APPENDIX

A TRAINING VISUO-TACTILE WORLD MODEL

A.0.1 TRAINING DATASET

To train our multi-task visuo-tactile world model, we collect a dataset of teleoperated robot arm trajectories performing fundamental contact-rich manipulation actions, such as pick and place, pushing, and insertion. Our hardware setup consists of a table-top Franka Panda arm with an Allegro Hand as the end-effector and a Digit 360 sensor mounted on each fingertip. An exocentric view from a camera captures the global context of the robot's interaction with objects on the table.

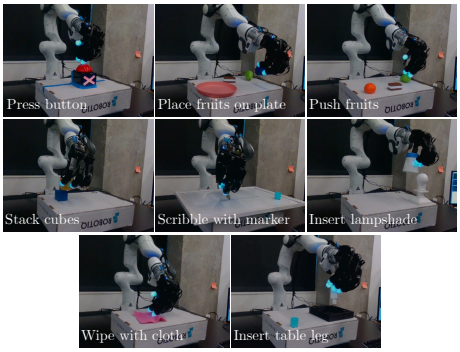

Figure 9: Multitask Vision-Tactile Dataset. Trajectories for training the world model collected via teleoperation, including both successful and failure sequences.

Through teleoperation, we collect a diverse set of trajectories, without discriminating between successes and failures, for eight distinct contact-rich tasks (see Fig. 9): pick and place on a plate, reach and press a button, push, wipe with a cloth, lampshade insertion, table leg insertion, cube stacking, and scribbling with a marker. For each task, we recorded successful and failure demonstrations. Each sequence contains multimodal data streams: proprioceptive information (wrist pose, joint positions), exocentric video from the camera, and video from each Digit 360 fingertip sensor. All data streams were synchronized using timestamps and downsampled to 6 FPS for training the world model. Our training dataset for V-WM and VT-WM consists of 124 demonstrations totaling 112k datapoints, with each demonstration averaging 40 seconds. For validation, we use 26 demonstrations spanning all tasks, comprising 17k datapoints.

A.1 TRAINING PARAMETERS

The model is optimized using AdamW (Loshchilov & Hutter, 2019) with parameters $\beta_1 = 0.9, \beta_2 = 0.95$ and a weight decay of 0.01. We use a learning rate scheduler with linear warmup for the first 10,000 gradient updates to a peak learning rate of $3e-4$, followed by cosine decay to $3e-7$ over a total of 80,000 updates. We use an effective batch size of 64 distributed over 32 A100 GPUs. We found that fine-tuning the Sparsh-X encoder was beneficial to account for sensor-specific variations, such as those arising from manufacturing tolerances and elastomer wear, while the Cosmos Tokenizer (Agarwal et al., 2025) was kept frozen during training. Our visuo-tactile world model has a total of 173M parameters, of which 96M were trained.

B CONTACT PERCEPTION WITH VISUO-TACTILE WORLD MODEL

We corroborate the world model's capacity to generate future states that are a reliable and predictable consequence of the given action conditioning. We study action controllability qualitatively by visualizing rollouts under simple, disentangled action commands: moving the end-effector along the Cartesian axes ($\pm x$, $\pm y$, $\pm z$) and opening/closing the hand. Actions conditioning is given to the VT-WM as deltas in the robot's proprioceptive state.

We observe in Fig. 10 that the VT-WM produces coherent rollouts aligned with the commanded actions. Translations along each axis result in consistent directional displacements of the end-effector in imagination (notice the reference frame in the figure), while hand open/close commands lead to corresponding changes in finger configurations. Notably, these behaviors emerge from the learned dynamics rather than explicit supervision of axis-aligned motion, indicating that the model internalizes the action-conditioned structure of the robot's kinematics. We compare ground-truth trajectories with VT-WM rollouts under the same action sequences and illustrate what the world model *imagines* in terms of contact.

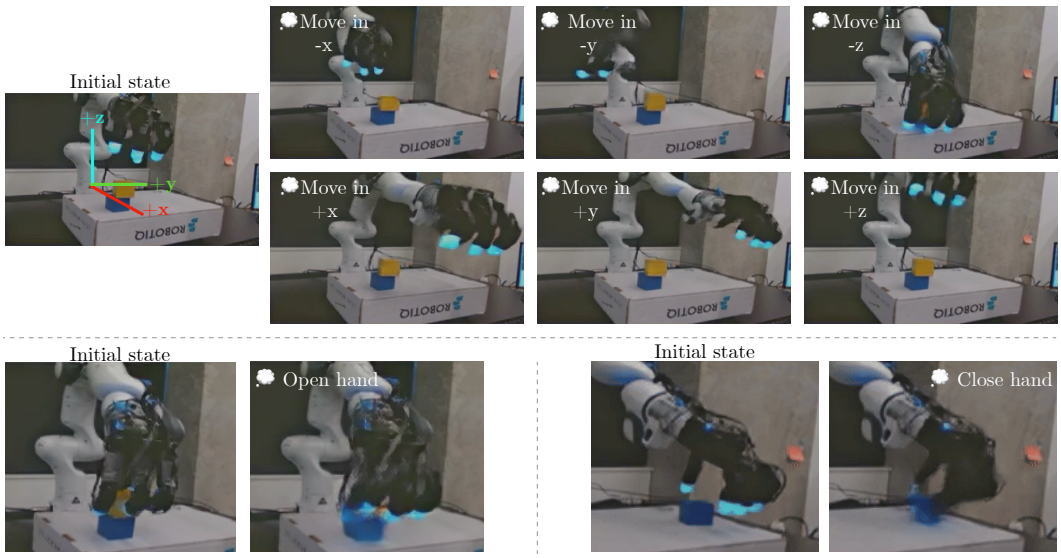

Figure 10: Visuo-Tactile World Model generates rollouts aligned with commanded actions along reference axes ($\pm x$, $\pm y$, $\pm z$) and for hand open/close.

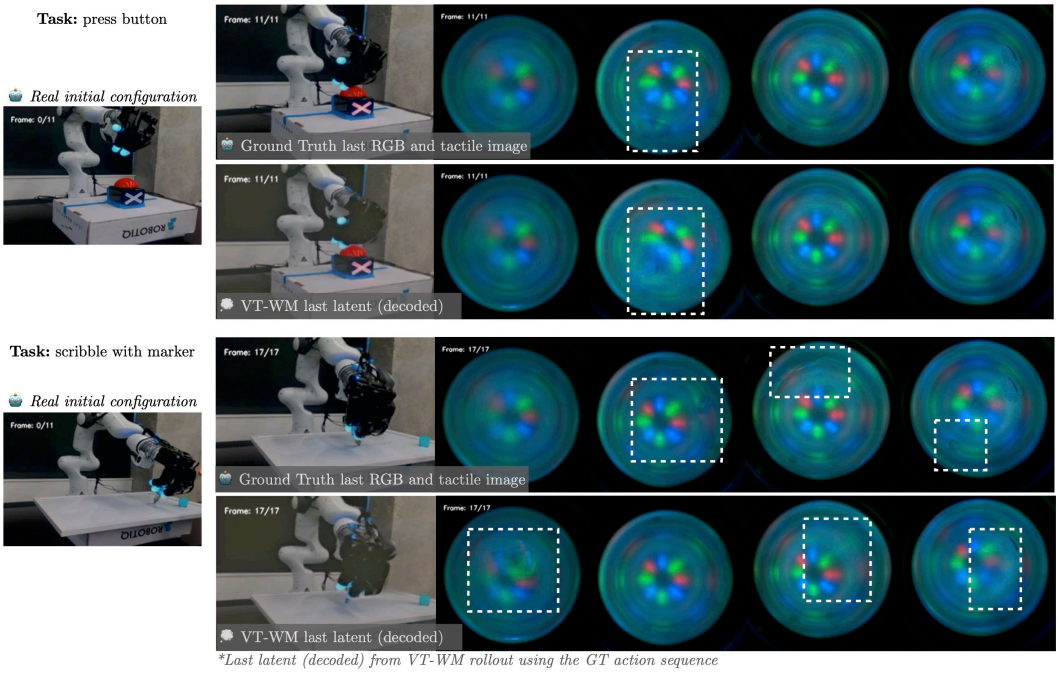

Figure 11: Visuo-Tactile World Model rollouts conditioned on ground-truth action sequences. Predicted visual states closely match the final RGB observations, while predicted tactile states capture plausible contact events and finger–object interactions.

To illustrate the predictive capability of the visuo-tactile world model, we evaluate rollouts conditioned on real robot action sequences. Specifically, we use held-out demonstrations from two tasks in our dataset: *press button* and *scribble with marker*. For each task, the VT-WM is queried autoregressively using the ground-truth sequence of control deltas.

Fig. 12 compares snapshots of a `Visuo-Tactile World Model` rollout for the *insert table leg* task, when grasping the object. Fig. 11 compares the final predicted states with the corresponding real-world outcomes. Since the model produces latent representations of future visual and tactile observations, we employ pretrained decoders to reconstruct these latents for visualization. Across both

tasks, the predicted visual states closely resemble the final RGB images of the real trajectories. The predicted tactile states also capture the key interaction events: although slight differences appear in the precise location of per-finger contacts, the rollouts consistently indicate whether contact occurs and depict plausible patterns of hand–object interaction. This demonstrates that the VT-WM, when guided by real action sequences, generates in imagination physically meaningful futures across both visual and tactile modalities.

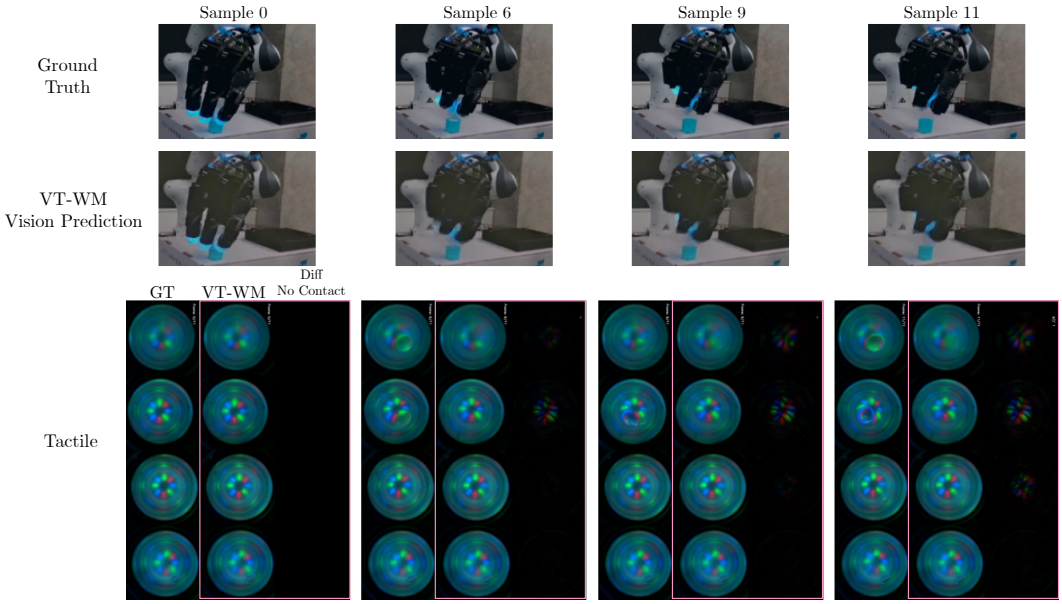

Figure 12: Snapshots of a Visuo-Tactile World Model rollout for *insert table leg* task conditioned on ground-truth action sequences for a 2s horizon. *Top:* ground truth vision state. *Middle:* predicted vision state across rollout. *Bottom:* ground truth tactile signatures, predicted tactile and its difference with respect the no contact state. Notice that fingers that are in contact match between ground truth and VT-WM predicted signatures.

By producing consistent visuo-tactile rollouts under real control sequences, VT-WM demonstrates the ability to represent both global visual context and local contact dynamics in a unified predictive framework useful for planning.

C Zero-shot planning with World Models

C.1 CEM algorithm for planning with World Models

The algorithm 1 performs planning in a world model (WM) imagination using the Cross-Entropy Method (CEM). Given a goal image and the current multimodal context (vision and tactile), the algorithm first encodes these inputs into latent representations. CEM is then used to optimize a sequence of actions over a finite prediction horizon by iteratively sampling action sequences (particles), rolling them out in the world model, and evaluating their predicted visual outcomes against the goal latent state using an ℓ_2 distance. The top-performing action sequences are used to update the mean and variance of the action distribution, refining the search over multiple iterations. After convergence, the best action sequence is executed on the robot, and the process can be repeated over several trials with updated context.

Next, we describe the goal of each of the tasks we use to evaluate the planning capabilities of the world models and discuss their results.

Reach Button: In this task, the robot must approach and press the center of a button starting from varied initial poses directly above it. Success therefore requires planning a sequence of actions that align the end-effector laterally with the button and then move downward to establish contact. Fig. 14 illustrated the plans produced by each world model using CEM rollouts in imagination,

Algorithm 1 Planning in WM imagination via CEM

Require: WM (world model)
Require: Goal Image X^{goal}_{rgb}
Require: Context (current state) X^0_{rgb}, X^0_{touch}
$f \leftarrow 6$ ▷ WM frequency
$H \leftarrow 2$ ▷ Prediction horizon in seconds
$P \leftarrow 36$ ▷ Number of particles for CEM algorithm
$N \leftarrow 10$ ▷ Number of iterations for CEM algorithm
$d \leftarrow 7$ ▷ Action dimensionality [X,Y,Z,roll,pitch,yaw,gripper]
max-trials $\leftarrow 3$ ▷ Number of calls to CEM algorithm
for trials $<$ max-trials **do**
  Update context (current state) X^0_{rgb}, X^0_{touch} ▷ Read from sensors
  $Z^{goal} \leftarrow$ vision-encoder(X^{goal}_{rgb}) ▷ Encode goal image
  $Z^0_{rgb} \leftarrow$ vision-encoder(X^0_{rgb}) and $Z^0_{touch} \leftarrow$ touch-encoder(X^0_{touch}) ▷ Encode context
  ▷ Initialize CEM action distribution parameters
  $\mu \leftarrow zeros(1, H * f, d)$ and $\sigma \leftarrow ones(1, H * f, d)$
  best-cost $\leftarrow \infty$
  best-action $\leftarrow None$
  for $n < N$ **do**
    ▷ Generate action particles
    actions $\leftarrow (\mu.repeat(N) + \sigma.repeat(N)) * rand(N, H * f, d)$
    $\hat{Z}^{1:H}_{rgb}, \hat{Z}^{1:H}_{touch} \leftarrow$ WM.rollout$(Z^0_{rgb}, Z^0_{touch}, actions)$ ▷ Rollout WM
    $costs \leftarrow \ell_2\left(Z^{goal}, \hat{Z}^H_{rgb}\right)$ ▷ Compute distance with target latent state
    elite-actions $\leftarrow actions[topk(costs)]$ ▷ Choose top 5 particles with lowest cost
    ▷ Update distribution parameters
    $\mu \leftarrow$ elite-actions$.mean()$ and $\sigma \leftarrow$ elite-actions$.std()$
    if $costs.min() <$ best-cost **then**
      best-action $\leftarrow actions[costs.argmin()]$ ▷ Found a new action that gets closer to goal
    end if
  end for
  Execute sequence of robot commands from best-action
end for

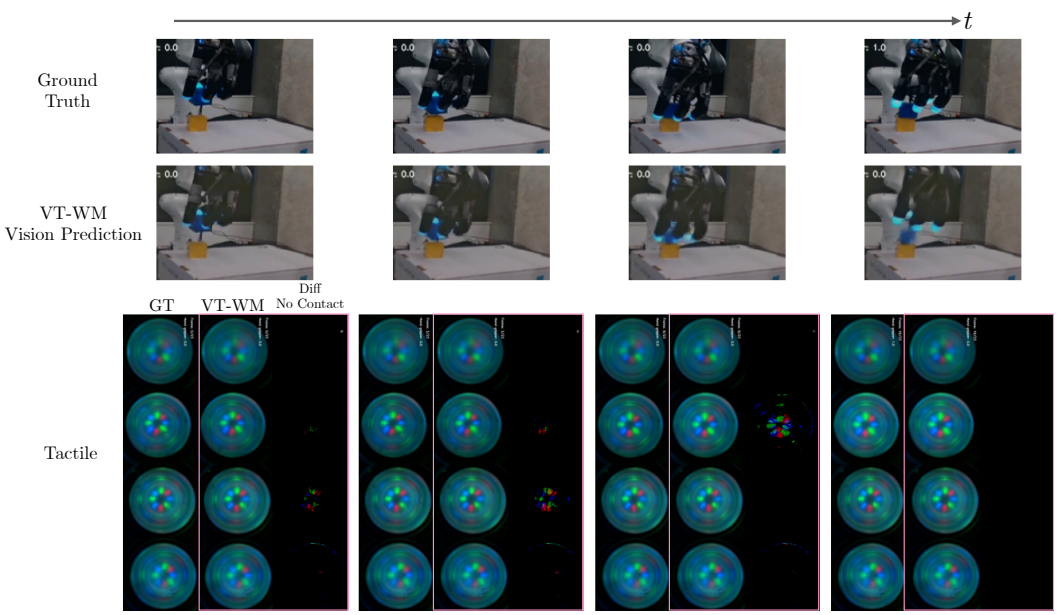

Figure 13: Snapshots of a Visuo-Tactile World Model rollout for *cube stacking* task. *Top:* ground truth vision state. *Middle:* predicted vision state across rollout. *Bottom:* ground truth tactile signatures, predicted tactile and its difference with respect the no contact state. Notice that fingers that are in contact match between ground truth and VT-WM predicted signatures.

alongside the corresponding executions on the real robot. We observe that both V-WM and VT-WM generate feasible trajectories that transfer zero-shot to the real system. This is expected since reaching primarily involves spatial reasoning and gross kinematic alignment, which vision alone can capture reliably.

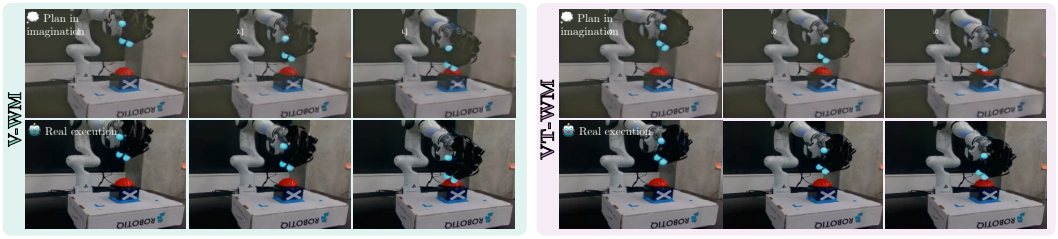

Figure 14: Reach Button task. Plans generated by V-WM and VT-WM with CEM in imagination (top) and their zero-shot executions on the real robot (bottom). Both models produce feasible trajectories, as reaching relies mainly on spatial reasoning and kinematic alignment.

Push Fruits: In this task, the robot hand begins directly in front of a target object that must be pushed downwards (toward the robot base). A successful plan requires maintaining persistent but gentle contact, allowing the object to slide across the table rather than topple.

Fig. 15 compares imagined rollouts and real executions for both models. While both V-WM and VT-WM produce plausible plans, we observe notable artifacts in the imagined rollouts, most prominently visual distortions of the green fruit when the hand occludes it. These artifacts are less pronounced in VT-WM, which better preserves the object's geometry in imagination. The deployment of the V-WM plan not only results in shorter object displacement but also lead to physical failures in execution, where the object occasionally topples instead of sliding without orientation changes.

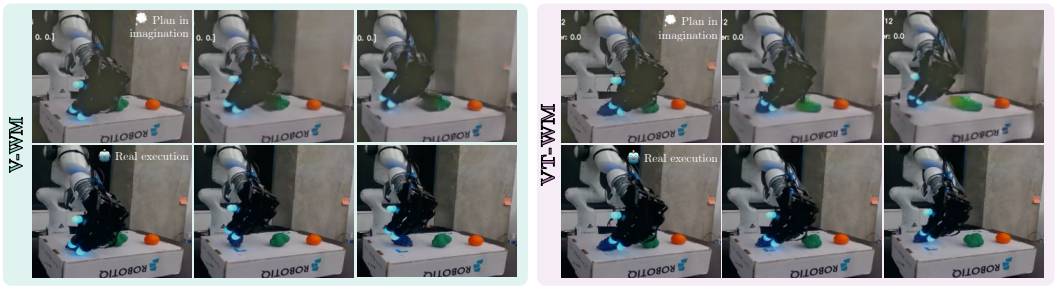

Figure 15: Push Fruits task. VT-WM preserves object geometry in imagination and transfers to stable sliding, while V-WM introduces distortions and often causes toppling in execution.

Reach & Push: This task requires a two-stage plan, first reaching the object to establish contact, then pushing it downward toward the robot base. Both subgoals are illustrated in Fig. 16, which shows the imagined plans and real executions.

In the V-WM rollout, the hand consistently hovers slightly above the object during the reach phase. As a result, the subsequent push proceeds without contact, and the execution on the real robot fails to move the object. By contrast, the VT-WM rollout explicitly brings the hand into contact during the reach, enabling the push plan to apply move the object effectively. When deployed, this produces the desired behavior, with both the reach and push subgoals successfully achieved. This highlights how tactile grounding resolves cases of visual aliasing, ensuring reliable contact in imagination and execution.

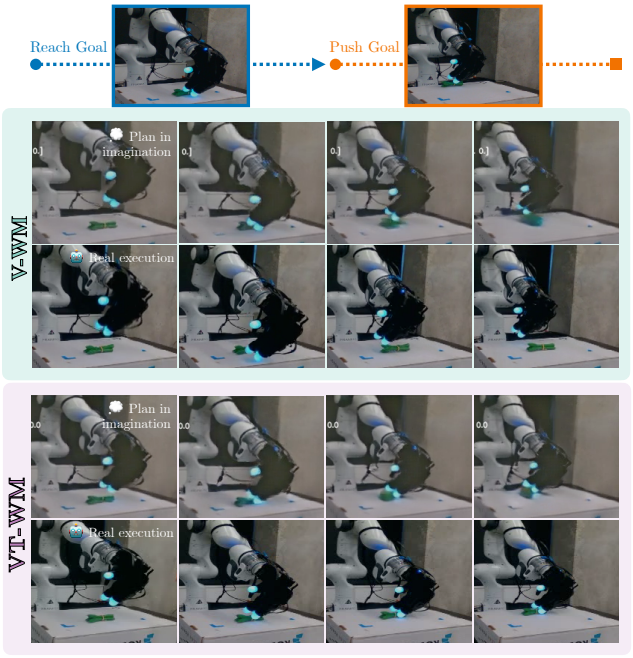

Figure 16: Reach & Push task. V-WM fails to establish contact, leading to ineffective pushes, while VT-WM ensures contact in imagination and execution, successfully completing both subgoals.

Wipe Cloth: This task consists of two subgoals, first reaching the cloth to establish contact, and then wiping it horizontally across the table. Both stages are illustrated in Fig. 17, which compares imagined rollouts with real executions.

In the V-WM rollout, the reach phase frequently results in the hand hovering slightly above the cloth, leading to ineffective wiping during execution. Even when a wiping trajectory is imagined, the visualizations exhibit noticeable artifacts such as geometric distortions of the cloth and hand. These artifacts reflect the model's uncertainty about contact dynamics and correspond to execution failures where the cloth barely moves.

In contrast, the VT-WM rollout shows clearer geometry and maintains consistent contact with the cloth in imagination. As a result, the subsequent wiping action produces a stable horizontal displacement of the cloth when deployed on the real robot. This example underscores the advantages of visuo-tactile world model in tasks that require sustained contact to manipulate objects.

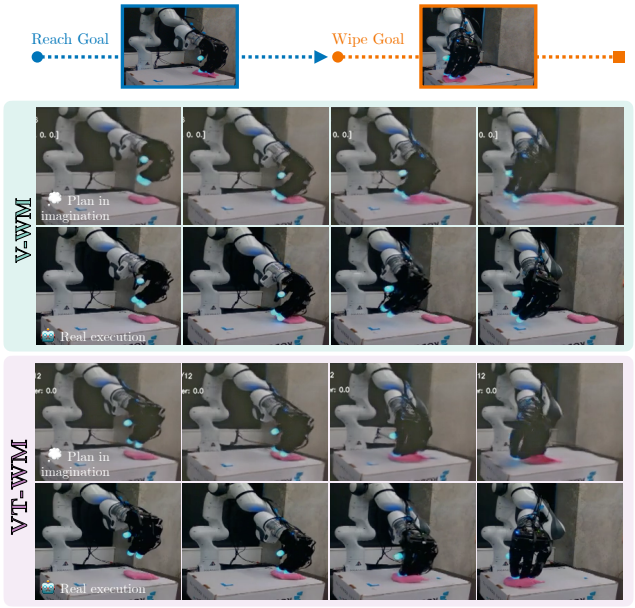

Figure 17: Wipe Cloth task. V-WM rollouts show artifacts and miss contact, leading to ineffective wiping, while VT-WM maintains contact and produces consistent cloth displacement in execution.

Stack Cubes: This task requires transporting a blue cube to the stacking location and then placing it stably on top of a yellow cube. Both subgoals are illustrated in Fig. 18, which shows imagined rollouts alongside real executions.

While the V-WM generates reasonable transport trajectories, failures arise during placement. In imagination, the cube intermittently disappears from the hand, revealing artifacts that indicate the model is tracking only the hand–scene geometry (e.g., alignment with the target yellow cube) rather than maintaining a consistent hand–object relationship. This disconnect leads to execution failures, where the cube is not reliably placed.

However, visuo-tactile world model accurately captures the object–hand interaction, throughout both transport and placement, the cube remains consistently represented in the rollout. When transferred zero-shot to the real robot, these plans result in stable stacking, highlighting the advantage of VT-WM for tasks that demand precise, contact-rich manipulation.

D LIMITATIONS

While our results demonstrate clear benefits of visuo-tactile world model, following limitations point to promising directions for future research. First, our tactile modality is limited to vision-based tactile sensing, specifically the Digit 360 sensor. However, the VT-WM framework applies to other tactile modalities as well, provided that an appropriate pretrained tactile encoder is available. Second, evaluation of contact perception uses unseen robot trajectories but only within tasks from the training distribution, leaving open the question of how well the model generalizes to entirely novel

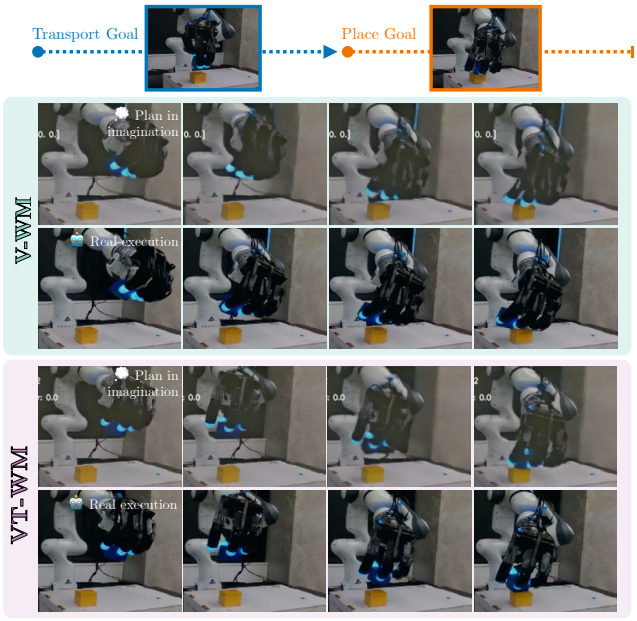

Figure 18: Stack Cubes task. V-WM imagined rollouts during planning lose track of the cube for the placement subgoal, leading to failed stacking, while VT-WM preserves hand–object interaction and transfers to successful stacks.

manipulation tasks or object characteristics. Third, our planning experiments randomize initial robot states but are limited to the same scene and objects, without testing generalization to objects with different visual or physical properties such as size, shape, or color. Planning with world models via CEM remains computationally expensive, as it requires generating many autoregressive rollouts per particle. This leads to open-loop execution in trajectory chunks, unlike classical policies that operate in closed-loop at higher control frequencies. Finally, our comparison against a single task behavior cloning (BC) policy does not fully rule out the possibility that a multi-task BC policy could also exhibit strong data efficiency for the new task.

E Additional Notes

About the use of large language models: Large Language Models (LLMs) were used exclusively to assist with grammar correction and refinement of writing style (flow, academic tone, and conciseness), based on drafts authored by the researchers. LLMs were not employed for data generation, or in any stage of the proposed model's design, training, or evaluation.