Solving New Tasks by Adapting Internet Video Knowledge

Anonymous authors
Paper under double-blind review

Abstract

Video generative models, beyond enabling the production of astounding visual creations, offer a promising pathway for unlocking novel, text-conditioned robotic behaviors, whether utilized as a video planner or as a policy supervisor. When pretrained on internet-scale datasets, such video models intimately understand alignment with natural language, and can thus facilitate novel text-conditioned behavior generalization. At the same time, however, they may not be sensitive to the specificities of the particular environment in which a policy of interest is to be learned. On the other hand, video modeling over in-domain examples of robotic behavior naturally encodes environment-specific intricacies, but the scale of available demonstrations may not be sufficient to support generalization to unseen tasks via natural language specification. In this work, we investigate different adaptation techniques that integrate in-domain information into large-scale pretrained video models, and explore the extent to which they enable novel text-conditioned generalization for robotic tasks. Furthermore, we highlight the individual data and training requirements of each approach, which range from utilizing only a few still frames illustrating the subject of interest, to direct finetuning over videos labeled with text descriptions. We successfully demonstrate across robotic environments that adapting powerful video models with small scales of example data can successfully facilitate generalization to novel behaviors, both when utilized as policy supervisors, and as visual planners.

1 Introduction

In the past year, video generative models trained explicitly on in-domain demonstrations have demonstrated accurate encoding of environment-specific visual details and dynamics, and have been popular choices to utilize for robotic learning (Du et al., 2024b; Huang et al., 2023; Yang et al., 2023b; Ko et al., 2024; Liang et al., 2024). With visual frames optimized on expert video demonstrations, their encoded understanding of expert behavior can be directly used to supervise the learning of high-performing policies (Huang et al., 2023; Escontrela et al., 2024). Furthermore, they have been applied as performant visual planners (Du et al., 2024a) in robotic settings. However, for arbitrary robotic environments, there is usually a severe difference in scale of tractably available expert demonstration data, especially with associated text labelling, in comparison with general internet-scale datasets of videos paired with natural language. As a result, such in-domain video generative models usually suffer from weaker generalization capability, across novel text specifications and motions of interest.

Instead of directly training a video model on demonstration data, we can obtain much better generalization performance by using existing text-to-video models pretrained on internet-scale data. Having summarized powerful priors over visual styles, natural motion, and alignment with natural language from large-scale data, such models can be leveraged to supervise policies that behave flexibly conditioned on text, in accordance with natural motion priors, and across multiple environment visual styles without modification (Luo et al., 2024). However, policies are often deployed in fixed environment with specific visual characteristics and potentially unique interaction dynamics, which a general environment-agnostic video model may not inherently understand or respect. Thus, directly applying a large-scale pretrained video generative model without modification comes with a potential drawback; they may not understand the intricacies of particular environments of interest to supervise the learning of high-performing policies within them.

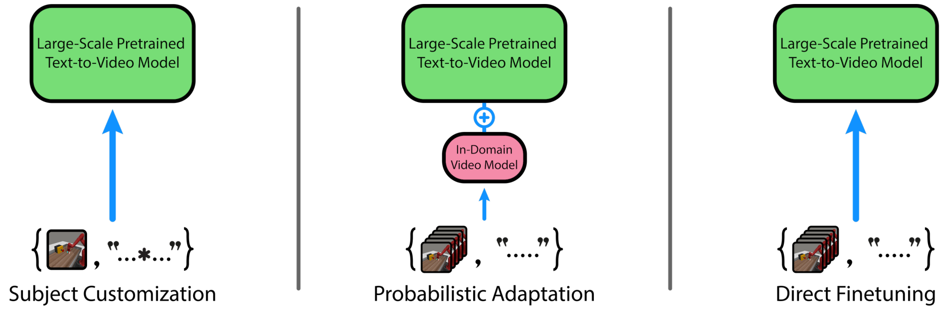
Large-Scale Pretrained Text-to-Video Model
"...*..."
Subject Customization
Large-Scale Pretrained Text-to-Video Model
In-Domain Video Model
"....."
Probabilistic Adaptation
Large-Scale Pretrained Text-to-Video Model
"....."
Direct Finetuning

Figure 1: **Adaptation Techniques.** We explore how in-domain information can be integrated into large-scale text-to-video models through three different adaptation techniques: Subject Customization, Probabilistic Adaptation, and Direct Finetuning. Subject Customization only modifies the image and text encoder, rather than the motion module, and is lightweight in terms of data requirements: it only utilizes pairs of static images and text annotated with a special identifier. Probabilistic Adaptation learns a small in-domain model from paired video data, which is then used through score composition with a large-scale video model that is kept frozen. The small in-domain model can be flexibly parameterized to consider available training resources. Direct Finetuning seeks to update the motion module of the large-scale video model with in-domain paired video data.

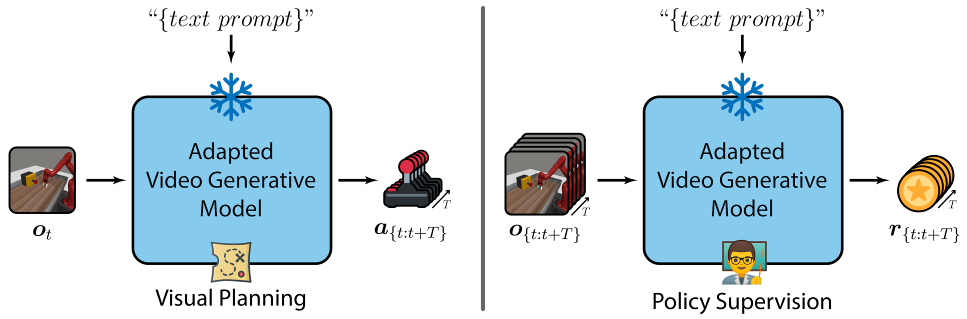

Figure 2: **Downstream Task Evaluation.** We evaluate how adapted video models can enable text-conditioned generalization via two approaches: **visual planning** and **policy supervision**. For visual planning, the adapted video model synthesizes a text-conditioned video plan into the future, which is then converted into actions to follow. In policy supervision, the adapted video model is used in a discriminative manner to evaluate frames achieved by the policy; these are converted into text-conditioned rewards, which the policy is optimized to maximize.

These considerations naturally motivate the investigation of ways to mutually cover the independent deficiencies of each approach. In this work, we perform a thorough study on novel task generalization via adapting internet video knowledge; we seek to illuminate how in-domain information can be best integrated into large-scale pretrained text-to-video models, such that powerful zero-shot text-conditioned generalization capabilities are enabled while considering environment-specific knowledge pertaining to visual styles and interaction dynamics. We compare the downstream robotic performance of multiple adaptation techniques and contrast their respective requirements on in-domain data samples, which range from utilizing only a few still-frames of the agent to text-labelled video demonstrations, and training resources, which span from direct finetuning of the large-scale video model to utilizing it only for inference without any updates. We provide these as valuable insights to the practitioner interested in balancing performance with resource availability.

We perform standardized evaluations across both robotic manipulation tasks (Yu et al., 2020) and continuous control (Tassa et al., 2018), and demonstrate that adapted video generative models are able to successfully act as accurate video planners for novel text-conditioned specifications across a variety of robotic tasks, and can also supervise the learning of novel text-conditioned policies. We commit to open sourcing our models and code. More qualitative examples are available at https://sites.google.com/view/videoadapt-iclr25.

2 Related Work

Adaptation Techniques for Diffusion Models. Although many large-scale pretrained text-to-video models (Ho et al., 2022b; Guo et al., 2023; Ramesh et al., 2022; Brooks et al., 2024; Xing et al., 2023; Ho et al., 2022a; Villegas et al., 2022; Singer et al., 2022; Khachatryan et al., 2023) have demonstrated strong capabilities of synthesizing high-quality videos following the given prompts, it is often desirable to perform adaptation for specialized tasks, such as customizing video generation with specific subjects or styles.

DreamBooth (Ruiz et al., 2023) finetunes text-to-image diffusion models to connect a unique identifier to a subject of interest, using a few images of that specific subject. The subject will be implanted into the output space of the diffusion model after finetuning, enabling novel view synthesis with the subject via prompting with its corresponding identifier. In DreamVideo (Wei et al., 2024), this idea is extended to facilitate novel video generation with respect to a particular subject of interest. It learns subject customization for a pretrained video diffusion model through a few provided static images, which is achieved by combining textual inversion with finetuning an identity adapter.

Prior work on large-to-small adaptation of video models, through composing predicted scores, has demonstrated successful transfer of artistic styles while maintaining powerful text-conditioning behavior (Yang et al., 2023a). In this work, we evaluate this approach to explore the degree to which in-domain environment dynamics and notions of expert behaviors similarly generalize through adaptation with large-scale pretrained video models, conditioned flexibly on natural language. Furthermore, we propose and evaluate a novel probabilistic adaptation technique, which performs score composition in an inverted manner from that presented in (Yang et al., 2023a).

Prior adaptation works mostly seek improvements over visual quality and its related metrics, such as FID (Heusel et al., 2017) and FVD (Thomas et al., 2018). Here, we focus on a new application domain to evaluate adaptation: robotic task performance. We study how the text-conditioning capabilities of large-scale pretrained video generative models can be combined with environment-specific information to deliver further improvements on text-conditioned task generalization.

Video Models for Decision Making. A large body of recent work has explored how video models may be used for decision making (Yang et al., 2024; McCarthy et al., 2024). One line of work explores how video generative models can provide rewards, particularly through a pixel interface (Sermanet et al., 2016; Ma et al., 2022). In VIPER (Escontrela et al., 2024), a video model is trained on expert in-domain demonstrations; it is then utilized to provide dense rewards to supervise downstream policies by evaluating the likelihood of achieved frames during interaction. Similarly, expert demonstrations are also used in Diffusion-Reward (Huang et al., 2023), but a diffusion model is trained instead. Rewards are once again provided through achieved frames, but through a novel cross-entropy computation. In Video-TADPoLe, a large-scale pretrained video diffusion model is used to provide text-conditioned rewards through achieved environment-rendered frames. In this work, we also seek to use video generative models as supervisors for policy learning, but we treat it as a method to evaluate the efficacy of different techniques for adapting large-scale pretrained video models to in-domain data.

A separate line of work utilizes video models as pixel-based planners (Ko et al., 2024; Du et al., 2024a;b; Ajay et al., 2023; Wen et al., 2023; Liang et al., 2024; Yang et al., 2023b; Zhou et al., 2024b; Wang et al., 2024a; Zhou et al., 2024a). In such works, the video model can be directly used to generate a visual plan to solve a task, which can be converted into actions using an inverse dynamics model (Du et al., 2024a) or through dense 3D correspondences (Ko et al., 2024). Alternatively, the video model can also be used as a visual dynamics model as part of a more complex planning routine (Ajay et al., 2023; Du et al., 2024b), to form more complex, long horizon video plans. We utilize video models as visual planners for robotic tasks to understand the quality of different adaptation techniques in integrating in-domain data into large-scale pretrained text-to-video models.

3 Method

Video models pretrained on internet-scale data exhibit strong zero-shot generalization capabilities across diverse visual scenarios, which make them attractive to leverage for downstream robotic tasks. However, the general nature of their pretraining may not inherently enable them to understand domain-specific nuances of the environment within which we would like to learn robotic behavior. We investigate how this can be addressed by integrating in-domain information into large-scale text-to-video models; in Section 3.1 we describe three adaptation approaches of interest, each with separate requirements on in-domain training data and optimization cost. Then, in Section 3.2, we describe two techniques through which we can evaluate the novel task generalization capabilities of these adapted video models in a standardized manner.

3.1 Adaptation Techniques

We aim to adapt pretrained large video models to in-domain data, while maintaining and generalizing their internet-scale knowledge to solving downstream robotic tasks. We utilize an AnimateDiff (Guo et al., 2023) checkpoint as our large video model, which is designed to effectively animate pretrained text-to-image diffusion models such as StableDiffusion (Rombach et al., 2022). At its core, AnimateDiff features a motion module pretrained on a large-scale video dataset, providing powerful motion priors to guide video generation. We investigate three different techniques for in-domain adaptation: ***direct finetuning***, ***subject customization***, and ***probabilistic adaptation***.

3.1.1 Direct Finetuning

Directly finetuning a generally-pretrained text-to-video model is one of the most straightforward ways to mitigate potential domain gaps. Given a video τ_0 sampled from in-domain data distribution $p(\tau_0)$, a randomly sampled Gaussian noise $\epsilon \sim \mathcal{N}(0, 1)$ and a schedule of noise levels β_t that are indexed by timestep $t \in [0, T]$, a text-conditioned video diffusion model can be trained or finetuned as a denoising function $\epsilon_\theta(\cdot)$ by optimizing the denoising objective (Ho et al., 2020) below:

$$\mathcal{L}_{\text{denoise}}(\theta) = \mathbb{E}_{\tau_0,\epsilon,t}[||\epsilon - \epsilon_\theta(\tau_t, t \mid \text{text})||^2] \tag{1}$$

in which τ_t is a noise corrupted video obtained by perturbing τ_0 with sampled Gaussian noise ϵ and noise level t, and training is done with text-conditioning dropout to implement classifier-free guidance (Ho & Salimans, 2022). AnimateDiff is implemented as a trained motion module built on top of pretrained StableDiffusion components; in our study we keep these reused parts unchanged and only adjust the motion module. Direct finetuning involves additional training with labelled pairs of in-domain video demonstrations, allowing the model to update the parameters of its motion module to arbitrary extents and shift its output space towards the target domain. However, direct finetuning of large video models may cause issues such as model collapse or catastrophic forgetting to emerge, especially when demonstration samples are limited.

3.1.2 Subject Customization

Performing direct finetuning on pretrained large video models can often be a computationally expensive endeavour. Furthermore, it requires labelled in-domain video demonstrations, the ready availability of which cannot always be assumed in adaptation scenarios of interest for downstream robotic tasks. We therefore investigate cheaper alternatives for adaptation with respect to data and optimization cost. Customized generation (Ruiz et al., 2023; Gal et al., 2023; Wei et al., 2024) has been widely used for synthesizing subjects and scenes that accommodate user preferences, utilizing only a few static images of a subject or style. In this work, we also explore how this technique can be used to inject subject-centric information into pretrained video models, potentially enabling them to better supervise robotic task behavior. We use DreamBooth (Ruiz et al., 2023) to customize the generation process due to its simplicity and data efficiency. This method binds a unique text identifier to a specific subject, examples of which are provided using still images *without* text descriptions about the motions, and enables novel view synthesis of the subject contextualized in different scenes.

Following DreamBooth (Ruiz et al., 2023), we design special prompts by using a rare token (e.g. "[D]") as the unique identifier for each in-domain subject (e.g."a photo of [D] robot arm"). We finetune StableDiffusion with DreamBooth on static images of the subject paired with this special text prompt. We then instantiate AnimateDiff with the DreamBooth-finetuned UNet and text encoder for subject-informed video generation. Unlike direct finetuning which requires labelled video data, this approach performs few-shot customization just through the use of static images; the adaptation is possible without requiring expert video demonstrations, when only still observations of the environment and its subject are available. Since this adaptation technique will not expose any subject motions to the video model, it also allows us to study whether the pretrained video model can transfer its *motion prior*, which is obtained from domain-agnostic pre-training, onto arbitrary in-domain subjects of interest and facilitate generalization over downstream robotic tasks.

3.1.3 Probabilistic Adaptation

Under some circumstances, the large-scale pretrained model is available for inference, but adjusting it in any way may not be feasible or desirable. In such scenarios, we consider Probabilistic Adaptation (Yang et al., 2023a), where a small sample of demonstrations is utilized to train an in-domain video model, which can be flexibly parameterized to accommodate available modeling resource constraints. Adaptation is then performed by combining the predicted scores from the pretrained, frozen large-scale video model $\epsilon_{\text{pretrained}}(\tau_t, t \mid \text{text})$ with those of the lightweight domain-specific video model $\epsilon_\theta(\tau_t, t \mid \text{text})$ during inference. We use low-temperature sampling (Yang et al., 2023a) to compute the adapted score following the denoising function below:

$$\tilde{\epsilon} = \epsilon_\theta(\tau_t, t) + \alpha\Big(\epsilon_\theta(\tau_t, t \mid \text{text}) + \gamma\epsilon_{\text{pretrained}}(\tau_t, t \mid \text{text}) - \epsilon_\theta(\tau_t, t)\Big) \tag{2}$$

where γ is the prior strength, and α is the guidance scale of text-conditioning. This method only requires the training of a small component with limited in-domain data, and allows the pretrained large video model to serve as a probabilistic prior which guides the generation process of the domain-specific model through score composition.

Moreover, we extend Equation 2 to its inverse version, in which the adaptation direction between $\epsilon_\theta(\tau_t, t \mid \text{text})$ and $\epsilon_{\text{pretrained}}(\tau_t, t \mid \text{text})$ is inverted:

$$\tilde{\epsilon}_{\text{inv}} = \epsilon_{\text{pretrained}}(\tau_t, t) + \alpha\Big(\epsilon_{\text{pretrained}}(\tau_t, t \mid \text{text}) + \gamma\epsilon_\theta(\tau_t, t \mid \text{text}) - \epsilon_{\text{pretrained}}(\tau_t, t)\Big). \tag{3}$$

In inverse probabilistic adaptation, the pretrained video model controls the generation process, while consulting the small model for domain-specific information. Both probabilistic adaptation formulations allow more flexible and low-cost adaptation in the video space compared to direct finetuning; empirically, we find that one direction may work better than the other in certain circumstances.

3.2 Evaluating Task Generalization Capabilities of Video Models

To measure the quality of adaptation, samples from the adapted video models can be judged with respect to in-domain examples in terms of Fréchet Video Distance (FVD) scores (Yang et al., 2023a; Thomas et al., 2018). However, beyond simply assessing surface-level visual style, we propose further evaluating adaptation quality via their ability to facilitate downstream robotic performance. For tasks with predefined evaluation schemes, this provides a quantifiable metric in terms of achieved performance and success, and can deliver additional insights into the capabilities of video models beyond appealing visual content generation. In this work we consider two approaches, depicted in Figure 2, for applying video models to decision making – ***visual planning*** and ***policy supervision***. Under both scenarios, we can measure to what degree downstream robotic performance and text-conditioned generalization may be enabled via different adaptation techniques.

3.2.1 Video Models as Visual Planners

Synthesizing a visual plan in imagination and then executing it by converting it into actions is an intuitive and effective way to utilize video generative models for decision making. Prior work has applied text-guided video generation successfully for task planning (Du et al., 2024a;b; Ajay et al., 2023), with action translation facilitated via an inverse dynamics model. However, as the performance of the video planner can be highly dependent on both the visual quality of the imagined plan and the robustness of the inverse dynamics model, prior work has only utilized video models trained on in-domain demonstrations. In this work, we investigate whether generally-pretrained models, which may inherently produce out-of-distribution visual plans, can capture the underlying environment dynamics through cheap adaptation to in-domain data and act as powerful in-domain planners with novel text-conditioned generalization capabilities.

We based our implementation on the UniPi framework (Du et al., 2024a), in which the adapted text-to-video model is used to synthesize a text-conditioned sequence of future frames as a task plan. To execute the plan, we use an inverse dynamics model to translate sequential pairs of visual frames into executable robotic actions, which are then directly performed in interaction with the environment.

3.2.2 Video Models as Policy Supervisors

When learning behaviors within environments that have visual rendering capabilities, video generative models can be used as policy supervisors. In this approach, the video model is utilized to evaluate frames achieved by the agent during interaction in a discriminative manner; these signals can then be converted into rewards with which to optimize the policy. While many prior works require video models that are trained solely on expert in-domain demonstrations (Huang et al., 2023; Escontrela et al., 2023), with rewards computed against a summarized notion of "expertness", here we seek to extract accurate text-conditioned rewards from text-to-video models. Video-TADPole (Luo et al., 2024) measures text-alignment of robotic trajectories by noise-corrupting achieved pixel observations, and evaluating how likely a large-scale pretrained text-to-video model would reconstruct the video interactions conditioned on the provided natural language prompt (additional details provided in Appendix B). We train a text-conditioned policy by maximizing cumulative Video-TADPoLe rewards through reinforcement learning. Successful optimization of a text-conditioned policy enables us to evaluate the ability of adapted large-scale text-to-video models in facilitating novel task generalization, as specified by natural language.

After adapting to limited data samples, synthesizing coherent high-quality in-domain videos can still be challenging for video models. This may pose issues in visual planning, where the generated plan must appear sufficiently in-domain for an inverse dynamics model to be able to accurately translate into meaningful actions. On the contrary, video models do not necessarily need the ability to create high-quality in-domain videos from scratch to behave as effective policy supervisors; they simply need to be able to critique the quality of achieved in-domain frames. Thus, expressing adapted knowledge through rewards may allow the detachment of downstream policy performance from demands on video generation quality. Conversely, a potential drawback of this approach in comparison to visual planning is the high variance commonly observed in the policy learning process.

4 Experiments

4.1 Experimental Setup and Evaluation

Benchmarks: We evaluate to what degree adapted video models can facilitate downstream robotic behavior generalization across a variety of environments and tasks, spanning robotic manipulation to continuous control. We focus the bulk of our explorations on MetaWorld-v2 (Yu et al., 2020), which offers a suite of robotic manipulation tasks with different levels of complexity. This benchmark allows us to thoroughly assess the generalization capabilities of adaptation methods across a wide selection of tasks. To study the effectiveness of adaptation techniques in a *low data* regime, we curate a small dataset of in-domain examples from 7 MetaWorld tasks (denoted with an asterisk in Table A2) to adapt pretrained video models. For each task, we utilize 25 expert videos for direct finetuning and probabilistic adaptation, while sampling a small set of non-consecutive observations for subject customization. During inference, we evaluate the adapted video models on 9 tasks, 7 of which are novel tasks that are not exposed during adaptation (denoted with no asterisk in Table A2).

Additionally, we extend our evaluation to Humanoid and Dog environments from the DeepMind Control Suite (Tassa et al., 2018). We select "Dog walking" and "Humanoid walking", which offer quantitative evaluation through ground-truth rewards, as tasks where we collect 20 demonstrations for adaptation. Following Video-TADPoLe (Luo et al., 2024), we evaluate the adapted models on walking as well as other behavior achievement tasks specified by novel text prompts. We provide a detailed list of text prompts used for both adaptation and task evaluation in Table A2.

Implementation details of adaptation: In our experiments, we use AnimateDiff (Guo et al., 2023) (∼1.5B parameters) as our pretrained text-to-video model, which combines StableDiffusion with a motion module pretrained on WebVid-10M (Bain et al., 2021) for high-quality video generation. To perform direct finetuning on AnimateDiff, we follow the training pipeline provided by the authors and only update the parameters of its motion module with a small in-domain dataset. For subject customization, we utilize 20 static images to finetune StableDiffusion with DreamBooth for each environment. In addition, we adopt DreamBooth LoRA (Hu et al., 2022) for lower memory usage and better training efficiency. In probabilistic adaptation, we implement our small in-domain video model based on AVDC (Ko et al., 2024), a text-to-video model that diffuses over pixel space; implemented using ∼109M parameters, this is comparable in size to that of the small models used in

Episode Return	Vanilla AnimateDiff	In-Domain-Only	Direct Finetuning	Subject Customization	Prob. Adaptation	Inverse Prob. Adaptation
Humanoid Walking	145.8 ± 48.2	2.4 ± 0.3	111.5 ± 106.4	174.7 ± 42.7	1.8 ± 0.2	92.6 ± 51.1
Dog Walking	60.2 ± 8.8	76.2 ± 29.5	44.6 ± 44.3	117.9 ± 49.7	11.3 ± 0.9	88.7 ± 9.0
Overall	103	39.3	78.1	**146.3**	6.5	90.7

Table 1: **Policy Learning on Continuous Control.** We report ground-truth episode return achieved by policies optimized using the listed adapted video models, aggregated over 5 seeds. We observe that direct finetuning produces marginal improvement over a vanilla AnimateDiff model, and surprisingly, despite adaptation on just static images, subject customization is able to substantially improve continuous locomotion performance over the base pretrained video model.

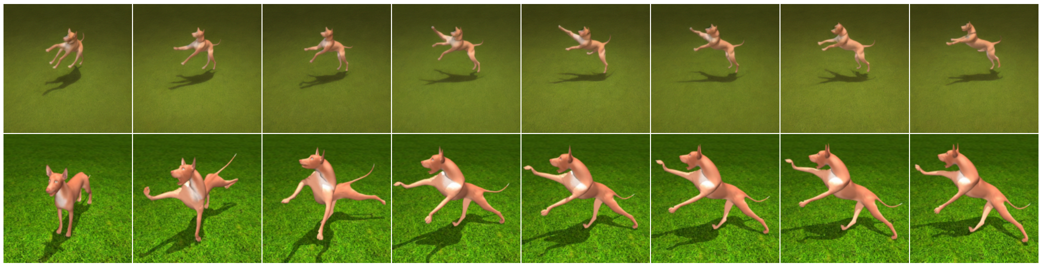

Figure 3: **Novel Text-Conditioned Generalization.** In the top row, we visualize a free-form video generation from a directly finetuned AnimateDiff model for the novel text prompt “a dog jumping”. This was a behavior unseen during adaptation. When using this adapted video model for policy supervision, we showcase that it can successfully supervise a downstream Dog agent to behave according to novel text specifications in a zero-shot manner (policy rollout shown in bottom row).

prior work (Yang et al., 2023a). To enable direct score composition between the in-domain model and AnimateDiff, we modify the AVDC model to diffuse over the same latent space used by StableDiffusion. We include detailed hyperparameters for in-domain model training in Appendix D.

Evaluation metrics: For robotic manipulation tasks in MetaWorld, we report the “success rate”, computed as the proportion of evaluation rollouts in which the agent successfully completes the given task. For Dog and Humanoid evaluation, we follow the setup in Video-TADPoLe (Luo et al., 2024) to report quantitative performance for walking tasks, which have ground-truth reward functions, while providing qualitative results for novel behavior achievement where ground-truth reward functions are unavailable. As in prior work (Yang et al., 2023a), we also use FVD (Thomas et al., 2018) to measure the visual quality of videos generated by adapted video models.

4.2 Policy Supervision

We implement video models as policy supervisors following the setup in Video-TADPoLe (Luo et al., 2024). We reuse the same TDMPC (Hansen et al., 2022) backbone for policy optimization, along with the same hyperparameter settings and training steps, which are tabulated in Section D.

Continuous Control: In Table 1, we report walking performance as evaluated by the ground-truth reward function across all adaptation techniques, in comparison to Video-TADPoLe using default AnimateDiff. In terms of Video-TADPoLe parameters, we utilize context window size of 8, stride length of 4, and noise level of 700 for Humanoid experiments, and a context window of 4, stride length of 2, and a noise level of 500 for Dog experiments; these settings were discovered in an offline manner using ***policy discrimination***, which is described in detail in Appendix E, and were kept across all adaptation methods. The text prompts utilized in these tasks can be found in the first two rows of Table A2. We discover that Subject Customization is able to do better on default walking behavior on both environments, with significant improvement in the case of Dog. This is a striking result, as Subject Customization only utilizes static images of the scene for adaptation and reuses the default motion module pretrained from AnimateDiff. Direct Finetuning also demonstrates a slight performance increase. On the other hand, probabilistic adaptation fails to learn a meaningful policy. We believe that our small in-domain model has difficulty modeling Humanoid and Dog motions, and more capacity for the in-domain model would yield better results through Video-TADPoLe.

However, for novel text-conditioned generalization to new poses, we find that direct finetuning is the best. We find that for a novel text prompt and motion, such as “a dog jumping”, a directly finetuned

Success Rate (%) w/	Door Close*	Door Open	Window Close	Window Open	Drawer Close
In-Domain-Only	100.0 ± 0.0	31.1 ± 44.0	0.0 ± 0.0	33.3 ± 47.1	74.4 ± 36.2
Vanilla AnimateDiff	100.0 ± 0.0	0.0 ± 0.0	33.3 ± 47.1	31.1 ± 44.0	98.9 ± 1.5
Direct Finetuning	100.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	47.8 ± 41.4	95.6 ± 7.7
Subject Customization	100.0 ± 0.0	0.0 ± 0.0	100.0 ± 0.0	60.0 ± 42.5	100.0 ± 0.0
Prob. Adaptation	100.0 ± 0.0	33.3 ± 47.1	33.3 ± 47.1	64.4 ± 45.6	100.0 ± 0.0
Inverse Prob. Adaptation	100.0 ± 0.0	64.4 ± 45.6	100.0 ± 0.0	58.9 ± 42.6	100.0 ± 0.0

Success Rate (%) w/	Drawer Open	Coffee Push*	Soccer	Button Press	**Overall**
In-Domain-Only	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	33.3 ± 47.1	30.2
Vanilla AnimateDiff	33.3 ± 47.1	28.9 ± 23.2	0.0 ± 0.0	33.3 ± 47.1	39.8
Direct Finetuning	0.0 ± 0.0	30.0 ± 26.0	5.6 ± 9.6	0.0 ± 0.0	31.0
Subject Customization	0.0 ± 0.0	15.6 ± 22.0	20.0 ± 17.8	0.0 ± 0.0	44.0
Prob. Adaptation	0.0 ± 0.0	0.0 ± 0.0	6.7 ± 5.4	0.0 ± 0.0	37.5
Inverse Prob. Adaptation	0.0 ± 0.0	16.7 ± 16.7	0.0 ± 0.0	33.3 ± 47.1	**52.6**

Table 2: **Policy Learning on MetaWorld.** We report the mean success rate across 9 manipulation tasks in MetaWorld, over 3 seeds. "$*$" denotes seen tasks during adaptation. We observe that inverse probabilistic adaptation achieves the highest overall performance, both in averaged success rate over the entire task suite, as well as successful generalization to the highest number of novel tasks. Subject customization also achieves surprisingly high aggregate success given its cheap data cost.

video model is able to supervise the learning of an associated policy. We provide a visual of the achieved policy rollout in Figure 3. These results indicate that for continuous locomotion settings, direct finetuning may be the best balance in terms of preserving performance but also enabling interesting text-conditioned generalization; however, subject customization is also a low-cost yet performant approach to consider.

Robotic Manipulation: In Table 2, we report the average success rate on MetaWorld tasks across adaptation techniques, using a standardized context window of 8, stride length of 4, and noise level of 700. We discover that inverse probabilistic adaptation has the best performance. It is able to solve 7 tasks, 5 of which are previously unseen during adaptation, with the highest average success rate of 52.6%. By default, utilizing vanilla AnimateDiff through Video-TADPoLe is able to achieve decent performance, highlighting the default text-conditioned generalization capabilities of large pretrained models. We also believe that integrating it with an in-domain model that supervises the motion of the particular dynamics of the environment, enables better generalization on more challenging tasks (e.g. Door Open). However, the default probabilistic adaptation formulation may heavily rely on the in-domain text-conditioned score, which may be inaccurate when handling novel task prompts due to the small scale of its pretraining. We therefore hypothesize that this explains why inverse probabilistic adaptation is more performant; it may be more robust to novel text-conditioning, as more weight is put on leveraging textual priors from the pretrained model.

FVD Scores (MetaWorld) ↓	Continued Denoising		Free-form Generation	
	Seen	Unseen	Seen	Unseen
Vanilla AnimateDiff	2700.4	2643.2	4625.3	4469.7
In-Domain-Only	**602.8**	**610.1**	2987.2	3080.7
Direct Finetuning	1004.6	978.5	946.0	**915.0**
Subject Customization	1078.9	1711.8	2212.8	2316.0
Prob. Adaptation	622.6	630.6	**848.3**	1237.6
Inv Prob. Adaptation	627.4	681.8	928.6	1250.3

Table 3: **FVD Scores.** We report FVD scores for videos of MetaWorld tasks, produced by Continued Denoising and Free-form Generation via the video generative models of interest. This is computed for both seen and unseen task sets, each with 7 tasks, aggregating results over 1000 synthetic videos.

Additional Metrics: In diffusion-based policy supervision, rewards are extracted from the procedure of corrupting frames achieved by the policy with some level of Gaussian noise and then making denoising predictions using the video model (Huang et al., 2023; Luo et al., 2024). For additional insight, we propose a visualization technique called ***continued denoising***, and report FVD scores for videos generated in such a manner. In continued denoising, rather than extracting a scalar

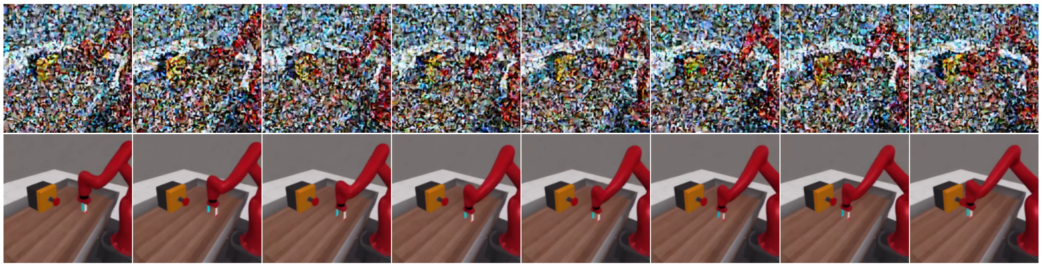

Figure 4: **Continued Denoising.** We visualize frames from a task unseen during adaptation, corrupted with a level of Gaussian noise (top row). We then show the result of continued denoising using an inverse probabilistic adaptation model to verify it can visually generalize to fill in novel in-domain information. Despite not having seen a button, it is able to reconstruct it conditioned on text. This figure is for intuition; in practice, a much higher noise level is used, shown in Figure A1.

from components of the denoising prediction as in Video-TADPoLe, we treat the noised video as an initialization and iteratively continue sampling to produce a final clean video prediction - thus, "continuing" the denoising procedure. In our experiments we perform continued denoising conditioned on a desired text prompt, a noise level of 700, a total frame length of 16, and 10 denoising steps.

As mentioned in Section 3.2.2, policy supervision does not necessarily require strong free-form generation of in-domain videos; rather it evaluates observed frames achieved by following the current policy. For qualitative purposes, continued denoising provides us a visual sense of how this evaluation of achieved frames is done (examples in Figure 4), as well as a sanity check on the integration of in-domain information through adaptation. Furthermore, it enables quantitative comparison through FVD scores, which provides an idea on the capability of adapted video models to reconstruct in-domain-like videos conditioned on text. It is intuitive to hypothesize that a lower FVD score correlates with better in-domain adaptation, as it understands how to accurately complete the provided in-domain frames from a heavy noise corruption.

In Table 3, we report continued denoising FVD scores for both seen and unseen tasks in MetaWorld benchmark. In each setup, we generate 1,000 synthetic videos over 7 robotic manipulation tasks. We discover that the lowest FVD score for continued denoising is achieved by the in-domain model, which is unsurprising as it was explicitly trained on such examples. The next-best FVD scores are achieved by probabilistic adaptation and its inverse. This is significant because it supports the finding that with adaptation, generalization to unseen tasks is possible, and suggests that accurate domain-specific rewards can be supplied through policy supervision. Indeed, this aligns with our result in Table 2, where Inverse Probabilistic Adaptation achieves the best overall task performance through policy supervision. However, a lower FVD alone is not sufficient for successful policy learning, as demonstrated by the relatively weak performance of Probabilistic Adaptation.

4.3 Visual Planning

We implement video models as visual planners following the framework in UniPi (Du et al., 2024a). To generate a plan, we synthesize a sequence of 8 future frames conditioned on both the current visual observation from the environment and the text prompt specifying the task. This is then translated into an executable action sequence via an inverse dynamics model. To mitigate the potential error accumulation problem, we evaluate our visual planner in a closed-loop manner, in which we only execute the first inferred action for every environment step. We provide detailed hyperparameters for video planning, and the implementation of the inverse dynamics model, in Appendix D.

Robotic Manipulation: We evaluate visual planners with different adaptation techniques across 9 selected tasks in MetaWorld, of which 7 are not encountered during adaptation, and report the success rates in Table 4. Among all adaptation techniques that have been evaluated, we observe that probabilistic adaptation and its inverse version achieves the highest overall performance, within which probabilistic adaptation achieves the highest number of non-zero performance on novel tasks. We also find that subject customization improves in-domain performance in comparison to Vanilla AnimateDiff, which is notable given that the same pretrained motion module is reused for planning, with only static images during adaptation to inform in-domain visual details. We hypothesize that

Success Rate (%) w/	Door Close*	Door Open	Window Close	Window Open	Drawer Close
In-Domain-Only	93.3 ± 14.9	0.0 ± 0.0	53.3 ± 29.8	6.7 ± 14.9	20.0 ± 29.8
Vanilla AnimateDiff	100.0 ± 0.0	0.0 ± 0.0	13.3 ± 18.3	40.0 ± 27.9	46.7 ± 29.8
Direct Finetuning	13.3 ± 18.3	0.0 ± 0.0	66.7 ± 33.3	20.0 ± 29.8	60.0 ± 14.9
Subject Customization	100.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	20.0 ± 29.8	25.0 ± 31.9
Prob. Adaptation	100.0 ± 0.0	0.0 ± 0.0	73.3 ± 27.9	13.3 ± 18.3	40.0 ± 43.5
Inverse Prob. Adaptation	100.0 ± 0.0	0.0 ± 0.0	53.3 ± 18.3	0.0 ± 0.0	53.3 ± 38.0
Success Rate (%) w/	Drawer Open	Coffee Push*	Soccer	Button Press	**Overall**
In-Domain-Only	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	40.0 ± 14.9	23.7
Vanilla AnimateDiff	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	22.2
Direct Finetuning	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	17.8
Subject Customization	0.0 ± 0.0	13.3 ± 29.8	0.0 ± 0.0	0.0 ± 0.0	17.6
Prob. Adaptation	6.7 ± 14.9	6.7 ± 14.9	0.0 ± 0.0	33.3 ± 23.6	**30.4**
Inverse Prob. Adaptation	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	26.7 ± 27.9	25.9

Table 4: **Visual Planning on MetaWorld.** We report the mean success rate via visual planning across 9 tasks, aggregated over 5 seeds each. We discover that both probabilistic adaptation and its inverse are able to act as performant visual planners, and substantially improve over default AnimateDiff; notably, probabilistic adaptation achieves success on more unseen tasks than alternatives.

direct finetuning of AnimateDiff fails to produce useful in-domain plans due to the small scale of demonstration data being considered; for such settings with cheap amounts of in-domain data, probabilistic adaptation, its inverse, and subject customization appear comparatively more promising.

Additional Metrics: Visual planning relies on synthesizing coherent high-quality video plans that can be accurately interpreted by the inverse dynamics model. Therefore, we may be interested in measuring the quality of visual plans created by adapted video models in a free-form manner, in other words, generated from pure noise. In Table 3, we report the FVD scores for the same set of seen and unseen tasks that are evaluated in continued denoising experiments. We observe that probabilistic adaptation and its inverse have strong FVD scores compared to other adaptation techniques. At the same time, high FVD scores achieved by direct finetuning fails to translate to high task success rate through visual planning; this suggests that while its plans are of high visual quality, they may not necessarily model the correct motions for solving tasks in a text-conditioned manner. This further suggests that beyond just gauging visual metrics alone, using downstream robotic tasks can provide meaningful and additional insights to evaluate adaptation performance of video models.

Studying Data Quality: In probabilistic adaptation, our in-domain model is trained solely on limited expert demonstrations, which can still be prohibitively expensive to collect in some scenarios. By combining with pretrained video models through adaptation, we provide further investigation on whether the knowledge (e.g. motion priors) obtained from large-scale pretraining can bridge the gap between the suboptimality of in-domain data and task evaluation performance in Appendix A. The same number of demonstrations is reused for adaptation, but trajectories are instead generated by a suboptimal agent that takes an expert action only 30% of the time, and a random action otherwise. To our surprise, the planning performance remains robust, and even sometimes better, despite the use of suboptimal video examples. Moreover, the benefit of pretrained video priors becomes more prominent, and inverse probabilistic adaptation outperforms the in-domain only model by **37%**.

5 Conclusion and Future Work

In this work, we have explored several methods through which internet-scale video models may be adapted to model the appearance and dynamics of novel environments, and subsequently perform new behaviors or accomplish unseen tasks conditioned on text prompts. Our considered adaptation techniques vary in their data and resource requirements, and we introduce the inverse probabilistic adaptation technique to better utilize the pretrained video prior. We have conducted extensive evaluations on Meta-World and DeepMind Control Suite under both policy learning and visual planning setups. We uncover a surprising observation that subject customization, despite relying only on a few static images unaware of the in-domain motions and their text descriptions, serve as a strong and data-efficient baseline across the environments, especially under the policy learning setup. For robotic tasks, our introduced inverse probabilistic adaptation achieves the overall best performance on both task completion, and also visual quality (as measured by FVD). Finally, we observe promising preliminary signals on adaptation with suboptimal video data, highlighting the importance of leveraging pretrained, internet-scale video priors for solving new tasks.

6 Reproducibility

All adaptation techniques in our work are implemented using available open-sourced components. As mentioned in Section 4.1, we use publicly available checkpoints for pretrained large models, such as AnimateDiff (Guo et al., 2023) as well as StableDiffusion (Rombach et al., 2022). We utilize DreamBooth (Ruiz et al., 2023) for subject customization. Furthermore, we reuse the codebase provided by the authors of AVDC (Ko et al., 2024) for in-domain model training, with minimal adjustments to enable the latent diffusion. For policy learning, we follow Video-TADPoLe (Luo et al., 2024) framework, which itself is built off of publicly available AnimateDiff and TDMPC (Hansen et al., 2022). Furthermore, we release detailed hyperparameter settings and implementation details in Appendix D, as well as elaborate on techniques on how they were discovered or selected in Appendix E. We believe that the simplicity of our approach, along with the utilization of open-sourced checkpoints, makes this work highly reproducible. We also commit to open-sourcing our code, to support further reproducibility efforts in the community.

References

Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 3, 5

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 6

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators. 3

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024a. 1, 3, 5, 9

Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, brian ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *International Conference on Learning Representations (ICLR)*, 2024b. 1, 3, 5

Kolve Eric, Mottaghi Roozbeh, Han Winson, VanderBilt Eli, Weihs Luca, Herrasti Alvaro, Deitke Matt, Ehsani Kiana, Gordon Daniel, Zhu Yuke, Kembhavi Aniruddha, Gupta Abhinav, and Farhadi Ali. AI2-THOR: an interactive 3d environment for visual AI. *arXiv*, 1712.05474, 2017. 19

Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 6

Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations (ICLR)*, 2023. 4

Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 3, 4, 6, 11

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022. 7, 11, 17

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 3

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 4

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a. 3

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022b. 3

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 6

Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. *arXiv preprint arXiv:2312.14134*, 2023. 1, 3, 6, 8

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023. 3

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 3, 6, 11, 20

Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024. 1, 3

Calvin Luo, Mandy He, Zilai Zeng, and Chen Sun. Text-aware diffusion for policy learning. In *NeurIPS*, 2024. 1, 6, 7, 8, 11, 15

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 3

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 18

Robert McCarthy, Daniel CH Tan, Dominik Schmidt, Fernando Acero, Nathan Herr, Yilun Du, Thomas G Thuruthel, and Zhibin Li. Towards generalist robot learning from internet video: A survey. *arXiv preprint arXiv:2404.19664*, 2024. 3

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. 4, 11

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3, 4, 11

Pierre Sermanet, Kelvin Xu, and Sergey Levine. Unsupervised perceptual rewards for imitation learning. *arXiv preprint arXiv:1612.06699*, 2016. 3

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 3

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. 17

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 2, 6

Unterthiner Thomas, van Steenkiste Sjoerd, Kurach Karol, Marinier Raphael, Michalski Marcin, and Gelly Sylvain. Towards accurate generative models of video: A new metric & challenges. *arXiv*, 1812.01717, 2018. 3, 5, 7

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 3

Boyang Wang, Nikhil Sridhar, Chao Feng, Mark Van der Merwe, Adam Fishman, Nima Fazeli, and Jeong Joon Park. This&that: Language-gesture controlled video generation for robot planning. *arXiv preprint arXiv:2407.05530*, 2024a. 3

Fu-Yun Wang, Zhaoyang Huang, Weikang Bian, Xiaoyu Shi, Keqiang Sun, Guanglu Song, Yu Liu, and Hongsheng Li. AnimateLCM: computation-efficient personalized style video generation without personalized video data. *arXiv*, 2402.00769, 2024b. 19, 20

Yujie Wei, Shiwei Zhang, Zhiwu Qing, Hangjie Yuan, Zhiheng Liu, Yu Liu, Yingya Zhang, Jingren Zhou, and Hongming Shan. Dreamvideo: Composing your dream videos with customized subject and motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6537–6549, 2024. 3, 4

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 3

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023. 3

Mengjiao Yang, Yilun Du, Bo Dai, Dale Schuurmans, Joshua B Tenenbaum, and Pieter Abbeel. Probabilistic adaptation of text-to-video models. *arXiv preprint arXiv:2306.01872*, 2023a. 3, 5, 7

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023b. 1, 3

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. *arXiv preprint arXiv:2402.17139*, 2024. 3

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020. 2, 6

Enshen Zhou, Yiran Qin, Zhenfei Yin, Yuzhou Huang, Ruimao Zhang, Lu Sheng, Yu Qiao, and Jing Shao. Minedreamer: Learning to follow instructions via chain-of-imagination for simulated-world control. *arXiv preprint arXiv:2403.12037*, 2024a. 3

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024b. 3

A Visual Planning via Suboptimal Adaptation Data

Success Rate (%) w/	Door Close*	Door Open	Window Close	Window Open	Drawer Close
In-Domain-Only	93.3 ± 14.9	0.0 ± 0.0	40.0 ± 27.9	0.0 ± 0.0	33.3 ± 23.6
Prob. Adaptation	100.0 ± 0.0	0.0 ± 0.0	60.0 ± 36.5	13.3 ± 18.3	46.7 ± 29.8
Inverse Prob. Adaptation	93.3 ± 14.9	0.0 ± 0.0	53.3 ± 29.8	0.0 ± 0.0	93.3 ± 14.9

Success Rate (%) w/	Drawer Open	Coffee Push*	Soccer	Button Press	**Overall**
In-Domain-Only	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	13.3 ± 18.3	20.0
Prob. Adaptation	6.7 ± 14.9	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	25.2
Inverse Prob. Adaptation	6.7 ± 14.9	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	**27.4**

Table A1: **Probabilistic Adaptation w/ Suboptimal Data for Video Planning.** We report the mean success rate via visual planning across 9 tasks, aggregated over 5 seeds each. Compared to results in Table 4, we discover that the overall performance of the in-domain model and probabilistic adaptation is severely impacted by the suboptimality of the training data. In contrast, the performance of inverse probabilistic adaptation gets improved under the suboptimal setup.

A natural question that arises through our study is whether or not it is important for expert demonstrations to be available during adaptation, or if the priors captured within large-scale pretrained video models (whether it be from powerful text-conditioning capabilities or natural motion priors summarized from vast pretraining) can be leveraged to bridge the gap from suboptimal in-domain demonstrations. In these experiments, we perform planning with probabilistic adaptation and its inverse, where the available adaptation data is produced by a suboptimal agent. The suboptimal agent takes an expert action only 30% of the time, and a random action 70% of the time. In consistency with the previous setup, we collect 25 (now suboptimal) demonstrations from the same 7 tasks denoted with asterisks in Table A2.

We observe that surprisingly, the overall average task success rate increases for inverse probabilistic adaptation, whereas the in-domain performance decreases overall. This is a promising sign that in adapting large-scale text-to-video models for robotic downstream tasks, expert demonstration may not be explicitly needed. This potentially opens up opportunities for applying large-scale video models to novel robotic tasks, where only random or suboptimal demonstrations are tractably available.

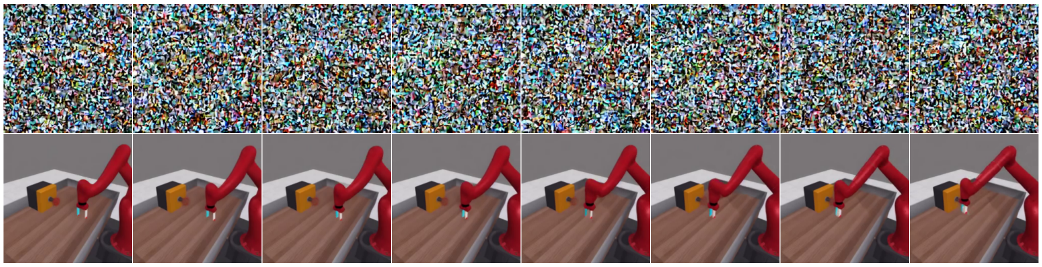

Figure A1: **Continued Denoising (in practice).** In practice, an aggressive level of Gaussian corruption is usually used on achieved frames for reward computation (700 for MetaWorld). However, because to the human eye this may look virtually indistinguishable from pure noise, we supply an illustrative example in Figure 4 using a noise level of 400. Here, we showcase visuals of the same unseen task corrupted with a practical noise level of 700. We then show the result of continued denoising to visually verify the model integrates adapted in-domain information successfully. When performing continued denoising from such a high corruption, conditioned on the text prompt “a robot arm pushing a button”, it is therefore quite surprising the level of detail with which the adapted text-to-video model is able to reconstruct novel in-domain features such as the button - which it has not even seen during adaptation. The resulting continued denoising video can also be evaluated against in-domain examples via FVD for further insights.

B Video-TADPoLe Reward Computation

Video-TADPoLe (Luo et al., 2024) rewards are densely computed for a trajectory achieved by a policy, in terms of their rendered frames. For arbitrary start index i and end index j inclusive of the trajectory, for $i \leq j$, let $\mathbf{o}_{[i+1:j+1]}$ denote the associated sequence of rendered frames. Video-TADPoLe then utilizes a source noise vector $\boldsymbol{\epsilon}_0 \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I}_{j-i+1})$ of the same dimensionality as a Gaussian corruption to produce noisy observation $\tilde{\mathbf{o}}_{[i+1:j+1]}$. Then, Video-TADPoLe computes a batch of *alignment reward* terms through one inference step of the text-to-video diffusion model as:

$$r^{\text{align}}_{[i:j]} = \left\| \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{o}}_{[i+1:j+1]}; \texttt{t}_{\text{noise}}, y) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{o}}_{[i+1:j+1]}; \texttt{t}_{\text{noise}}) \right\|_2^2 ,$$

and a batch of *reconstruction reward* terms as:

$$r^{\text{rec}}_{[i:j]} = \left\| \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{o}}_{[i+1:j+1]}; \texttt{t}_{\text{noise}}) - \boldsymbol{\epsilon}_0 \right\|_2^2 - \left\| \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{o}}_{[i+1:j+1]}; \texttt{t}_{\text{noise}}, y) - \boldsymbol{\epsilon}_0 \right\|_2^2 .$$

For a provided context window of size n, Video-TADPoLe calculates the reward at each timestep t utilizing each context window that involves achieved observation $\mathbf{o}_{t+1}$:

$$r_t = \frac{1}{n} \sum_{i=1}^{n} \texttt{symlog}\left(w_1 * r^{\text{align}}_{[t-i+1:t-i+n]}[i-1] \right) + \texttt{symlog}\left(w_2 * r^{\text{rec}}_{[t-i+1:t-i+n]}[i-1] \right).$$

A stride term s can be used to make this computation tractable across long trajectories, where the context window skips s timesteps before computing a sequence of Video-TADPoLe rewards again. The context window n, stride s, and noise level $\texttt{t}_{\text{noise}}$ are hyperparameters to be set by the user; in practice, good settings for such hyperparameters can be found in an offline manner through *policy discrimination* (Section E).

C TEXT PROMPTS

Task	In-Domain Prompts	AnimateDiff Prompts	DreamBooth Identifier
Dog Walking	a dog/pharaoh hound walking	a dog/pharaoh hound walking	a [D] dog
Humanoid Walking	a(n) humanoid/action figure walking	a(n) humanoid/action figure walking	a [D] action figure
Assembly*	assembly	a robot arm placing a ring over a peg	a [D] robot arm
Dial Turn*	dial turn	a robot arm turning a dial	
Reach*	reach	a robot arm reaching a red sphere	
Peg Unplug Side*	peg unplug side	a robot arm unplugging a gray peg	
Lever Pull*	lever pull	a robot arm pulling a lever	
Coffee Push*	coffee push	a robot arm pushing a white cup towards a coffee machine	
Door Close*	door close	a robot arm closing a door	
Door Open	door open	a robot arm opening a door	
Window Close	window close	a robot arm closing a window	
Window Open	window open	a robot arm opening a window	
Drawer Close	drawer close	a robot arm closing a drawer	
Drawer Open	drawer open	a robot arm open a drawer	
Soccer	soccer	a robot arm pushing a soccer ball into the net	
Button Press	button press	a robot arm pushing a button	

Table A2: **Task-Prompt Pairs.** We include a comprehensive list of tasks and their text prompts for adaptation and evaluation. “$*$” denotes tasks seen during adaptation.

Task	In-Domain Prompts	AnimateDiff Prompts
Spatula in Kitchen*	spatula	find the spatula
Toaster in Kitchen*	toaster	find the toaster
Painting in Living Room*	painting	find the painting
Blinds in Bedroom*	blinds	find the blinds
ToiletPaper in Bathroom*	toilet paper	find the toilet paper
Pillow in Living Room	pillow	find the pillow
DeskLamp in Living Room	desk lamp	find the desk lamp
Mirror in Bedroom	mirror	find the mirror
Laptop in Bedroom	laptop	find the laptop

Table A3: **Task-Prompt Pairs for iTHOR.** We include a comprehensive list of iTHOR tasks and their text prompts for adaptation and evaluation. “$*$” denotes tasks seen during adaptation.

D IMPLEMENTATION DETAILS

We include the default hyperparameters from the TD-MPC implementation in Table A6 for completeness. We do not modify the default recommended settings for both Humanoid and Dog environments, as well as the Meta-World experiments.

Component	# Parameters (Millions)
VAE (Encoder)	34.16
VAE (Decoder)	49.49
U-Net	865.91
Text Encoder	340.39

Table A4: **StableDiffusion Components.** For completeness, we list sizes of the components of the StableDiffusion v2.1 checkpoint used in Video-TADPoLe experiments. The checkpoint is used purely for inference, and is not modified or updated in any way. Note that the VAE Decoder is not utilized in our framework.

Component	# Parameters (Millions)
VAE (Encoder)	34.16
VAE (Decoder)	49.49
U-Net	1312.73
Text Encoder	123.06

Table A5: **AnimateDiff Components.** For completeness, we list sizes of the components of the AnimateDiff checkpoint used in Video-TADPoLe experiments. The checkpoint is used purely for inference, and is not modified or updated in any way. Note that the VAE Decoder is not utilized in our framework.

Hyperparameter	Value
Discount factor (γ)	0.99
Seed steps	$5,000$
Replay buffer size	Unlimited
Sampling technique	PER ($\alpha = 0.6, \beta = 0.4$)
Planning horizon (H)	5
Initial parameters (μ^0, σ^0)	$(0, 2)$
Population size	512
Elite fraction	64
Iterations	12 (Humanoid) 8 (Dog)
Policy fraction	5%
Number of particles	1
Momentum coefficient	0.1
Temperature (τ)	0.5
MLP hidden size	512
MLP activation	ELU
Latent dimension	100 (Humanoid, Dog)
Learning rate	3e-4 (Dog) 1e-3 (Humanoid)
Optimizer (θ)	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Temporal coefficient (λ)	0.5
Reward loss coefficient (c_1)	0.5
Value loss coefficient (c_2)	0.1
Consistency loss coefficient (c_3)	2
Exploration schedule (ϵ)	0.5 → 0.05 (25k steps)
Planning horizon schedule	1 → 5 (25k steps)
Batch size	2048 (Dog) 512 (Humanoid)
Momentum coefficient (ζ)	0.99
Steps per gradient update	1
θ^- update frequency	2

Table A6: **TD-MPC hyperparameters.** We use the official implementation TD-MPC (Hansen et al., 2022) with no adjustments to the hyperparameters, but list it below for completeness. We set the number of training steps to 2 million for continuous control experiments using TD-MPC, and 700k steps for MetaWorld experiments.

Hyperparameter	Value
Training Objective	`pred_noise`
Number of Training Steps	60000
Loss Type	L2
Learning Rate	1e-4
Beta Schedule	Linear schedule (0.0085, 0.012)
Timesteps	1000
EMA Decay	0.99
EMA Update Steps	10

Table A7: **Hyperparameters for In-Domain Model Training.**

Visual Planning Hyperparameters: To generate a video plan with adapted video models, we perform DDIM (Song et al., 2021) sampling for 25 steps. We use 7.5 as the text-conditioning guidance scale for directly finetuned AnimateDiff, and use 2.5 for other adaptation techniques. Additionally, we use 0.1 as the prior strength for probabilistic adaptaion and 0.5 for its inverse version.

Hyperparameter	Value
Input Dimension	1536
Output Dimension	4
Training Epochs	20
Learning Rate	3e-5
Optimizer	AdamW

Table A8: **Hyperparamters of Inverse Dynamics Model Training**

Inverse Dynamics: We employ a small MLP network as our inverse dynamics model. The model takes in the embeddings of two consecutive video frames, which are extracted using VC-1 (Majumdar et al., 2023), and predicts the action that enables the transition between the provided frames. We train the inverse dynamics model on a dataset comprising a mixture of expert and suboptimal trajectories rendered from the environment, using the same set of tasks and data volumn as used for adaptation. For fairness, we reuse the same dynamics model across all adaptation techniques during evaluation. We provide the detailed hyperparameters of inverse dynamics training in Table A8.

E Policy Discrimination

Rather than performing an expensive sweep over Video-TADPoLe hyperparameters directly by launching policy supervision experiments across each adapted video model technique, which can be expensive, we look for an offline method to determine reasonable hyperparameter settings. For each environment, we therefore utilize an example expert quality demonstration video as well as an example poor quality demonstration video (with arbitrary quality levels in-between, if available). Then, we can perform a search over Video-TADPoLe parameters by computing Video-TADPoLe rewards for these trajectories using an adapted video model, conditioned on the task-relevant text prompt, with respect to different context window, stride, and noise level settings. We seek parameter settings that, through the adapted video model's Video-TADPoLe reward computation, can correctly distinguish between the expert, text-aligned video demonstration from the poor, text-unaligned video demonstration; this can be done by comparing the predicted Video-TADPoLe rewards. Once identified in this offline manner, we can subsequently use the discovered settings of context window, stride, and noise level for learning text-conditioned policies. In practice, we have found that these settings can be reused for novel text-conditioning within the same environment without issue.

F Visual Planning with Additional Pretrained Video Models

Success Rate (%) w/	Door Close*	Door Open	Window Close	Window Open	Drawer Close	Drawer Open	Coffee Push*	Soccer	Button Press	**Overall**
In-Domain-Only	93.3 ± 14.9	0.0 ± 0.0	53.3 ± 29.8	6.7 ± 14.9	20.0 ± 29.8	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	40.0 ± 14.9	23.7
Vanilla AnimateLCM	100.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	20.0 ± 18.3	40.0 ± 27.9	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	17.8
Prob. Adaptation	100.0 ± 0.0	0.0 ± 0.0	53.3 ± 38.0	0.0 ± 0.0	53.3 ± 29.8	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	6.7 ± 14.9	23.7
Inverse Prob. Adaptation	100.0 ± 0.0	0.0 ± 0.0	40.0 ± 14.9	0.0 ± 0.0	93.3 ± 14.9	0.0 ± 0.0	0.0 ± 0.0	6.7 ± 14.9	26.7 ± 27.9	**29.6**

Table A9: **Visual Planning on MetaWorld with AnimateLCM.** We report the mean success rate across 9 manipulation tasks in MetaWorld. Each table entry shows the average success rate aggregated from 5 seeds.

We provide visual planning results on MetaWorld with an additional video diffusion model, AnimateLCM (Wang et al., 2024b), in Table A9. We observe both probabilistic adaptation and its inverse version bring improvements in overall success rate compared to Vanilla AnimateLCM. Specifically, inverse probabilistic adaptation achieves the best overall performance and outperforms the in-domain-only baseline by 24.9%, reconfirming the efficacy of adaptation in improving in-domain task performance. This further demonstrates that adaptation as an approach can be applied flexibly across different backbone text-to-video models for successful downstream robotic applications.

G Visual Planning for Object Navigation in iTHOR Environments

Success Rate (%) w/	Spatula in *Kitchen**	Toaster in *Kitchen**	Painting in *Living Room**	Blinds in *Bedroom**	ToiletPaper in *Bathroom**	Pillow in *Living Room*	DeskLamp in *Living Room*	Mirror in *Bedroom*	Laptop in *Bedroom*	**Overall**
In-Domain-Only	13.3 ± 29.8	33.3 ± 33.3	0.0 ± 0.0	13.3 ± 29.8	40.0 ± 36.5	6.7 ± 14.9	6.7 ± 14.9	0.0 ± 0.0	26.7 ± 27.9	15.6
Prob. Adaptation	20.0 ± 29.8	60.0 ± 27.9	0.0 ± 0.0	26.7 ± 36.5	73.3 ± 14.9	13.3 ± 18.3	13.3 ± 18.3	0.0 ± 0.0	53.3 ± 29.8	**28.9**
Inverse Prob. Adaptation	13.3 ± 18.3	33.3 ± 23.6	0.0 ± 0.0	33.3 ± 33.3	40.0 ± 14.9	6.7 ± 14.9	13.3 ± 18.3	6.7 ± 14.9	60.0 ± 27.9	23.0

Table A10: **Visual Planning on iTHOR.** We report the mean success rate across 9 object navigation tasks in iTHOR. Each table entry shows the average success rate aggregated from 5 seeds. "$*$" denotes seen tasks during adaptation.

We provide additional experimentation of adaptation techniques on iTHOR (Eric et al., 2017), in which a mobile robotic agent is asked to perform egocentric navigation to a specified target object in different scenes. This benchmark poses challenges of navigating in partially observable settings and allows us to further evaluate adaptation methods on in-domain video generation from egocentric views. To perform adaptation, we reuse the video dataset provided by AVDC (Ko et al., 2024), which spans 12 target objects and includes 25 successful navigation trajectories for each object. We provide the success rates of visual planning across 9 navigation tasks in Table A10, in which 4 tasks are unseen during adaptation. We provide a detailed list of iTHOR tasks along with their corresponding text prompts in Table A3. In Table A10, we again observe that the overall performance of both probabilistic adaptation and its inverse outperform that of in-domain-only baseline by a large margin, highlighting that the internet knowledge of pretrained video models can be effectively utilized for various downstream robotic applications through proper adaptation. This result further highlights how adaptation can be flexibly applied across varied robotic settings.

H Step Counts to Task Success in Close-loop Visual Planning

Step Count w/	Door Close*	Door Open	Window Close	Window Open	Drawer Close
In-Domain-Only	80.0	-	176.0	344.0	25.3
Vanilla AnimateDiff	122.3	-	323.0	217.3	160.9
Direct Finetuning	96.0	-	159.2	333.3	63.8
Subject Customization	150.5	-	-	297.3	238.7
Prob. Adaptation	75.4	-	171.5	312.0	31.0
Inverse Prob. Adaptation	87.0	-	222.6	-	35.8

Step Count w/	Drawer Open	Coffee Push*	Soccer	Button Press
In-Domain-Only	-	-	-	204.0
Vanilla AnimateDiff	-	-	-	-
Direct Finetuning	-	-	-	-
Subject Customization	-	155.0	-	-
Prob. Adaptation	244.0	52.0	-	183.2
Inverse Prob. Adaptation	-	-	144.0	192.0

Table A11: **Step Counts of Visual Planning on MetaWorld.** We report the average number of taken steps in successful evaluation rollouts across 9 manipulation tasks in MetaWorld. Unsuccessful rollouts are omitted. We observed that probabilistic adaptation in general achieves task success using fewer number of steps.

I Policy Supervision with Additional Pretrained Video Models

Success Rate (%) w/	Door Close*	Door Open	Window Close	Window Open	Drawer Close
In-Domain-Only	100.0 ± 0.0	31.1 ± 44.0	0.0 ± 0.0	33.3 ± 47.1	74.4 ± 36.2
Vanilla AnimateLCM	100.0 ± 0.0	0.0 ± 0.0	98.9 ± 1.9	33.3 ± 29.1	100.0 ± 0.0
Prob. Adaptation	100.0 ± 0.0	0.0 ± 0.0	66.7 ± 57.7	0.0 ± 0.0	100.0 ± 0.0
Inverse Prob. Adaptation	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	94.4 ± 9.6	100.0 ± 0.0

Success Rate (%) w/	Drawer Open	Coffee Push*	Soccer	Button Press	**Overall**
In-Domain-Only	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	33.3 ± 47.1	30.2
Vanilla AnimateLCM	0.0 ± 0.0	5.6 ± 9.6	0.0 ± 0.0	0.0 ± 0.0	37.5
Prob. Adaptation	0.0 ± 0.0	32.2 ± 28.0	4.4 ± 7.7	0.0 ± 0.0	33.7
Inverse Prob. Adaptation	16.7 ± 29.0	41.1 ± 15.0	4.4 ± 5.1	30.0 ± 52.0	**65.2**

Table A12: **Policy Learning on MetaWorld with AnimateLCM.** We report the mean success rate across 9 manipulation tasks in MetaWorld, aggregated over 3 seeds.

We also provide policy supervision results on MetaWorld with AnimateLCM (Wang et al., 2024b) in Table A12. Similar to AnimateDiff, vanilla AnimateLCM is also able to achieve decent success rates through Video-TADPoLe. Furthermore, we discover that inverse probabilistic adaptation consistently achieves the best performance with both AnimateDiff and AnimateLCM. With AnimateLCM, inverse probabilistic adaptation obtains the highest overall success rate of **65.2%**, surpassing all other evaluated video models and adaptation techniques, with non-zero success rates across all evaluated tasks.

J Limitations

We observe that the FVD visual quality metric, while overall correlated with task completion success rate, is not always a clear indicator of an adapted model's performance on policy learning or planning. A better metric that can be offline computed is needed to select proper model checkpoints, and also to better understand the connection between generation quality and utility for robotic tasks. Similarly, we do not yet fully understand the performance differences when the same adapted model is utilized to solve the tasks under different setups. For example, while inverse probabilistic adaptation outperforms its inverse version significantly under the policy learning setup, both achieve similar success rates under the video planning setup. Finally, a natural next step is to further study the use of suboptimal data, which can be offline collected by running random policies on seen and unseen tasks, or iteratively augmented from on-policy observations.