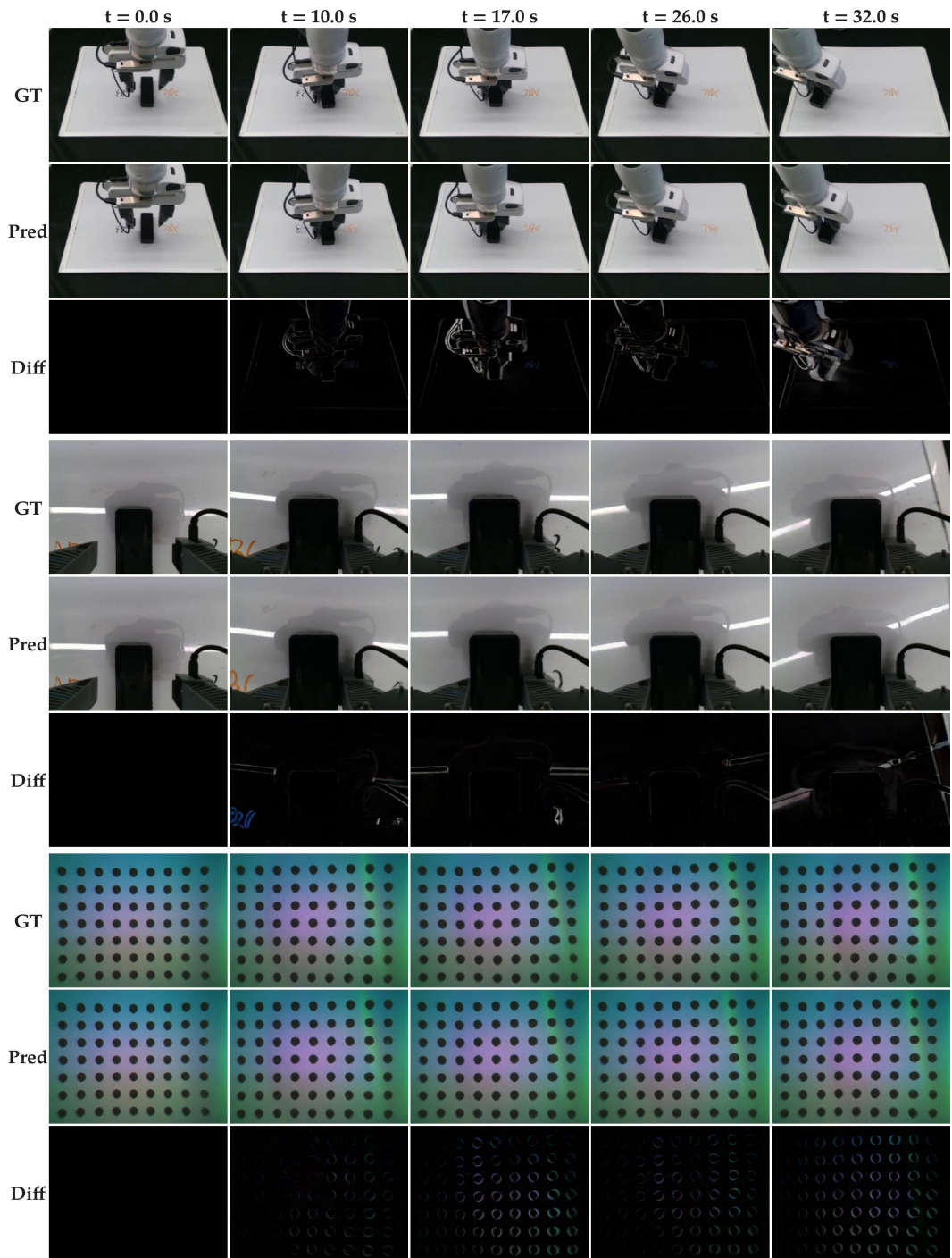


Figure 17: **ViTaL Visuo-Tactile World Model Rollouts For Successfully Wiping the Black Marks From the Board.** We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

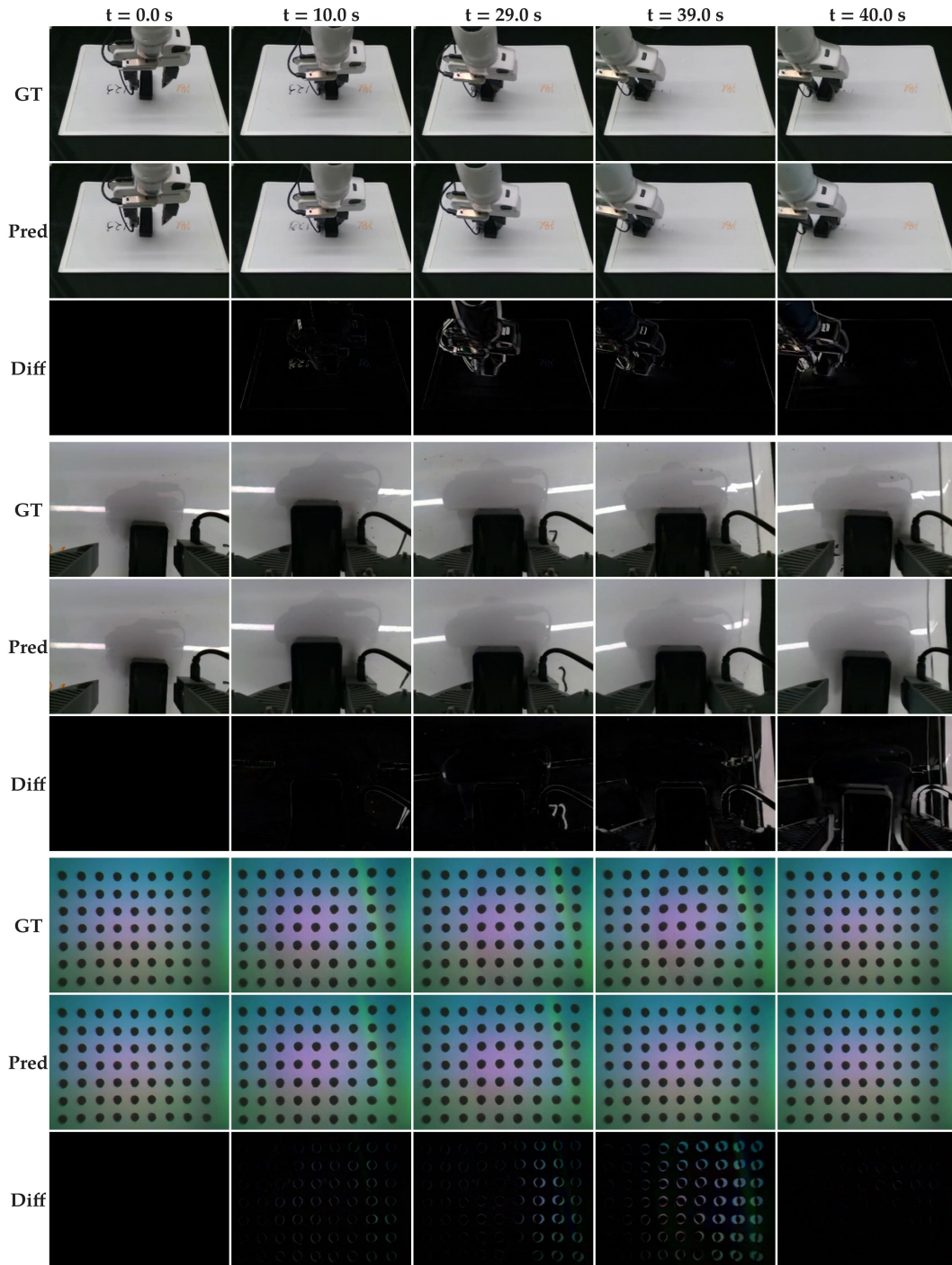


Figure 18: ViTaL Visuo-Tactile World Model Rollouts For Failing to Wipe the Black Marks From the Board. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

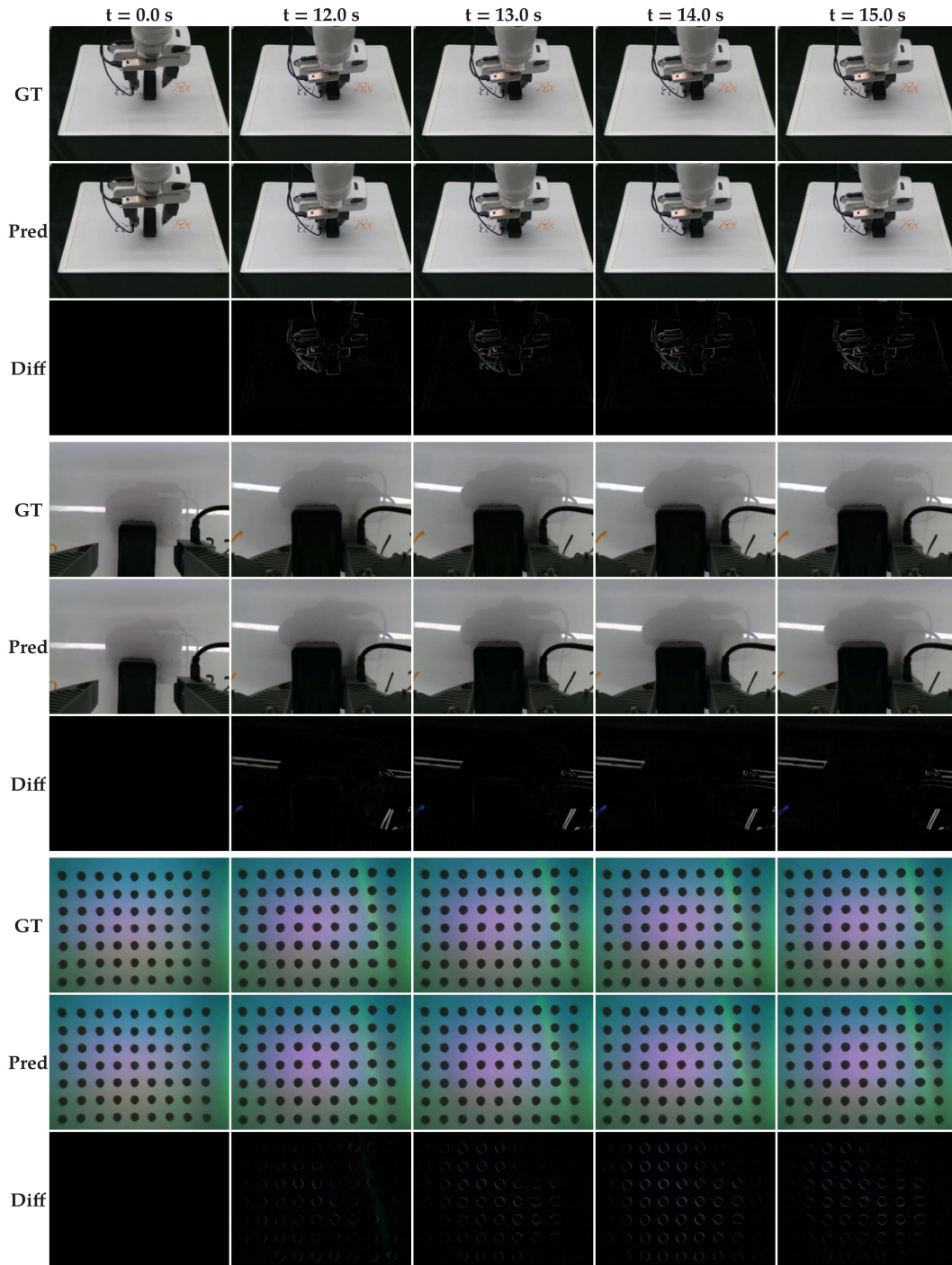


Figure 19: ViTaL Visuo-Tactile World Model Rollouts For Sticking in the Middle of the Board. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

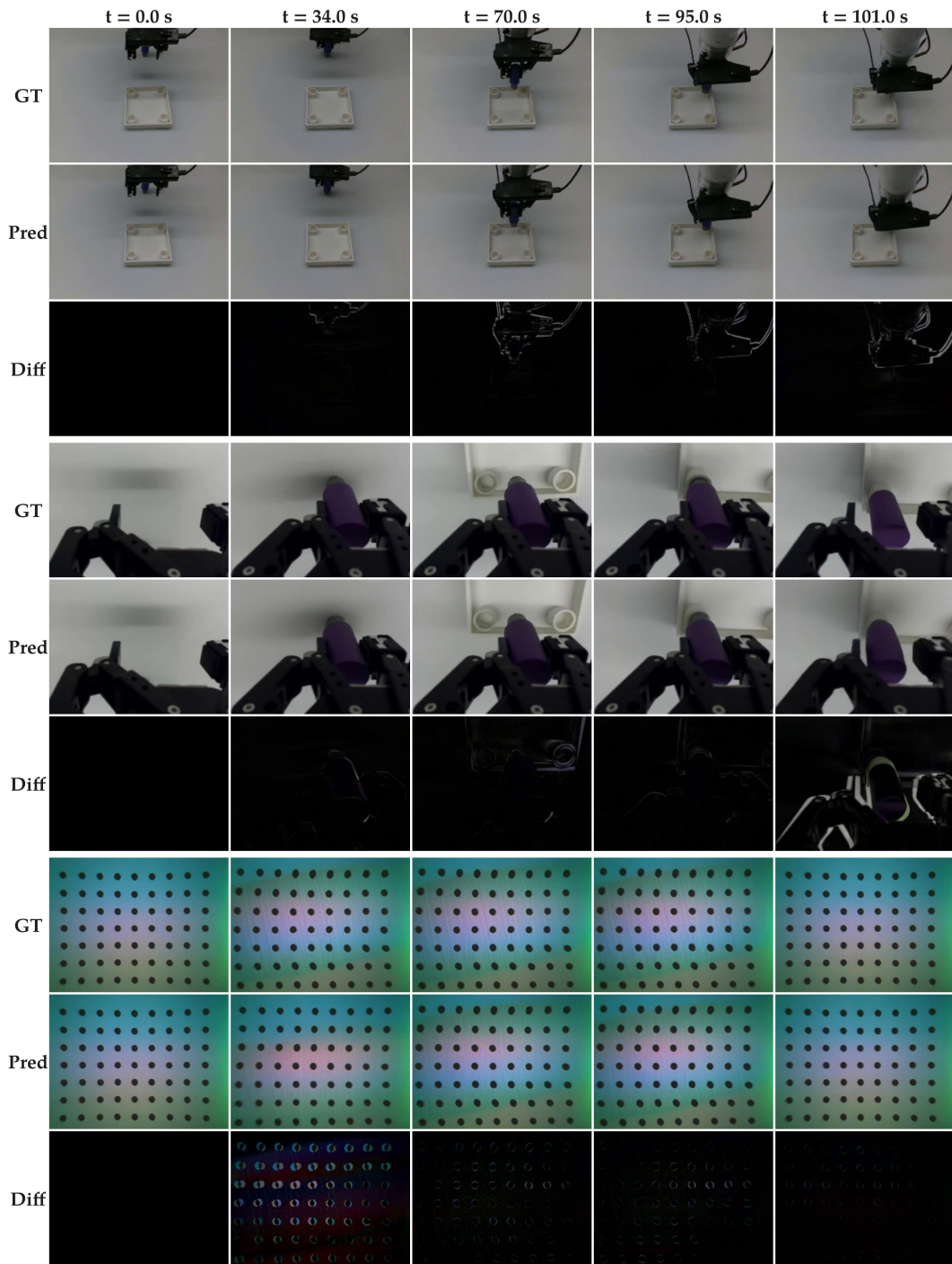


Figure 20: ViTaL Visuo-Tactile World Model Rollouts For Successfully Inserting the Peg into the Right Hole. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

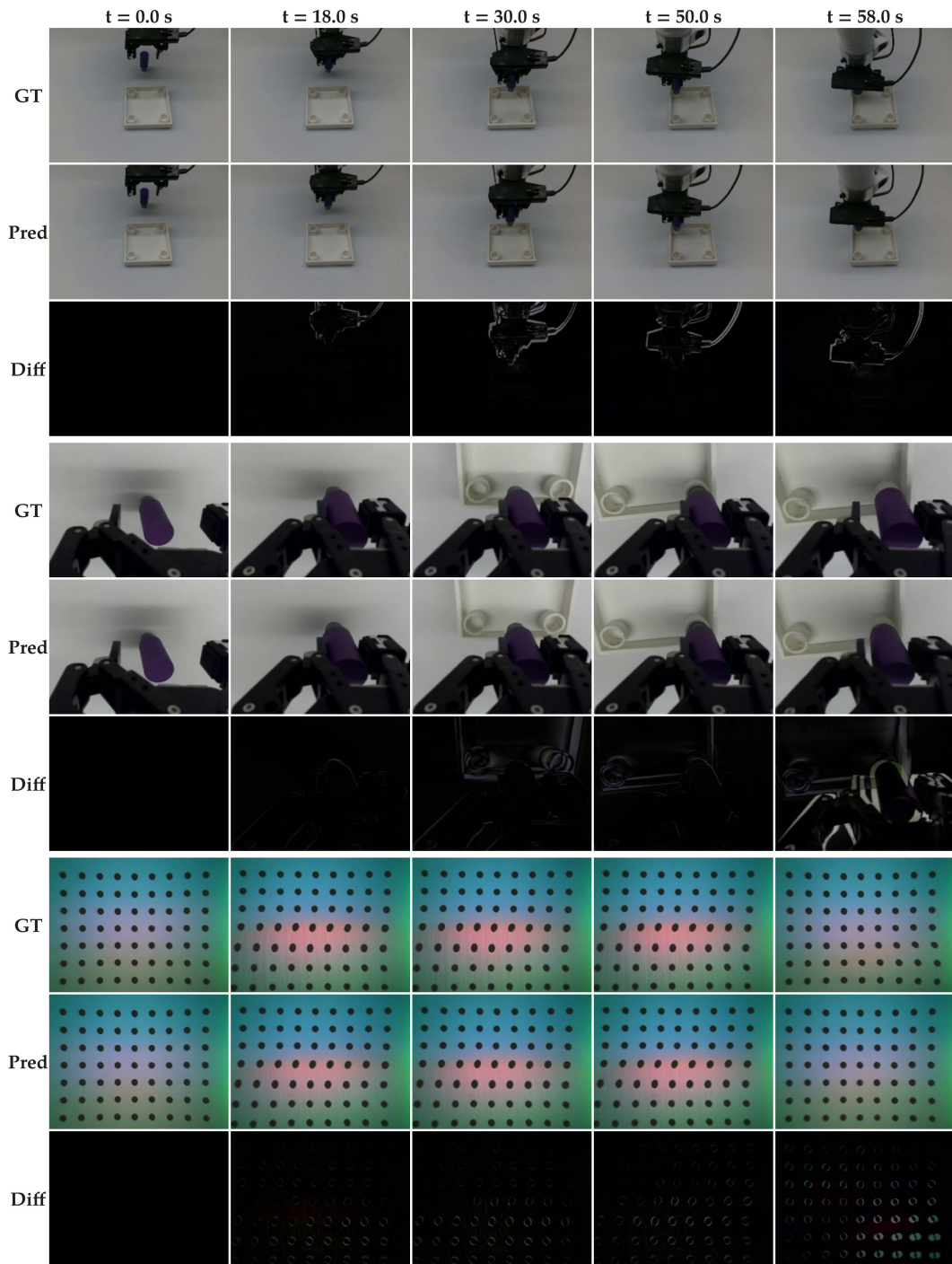


Figure 21: ViTaL Visuo-Tactile World Model Rollouts For Successfully Inserting the Peg into the Left Hole. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

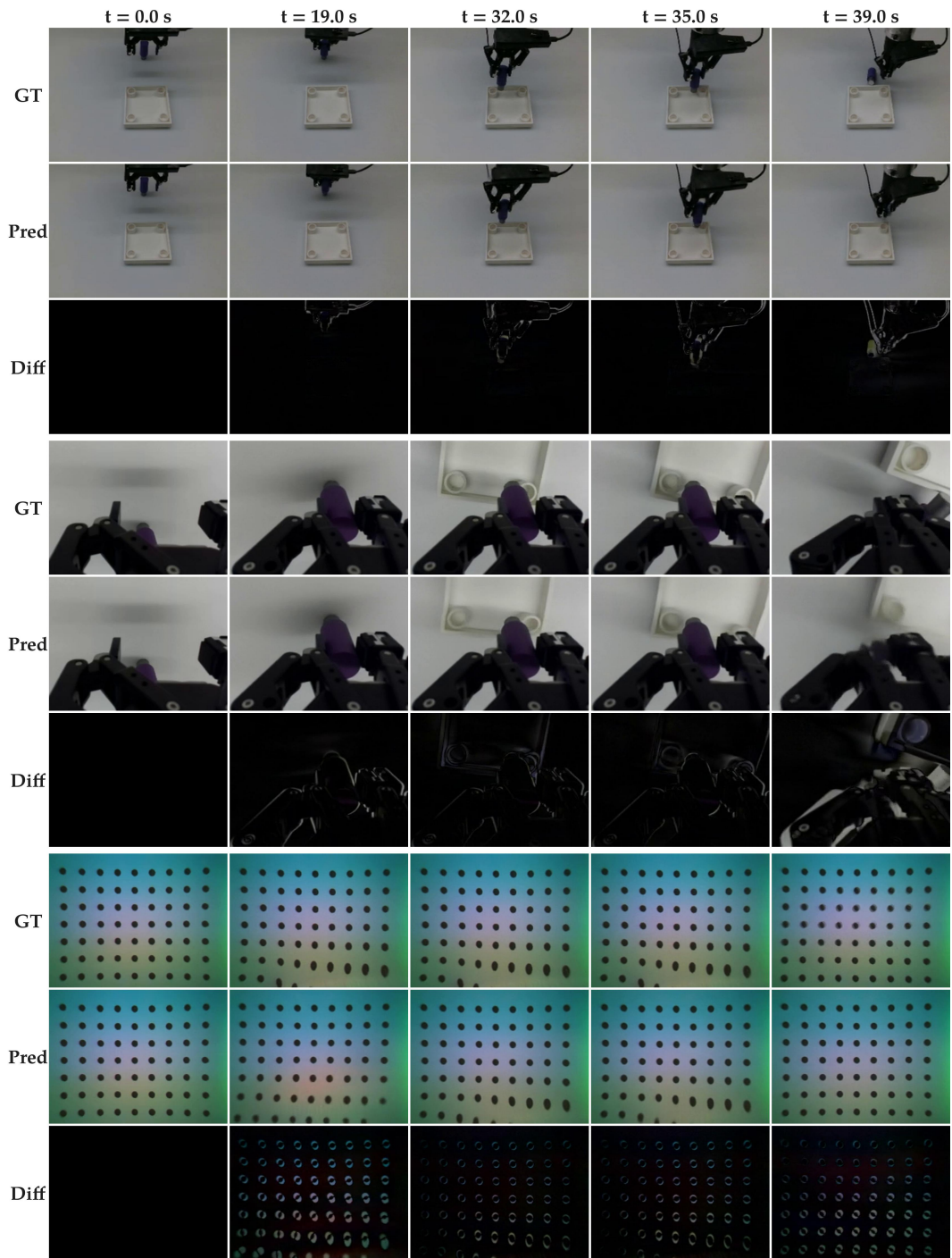


Figure 22: ViTaL Visuo-Tactile World Model Rollouts For Failing to Insert the Peg into the Right Hole. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

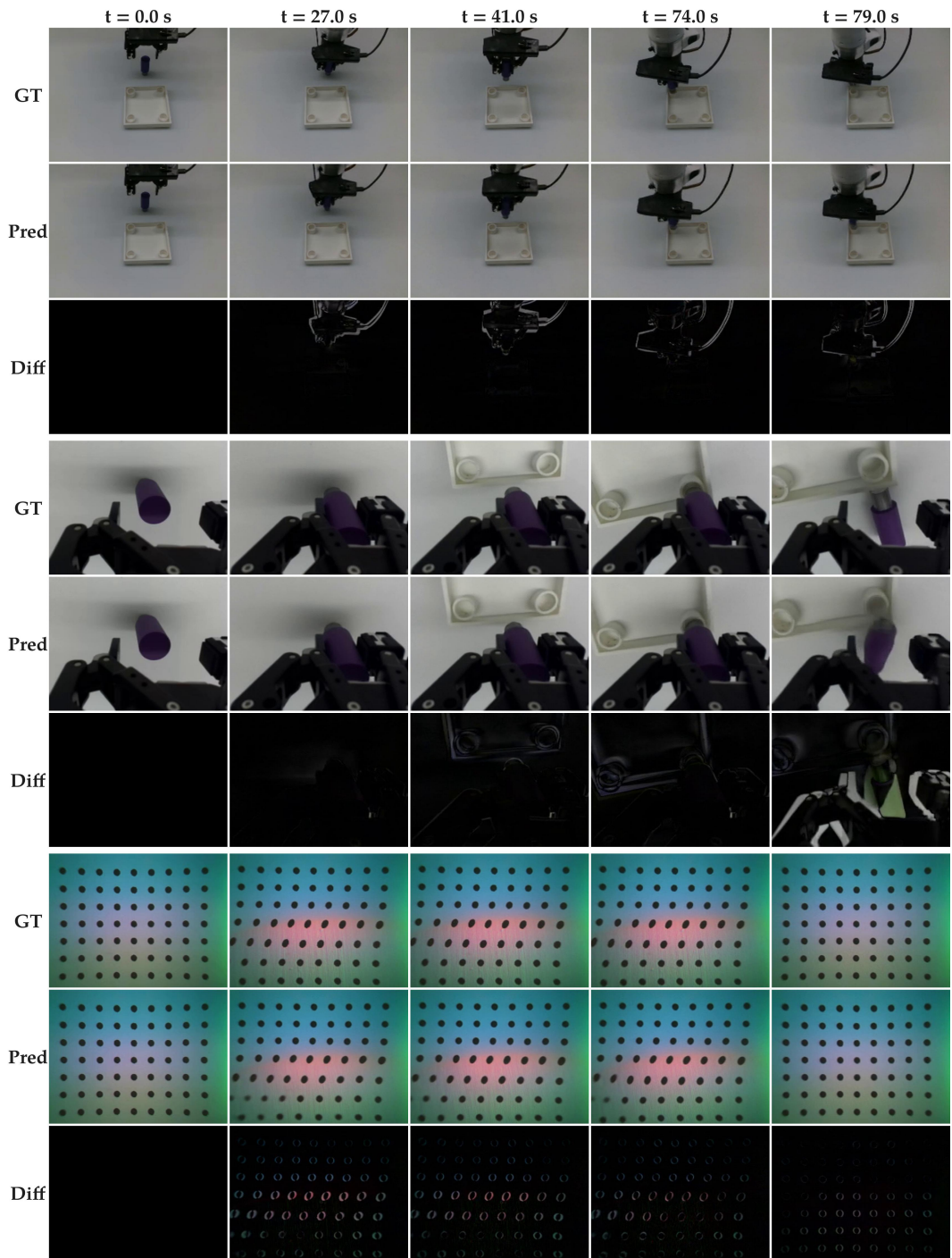


Figure 23: ViTaL Visuo-Tactile World Model Rollouts For Failing to Insert the Peg into the Left Hole. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

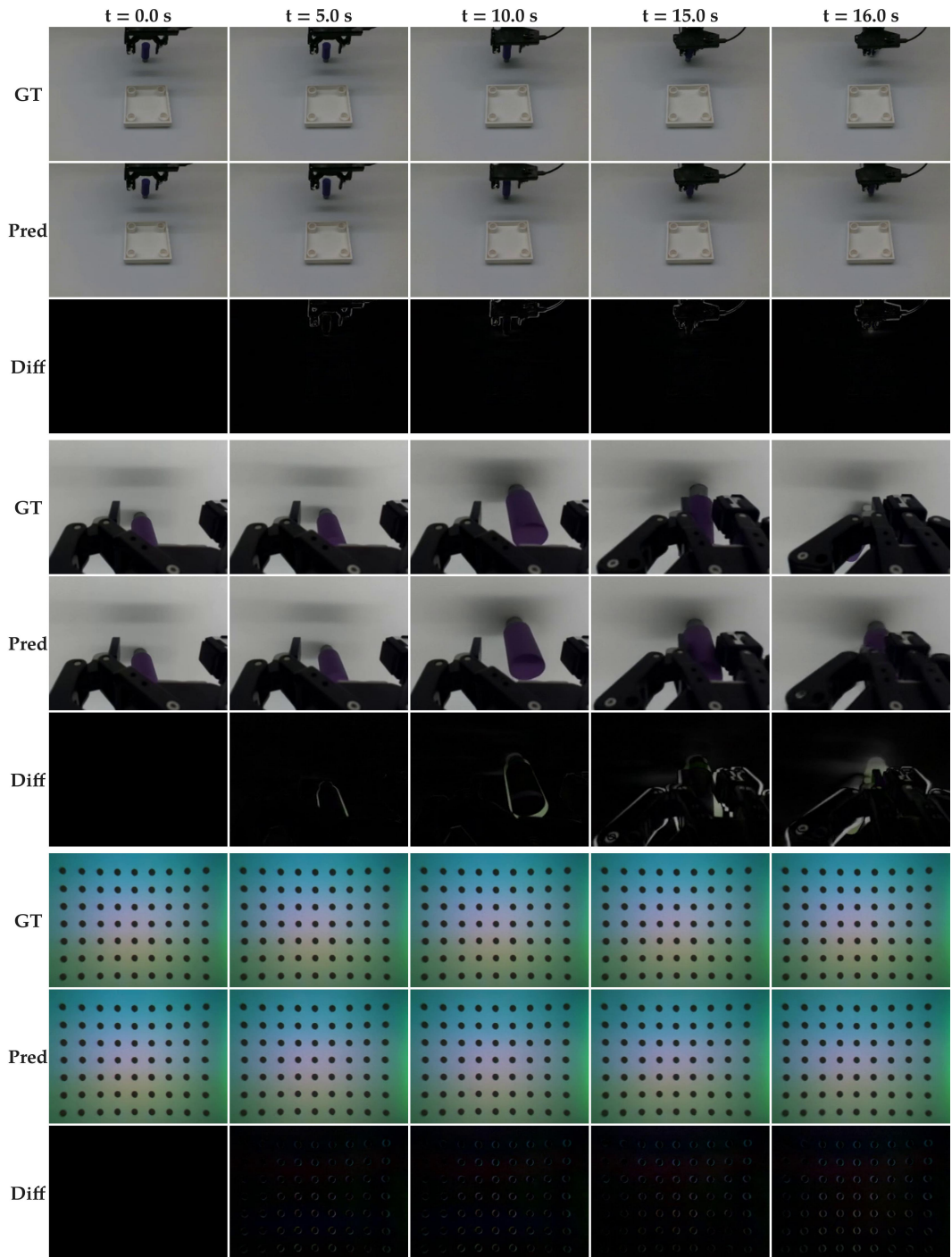


Figure 24: ViTaL Visuo-Tactile World Model Rollouts For Failing to Grasp the Peg. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

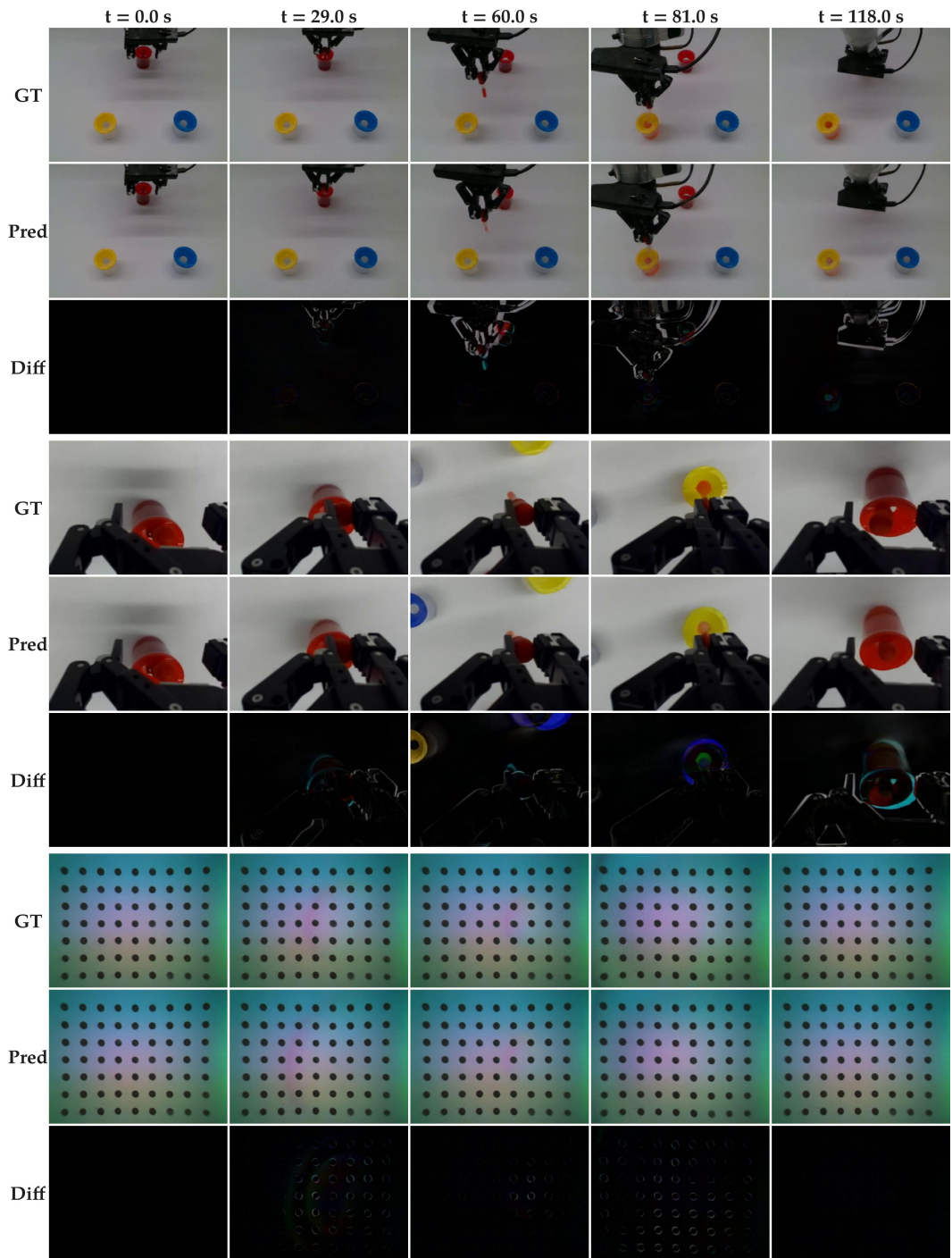


Figure 25: ViTaL Visuo-Tactile World Model Rollouts For Successfully Transferring Liquid to the Yellow Cup and Coming Back to the Red Cup. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

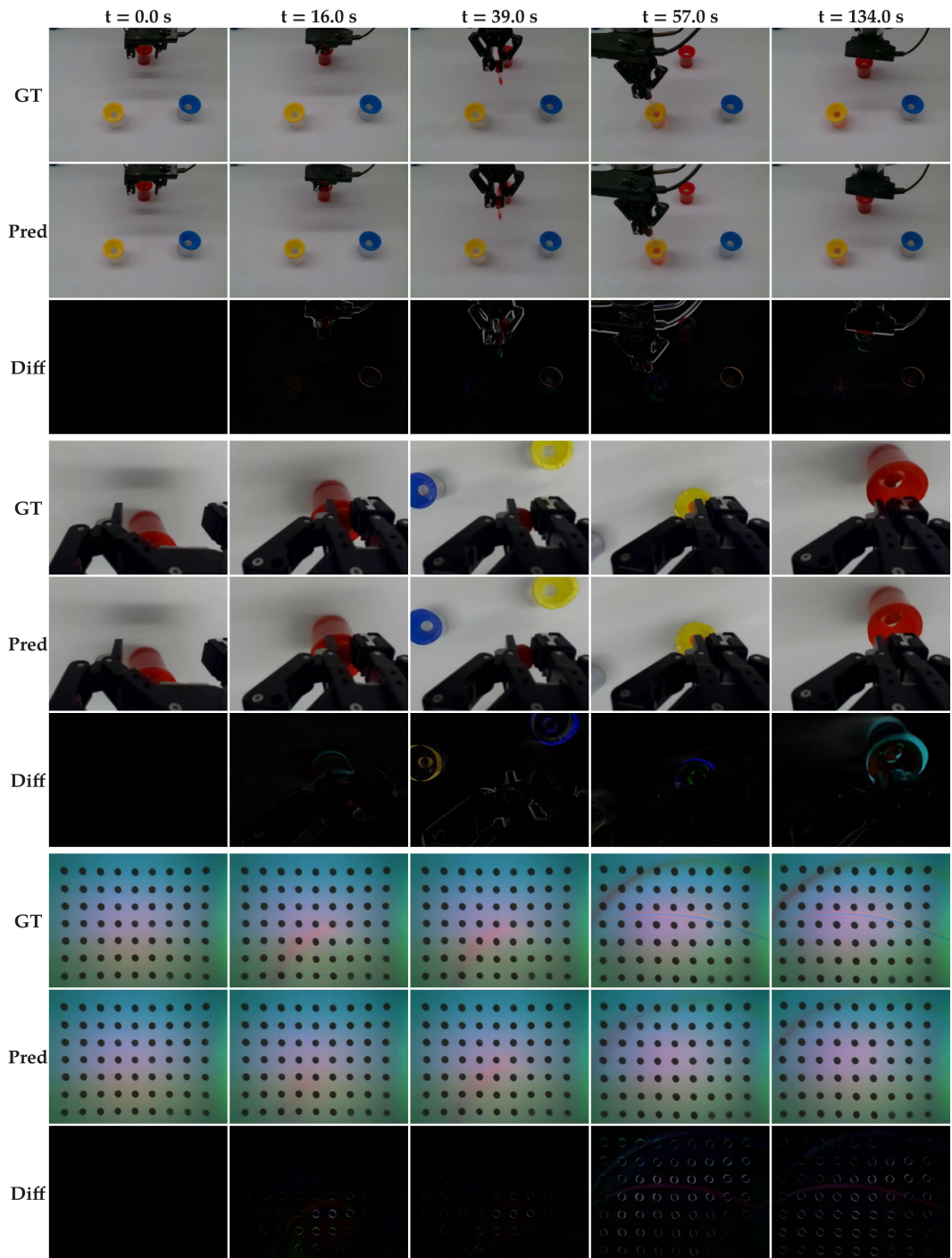


Figure 26: ViTaL Visuo-Tactile World Model Rollouts For Failing to Put the Dropper Back to the Red Cup After Squeezing into the Yellow Cup. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

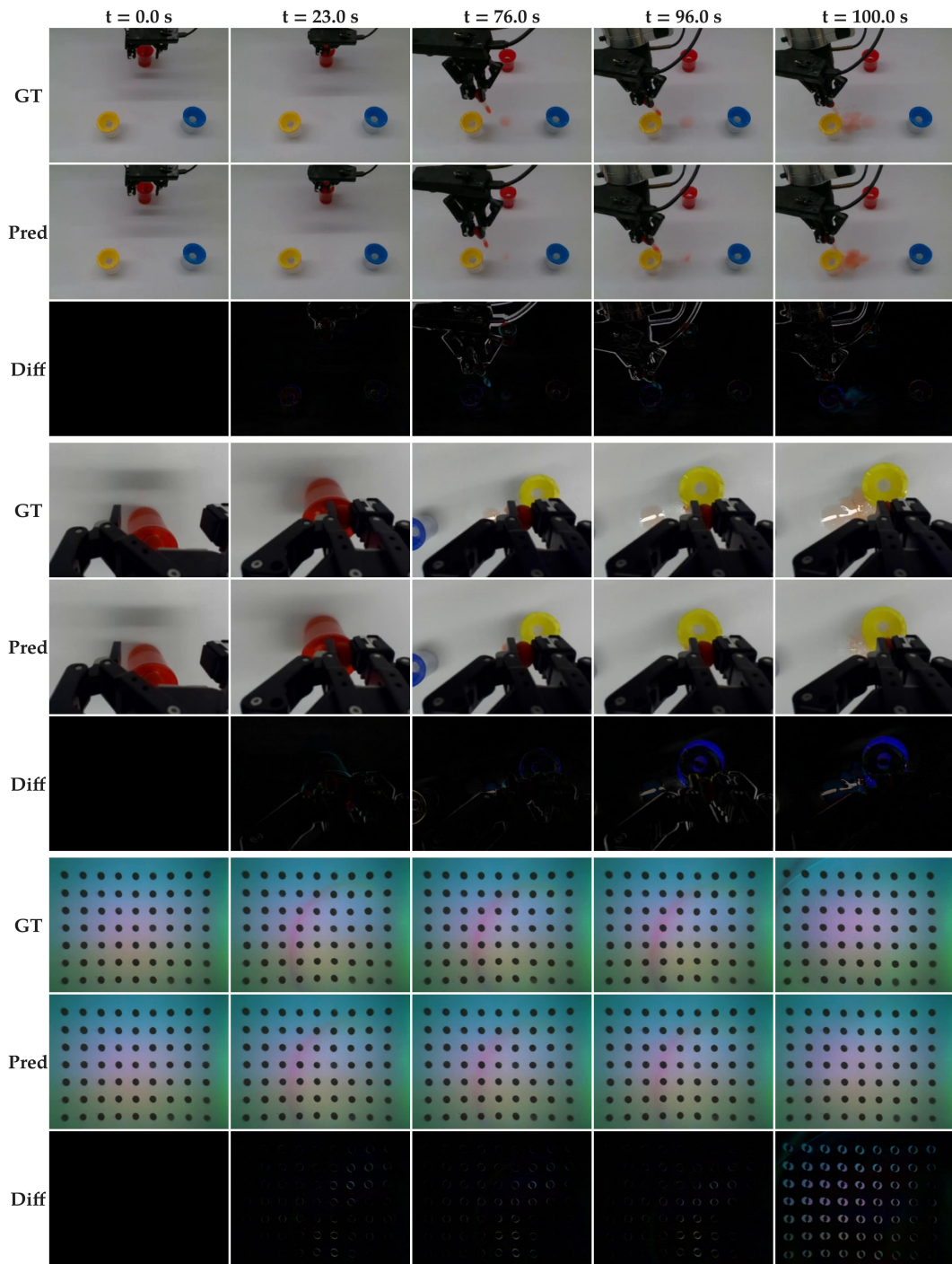


Figure 27: **ViTaL Visuo-Tactile World Model Rollouts For Spilling the Liquid Outside the Yellow Cup.** We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

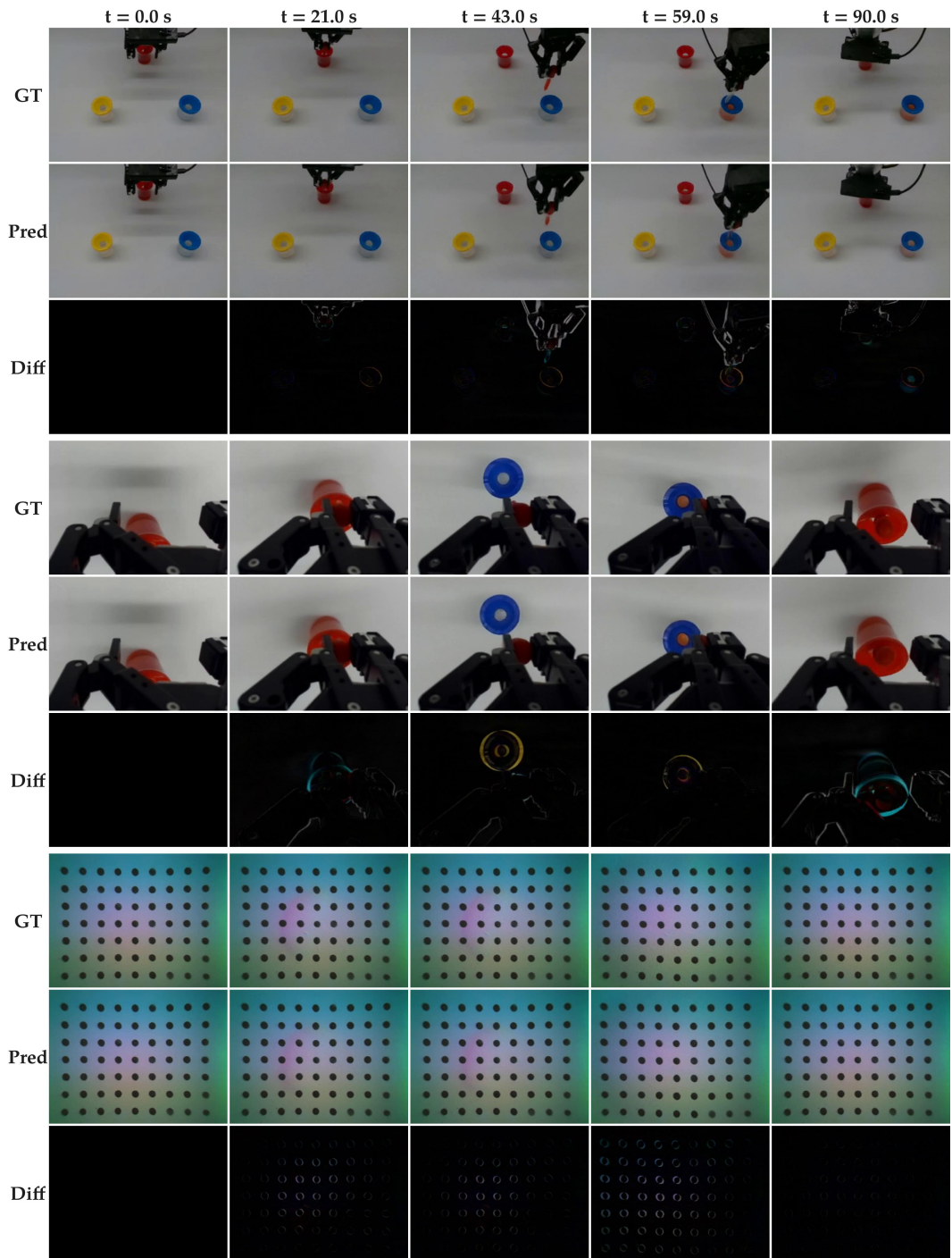


Figure 28: ViTaL Visuo-Tactile World Model Rollouts For Successfully Transferring Liquid to the Blue Cup and Coming Back to the Red Cup. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

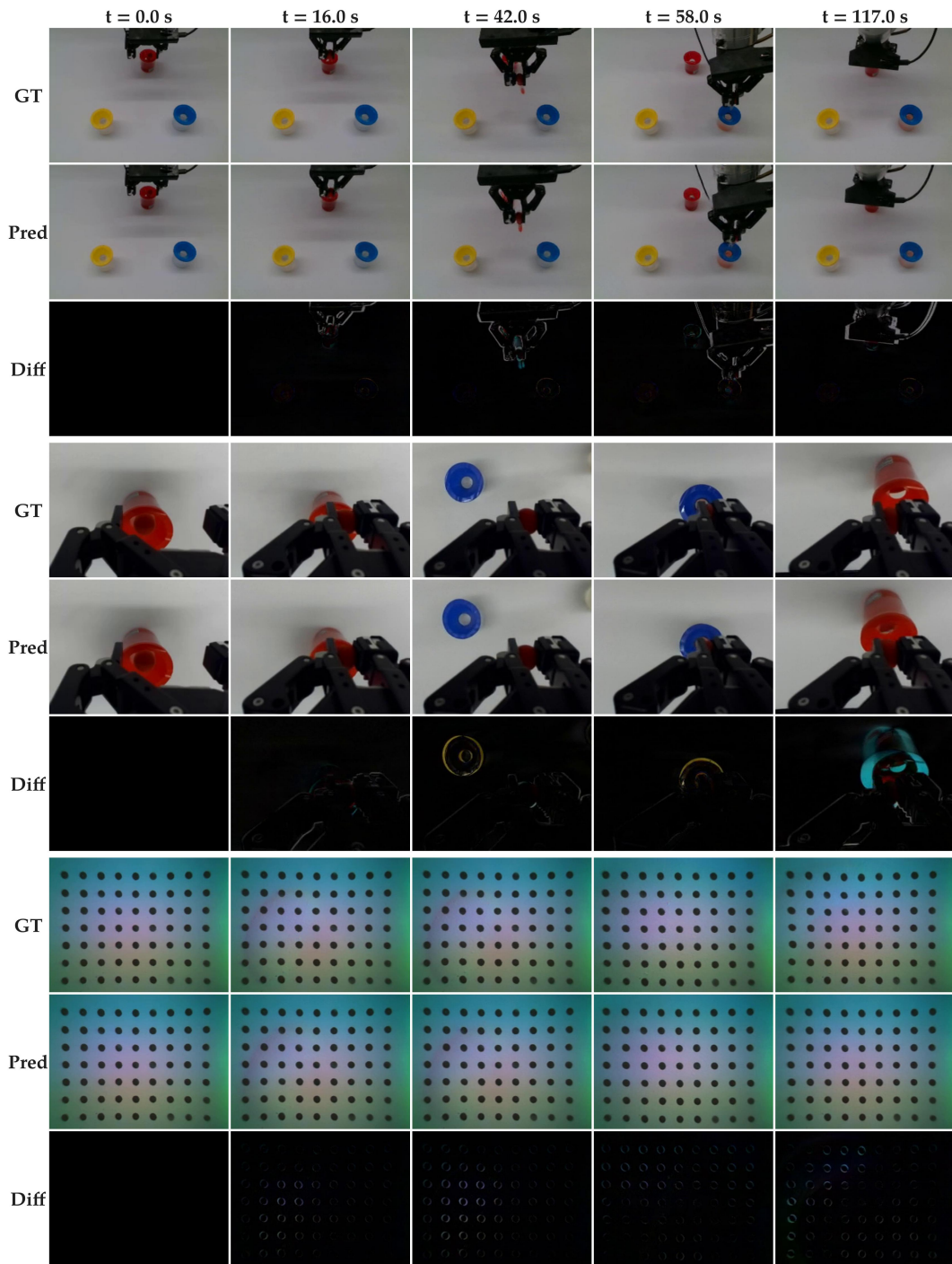


Figure 29: ViTaL Visuo-Tactile World Model Rollouts For Failing to Put the Dropper Back to the Red Cup After Squeezing into the Blue Cup. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

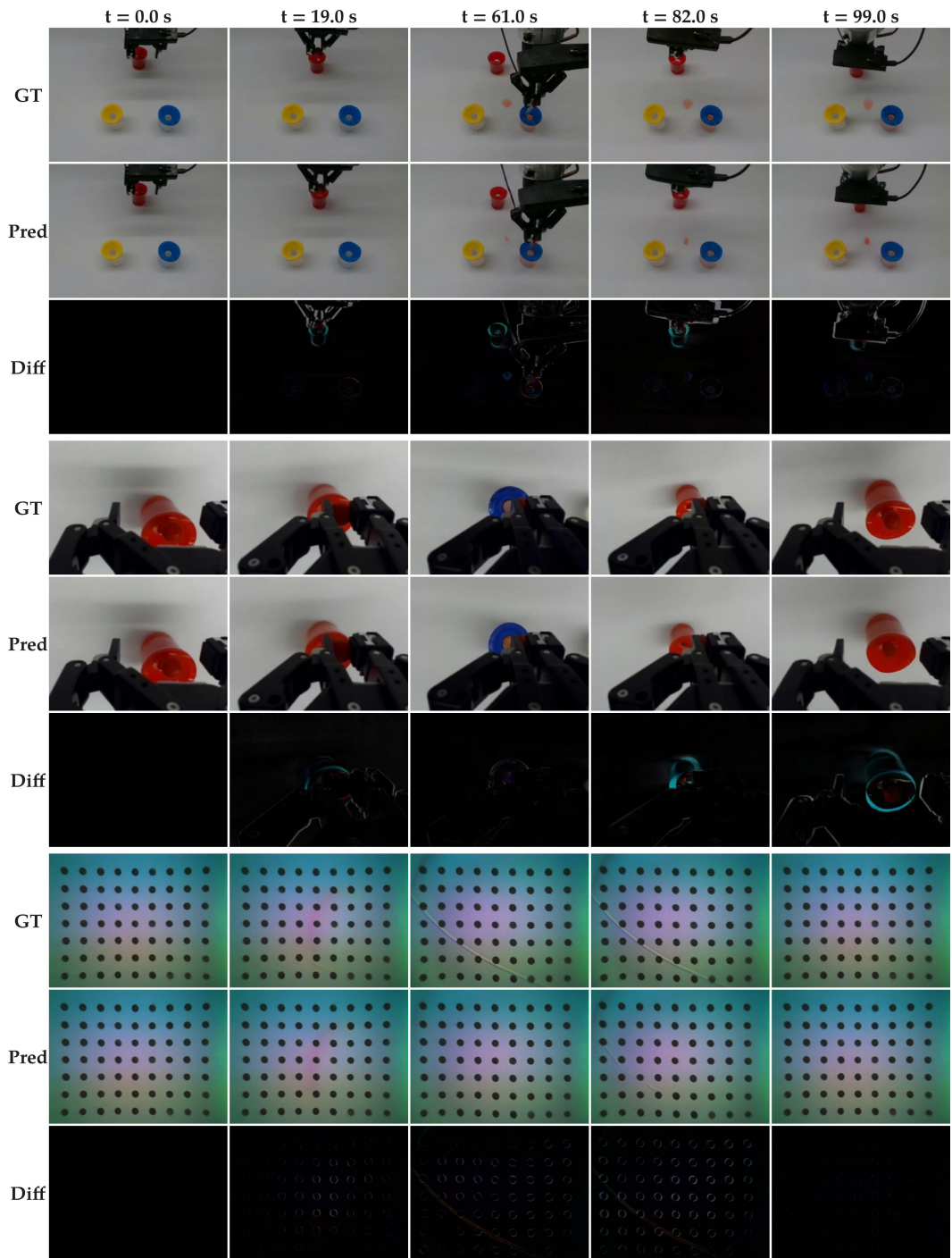


Figure 30: ViTaL Visuo-Tactile World Model Rollouts For Spilling the Liquid Outside the Blue Cup. We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.

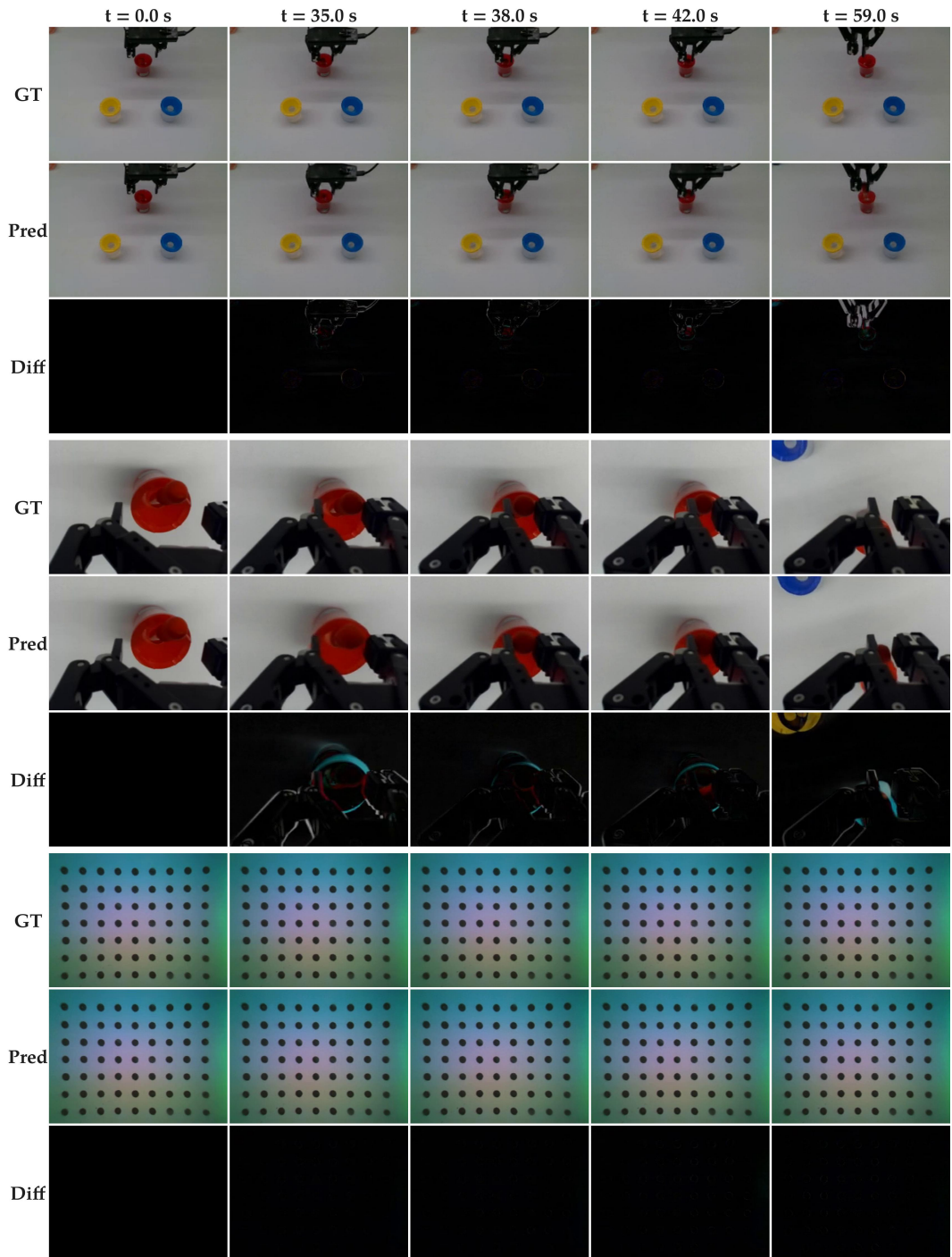


Figure 31: **ViTaL Visuo-Tactile World Model Rollouts For Failing to Have a Stable Grasp.** We show the ground-truth rollouts, predicted rollouts for front view (top), wrist view (middle) and the gelsight Mini tactile image (bottom) as well as the differences between ground-truth and predictions.