Presentation of ViViDex: Learning Vision-based Dexterous Manipulation from Human Videos

Victorin Turnel, Thomas Winninger
Master MVA, ENS Paris-Saclay

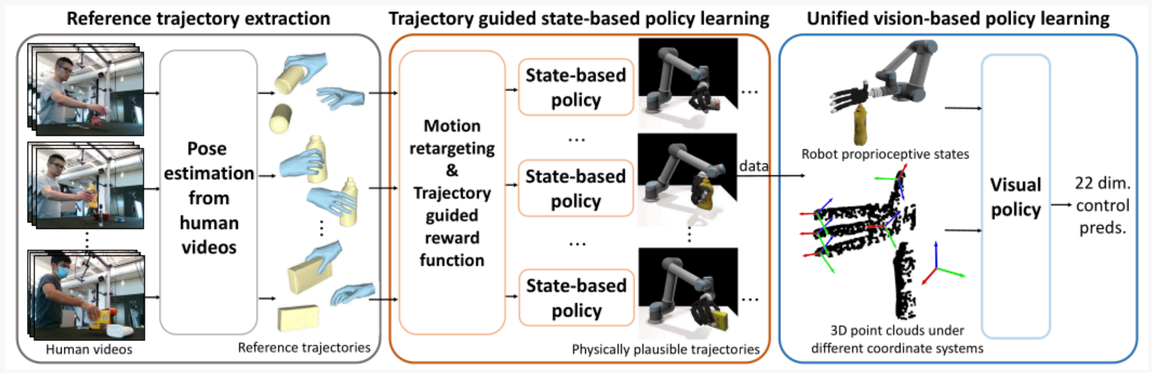
Context

Dexterous manipulation with multi-fingered robot hands remains a significant challenge in robotics, particularly when manipulating a variety of objects in diverse poses. A major limitation of prior works is their reliance on privileged information, such as ground-truth object states or CAD models. Obtaining such data is often unrealistic in real-world scenarios due to sensor limitations.

To address these challenges, ViViDex consists of three main modules:

- Reference trajectory extraction
- Trajectory-guided state-based policy learning
- Unified vision-based policy learning

Method



Reference trajectory extraction

The goal of this module is to extract hand and object poses from video demonstrations. The method relies on MANO to represent the poses and shapes of the hand.

$$\min_{q_r^t} \sum_{t=1}^{T} \left\| \hat{x}_{rj}^t(q_r^t) - \psi_{hj}^t \right\|_2^2 + \alpha \| q_r^t - q_r^{t-1} \|$$

The first term aims to minimize the L^2 difference between the 3D locations of the robot joints and the human hand. The second term acts as a regularization to prevent sudden changes in the robot pose over time.

Trajectory-guided state-based policy learning

The previously generated trajectories are visually plausible but are not necessarily physically plausible. To address this challenge, a state-based policy is trained using RL to recover physically plausible trajectories, using the reference trajectories as a reward function.

The RL training process is divided into two stages. The first step, called the "pre-grasp" stage, focuses on learning a coarse motion to quickly approach the object without establishing physical contact. To achieve this, the following reward function is defined:

$$R_p = \sum_{t=1}^{T_p} 10 \cdot \exp\left(-10 \cdot \| x_{rt}^t(q_r^t) - \hat{x}_{rt}^t \|_2^2\right)$$

The manipulation stage begins when the robot successfully reaches its pre-grasp configuration. In this stage, the objective is to refine the trajectory with finer and more constrained motions to bring the object to the desired configuration. This step is guided by the following reward function:

$$R_m = \sum_{t=T_p+1}^{T_r} \lambda_1 R_m^h + \lambda_2 R_m^o + \lambda_3 \mathbf{1}_{\text{cont}} + \lambda_4 \mathbf{1}_{\text{lift}}$$

Unified vision-based policy learning

To bridge the gap to real-world application, privileged state access is replaced with a vision-based policy relying on 3D point clouds and proprioception. Using an expert state-based policy to generate training trajectories, raw point clouds are transformed into the target and robot joint coordinate systems (palm, fingertips) before encoding them via PointNet.

These extracted visual features serve as inputs to predict robot commands. Two imitation learning strategies were implemented: direct prediction via Behavior Cloning and a generative approach using a 3D Diffusion Policy, both trained to mimic the expert trajectories.

Limitations & Improvements

Unified reward loss for state-based policy

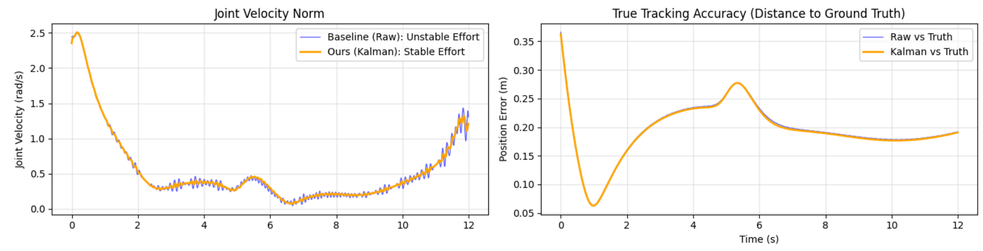
The current method artificially divides the task into two distinct phases: the approach phase (R_p) and the manipulation phase (R_m). These two stages are rigidly separated by a configuration threshold that triggers the switch from one step to the other. This binary separation renders the overall motion unsmooth by creating two independent and "specialized" movements.

$$R_{\text{unified}(t)} = R_{\text{reach}(t)} + w(d_t) \cdot R_{\text{hand\_shape}(T_p)} + \mathbf{1}_{\text{contact}} \cdot R_{\text{object\_track}(t)}$$

To create a unified reward function, we introduce a transition term that facilitates a smooth progression between the two stages. Specifically, this term enables the hand to progressively adopt the optimal pose for the subsequent manipulation as the distance between the hand and the object decreases.

Improving Signal Quality via Kalman Filtering

The standard optimization framework operates directly on raw pose estimations with minimal regularization ($\alpha \approx 0.004$), causing the robot to track high-frequency sensor noise (jitter) and vibrate continuously. To solve this, we introduce a Linear Kalman Filter as a pre-processing layer. By decoupling motion intention from measurement noise before optimization, the filter provides smoothed target trajectories. This effectively eliminates mechanical instability.



The Kalman filter effectively stops motor vibrations caused by sensor noise. This ensures the robot remains physically stable. However, its necessity is debatable. Since the State-Based Policy is already trained to produce smooth and plausible movements, it might naturally filter this noise. Consequently, adding this explicit filtering layer could be redundant.

Curriculum Learning and Physical Domain Randomization

The current "flat" training strategy has two major drawbacks:

- *Optimization Inconsistency*: The network is forced to learn how to handle complex non-convex geometries (like the handle of a mug) simultaneously with simple geometries. This steep learning curve can lead to sub-optimal convergence or "forgetting" of simple tasks.
- *Sim-to-Real Gap*: The lack of variation in physical parameters (friction, mass, damping) makes the policy brittle when deployed on a real robot, where physical properties inevitably differ from the simulator settings.

Instead of shuffling all object data randomly, we organize the training data by geometric complexity : Simple Convex Synthetic Objects → Simple Convex Real Objects → Complex Non-Convex Real Objects → Complex Non-Convex Synthetic Objects.

To prepare the vision-based policy for real-world deployment, we generate the training data (rollouts from the state-based policy) by varying the physical properties of the simulation environment: Object mass, inertia, surface friction.

Consistent object localization, appearance descriptors, and 3D geometric cues for the vision-based policy

The original ViViDex pipeline relies on 3D scene point-cloud features and a learned pose estimator, which can be noisy, especially when using a single video. To strengthen this pipeline, we propose to add a perception module to enrich the visual observations used during the trajectory extraction and the RL phase.

Object segmentation and identification can be used to extract only the relevant information (hand and object position), and remove everything else.

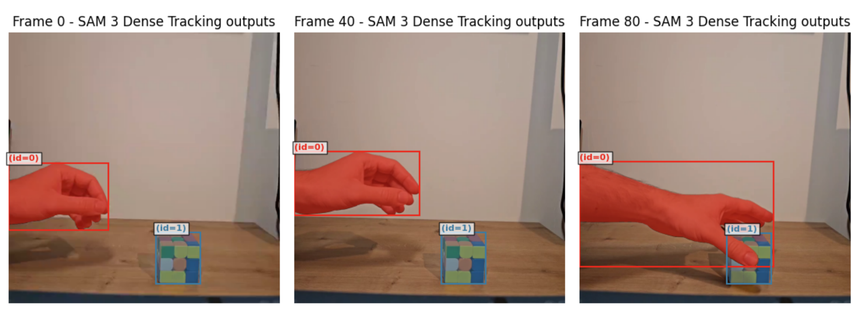


Figure 2: Segmentation and tracking with SAM v3 [1]. The model is provided the video, along with textual clues, in this case "Hand and object". The textual clues can also be used to avoid tracking a specific object.

Dense per-pixel feature embeddings can provide useful information about the physics of the objects, and may help generalisation.

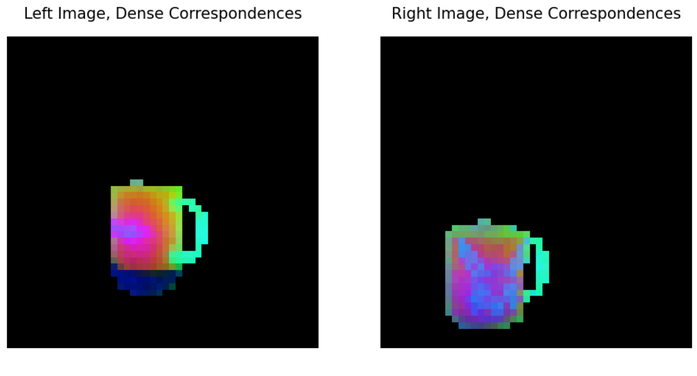


Figure 3: Dense matching with DINOv3 [2].

Conclusion

The ViViDex framework is a promising approach to vision-based dexterous manipulation from human videos as it does not require huge amount of labelled data as input. Using an unsupervised perception module may help leveraging unlabelled internet videos to scale and generalize the manipulation skills. Moreover, this approach holds significant potential for further optimization, notably through the refinement of RL loss functions and the integration of semantic scene information to enhance interaction robustness.

References

[1] N. Carion *et al.*, "SAM 3: Segment Anything with Concepts." [Online]. Available: https://arxiv.org/abs/2511.16719

[2] O. Siméoni *et al.*, "DINOv3." [Online]. Available: https://arxiv.org/abs/2508.10104

victorin.turnel@ens-paris-saclay.fr, thomas.winninger@telecom-sudparis.eu