SOLVING ROBOTIC TASKS VIA SELF-ADAPTING IMPROVEMENT LOOPS WITH INTERNET VIDEO KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generative models have been recently applied to robotic applications as visual planners. However, such visual planning models are generally trained on in-domain expert data, which may potentially be expensive to collect. Recent work has shown that instead, an in-domain video model trained on suboptimal data can be composed with a video model trained on internet-scale data to produce a performant video planner capable of generating high-quality trajectories during interaction with the environment. In this work, we investigate if utilizing these improved trajectories to update the in-domain model in a virtuous cycle can facilitate further downstream robotic task performance over multiple iterations. We present the Self-Adapting Improvement Loop (SAIL), where an in-domain model initially trained on only suboptimal demonstration data is iteratively adapted to the trajectories synthesized when using it as an adapted visual planner, *without any reward annotation or heuristical data filtering*. We apply SAIL on a large suite of MetaWorld tasks unseen during initial in-domain training, and find that improvements do continuously emerge over multiple iterations, thus demonstrating a way to iteratively bootstrap a high-performance video model for solving novel robotic tasks from cheap, suboptimal data through self-improvement.

1 INTRODUCTION

Advancements in video generative modeling capabilities have directly led to their increased application as visual planners for robotic applications (Du et al., 2024b; Yang et al., 2023b; Ko et al., 2024; Liang et al., 2024). Such visual planners are usually trained explicitly on expert in-domain demonstrations, which communicate not only environment-specific visual characteristics, physics, and interaction dynamics to the generative model during optimization, but also a notion of success and optimal behavior. However, for arbitrary environments, in-domain data can be expensive to collect and curate at scale, particularly when trajectories are required to be of expert quality. A potential consequence of training on a small dataset is limited generalization capability. On the other hand, suboptimal demonstration data, such as utilizing random actions during the collection procedure, may generally be cheaper to gather at scale; however, training on a large dataset of poor-quality data may not result in a performant visual planning model capable of generating plans worth following.

Recent work (Luo et al., 2025) has proposed a way to address these considerations on data quality and dataset size simultaneously. They demonstrate that a powerful, generalizable visual planner can be created by adapting a large-scale model pretrained on web-scale video data with a video model trained on a small set of in-domain demonstrations via score composition. At a high level, the adapted video model draws upon large-scale motion priors and powerful zero-shot text conditioning capabilities from the web-pretrained video model to facilitate generalization. Simultaneously, it can leverage the in-domain video model to better generate visual plans that respect the environment-specific visual characteristics and dynamics of the robotic setting.

Crucially, in investigating the extent that integrating in large-scale motion priors benefits visual planning, it was discovered that adaptation reduces a reliance on expert-quality data. Combining an in-domain model trained on suboptimal-only data with large-scale video model via score composition adaptation was shown to still successfully generate performant video plans. As neither in-domain

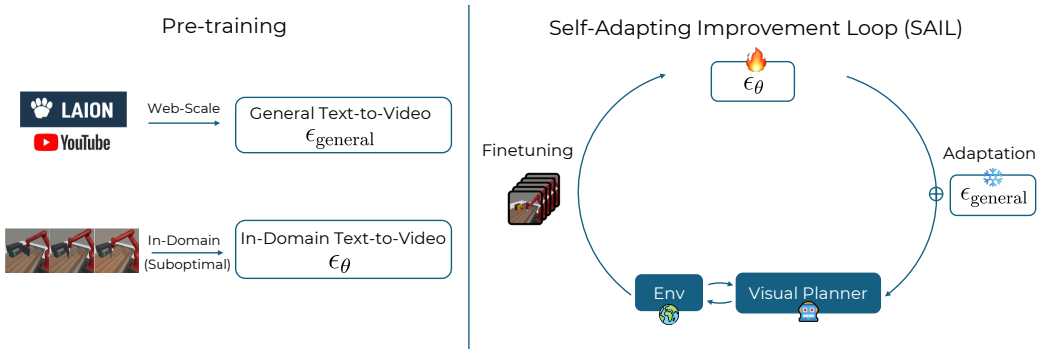

Figure 1: In the pretraining stage, a general text-to-video model is trained on web-scale text-annotated video datasets, and an in-domain text-to-video model is trained on a small set of suboptimal demonstrations of task behavior. Composing these two components results in a visual planner, which when utilized to interact with the environment, produces trajectories with improved success rate. In the Self-Adapting Improvement Loop (SAIL), these trajectories are then fed back to finetune the in-domain model, thus improving the overall quality of the adapted visual planner as a whole. In such a way, performance on a novel task can be iteratively improved upon, bootstrapping from two models that have never initially seen successful expert task demonstrations during the pretraining stage.

model nor generally-pretrained model had observed successful task demonstrations during training, this suggests that large-scale motion priors, and specified by text conditioning, may help to mitigate the optimality gap.

In this work, we leverage this discovery to design a Self-Adapting Improvement Loop (SAIL) for solving novel robotic tasks. As the resulting trajectories are of higher quality than the initial dataset used to train the in-domain model, we propose using them to further finetune the in-domain model. In such a way, the overall visual planner improves its performance by adapting itself to trajectories collected through its application.

We perform extensive evaluations of SAIL on the MetaWorld task suite, including on novel tasks unseen during initial training of the in-domain model. We discover that the success rate of following such visual adapted plans indeed improves over iterations. We highlight that this is accomplished without requiring any data filtering or task-reward labeling; the in-domain model is bootstrapped purely from a set of suboptimal demonstration data and improved using the trajectories it collects from its own performance as an adapted visual planner. Our work therefore highlights how a robotic planner can iteratively adapt to its own collected samples, and learn to improve its task performance without expensive human curation or overhead.

2 Related Work

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.

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., 2024; 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 solving robotic control tasks and evaluate whether a high-performing visual planner can be bootstrapped from a suboptimal visual planner via a self-adapting improvement loop.

Adaptation Techniques for Video 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. DreamVideo (Wei et al., 2024) 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). Furthermore, a variant of the probabilistic adaptation technique is proposed by (Luo et al., 2025) to perform score composition in an inverted direction from that presented in (Yang et al., 2023a). In this work, we evaluate probabilistic adaptation and its inverse to explore the degree to which a suboptimal in-domain video model can be improved iteratively through a self-adaptation loop.

3 Method

We introduce Self-Adapting Improvement Loop (SAIL), in which we iteratively improve a visual planning model initially trained on suboptimal in-domain demonstrations to a performant one in a self-adaptive manner. In Section 3.1, we first introduce how video models can be used as visual planners for solving decision making problems. In Section 3.2, we describe the probabilistic adaptation techniques that integrate a small in-domain video model with one generally pretrained on web-scale data to produce a strong, generalizable in-domain visual planner. Finally, we demonstrate how, through an iterative fine-tuning loop, we can bootstrap a suboptimal in-domain video model into a high-performing visual planner that is able to solve novel robotic control tasks in Section 3.3.

3.1 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), across a variety of robot configurations and environment settings.

Specifically, we base our implementation on the UniPi framework (Du et al., 2024a), in which the text-to-video model is used to synthesize a text-conditioned sequence of future frames as a task plan. To physically realize 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 Probabilistic Adaptation and Its Inverse

Probabilistic Adaptation (Yang et al., 2023a) is a training-free approach that adapts generally pretrained text-to-video models for domain-specific video generation. To perform adaptation, the score predicted by an in-domain video model ϵ_θ trained on a small sample of demonstrations, is composed with the score prediction of a web-scale pretrained model $\epsilon_{\text{general}}$ during the sampling procedure, as depicted in the function below:

$$\tilde{\epsilon} = \epsilon_\theta(\tau_t, t) + \alpha\Big(\epsilon_\theta(\tau_t, t \mid \text{text}) + \gamma\epsilon_{\text{general}}(\tau_t, t \mid \text{text}) - \epsilon_\theta(\tau_t, t)\Big) \tag{1}$$

where γ is the prior strength, and α is the guidance scale of text-conditioning. Intuitively, the general text-to-video model serves as a probabilistic knowledge prior that guides the generation process of the small in-domain model during sampling. Moreover, Probabilistic Adaptation has been extended to its inverse version (Luo et al., 2025) by inverting the adaptation direction:

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

where the small video model now serves as the probabilistic prior to facilitate adapted video generation. While Probabilistic Adaptation is initially proposed for generating high-quality yet specialized videos, prior work (Luo et al., 2025) has adopted this technique and its inverse to construct visual planners for solving robotic control tasks. When using expert demonstrations for in-domain training, these adapted video planners exhibit both strong generalization capability and in-domain understanding, allowing them to effectively solve even tasks unseen during the training of the video models.

3.3 Self-Adapting Improvement Loop

Collecting expert demonstrations for robotic control tasks can often be costly in many scenarios, making it difficult to scale up the in-domain training process. On the other hand, although suboptimal demonstrations lack the expert examples that visual planners can copy from to solve the task directly, they still contain valuable in-domain information, such as visual characteristics and environment dynamics, which remain crucial for visual planning.

More importantly, prior work (Luo et al., 2025) discovered that probabilistic adaptation techniques are able to mitigate this data optimality gap by leveraging the large-scale pre-trained video prior, and consistently improve visual planning performance even when only suboptimal demonstrations are available for in-domain training. Inspired by this, we propose Self-Adapting Improvement Loop (SAIL), where we can iteratively improve a suboptimal in-domain model via a self-adaptive manner (as in Figure 1 right).

Specifically, we initialize SAIL with an in-domain video model ϵ_θ pre-trained on suboptimal demonstrations. In each iteration, the in-domain video model is adapted with a large-scale pretrained video model $\epsilon_{\text{general}}$ through probabilistic adaptation techniques. The adapted video model serves as a visual planner to interact with the environment and solve novel tasks that are not observed in the initial training stage. During visual planning, we collect a small dataset of trajectories rendered by the environment for further in-domain finetuning. Based on the discovery in (Luo et al., 2025), the adapted video model can achieve better task performance than the unadapted in-domain video model when functioning as a visual planner, indicating more performant trajectories are collected when using the adapted visual planner. We then leverage these high-performing trajectories to refine the in-domain video model and eventually bootstrap a strong in-domain model through the self-adapting improvement cycle.

4 Experiments

4.1 Experimental Setup and Evaluation

Benchmarks: We evaluate to what degree SAIL can improve the in-domain video model initially trained on suboptimal demonstrations and further solve novel robotic control tasks. We choose MetaWorld-v2 (Yu et al., 2020) as our evaluation benchmark, which offers a suite of robotic manipulation tasks with different levels of complexity. This benchmark allows us to thoroughly assess visual planning performance with SAIL across a wide selection of tasks. We curate a small dataset of in-domain demonstrations from 7 MetaWorld tasks (denoted with an asterisk in Table A1) for initial in-domain training, in which 25 suboptimal videos are used for each task. During inference, we evaluate the adapted visual planners on 9 tasks, 7 of which are novel tasks that are not exposed during initial in-domain training (denoted with no asterisk in Table A1). In each SAIL iteration, we collect 25 trajectories rendered from the environment during visual planning for further in-domain finetuning.

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. 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 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.

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.

In-Domain Only	Iteration 0	Iteration 1	Iteration 2
Door-Close*	98.7 ± 2.3	100 ± 0	96.0 ± 4.0
Door-Open	0 ± 0	0 ± 0	0 ± 0
Drawer-Close	13.3 ± 8.3	13.3 ± 10.6	28.0 ± 14.4
Drawer-Open	1.3 ± 2.3	2.6 ± 2.3	1.3 ± 2.3
Window-Close	53.3 ± 16.7	52.0 ± 16.0	46.7 ± 9.2
Window-Open	16.0 ± 8.0	9.3 ± 4.6	13.3 ± 8.3
Coffee-Push*	0 ± 0	0 ± 0	0 ± 0
Button-Press	0 ± 0	2.7 ± 2.3	6.7 ± 2.3
Soccer	0 ± 0	0 ± 0	0 ± 0
Average	20.3	20.0	21.3

Table 1: **SAIL without Adaptation** We report the mean success rate via visual planning across 9 tasks, aggregated over 3 seeds each. No apparent improvement over average performance can be observed across iterations.

Prob. Adaptation	Iteration 0	Iteration 1	Iteration 2
Door-Close*	98.7 ± 2.3	93.3 ± 4.6	96.0 ± 4.0
Door-Open	0 ± 0	0 ± 0	0 ± 0
Drawer-Close	21.3 ± 2.3	48.0 ± 13.8	41.3 ± 11.5
Drawer-Open	0 ± 0	2.7 ± 2.3	0 ± 0
Window-Close	49.3 ± 6.1	65.3 ± 8.3	68.0 ± 6.9
Window-Open	6.7 ± 6.1	5.3 ± 6.1	9.3 ± 2.3
Coffee-Push*	0 ± 0	0 ± 0	0 ± 0
Button-Press	0 ± 0	2.7 ± 4.6	1.3 ± 2.3
Soccer	0 ± 0	0 ± 0	0 ± 0
Average	19.6	24.1	24.0

Table 2: **SAIL with Probabilistic Adaptation** We report the mean success rate via visual planning across 9 tasks, aggregated over 3 seeds each. Performance on Window-Close continuously improves over iterations, whereas the average performance shows improvement over the first few iterations and quickly saturates in the last iteration.

4.2 Visual Planning with SAIL

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 B.

We evaluate SAIL with three settings: Probabilistic Adaptation, Inverse Probabilistic Adaptation and no adaptation. For each setting, we initialize with the same in-domain video model, perform

Inverse Prob. Adaptation	Iteration 0	Iteration 1	Iteration 2
Door-Close*	100 ± 0	100 ± 0	100 ± 0
Door-Open	0 ± 0	0 ± 0	0 ± 0
Drawer-Close	22.6 ± 4.6	36.0 ± 14.4	56.0 ± 18.3
Drawer-Open	1.3 ± 4.3	1.3 ± 2.3	1.3 ± 2.3
Window-Close	34.7 ± 2.3	66.7 ± 14.8	68.0 ± 10.6
Window-Open	24.0 ± 8.0	10.7 ± 10.1	10.7 ± 2.3
Coffee-Push*	0 ± 0	0 ± 0	0 ± 0
Button-Press	0 ± 0	5.3 ± 2.3	8.0 ± 4.0
Soccer	2.6 ± 4.6	1.3 ± 2.3	1.3 ± 2.3
Average	20.6	24.6	**28.4**

Table 3: **SAIL with Inverse Probabilistic Adaptation** We report the mean success rate via visual planning across 9 tasks, aggregated over 3 seeds each. Continuous improvements are observed in Drawer-Close, Window-Close, Button-Press as well as the average performance over multiple iterations. Furthermore, it achieves the best visual planning performance in the last SAIL iteration across all evaluation settings.

three SAIL iterations, and report the visual planning performance in every iteration in the tables below. While improvements can be barely observed across iterations without any adaptation in Table 1, Table 2 and Table 3 consistently show an improving trend over average performance, highlighting the effectiveness of self-adaptation. From Table 3, we discover that Inverse Probabilistic Adaptation enables a continuously improving behavior over iterations on three unseen tasks and average performance, and achieves the highest success rate in Iteration 2. On the other hand, in Table 2, only one unseen task shows a similar improving behavior with probabilistic adaptation, and the average performance also quickly saturates after Iteration 1. Compared to probabilistic adaptation, we believe its inverse variant serves as a more robust adaptation technique, especially with suboptimal in-domain initialization, which allows more performant trajectories to be collected through visual planning, constantly improving the in-domain video model through the self-adaptation loop.

5 Conclusion and Future Work

In this work, we propose SAIL, a self-adapting improvement loop for solving novel robotic tasks via visual planning. SAIL initially starts from an in-domain video model pretrained on a small set of suboptimal data, as well as a large-scale video model pretrained on general internet data. By utilizing the composition of these two models as a visual planner, then executing synthesized visual plans in an environment for a desired task, performant trajectories can be generated. SAIL utilizes these improved trajectories to further finetune the indomain model, and repeats this sequence of interactions and updates over multiple iterations. We evaluate SAIL on an extensive suite of MetaWorld tasks, with task performance increasing with iteration count in particular for inverse probabilistic adaptation. Notably, we successfully deploy SAIL to novel robotic tasks, where even suboptimal demonstrations are not provided during in-domain model training. Furthermore, no human data filtering is applied on the trajectories produced by following the visual planner, and reward information from the environment is not utilized in any capacity. We therefore highlight how SAIL can be used to iteratively adapt itself to solve novel robotic tasks, starting from a model trained on cheaply generated suboptimal data.

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. 2, 3

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. 5

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. 2, 3, 5

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, 2, 3

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. 2

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, 5

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

Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. *arXiv preprint arXiv:2312.14134*, 2023. 2

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, 2, 3, 5

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, 2

Calvin Luo, Zilai Zeng, Yilun Du, and Chen Sun. Solving new tasks by adapting internet video knowledge. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=p01BR4njlY. 1, 3, 4

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. 2

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. 10

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. 2

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

Pierre Sermanet, Kelvin Xu, and Sergey Levine. Unsupervised perceptual rewards for imitation learning. *arXiv preprint arXiv:1612.06699*, 2016. 2

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. 9

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*, 2024. 2

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

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. 2

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

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, 2

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. 2

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. 4

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. 2

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. 2

A TEXT PROMPTS

Task	In-Domain Prompts	AnimateDiff Prompts
Assembly*	assembly	a robot arm placing a ring over a peg
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 A1: **Task-Prompt Pairs.** We include a comprehensive list of tasks and their text prompts for adaptation and evaluation. "$*$" denotes tasks seen during adaptation.

B IMPLEMENTATION DETAILS

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

Table A2: **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 A3: **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
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 A4: **Hyperparameters for In-Domain Model Training.**

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

Table A5: **Hyperparamters of Inverse Dynamics 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 2.5 as the text-conditioning guidance scale. Additionally, we use 0.1 as the prior strength for probabilistic adaptaion and 0.5 for its inverse version.

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 A5.