LATENT DIFFUSION PLANNING FOR IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent progress in robotic imitation learning has been enabled by policy architectures that scale to complex visuomotor tasks, multimodal distributions, and large datasets. However, these methods rely on supervised learning of actions from expert demonstrations, which can be challenging to scale. We propose Latent Diffusion Planning, which forecasts future states as well as actions via diffusion. This objective can scalably leverage heterogeneous data sources and provides a denser supervision signal for learning. To plan over images, we learn a compact latent space through a variational autoencoder. We then train a planner to forecast future latent states, and an inverse dynamics model to extract actions from the plans. As planning is separated from action prediction, LDP can leverage suboptimal or action-free data to improve performance in low demonstration regimes. On simulated visual robotic manipulation tasks, LDP outperforms state-of-the-art imitation learning approaches as they cannot leverage such additional data.[1]

1 INTRODUCTION

Combining large-scale expert datasets and powerful imitation learning policies has been a promising direction for robot learning. Recent methods using transformer backbones or diffusion heads (Octo Model Team et al., 2024; Kim et al., 2024; Zhao et al., 2024; Chi et al., 2023) have capitalized on new robotics datasets pooled together from many institutions (Khazatsky et al., 2024; Open X-Embodiment Collaboration et al., 2023), showing potential for learning generalizable robot policies. However, this recipe is fundamentally limited by data, as robotics demonstration data is limited and expensive to collect. While it is often easier to collect in-domain data that is suboptimal or action-free, these methods are not designed to use such data, as they rely on directly modeling optimal actions.

Prior work in reinforcement learning has explored using heterogeneous data sources. Approaches that can be scaled to the imitation learning setting include conditioning the policy on optimality (Chen et al., 2021), and relabeling action-free trajectories using an inverse model (Baker et al., 2022). However, many of these approaches have not been shown to be competitive with state-of-the-art robotic imitation learning (Mirchandani et al., 2024). Recent work in robotics has leveraged heterogeneous data for pretraining via representation learning (Radosavovic et al., 2023; Wu et al., 2023b; Cui et al., 2024). However, only using the data for representation learning is limited, as it does not necessarily improve the planning capabilities of the method. In this work, we investigate how a simple planning-based method can leverage heterogeneous data in a principled way be decoupling forecasting future states from extracting actions.

We propose Latent Diffusion Planning (LDP), which learns a planner that can be trained on data does not require actions; and an inverse dynamics model that can be trained on data that may be suboptimal. While prior planning-based works (Du et al., 2023a; Black et al., 2023) improve high-level decision making by producing subgoals, we focus on forecasting a dense trajectory of latent states as an alternative method for imitation learning. As diffusion objectives proved to be effective for imitation learning (Chi et al., 2023), we use diffusion for both forecasting state and actions, which enables competitive performance. LDP plans across latent image embeddings, scaling up gracefully

[1] We include visualizations of plans and rollouts in `https://sites.google.com/view/latent-diffusion-planning/home`

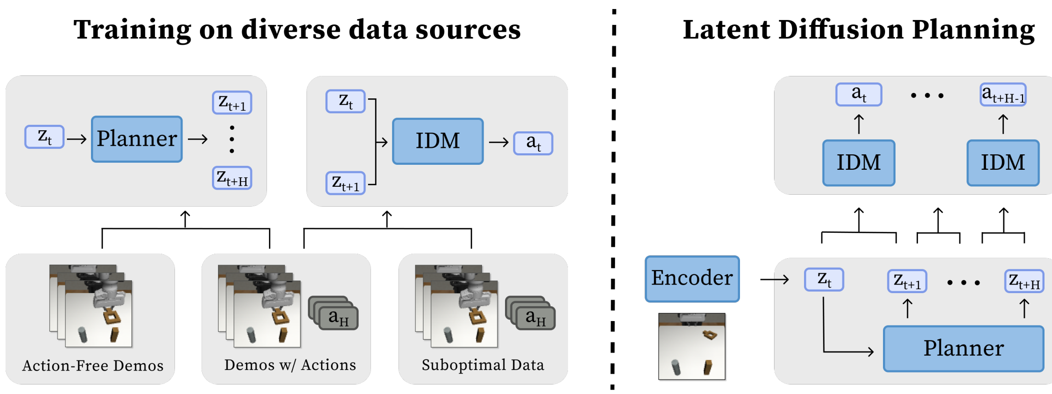


Figure 1: Latent Diffusion Planning. *Left*: LDP separates the control problem into forecasting future states with a diffusion-based planner, and extracting actions with a diffusion-based inverse dynamics model (IDM). This design enables training on heterogeneous sources of data, including suboptimal data and action-free data. *Right*: Unlike action imitation methods such as diffusion policy, LDP is based on forecasting a dense temporal sequence of latent states as well as actions. Using powerful diffusion models for both of these objectives enables LDP to have competitive performance to state-of-the-art imitation learning. Further, unlike prior work on forecasting subgoals, LDP predicts a dense temporal sequence of latent states, which enables scalable closed-loop planning.

to vision-based domains without the computational complexities of video generation. First, it trains a variational autoencoder with an image reconstruction loss, producing compressed latent embeddings. Then, it learns an imitation learning policy through two components: (1) a planner, which consumes demonstration state sequences, which may be action-free, and (2) an inverse dynamics model, trained on in-domain, possibly suboptimal, environment interactions. To maximally capture expressivity, both the planner and inverse dynamics models are implemented as diffusion models. Furthermore, our method is closed-loop and reactive, as planning over latent space is much faster than generating visually and physically consistent video frames.

In summary, our main contributions are threefold:

- We propose a novel imitation learning algorithm, Latent Diffusion Planning, a simple, diffusion planning-based method comprised of a learned visual encoder, latent planner, and an inverse dynamics model.
- We show that Latent Diffusion Planning can be trained on suboptimal or action-free data, and improves from learning on such data in the regime where demonstration data is limited. LDP can leverage such data better than prior work based on optimality conditioning or representation learning.
- We experimentally show that our method outperforms prior planning-based work by leveraging temporally dense predictions in a latent space, which enables closed-loop planning.

2 Related Work

Imitation Learning in Robotics. One popular approach to learning robot control policies is imitation learning, where policies are learned from expert-collected demonstration datasets. This is most commonly done via behavior cloning, which reduces policy learning to a supervised learning objective of mapping states to actions. Recently, Diffusion Policy (Chi et al., 2023) and Action Chunking with Transformers (Zhao et al., 2023) have shown successful results in complex manipulation tasks using action chunking and more expressive architectures. Similarly, Behavior Transformer (Shafiullah et al., 2022) and VQ-BeT (Lee et al., 2024) have focused on improving the ability of policies to capture multimodal behaviors. In this work, we focus on forecasting a sequence of future states instead of actions, and use diffusion to capture multimodal trajectories.

Learning from Unlabelled Suboptimal and Action-Free Data. Learning from suboptimal data has long been a goal of many robot learning methods, including reinforcement learning. A typical approach is offline reinforcement learning, which considers solving a Markov decision process from an offline dataset of states, actions, and reward (Levine et al., 2020; Kumar et al., 2020; Kostrikov et al., 2021; Hansen-Estruch et al., 2023; Yu et al., 2022). Particularly relevant are the approaches that use supervised learning conditioned on rewards (Schmidhuber, 2019; Kumar et al., 2019a; Chen et al., 2021). In this work, we want to leverage suboptimal, reward-free data, such as play data or failed trajectories. In addition, we would like to avoid the additional complexity of annotating the data with rewards or training a value function which the offline RL methods rely on.

Autonomous imitation learning methods seek to self-bootstrap from a pretrained imitative policy. Typically, these methods assume learning from online, autonomous rollouts and reward labels from trained classifiers or vision-language models (Konstantinos Bousmalis* & Heess, 2023; Zhou et al., 2024b; Mirchandani et al., 2024). Unlike these works, we assume access to a static, offline dataset, and we do not label the dataset with pseudo-rewards.

Several works have also addressed learning from action-free data, such as using inverse models (Torabi et al., 2018; Baker et al., 2022), latent action models (Edwards et al., 2019; Schmeckpeper et al., 2020; Bruce et al., 2024), or representation learning (Radosavovic et al., 2023; Wu et al., 2023b; Cui et al., 2024). In this work we focus on a simple recipe for robotic imitation learning that is naturally able to leverage action free data through state forecasting.

Diffusion and Image Prediction in Robot Learning. Diffusion models, due to their expressivity and training and sampling stability, have been applied to robot learning tasks. Diffusion has been used in offline reinforcement learning (Hansen-Estruch et al., 2023) and imitation learning (Chi et al., 2023). Diffuser (Janner et al., 2022) learns a denoising diffusion model on trajectories, including both states and actions, in a model-based reinforcement learning setting. Decision Diffuser (Ajay et al., 2023) extends Diffuser by showing compositionality over skills, rewards, and constraints, and instead diffuses over states and uses an inverse dynamics model to extract actions from the plan. Due to the complexity of modeling image trajectories, Diffuser and Decision Diffuser restrict their applications to low-dimensional states.

To scale up to diffusing over higher-dimensional plans, UniPi (Du et al., 2023a; Ko et al., 2023) adapts video models for planning. Unlike works that rely on foundation models and video models for planning (Du et al., 2023b; Yang et al., 2024; Zhou et al., 2024a), our method avoids computational and modeling complexities of generative video modeling by planning over latent embeddings instead.

Previous works have used world models to plan over images in a compact latent space (Hansen et al., 2024; Hafner et al., 2019; 2020). In contrast with these works, we focus on single task imitation instead of reinforcement learning.

Many prior works argued that state forecasting objectives are uniquely suitable for robotics to improve planning quality with trajectory optimization or reinforcement learning Finn & Levine (2017); Yang et al. (2023), by using the model directly to plan future states Du et al. (2023b;a), as well as representation learning (Wu et al., 2023a; Radosavovic et al., 2023). We follow this line of work by proposing a planning-based method competitive to state-of-the-art robotic imitation learning that can leverage heterogeneous data sources.

3 Background

Diffusion Models Diffusion models are likelihood-based generative models that learn an iterative denoising process from a Gaussian prior to a data distribution (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020). Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) optimizes a variational lower bound on data likelihood, derived in a similar way to variational autoencoders (Kingma & Welling, 2014; Rezende et al., 2014). DDPMs are trained to reverse a single noising step, formally:

$$\mathcal{L}_{\text{DDPM}}(\phi, \mathbf{z}) = \mathbb{E}_{t,\epsilon}[||\epsilon_\phi(\alpha_t \, \mathbf{z} + \sigma_t \epsilon) - \epsilon||^2] \tag{1}$$

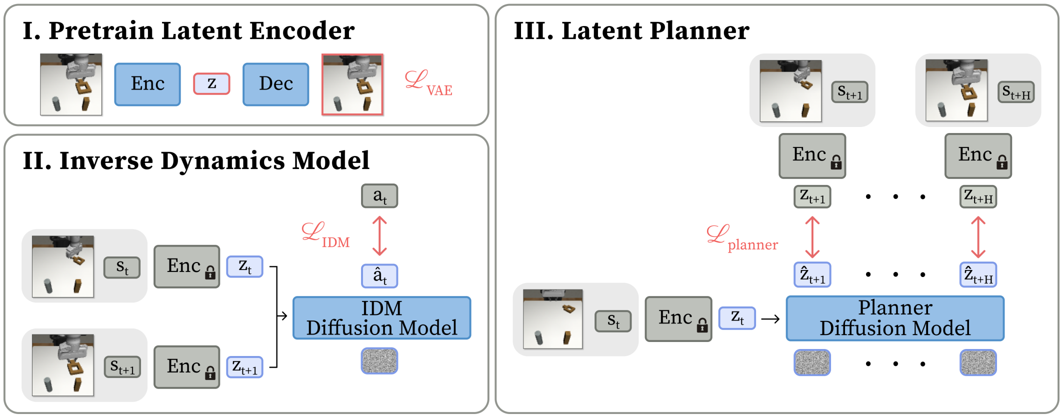


Figure 2: The architecture for Latent Diffusion Planning. *I.* We train a variational autoencoder on in-domain data to compress images into latents z. This allows for scalable closed-loop planning in the latent space. *II.* We train a inverse dynamics model (IDM) with a diffusion objective to directly extract the actions that will be used for control from pairs of latent states. *III.* We train a powerful latent diffusion model to forecast a chunk of future latent states. The planner and the IDM are used together to produce an action chunk, similar to Chi et al. (2023). By leveraging multi-step prediction and powerful diffusion models based on Chi et al. (2023), we can construct a method competitive to state-of-the-art imitation learning methods.

where $\mathbf{z}$ is a data sample, α_t, σ_t are noise schedule values indexed by timestep $t \in 1, 2, ..., T$, ϵ is randomly sampled Gaussian noise, and ϕ are learned parameters.

To reverse the diffusion process, the model iteratively denoises a sample drawn from the known prior $z_T \sim N(0, I)$. For example, DDPM samples the chain $z_T, ..., z_0$ according to:

$$p_\phi(\mathbf{z}_{t-1} \,|\, \mathbf{z}_t) = N(\mathbf{z}_{t-1} \,|\epsilon_\phi(\mathbf{z}_t, t), \sigma_t^2 I) \tag{2}$$

Diffusion models may also be conditioned on additional context $\mathbf{c}$. For example, text-to-image generative models are conditioned on text, Diffusion Policy is conditioned on visual observations, and Decision Diffuser can be conditioned on reward, skills, and constraints.

Recent generative models have used Latent Diffusion Models, which trains a diffusion model in a learned, compressed latent space (Rombach et al., 2022; Peebles & Xie, 2023; Blattmann et al., 2023) to improve computational and memory efficiency. The latent space is typically learned via an autoencoder, with encoder $\mathcal{E}$ and decoder $\mathcal{D}$ trained to reconstruct $x \approx \hat{x} = \mathcal{D}(\mathcal{E}(x))$. Instead of diffusing over x, the diffusion model is trained on diffusing over $z = \mathcal{E}(x)$.

Imitation Learning In the imitation learning framework, we assume access to a dataset of expert demonstrations, $\mathcal{D} \triangleq \{(s_0, x_0, a_0), .., (s_T, x_T, a_T)\}$, generated by π_E, an expert policy. s_i, x_i, a_i correspond to the state, image, and action at timestep i respectively. The imitation learning objective is to extract a policy $\hat{\pi}(a|s, x)$ that most closely imitates π_E. In robotics, this is typically approached through behavior cloning, which learns the mapping between states and actions directly via supervised learning. We consider single-task imitation, where the dataset corresponds to a single task.

Datasets of expert demonstrations often do not provide sufficient state distribution coverage to effectively solve a given task with imitation learning. However, there often exists additional data in the form of action-free or suboptimal data, which may consist of failed policy rollouts, play data, or miscellaneous environment interactions. Unfortunately, behavior cloning assumes access to data annotated with optimal actions, so such additional data cannot be easily incorporated into training.

4 LATENT DIFFUSION PLANNING

Latent Diffusion Planning consists of three stages, as shown in fig. 2: (1) Training an image encoder via an image reconstruction loss, (2) learning an inverse dynamics model to extract actions a_t from pairs of latent states z_t, z_{t+1}, and (3) learning a planner to forecast future latents z_t.

Algorithm 1 Inference with Latent Diffusion Planning

Input: Encoder $\mathcal{E}$, Planner ϵ_ψ, IDM ϵ_ξ, Planner Diffusion Timesteps T_p, IDM Diffusion Timesteps T_{IDM}, Planning Horizon H_p, Action Horizon H_a

Observe initial state s_0 and image x_0; $k = 0$
while not done **do**
  $\mathbf{z}_k \leftarrow (\mathcal{E}(x_k), s_k)$
  // Diffuse over latent embedding plan
  $\hat{\mathbf{z}}_{k+1}, ..., \hat{\mathbf{z}}_{k+H_p} \sim \mathcal{N}(0, I)$
  for $t = T_p \dots 1$ **do**
    $\hat{\epsilon} \leftarrow \epsilon_\psi(\hat{\mathbf{z}}_{k+1}, ..., \hat{\mathbf{z}}_{k+H_p}; \mathbf{z}_k, t)$
    Update $\hat{\mathbf{z}}_{k+1}, ..., \hat{\mathbf{z}}_{k+H_p}$ using DDPM update with $\hat{\epsilon}$
  end for
  // Diffuse over actions between latent embeddings
  for $i = 0 \dots H_a - 1$ **do**
    $\hat{a}_{k+i} \sim \mathcal{N}(0, I)$ // Predict action for each timestep in action horizon
    for $t = T_{\text{IDM}} \dots 1$ **do**
      $\hat{\epsilon} \leftarrow \epsilon_\xi(\hat{a}_{k+i}; \hat{\mathbf{z}}_{k+i}, \hat{\mathbf{z}}_{k+i+1}, t)$
      Update $\hat{a}_{k+i}$ using DDPM update with $\hat{\epsilon}$
    end for
  end for
  // Execute actions
  for $i = 0 \dots H_a - 1$ **do**
    $s_{k+i+1} \leftarrow$ env.step(s_{k+i}, $\hat{a}_{k+i}$)
  end for
  $k \leftarrow k + H_a$
end while

4.1 LEARNING THE LATENT SPACE

We circumvent planning over high-dimensional image observations by planning over a learned latent space. Similar to prior work in planning with world models (Watter et al., 2015; Ha & Schmidhuber, 2018; Hafner et al., 2020), we learn this latent space using an image reconstruction objective. Our planner thus becomes similar to video models that forecast image frames in a learned latent space (Yan et al., 2021; Hong et al., 2022; Blattmann et al., 2023).

In practical scenarios, we may have a limited demonstration dataset, but much larger and diverse suboptimal or action-free datasets. In this phase of learning, we can make use of the visual information in such datasets for training a more robust latent encoder.

In this work, we train a variational autoencoder (Kingma & Welling, 2014; Rezende et al., 2014) to obtain a latent encoder $\mathcal{E}$ and decoder $\mathcal{D}$. Specifically, we optimize the β-VAE (Higgins et al., 2017) objective:

$$\mathcal{L}_{\text{VAE}}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z} \,|\, \mathbf{x})}[\log p_\theta(\mathbf{x} \,|\, \mathbf{z})] - \beta \mathcal{D}_{\text{KL}}(q_\phi(\mathbf{z} \,|\, \mathbf{x}) || p(\mathbf{z})) \tag{3}$$

where **x** is our original image, **z** is our learned latent representation of the image, θ are the parameters for our decoder, ϕ are the parameters for our encoder, and β is the weight for the KL regularization term.

4.2 Planner and Inverse Dynamics Model

Our policy consists of two separate modules: (1) a planner over latent embeddings, and (2) an inverse dynamics model. The planner and IDM both optimize the DDPM objective.

The planner is conditioned on the current latent embedding, which consists of the concatenated latent image embedding and robot proprioception, and diffuses over a horizon of future embeddings. We use Diffusion Policy's Conditional U-Net architecture. Concretely, we optimize the following objective:

$$\mathcal{L}_{\text{planner}}(\psi, \mathbf{z}) = \mathbb{E}_{t,\epsilon}[||\epsilon_\psi(\hat{\mathbf{z}}_{k+1}, ..., \hat{\mathbf{z}}_{k+H}; \mathbf{z}_k, t) - \epsilon||^2] \tag{4}$$

where $\mathbf{z}_k$ is the latent embedding at timestep k of the trajectory; $\hat{\mathbf{z}}_{k+1}, ..., \hat{\mathbf{z}}_{k+H}$ is the noised latent embedding sequence, with corresponding noise ϵ; H is the maximum horizon of the forecasted latent plan; t is the diffusion noise timestep; and ψ are the parameters of the planner diffusion model.

Our inverse dynamics model is trained to reconstruct the action between a pair of states, conditioned on their associated latent embeddings.

$$\mathcal{L}_{\text{IDM}}(\xi, \mathbf{z}) = \mathbb{E}_{t,\epsilon}[||\epsilon_\xi(\hat{a}_k; \mathbf{z}_k, \mathbf{z}_{k+1}, t) - \epsilon||^2] \tag{5}$$

where $\mathbf{z}_k$ is the latent embedding at timestep k of the trajectory; $\hat{a}_k$ is the noised action, with corresponding noise ϵ; t is the diffusion noise timestep; and ξ are the parameters of the inverse dynamics diffusion model.

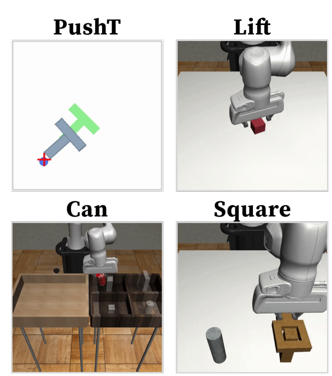
During inference, the planner forecasts a future horizon of states. Like Diffusion Policy, we employ receding-horizon control (Mayne & Michalska, 1988), and execute for a shorter horizon than the full forecasted horizon. We use the inverse dynamics model to extract actions from latent embedding pairs produced by the planner. We use DDPM sampling for both the planner and inverse dynamics models.

5 Experiments

We seek to answer the following questions:

- Is Latent Diffusion Planning an simple and effective imitation learning algorithm, compared to state-of-the-art imitation learning algorithms or methods that may leverage suboptimal data?
- Does our method leverage action-free data for improved planning?
- Does Latent Diffusion Planning enable us to effectively utilize and scale favorably with suboptimal data?

5.1 Experimental Setup



Tasks We focus our experiments on 4 image-based imitation learning tasks: (1) PushT, (2) Robomimic Lift, (3) Robomimic Can, and (4) Robomimic Square. PushT, adapted from IBC and Diffusion Policy (Florence et al., 2021; Chi et al., 2023), involves pushing a block to a target position with 2D end-effector control. Robomimic (Mandlekar et al., 2021) is a robotic manipulation and imitation benchmark.

Dataset To demonstrate the effectiveness of Latent Diffusion Planning, we assume a low-demonstration data regime. For PushT, Can, and Square, we filter 100 demonstrations out of the 200 total, and for Lift, we filter 3 demonstrations out of the 200 total. To further emphasize the importance of suboptimal data, these demonstrations cover a limited state space of the entire environment. For PushT, we filter demonstrations such that the agent never reaches the right third of the 2D state space. For Robomimic tasks, we filter demonstrations based on the initialization of the object of interest, such that the initialization does not cover the entire distribution of object initializations during evaluations.

Table 1: **Leveraging Suboptimal Data.** Latent Diffusion Planning, outperforms prior imitation learning works as it better utilizes suboptimal data via VAE representation learning and training the inverse dynamics model.

Method	PushT	Lift	Can	Square
DP	0.29 ± 0.002	0.36 ± 0.000	0.43 ± 0.010	0.37 ± 0.010
RC-DP	**0.58** ± 0.000	0.38 ± 0.08	0.43 ± 0.030	**0.50** ± 0.060
DP+Repr	0.19 ± 0.005	0.52 ± 0.020	**0.56** ± 0.020	0.42 ± 0.040
UniPi-OL	0.18 ± 0.006	0.47 ± 0.050	0.10 ± 0.020	0.15 ± 0.070
UniPi-CL	0.51 ± 0.032	0.14 ± 0.020	0.34 ± 0.020	0.11 ± 0.010
LDP + Subopt (ours)	0.20 ± 0.013	**0.83** ± 0.030	**0.58** ± 0.020	**0.47** ± 0.010

Our suboptimal data consists of failed trajectories from an under-trained behavior cloning agent. For simplicity, we assume an observation horizon of 1 and a single-view image input for all tasks. We use the third-person camera for Robomimic.

Baselines

- **Diffusion Policy** (**DP**) is a state-of-the-art imitation learning algorithm.
- **Reward-Conditioned Diffusion Policy** (**RC-DP**) utilizes suboptimal actions by conditioning the policy on a binary value indicating whether the action chunk comes from optimal demonstrations or not. This method is inspired by reward-conditioned approaches (Kumar et al., 2019b; Chen et al., 2021)
- **Diffusion Policy with Representation Learning** (**DP+Repr**) uses a VAE pretrained on demonstration and suboptimal data as the observation encoder. This is representative of the methods that leverage suboptimal data through representation learning.
- **Open-Loop UniPi** (**UniPi-OL**) is based off of UniPi (Du et al., 2023a), a video planner for robot manipulation. UniPi-OL generates a single video trajectory, extracts actions, and executes the actions in an open-loop fashion. We use a goal-conditioned behavior cloning agent to reach generated subgoals (Wen et al., 2024).
- **Closed-Loop UniPi** (**UniPi-CL**) is a modification that allows UniPi to perform closed-loop replanning over image chunks. Like LDP, UniPi-CL generates dense plans instead of waypoints, though in image space. We learn an inverse dynamics model to extract actions.

Table 2: **Leveraging Action-Free Data.** LDP outperforms prior imitation learning works as it better utilizes action free data via a state forecasting objective.

Method	Lift	Can	Square
DP	0.36 ± 0.000	0.43 ± 0.010	0.37 ± 0.010
UniPi-OL + Action-Free	0.48 ± 0.060	0.15 ± 0.010	0.21 ± 0.03
LDP + Action-Free (ours)	**0.55** ± 0.030	**0.99** ± 0.010	**0.40** ± 0.020

5.2 Imitation Learning with Suboptimal Data

In table 1, we present imitation learning results. First, LDP outperforms DP, which can only utilize data with optimal actions. LDP, which uses suboptimal data for the VAE and IDM, can leverage diverse data sources outside of the demonstration dataset.

RC-DP, a conditional variant of DP that utilizes suboptimal data, achieves competitive results for PushT Square, while struggling to improve for Lift or Can. We hypothesize that for the Square task, the primitive motions of reaching or grasping the object, which are partially covered by the suboptimal dataset, provides a useful visuomotor prior for the policy. In addition, the suboptimal data from PushT may provide a useful prior in how to interact with the object. LDP outperforms RC-DP as it is able to leverage additional data directly for better action extraction, whereas RC-DP and DP+Repr only use it to learn better representations.

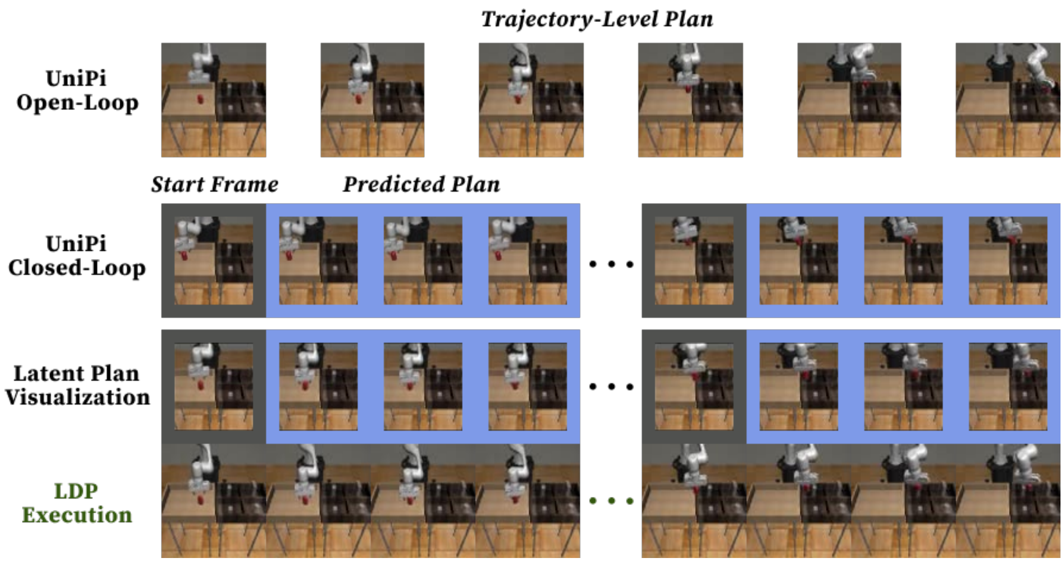


Figure 3: Visualizations of Generated Plans. UniPi-OL generates an entire trajectory for a GCBC agent to follow. UniPi-CL generates single-step image sequences. LDP generates latent plans (visualized via the VAE decoder), using an IDM to execute the plan.

In addition, we consider using a pretrained encoder to extract image features for DP (DP-Repr). We find that learning strong representations through pretraining leads to consistent improved performance for all Robomimic tasks.

Next, we compare against UniPi, which plans over image subgoals (OL) or image chunks (CL). Due to the low demonstration data regime, learning effective and accurate video policies is difficult, and LDP strongly outperforms UniPi-OL. LDP also outperforms UniPi-CL in all Robomimic environments. We hypothesize that this is due to the difficulties learning to forecast dense image chunks. For that reason, the high performance of UniPi-CL on PushT may be attributed to the simpler environment dynamics and observations.

5.3 Imitation Learning with Action-Free Data

Imitation learning policies that model actions, such as DP, are unable to use action-free data, while planning-based approaches can benefit from this additional data. We train the UniPi video with additional action-free demonstrations. We find that this leads to a slight boost in performance, compared to results from table 1, but it still does not outperform LDP. We also train the LDP planner with additional action-free demonstrations. We use a VAE pretrained with demonstration and suboptimal images, and only add the action-free data for the planner, to isolate the effect of action-free data for the planner. We find that action-free demonstrations leads to a large increase in performance in Can, although surprisingly, it does not improve performance for Lift or Square. The improvement in Can may be due to the visual complexity of the scene, where action-free data may provide additional reasoning. Both the third-person Lift and Square views have minimal table or background textures, whereas the Can task observations are comparatively zoomed out, with two unique table textures and a noticeable floor texture.

5.4 Ablations: Suboptimal Data

Suboptimal data can be used for (1) training the VAE (any image data, whether action-labeled or action-free, suffices), and (2) training the IDM. table 3 shows the effects of using each type of data across different suboptimal dataset sizes: no suboptimal data, 1,000 suboptimal trajectories, and 2,000 suboptimal trajectories.

Table 3: **Scaling the Amount of Suboptimal Data.** We investigate the effects of using suboptimal data for pretraining the VAE encoder and for training the IDM. The size of the suboptimal dataset minimally affects performance, but using the data for both encoder and the IDM leads to stronger performance.

Method	Lift	Can	Square
No Subopt	0.50 ± 0.030	0.40 ± 0.020	0.25 ± 0.030
Subopt 1k (Encoder)	0.74 ± 0.060	**0.63** ± 0.010	0.39 ± 0.010
Subopt 1k (Encoder & IDM)	**0.83** ± 0.030	0.54 ± 0.000	**0.47** ± 0.030
Subopt 2k (Encoder)	0.77 ± 0.090	0.58 ± 0.020	0.37 ± 0.010
Subopt 2k (Encoder & IDM)	**0.83** ± 0.030	0.58 ± 0.020	**0.47** ± 0.010

First, we find that not using suboptimal data drastically hurts performance. Only using suboptimal data for the VAE training dramatically improves performance, as it improves the quality of latent embeddings used for both planning and action extraction. Qualitatively and quantitatively, we find that image reconstruction on evaluation images from each environment improves with the use of suboptimal data. Visualized plans naturally look more cohesive, because the underlying latent embeddings used for planning are better structured.

Next, we find that using suboptimal data for inverse dynamics typically improves performance further. This suggests that additional environment interactions helps learn a more generalizable IDM, which can more faithfully extract actions from the diffused plan.

Scaling the amount of suboptimal data from 1,000 to 2,000 trajectories does not lead to noticeable improvements in performance. We hypothesize that the environments may be simple enough that further data does not bring extraordinary benefits. In addition, the difference between the two datasets may be minimal, and using even larger suboptimal datasets or different types of suboptimal data may lead to further improvements.

6 Discussion

We presented Latent Diffusion Planning, a simple planning-based method for imitation learning. We show that our design using powerful diffusion models for latent state forecasting enables competitive performance with state-of-the-art imitation learning. We further show this latent state forecasting objective enables us to easily leverage heterogeneous data sources. In low-demonstration data imitation learning regime, LDP outperforms prior imitation learning work that does not leverage such additional data as effectively.

Limitations. One limitation of the current approach is that the latent space for planning is simply learned with a variational autoencoder and might not learn the most useful features for control. Future work will explore different representation learning objectives. Furher, our method requires diffusing over states, which incurs additional computational overhead as compared to diffusing actions. However, we expect continued improvements in hardware and inference speed will mitigate this drawback. Finally, we did not explore applying recent improvements in diffusion models (Peebles & Xie, 2023; Lipman et al., 2022), which will be important to scale to real-world applications.

Future work. We have validated in simulation the hypothesis that latent state forecasting can leverage heterogeneous data sources. Future work will evaluate whether this can be used to improve practical real-world applications. One direction is to use a diverse dataset of human collected data, such as with handheld data collection tools (Young et al., 2021). Another approach would be to use autonomously collected robotic data (Konstantinos Bousmalis* & Heess, 2023). As these alternative data sources are easier to collect than demonstrations, they represent a different scaling paradigm that can outperform pure behavior cloning approaches. By presenting a method that can leverage such data, we believe this work makes a step toward more performant and general robot policies.

7 Reproducibility Statement

For reproducibility, we will open-source the implementations for our method. Our work primarily builds upon existing work Chi et al. (2023); Du et al. (2023a); Hansen-Estruch et al. (2023); Peebles & Xie (2023), which are also publicly available. In the Appendix, we include implementation details and hyperparameters.

References

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making?, 2023. URL https://arxiv.org/abs/2211.15657.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models, 2023. URL https://arxiv.org/abs/2310.10639.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021. URL https://arxiv.org/abs/2106.01345.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *arXiv preprint arXiv:2409.12192*, 2024.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation, 2023a. URL https://arxiv.org/abs/2302.00111.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning, 2023b. URL https://arxiv.org/abs/2310.10625.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International conference on machine learning*, pp. 1755–1763. PMLR, 2019.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning, 2021. URL https://arxiv.org/abs/2109.00137.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels, 2019. URL https://arxiv.org/abs/1811.04551.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024. URL https://arxiv.org/abs/2310.16828.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies, 2023. URL https://arxiv.org/abs/2304.10573.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022. URL https://arxiv.org/abs/2205.09991.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014. URL https://arxiv.org/abs/1312.6114.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to Act from Actionless Videos through Dense Correspondences. *arXiv:2310.08576*, 2023.

Dushyant Rao* Coline Devin* Alex X. Lee* Maria Bauza* Todor Davchev* Yuxiang Zhou* Agrim Gupta* 1 Akhil Raju Antoine Laurens Claudio Fantacci Valentin Dalibard Martina Zambelli Murilo Martins Rugile Pevceviciute Michiel Blokzijl Misha Denil Nathan Batchelor Thomas Lampe Emilio Parisotto Konrad Żołna Scott Reed Sergio Gómez Colmenarejo Jon Scholz Abbas Abdolmaleki Oliver Groth Jean-Baptiste Regli Oleg Sushkov Tom Rothörl José Enrique Chen Yusuf Aytar Dave Barker Joy Ortiz Martin Riedmiller Jost Tobias Springenberg Raia Hadsell† Francesco Nori† Konstantinos Bousmalis*, Giulia Vezzani* and Nicolas Heess. Robocat: A self-improving foundation agent for robotic manipulation. *ArXiv*, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning, 2021. URL https://arxiv.org/abs/2110.06169.

Aviral Kumar, Xue Bin Peng, and Sergey Levine. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*, 2019a.

Aviral Kumar, Xue Bin Peng, and Sergey Levine. Reward-conditioned policies, 2019b. URL https://arxiv.org/abs/1912.13465.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning, 2020. URL https://arxiv.org/abs/2006.04779.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions, 2024. URL `https://arxiv.org/abs/2403.03181`.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

David Q Mayne and Hannah Michalska. Receding horizon control of nonlinear systems. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pp. 464–465. IEEE, 1988.

Suvir Mirchandani, Suneel Belkhale, Joey Hejna, Evelyn Choi, Md Sazzad Islam, and Dorsa Sadigh. So you think you can scale up autonomous robot data collection? In *8th Annual Conference on Robot Learning*, 2024. URL `https://openreview.net/forum?id=XrxLGzF0lJ`.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL `https://arxiv.org/abs/2212.09748`.

Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, pp. 683–693. PMLR, 2023.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL `https://arxiv.org/abs/2112.10752`.

Karl Schmeckpeper, Annie Xie, Oleh Rybkin, Stephen Tian, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Learning predictive models from observation and interaction. In *European Conference on Computer Vision*, pp. 708–725. Springer, 2020.

Juergen Schmidhuber. Reinforcement learning upside down: Don't predict rewards–just map them to actions. *arXiv preprint arXiv:1912.02875*, 2019.

Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone, 2022. URL `https://arxiv.org/abs/2206.11251`.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL `https://arxiv.org/abs/1503.03585`.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2024. URL `https://arxiv.org/abs/2401.00025`.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pp. 2226–2240. PMLR, 2023a.

Philipp Wu, Arjun Majumdar, Kevin Stone, Yixin Lin, Igor Mordatch, Pieter Abbeel, and Aravind Rajeswaran. Masked trajectory models for prediction, representation, and control. In *International Conference on Machine Learning*, pp. 37607–37623. PMLR, 2023b.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Sherry Yang, Jacob Walker, Jack Parker-Holder, Yilun Du, Jake Bruce, Andre Barreto, Pieter Abbeel, and Dale Schuurmans. Video as the new language for real-world decision making, 2024. URL https://arxiv.org/abs/2402.17139.

Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. In *Conference on Robot Learning*, pp. 1992–2005. PMLR, 2021.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to leverage unlabeled data in offline reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25611–25635. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/yu22c.html.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

Tony Z. Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Seyed Kamyar Seyed Ghasemipour, Chelsea Finn, and Ayzaan Wahid. ALOHA unleashed: A simple recipe for robot dexterity. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=gvdXE7ikHI.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination, 2024a. URL https://arxiv.org/abs/2404.12377.

Zhiyuan Zhou, Pranav Atreya, Abraham Lee, Homer Walke, Oier Mees, and Sergey Levine. Autonomous improvement of instruction following skills via foundation models. *arXiv preprint arXiv:2407.20635*, 2024b.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

Diffusion Policy We use a Jax reimplementation of the convolutional Diffusion Policy, which we verify can reproduce reported Robomimic benchmark results. For improved performance, we process the 512-dimensional ResNet feature with an MLP [512, 256, 32] with ReLU activations and a final tanh activation.

UniPi We use the open-source implementation of UniPi (Ko et al., 2023). For UniPi-OL and UniPi-CL, we predict 7 future frames. During training time, for UniPi-OL, the 7 future frames are evenly sampled from a training demonstration. For UniPi-CL, the 7 future frames are the next consecutive frames.

The goal-conditioned behavior cloning agent is implemented as a goal-conditioned Diffusion Policy (Chi et al., 2023) agent, and it is trained on chunks of 16. The inverse dynamics model is based off of Hansen-Estruch et al. (2023), and shares the same architecture as the IDM used in LDP.

We train the video prediction models for 100k gradient steps with batch size 16.

LDP The LDP VAE is adapted from Diffusion Transformer (Peebles & Xie, 2023). The planner is based directly off of the convolutional U-Net from Diffusion Policy (Chi et al., 2023), with modifications to plan across latent embeddings instead of action chunks. The IDM is based off of Hansen-Estruch et al. (2023).

Table 4: Diffusion Policy Architecture Hyperparameters

	UniPi-OL GCBC	**DP and LDP**	**LDP - Square**
down_dims	[256, 512, 1024]	[256, 512, 1024]	[256, 512, 1024, 2048]
n_diffusion_steps	100	100	100
batch_size	512	256	256
lr	1e-4	1e-4	1e-4
n_grad_steps	200k	500k	500k

Table 5: IDM Architecture Hyperparameters

	UniPi-CL IDM	**LDP IDM**
n_blocks	3	5
n_diffusion_steps	100	100
batch_size	512	256
lr	1e-4	1e-4
n_grad_steps	200k	500k

Table 6: VAE Architecture Hyperparameters

	VAE
block_out_channels	[128, 256, 256, 256, 256, 256]
down_block_types	[DownEncoderBlock2D] x5
up_block_types	[UpDecoderBlock2D] x5
latent_channels	4
PushT Latent Dim	(3, 3, 4)
Robomimic Latent Dim	(2, 2, 4)
PushT KL Beta	1e-5
Lift KL Beta	1e-5
Can KL Beta	5e-6
Square KL Beta	5e-6
n_grad_steps	300k

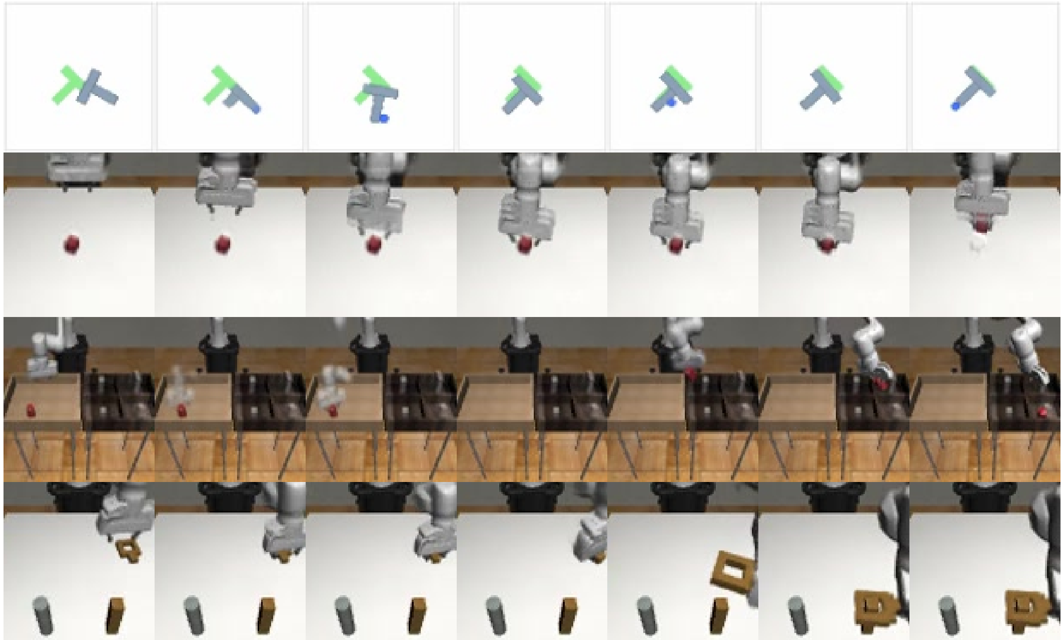


Figure 4: Visualizations of UniPi-OL plans. In PushT, there are small visual mistakes, such as a deformed T or a missing agent. Can, a more visually complex environment, also suffers from this challenge.

A.2 SIMULATION EXPERIMENTS

In our experiments, we report results on 2 seeds, across the best performing checkpoint from last 5 saved checkpoints. PushT results are based on environment reward, and Robomimic results are reported as success rates. For UniPi, we train two goal-conditioned or inverse dynamics models and report success from the 200k checkpoint.

For UniPi-OL evaluations, we predetermine the number of steps for the GCBC to reach each image subgoal based on the demonstration lengths. For PushT, evaluation episode lengths are 200 steps; Lift is 60 steps; Can is 140 steps; and Square is 160 steps. This maximum horizon is also enforced for UniPi-CL evaluations, for consistency.

A.3 UNIPI PLAN VISUALIZATIONS

We include visualizations of closed-loop replanning from UniPi-CL and LDP on our website: `https://sites.google.com/view/latent-diffusion-planning/home`

We include examples of non-cherrypicked UniPi-OL plans (trained w/o action-free data) in fig. 4.