When Should We Prefer State-to-Visual DAgger Over Visual Reinforcement Learning?

Tongzhou Mu*, Zhaoyang Li*, Stanisław Wiktor Strzelecki*, Xiu Yuan, Yunchao Yao, Litian Liang, Hao Su

University of California San Diego
{t3mu, zhl165, sstrzelecki, x1yuan, y8yao, l6liang, haosu}@ucsd.edu

Abstract

Learning policies from high-dimensional visual inputs, such as pixels and point clouds, is crucial in various applications. Visual reinforcement learning is a promising approach that directly trains policies from visual observations, although it faces challenges in sample efficiency and computational costs. This study conducts an empirical comparison of State-to-Visual DAgger — a two-stage framework that initially trains a state policy before adopting online imitation to learn a visual policy — and Visual RL across a diverse set of tasks. We evaluate both methods across 16 tasks from three benchmarks, focusing on their asymptotic performance, sample efficiency, and computational costs. Surprisingly, our findings reveal that State-to-Visual DAgger does not universally outperform Visual RL but shows significant advantages in challenging tasks, offering more consistent performance. In contrast, its benefits in sample efficiency are less pronounced, although it often reduces the overall wall-clock time required for training. Based on our findings, we provide recommendations for practitioners and hope that our results contribute valuable perspectives for future research in visual policy learning.

Code — https://github.com/tongzhoumu/s2v-dagger
Extended version — https://arxiv.org/abs/2412.13662

1 Introduction

Learning policies from high-dimensional visual observations, such as pixels and point clouds, is a crucial problem in fields like robotic manipulation (Nair et al. 2018; Ze et al. 2024; Hansen, Wang, and Su 2022), navigation (Gu et al. 2022), and autonomous driving (Hossain 2023). Visual reinforcement learning (RL) methods, which employ RL algorithms on visual observations, stand out as a leading approach for acquiring such visual policies. Despite their popularity, visual RL methods are generally more prone to issues related to sample efficiency and computational costs than their counterparts utilizing low-dimensional state observations (Chen et al. 2023). This is primarily because visual RL must address two challenges *concurrently*: 1) figuring out how to solve the task through trial and error; and 2) building a mapping from high-dimensional visual observations to the high-rewarding actions, a process that often involves training a large visual encoder.

A potential simplification of this problem is to tackle the two aforementioned challenges separately. Previous studies have utilized a two-stage approach for learning a visual policy, as illustrated in Fig. 1. In the first stage, a teacher policy is trained using RL with low-dimensional state observations, possibly incorporating privileged information to facilitate learning. In the second stage, a visual policy is learned by online imitating the teacher policy, akin to DAgger (Ross, Gordon, and Bagnell 2011). This two-stage framework has been applied across various applications, including dexterous manipulation (Chen, Xu, and Agrawal 2022; Chen et al. 2023), legged locomotion (Lee et al. 2020; Miki et al. 2022; Zhuang et al. 2023; Margolis et al. 2021), drone control (Loquercio et al. 2021), and autonomous driving (Chen et al. 2020). In our study, this two-stage framework is referred to as "State-to-Visual DAgger", highlighting the transition from *low-dimensional state observations* to *high-dimensional visual observations*.

While State-to-Visual DAgger can simplify the learning of visual policies by isolating focus at each stage, the added stage complicates training and may increase costs compared to single-stage visual RL methods. Therefore, our study explores the question: ***When should State-to-Visual DAgger be preferred over visual RL?***

We investigate this question empirically by comparing State-to-Visual DAgger against visual RL across diverse tasks and evaluation metrics. We selected **16** tasks from three benchmarks: ManiSkill (Gu et al. 2023), DMControl (Tassa et al. 2018), and Adroit (Rajeswaran et al. 2017). These tasks include stationary robot arm manipulation, mobile manipulation, dual-arm coordination, locomotion across different robot morphologies, classical control, and dexterous hand manipulation. Our comparison evaluates *asymptotic performance, sample efficiency, and computational cost*, offering a comprehensive assessment of both methods.

The fairness of the comparison also hinges on the implementations. Despite its usage in several publications, a standardized implementation of State-to-Visual DAgger has yet to be established. We meticulously developed our implementation of it, pinpointing several critical design decisions that significantly influence its performance. Further details

*These authors contributed equally.

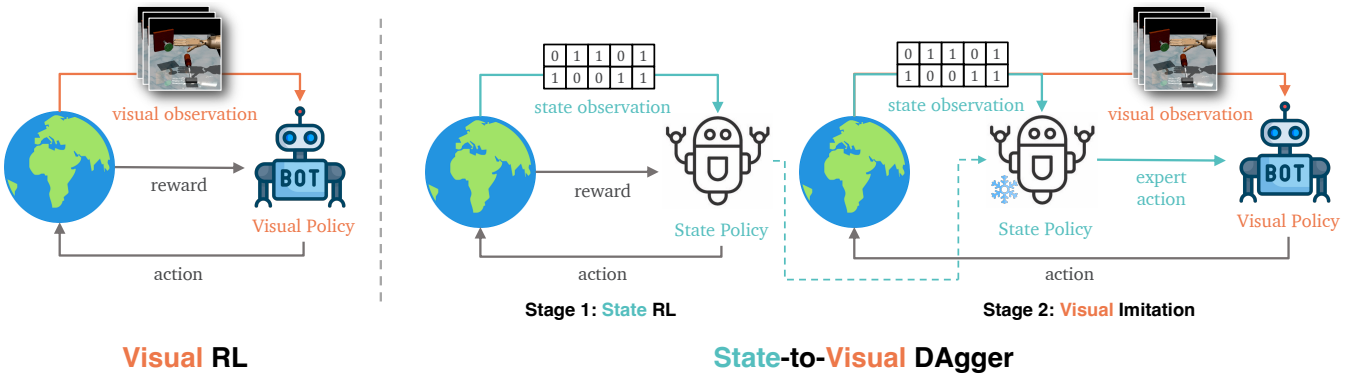

Figure 1: **Overview of Methods.** While Visual RL directly trains a visual policy using RL, State-to-Visual DAgger adopts a two-stage process: initially training a teacher policy with low-dimensional state observations, followed by teaching a visual policy via online imitation learning.

on this are elaborated in Sec. 3. Our evaluation revealed that State-to-Visual DAgger does not consistently outperform visual RL, with key findings summarized below.

Regarding asymptotic performance: State-to-Visual DAgger demonstrates significant advantages over visual RL in hard tasks, but only achieves similar or slightly worse performance in easy tasks. Notably, State-to-Visual DAgger usually provides more consistent and stable performance upon convergence.

Regarding efficiency: In scenarios where both State-to-Visual DAgger and visual RL are capable of effectively solving tasks, State-to-Visual DAgger does not distinctly outperform visual RL in terms of sample efficiency. Nevertheless, State-to-Visual DAgger significantly improves wall-clock efficiency across the most tasks.

For a more in-depth discussion of our findings, please refer to Sec. 5. Based on these empirical results, we also offer guidance for practitioners in Sec. 6. **Our contributions** can be summarized as follows:

- We delve into the crucial question of "when State-to-Visual DAggershould be preferred over visual RL," facilitated by a detailed comparison of a diverse set of tasks.
- Our analysis offers key insights and practical guidance for researchers and practitioners in visual policy learning.
- We offer a standardized implementation of State-to-Visual DAgger and meticulously analyze several key design choices that significantly influence its performance.

2 Related Works

Visual Reinforcement Learning: Visual reinforcement learning (visual RL) integrates complex visual inputs, such as pixels and point clouds, into reinforcement learning algorithms, enabling agents to make decisions based on these observations. Visual RL can be categorized into model-free and model-based approaches. Model-free methods are divided into value-based and policy-based approaches. Value-based methods, such as those in (Mnih et al. 2015; Silver, Wierstra, and Riedmiller 2013), combine Q-learning with deep neural networks to learn from raw pixel inputs using convolutional neural networks. Policy-based methods, including (Klimov 2017; Haarnoja et al. 2018), optimize agents using policy gradients. For model-based visual RL, the agent needs to learn a world model from visual observations. Approaches such as PlaNet (Hafner et al. 2019b), Dreamer (Hafner et al. 2019a), Dreamer-v2 (Hafner et al. 2020), and TD-MPC (Hansen, Wang, and Su 2022) focus on learning dynamics from images and planning actions in latent spaces, with enhancements for discrete and continuous environments. Representation learning enhances visual RL performance, with prior works exploring pre-training using single-view (Shah and Kumar 2021; Parisi et al. 2022), multi-view (Driess et al. 2022), and video data (Kulkarni et al. 2019). Additionally, self-supervised learning (Laskin, Srinivas, and Abbeel 2020) and data augmentation (Yarats, Kostrikov, and Fergus 2020) enhance performance without pre-training. Practical applications include QT-Opt (Kalashnikov et al. 2018) for real-world robotics manipulation and Akkaya et al.'s work enabling a robotic hand to solve a Rubik's Cube (Akkaya et al. 2019). However, visual RL faces challenges in sample efficiency and computational costs compared to low-dimensional approaches (Chen et al. 2023), and it struggles with computational efficiency and generalizability across different visual domains.

Utilize Privileged Information During RL Training: Privileged information can accelerate visual RL learning and improve sampling efficiency. While unavailable during deployment, it is often accessible during training and can be strategically utilized. For example, Kaufmann et al. (2023) uses privileged information about the highly accurate simulation of drone dynamics and environment and optimal race routes to help RL models train more effectively. Some methods, such as those described in (Pinto et al. 2017; Kumar et al. 2021), utilize simulation information to provide detailed and controlled feedback on actions within a simulated environment, thus enhancing the robustness of the RL policy. Besides using available privileged information during RL, some methods follow the teacher-student approach we call State-to-Visual DAgger, such as training the policy as the expert and then using the privileged information from the expert to supervise the student model. State-to-Visual

DAgger has been used in applications about autonomous driving (Chen et al. 2020), legged locomotion (Lee et al. 2020; Miki et al. 2022; Zhuang et al. 2023), drone control (Loquercio et al. 2021), dexterous grasp (Xu et al. 2023), and dexterous manipulation (Chen et al. 2023), which utilize the privileged information to depth. Although previous work does not investigate whether this State-to-Visual DAggerimproves learning efficiency, we focus on investigating the learning efficiency of State-to-Visual DAgger compared to single-stage visual RL, and clarify what situation we need to use State-to-Visual DAgger.

3 Methods

Our study aims to conduct a comparison between two distinct paradigms to learning visual policies: State-to-Visual DAgger and visual Reinforcement Learning. To provide a focused examination, we focus on representative methods within each paradigm. Given the absence of a standardized open-source implementation of State-to-Visual DAgger, we have carefully developed our version, identifying several crucial design choices that significantly affect its performance. Visual RL encompasses a wide range of approaches, each with its own strengths. For a fair comparison, we chose Asymmetric Actor Critic (Pinto et al. 2017) as the visual RL counterpart in this study. This method was selected due to its ability to incorporate privileged state information, similar to the advantage used by State-to-Visual DAgger. This section details the design and implementation of these methods.

State-to-Visual DAgger

State-to-Visual DAgger adopts a two-stage approach to learning visual policies, as depicted in Fig. 1. This method requires the training environment to *concurrently* supports two observation spaces: a low-dimensional state observation space denoted as $\mathbb{O}^{\mathcal{S}}$, and a high-dimensional visual observation space $\mathbb{O}^{\mathcal{V}}$. This approach usually assumes training in a simulator, which offers both the full system state and rendered images. However, the final visual policy learned by State-to-Visual DAggerdoes not rely on any simulator-specific privileged information.

Stage 1: Learning State Policy by RL. In the initial stage, we employ Soft Actor-Critic (SAC) (Haarnoja et al. 2018), a widely used RL algorithm, to train state-based a teacher policy $\pi^{\mathcal{S}}$ using low-dimensional state observations. Here, state observation refers to a low-dimensional vector that describes the current state, often incorporating privileged information not available during real-world policy deployment. For instance, in the PickCube task from ManiSkill, the state observation includes both the robot's proprioception data and the ground truth pose of the cube. While the robot proprioception data can be accessible in the real world, the ground truth pose of the cube typically is not. Our experiments directly employ the low-dimensional state observations provided by the environment's interface (more details in the Appendix B of the extended version.) The learned state-based teacher policy will guide the subsequent learning process of visual policy. In our experiments, the training of stage 1 is stopped upon convergence, and we save the latest checkpoint. Alternatively, the final checkpoint could be selected based on a predetermined computational budget.

Stage 2: Learning Visual Policy by DAgger. In Stage 2, the state policy acquired from Stage 1 serves as a teacher to guide the learning of the visual policy. This is achieved by using DAgger (Ross, Gordon, and Bagnell 2011), an online imitation learning algorithm. DAgger's primary advantage over traditional offline imitation methods lies in its ability to mitigate the covariate shift problem by leveraging an expanding from online interactions. The training of the visual policy $\pi^{\mathcal{V}}$ is done by minimizing the MSE loss on actions, formulated as:

$$\pi^{\mathcal{V}} = \text{argmin}_{\pi^{\mathcal{V}}} \|\pi^{\mathcal{V}}(o_t^{\mathcal{V}}) - \pi^{\mathcal{S}}(o_t^{\mathcal{S}})\|^2, \tag{1}$$

where $o_t^{\mathcal{V}}$ and $o_t^{\mathcal{S}}$ are paired visual observation and state observation. Our implementation of DAgger for learning visual policies incorporates two critical design decisions:

1. DAgger can be implemented in both on-policy and off-policy manners, analogous to the methods used in on-policy and off-policy reinforcement learning. The primary distinction lies in whether to incorporate off-policy trajectories into the training dataset via a replay buffer. Our experiments demonstrate that the off-policy version significantly outperforms the on-policy variant, likely due to its ability to retain a more diverse set of training examples.
2. Rather than employing a fixed number of gradient updates per training round, we utilize an early-stopping mechanism triggered when a predefined imitation loss threshold is reached. After early stopping, a new cycle of data collection is initiated through interaction with the environment, followed by the integration of this new data into the buffer. This approach reduces unnecessary training on patterns that have already been learned, thereby preventing overfitting and enhancing training efficiency.

For a detailed description of our State-to-Visual DAgger implementation algorithm and related details, please refer to the Appendix C in the extended version of our paper.

Visual Reinforcement Learning

While numerous visual RL algorithms exist (Espeholt et al. 2018; Kostrikov, Yarats, and Fergus 2020; Laskin et al. 2020; Hafner et al. 2019c), a direct comparison with the State-to-Visual DAgger method may not be entirely fair. This discrepancy arises because standard visual RL approaches do not leverage the privileged information that State-to-Visual DAgger capitalizes on, a factor that substantially aids in solving the tasks.

To ensure a more balanced comparison, we adopt the Asymmetric Actor Critic (Pinto et al. 2017) as the visual RL method for our study. This algorithm uniquely lets the critic take the state (including privileged information) as input, whereas the actor still operates on high-dimensional visual inputs. This design enables the utilization of privileged information without making the policy dependent on it. Originally, the Asymmetric Actor Critic employed DDPG

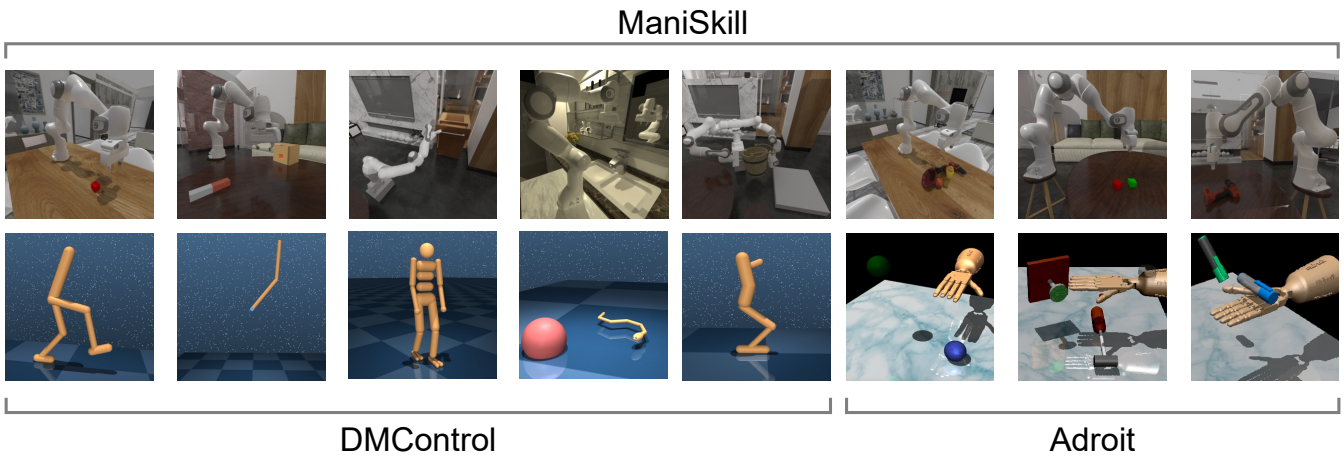

Figure 2: **Examples of Tasks.** We consider control tasks spanning **3** benchmarks. The first row contains tasks from ManiSkill (stationary and mobile robot arm manipulation, dual-arm coordination). The first five tasks in the second row are from DMControl (various robot morphologies for locomotion and classical control tasks), and the remaining three tasks in the second row are from Adroit (dexterous hand manipulation).

(Lillicrap et al. 2015) as its underlying RL algorithm; however, we opted for SAC to enhance performance. We refrained from incorporating advanced techniques for image-based feature extraction (Shang et al. 2021) and data augmentation (Laskin et al. 2020) that have been recently introduced, as their application to both State-to-Visual DAgger and visual RL would unlikely change the core findings of our study significantly.

Our empirical evaluations show that Asymmetric Actor Critic, when combined with SAC, matches the performance of state-of-the-art visual RL algorithms on the tasks we tested.This justifies its selection as the representative visual RL method for our comparisons. Details are in Appendix D of our extended version.

4 Experimental Setup

Our experimental setup is designed to thoroughly evaluate and compare the capabilities of two methods for learning visual policies, spanning a diverse range of tasks and employing specific evaluation metrics to gauge performance comprehensively. We discuss the details in this section.

Task Descriptions

Our experiments span **16** tasks across **3** benchmarks: ManiSkill (robotic manipulation; 8 tasks), DMControl (locomotion and control; 5 tasks), and Adroit (dexterous manipulation; 3 tasks). This diverse set includes stationary and mobile robot arm manipulation, dual-arm coordination, various robot morphologies for locomotion, classical control, and dexterous hand manipulation. The range ensures our conclusions are comprehensive and unbiased. Figure 2 illustrates all **16** tasks. Detailed task descriptions and setups are provided in Appendix D of the extended version of our work, summarized as follows:

ManiSkill: Features robotic manipulation tasks where low-dimensional state observations include robot proprioception (joint angles, joint velocities, end effector pose, base pose, etc.) and ground truth object or goal information, with visual observations from dual 64×64 RGBD cameras.

DMControl: We evaluate on locomotion and classical control tasks, following standard protocols (Kostrikov, Yarats, and Fergus 2020; Laskin et al. 2020; Hafner et al. 2019c). State observations primarily include robot proprioceptive data. Visual inputs are 84×84 RGB images, stacking 3 frames. We adopt action repeat parameters from (Kostrikov, Yarats, and Fergus 2020).

Adroit: Concentrates on dexterous manipulation tasks, with low-dimensional state observations detailing the information about all the joints as well as the pose of the palm and poses of other objects in the environment. Visual observations are 128×128 RGB images.

Evaluation Metrics

Our comparison of visual policy learning methods centers on two evaluation metrics: learning efficiency and asymptotic performance.

Learning Efficiency: We evaluate efficiency in terms of both sample efficiency (gauged by the number of environment steps) and computational cost (measured in wall-clock time), considering the cumulative costs of the two stages in State-to-Visual DAggerfor a balanced comparison. All experiments are standardized on the same hardware to ensure fair comparisons of computational costs. Our hardware setting: 32 CPU cores (Intel Xeon 2.1GHz) and 1 GPU (NVIDIA-GeForce-RTX-2080-Ti with 11GB).

Asymptotic Performance: To address the challenge of calculating asymptotic performance in RL experiments, we average data points over a window at the end of the learning curve to gauge this metric, with the window set at 3% of total environment steps.

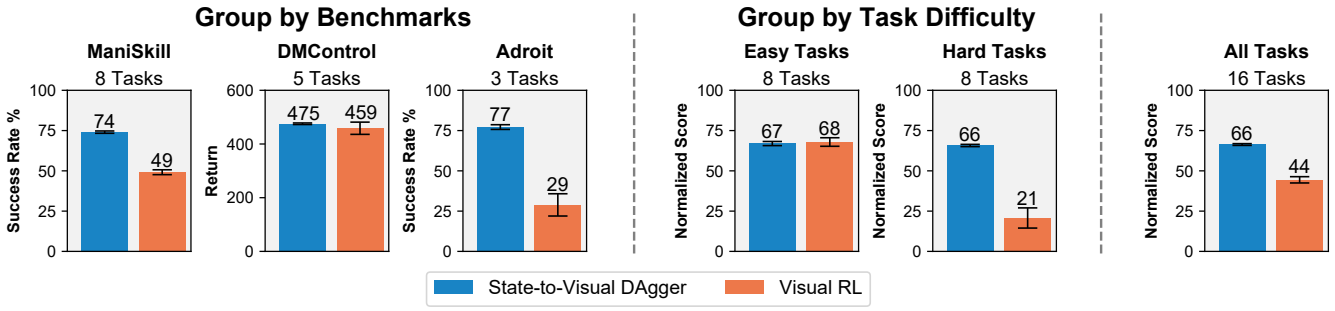

Figure 3: **Performance Overview.** The figure features histograms comparing average performance across different dimensions. On the left, three histograms present performance by benchmark (success rates for ManiSkill and Adroit, and returns for DMControl). In the center, two histograms categorize performance by task difficulty, utilizing normalized scores (success rate for ManiSkill and Adroit, return divided by 1000 for DMControl) to accommodate the varying metrics across benchmarks. The error bars represent the 95% CI over three seeds.

5 Results

In this section, We analyze the experimental results, highlighting key findings, with detailed implications and advice for practitioners in Sec. 6. All experiments use three random seeds, aggregating results across tasks for reliability.

Performance Comparison

Our results suggest that the efficacy of State-to-Visual DAgger compared to visual RL varies across tasks, as illustrated in Fig. 3 and Fig. 4. *There is no single approach that consistently outperforms the other across all tasks.* Specifically, State-to-Visual DAgger shows notable superiority in many tasks within the ManiSkill and Adroit benchmarks. Conversely, visual RL exhibits marginal benefits in the majority of tasks from the DMControl benchmark.

Given that previous works mainly apply State-to-Visual DAgger to exceptionally challenging tasks, such as dexterous manipulation (Chen et al. 2023) and drone control (Loquercio et al. 2021), categorizing tasks by their difficulty level may offer a clearer perspective. Here, we define "easy tasks" as those where state-based RL achieves convergence within 4 million environment steps, with all other tasks classified as "hard". Although this classification is not rigorous, it facilitates a more detailed comparison between State-to-Visual DAgger and visual RL. As illustrated in Fig. 3, *State-to-Visual DAgger markedly surpasses visual RL in hard tasks, but only achieves similar or slightly worse results in easier tasks.* The learning curves for each task, shown in Fig. 4, further validate this observation. While State-to-Visual DAgger excels at difficult tasks through imitation of state policies, its performance on easier tasks is comparable to or slightly below that of visual RL.

The performance gap in *hard tasks* stems from the disparity between state-based and visual RL. State-based RL with a simple MLP handles these tasks well (with dense rewards), while visual RL struggles. We hypothesize that noisy gradients during exploration impede CNN training with image observations.

Consistency and Stability

Our results also indicate that *State-to-Visual DAgger delivers more consistent performance at convergence*, as evidenced by the narrower confidence intervals observed across all benchmarks and difficulty levels, as illustrated in Fig. 3.

A closer look at individual task performances, as shown in Fig. 4, further shows that visual RL may exhibit fluctuating performance on certain tasks (e.g., ManiSkill OpenDrawer and Adroit Hammer), and may even unlearn (e.g., Adroit Pen). Conversely, *the performance of State-to-Visual DAgger (Stage 2) remains more stable upon convergence*, as indicated by the smoother learning curves. This stability is expected, as imitation learning in Stage 2 is inherently easier and more stable than reinforcement learning.It simplifies learning and streamlines deployment checkpoint selection.

Sample Efficiency (Environment Steps)

Comparing the sample efficiency of State-to-Visual DAgger and visual RL is not straightforward due to the inherent structure of State-to-Visual DAgger, where a visual policy is not trained until the second stage. Observations in Fig. 4 suggests State-to-Visual DAgger appears more sample-efficient than visual RL in hard tasks, primarily due to its superior asymptotic performance rather than true sample efficiency.

Conversely, when it comes to easier tasks where both methods converge to similar levels of performance, State-to-Visual DAgger does not demonstrate a clear advantage in sample efficiency over visual RL. This leads to the conclusion that the apparent higher sample efficiency of State-to-Visual DAgger in certain scenarios is more attributed to its enhanced asymptotic performance rather than an intrinsic efficiency advantage. Thus, *when both methods are capable of effectively solving tasks, State-to-Visual DAgger does not offer significant benefits in sample efficiency over visual RL.*

Computational Cost (Wall-clock Time)

Although State-to-Visual DAgger may not enhance sample efficiency, it excels in wall-clock time, consistently outperforming visual RL across all tasks as shown in Fig. 5.

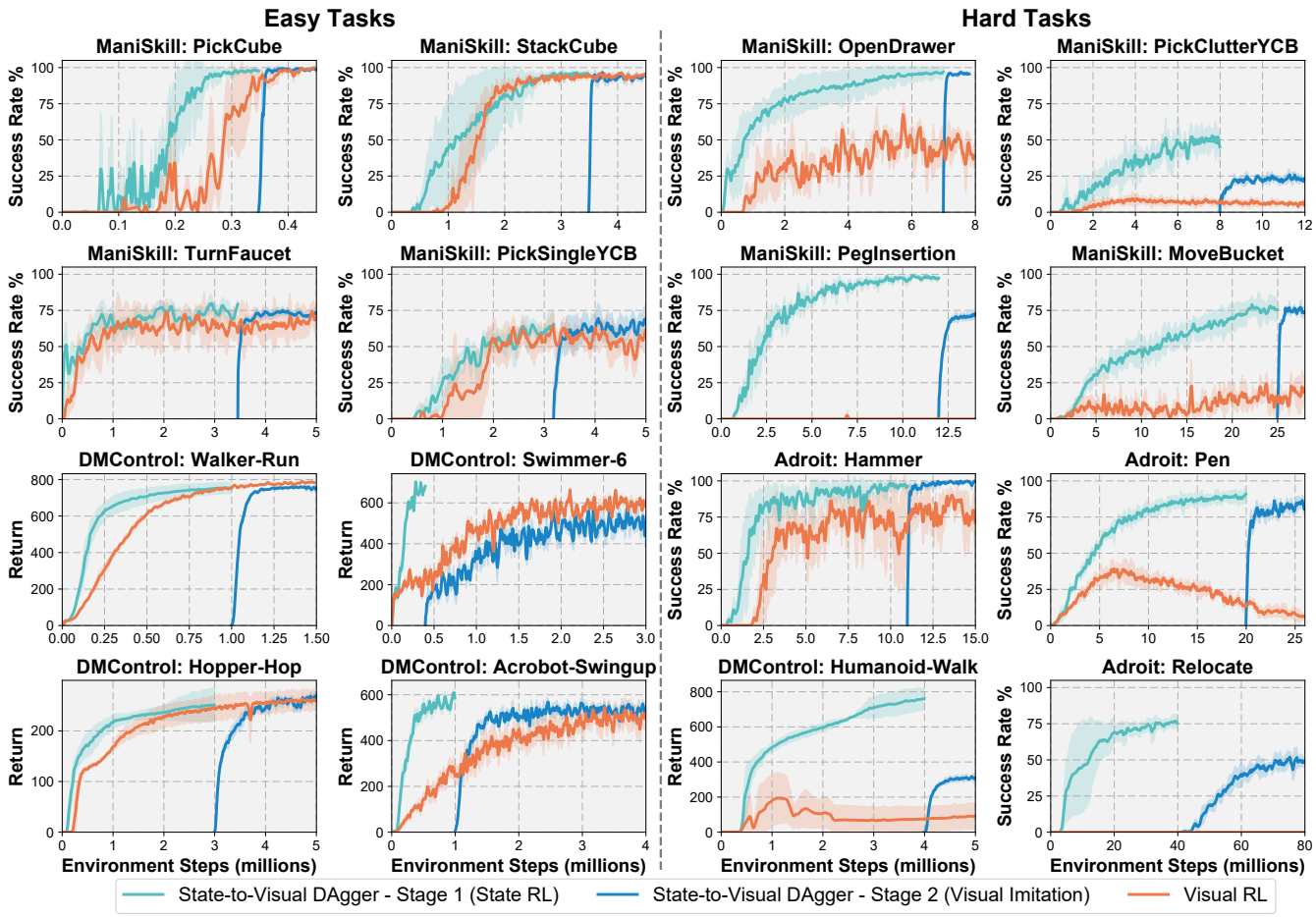

Figure 4: **Learning curves against environment steps.** Success rate (ManiSkill and Adroit) and return (DMControl) in each task. Tasks are categorized as easy if state-based RL converges within 4M steps, while the others are considered hard. State-to-Visual DAgger (Stage 2) comparisons with visual RL should account for the cost of Stage 1. The curve for stage 1 serves as a reference but is not directly comparable to others due to its state-based policy nature. The shaded region represents the 95% CI across three seeds.

State-to-Visual DAgger demonstrates notable time efficiency compared to visual RL across most tasks, including easier ones where it lacks superior sample efficiency. This advantage arises because visual RL requires training a visual encoder and rendering visual observations during training, both of which are time-intensive. In contrast, State-to-Visual DAgger confines these processes to its second stage, while the first stage leverages state-based RL, which is significantly faster in wall-clock time. Although factors like rendering speeds, network sizes, and hyperparameters influence time comparisons, State-to-Visual DAgger's time-saving benefits in the first stage are expected to generalize across settings.

6 Discussions and Recommendations

Our analysis reveals that no single method uniformly surpasses the other in every task, highlighting the distinct strengths and limitations of each approach. Below, we provide guidance for practitioners in visual policy learning, derived from our empirical findings. It is important to note, however, that these recommendations are based on observations from our experiments and should be considered as informed suggestions rather than definitive rules.

Recommend to Use State-to-Visual DAgger

Visual RL Struggles to Solve the Task: For challenging tasks where visual RL falls short, State-to-Visual DAgger is preferred, leveraging low-dimensional state information for effective policy learning before transitioning to high-dimensional visual inputs.

You Have Already Tried State RL: If you have state RL implemented and can extract or simulate low-dimensional state observations, transitioning to State-to-Visual DAgger is a natural next step, building on existing work without retraining a visual RL agent.

Focus on Wall-Clock Time Efficiency: For projects prioritizing computational cost and execution time, State-to-

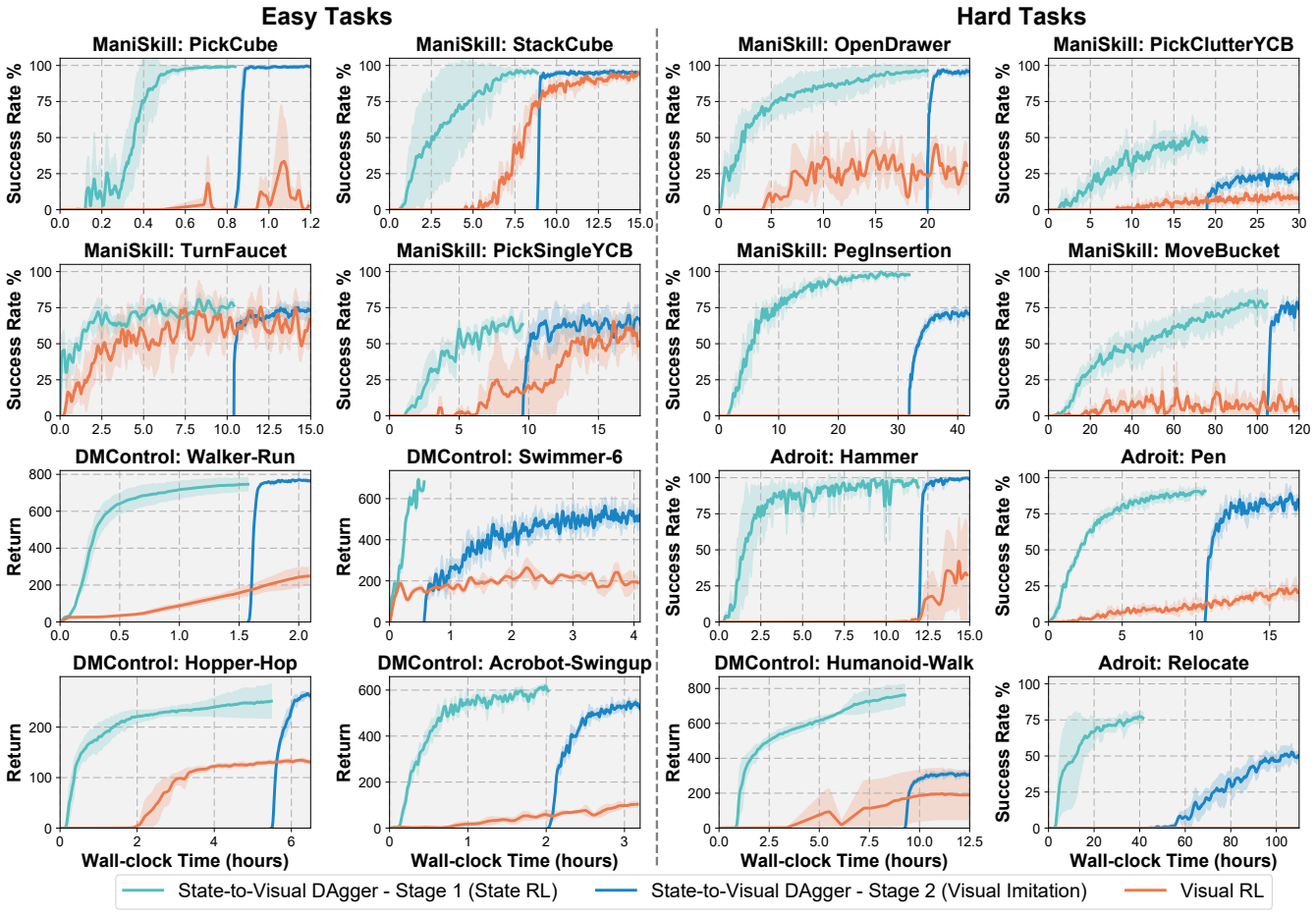

Figure 5: **Wall-clock Time.** Similar to Fig. 4, however, we use the wall-clock time as the x-axis instead of the environment steps. We find that State-to-Visual DAgger has better wall-clock time efficiency than visual RL on most tasks.

Visual DAgger is the optimal choice. Our experiments show that State-to-Visual DAgger significantly reduces wall-clock time compared to traditional visual RL methods, without compromising outcome quality.

Recommend to Use Visual RL

Low-Dimensional State Observations Are Not Available: If the environment does not provide, or it is not feasible to simulate, low-dimensional state observations necessary for the state-based teacher policy, visual RL becomes the more viable option. In such cases, direct learning from high-dimensional visual observations is the only path forward.

Preference for Minimal Intervention During Training: Visual RL provides a straightforward, hands-off approach to policy training, avoiding intermediate steps like interrupting state RL training to select checkpoints and switching to visual imitation. For a process requiring less intervention and manual oversight, visual RL may better suit your workflow.

Tasks Evidently Solvable by Visual RL: For simpler tasks where visual RL has been shown to be effective, starting with visual RL might be the most practical choice. It simplifies the setup process by removing the need for a two-stage training protocol and can achieve performance on par with State-to-Visual DAgger in these scenarios, making it an efficient and straightforward solution.

7 Conclusions

Our research compares State-to-Visual DAgger and visual RL on asymptotic performance, sample efficiency, and computational costs across tasks, highlighting their unique strengths and limitations to guide strategic application choices. We provide practical guidelines for selecting between State-to-Visual DAgger and visual RL, considering task complexity and context to determine the preferred method. However, our study has several limitations. Firstly, the categorization of tasks as difficult based on a threshold number of environmental steps is not rigorous. Additionally, we did not investigate the impact of different checkpoint selections on State-to-Visual DAgger's efficiency and performance, which could provide further insights into its adaptability. Future research should analyze checkpoint selection effects to ensure a fair and thorough comparison between State-to-Visual DAgger and visual RL.

References

Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; Ribas, R.; et al. 2019. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*.

Chen, D.; Zhou, B.; Koltun, V.; and Krähenbühl, P. 2020. Learning by cheating. In *Conference on Robot Learning*, 66–75. PMLR.

Chen, T.; Tippur, M.; Wu, S.; Kumar, V.; Adelson, E.; and Agrawal, P. 2023. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84): eadc9244.

Chen, T.; Xu, J.; and Agrawal, P. 2022. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, 297–307. PMLR.

Driess, D.; Schubert, I.; Florence, P.; Li, Y.; and Toussaint, M. 2022. Reinforcement learning with neural radiance fields. *Advances in Neural Information Processing Systems*, 35: 16931–16945.

Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; et al. 2018. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, 1407–1416. PMLR.

Gu, J.; Chaplot, D. S.; Su, H.; and Malik, J. 2022. Multi-skill mobile manipulation for object rearrangement. *arXiv preprint arXiv:2209.02778*.

Gu, J.; Xiang, F.; Li, X.; Ling, Z.; Liu, X.; Mu, T.; Tang, Y.; Tao, S.; Wei, X.; Yao, Y.; Yuan, X.; Xie, P.; Huang, Z.; Chen, R.; and Su, H. 2023. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In *International Conference on Learning Representations*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2019a. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019b. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2555–2565. PMLR.

Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019c. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, 2555–2565. PMLR.

Hafner, D.; Lillicrap, T.; Norouzi, M.; and Ba, J. 2020. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*.

Hansen, N.; Wang, X.; and Su, H. 2022. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*.

Hossain, J. 2023. Autonomous Driving with Deep Reinforcement Learning in CARLA Simulation. *arXiv preprint arXiv:2306.11217*.

Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. 2018. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, 651–673. PMLR.

Kaufmann, E.; Bauersfeld, L.; Loquercio, A.; Müller, M.; Koltun, V.; and Scaramuzza, D. 2023. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976): 982–987.

Klimov, P. D. A. R. O. 2017. John Schulman, Filip Wolski. *Proximal policy optimization algorithms. arXiv, abs/1707.06347*.

Kostrikov, I.; Yarats, D.; and Fergus, R. 2020. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*.

Kulkarni, T. D.; Gupta, A.; Ionescu, C.; Borgeaud, S.; Reynolds, M.; Zisserman, A.; and Mnih, V. 2019. Unsupervised learning of object keypoints for perception and control. *Advances in neural information processing systems*, 32.

Kumar, A.; Fu, Z.; Pathak, D.; and Malik, J. 2021. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*.

Laskin, M.; Lee, K.; Stooke, A.; Pinto, L.; Abbeel, P.; and Srinivas, A. 2020. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33: 19884–19895.

Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, 5639–5650. PMLR.

Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; and Hutter, M. 2020. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47): eabc5986.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Loquercio, A.; Kaufmann, E.; Ranftl, R.; Müller, M.; Koltun, V.; and Scaramuzza, D. 2021. Learning high-speed flight in the wild. *Science Robotics*, 6(59): eabg5810.

Margolis, G. B.; Chen, T.; Paigwar, K.; Fu, X.; Kim, D.; Kim, S.; and Agrawal, P. 2021. Learning to jump from pixels. *arXiv preprint arXiv:2110.15344*.

Miki, T.; Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; and Hutter, M. 2022. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62): eabk2822.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Nair, A. V.; Pong, V.; Dalal, M.; Bahl, S.; Lin, S.; and Levine, S. 2018. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31.

Parisi, S.; Rajeswaran, A.; Purushwalkam, S.; and Gupta, A. 2022. The unsurprising effectiveness of pre-trained vision models for control. In *international conference on machine learning*, 17359–17371. PMLR.

Pinto, L.; Andrychowicz, M.; Welinder, P.; Zaremba, W.; and Abbeel, P. 2017. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*.

Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; and Levine, S. 2017. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 627–635. JMLR Workshop and Conference Proceedings.

Shah, R.; and Kumar, V. 2021. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*.

Shang, W.; Wang, X.; Srinivas, A.; Rajeswaran, A.; Gao, Y.; Abbeel, P.; and Laskin, M. 2021. Reinforcement learning with latent flow. *Advances in Neural Information Processing Systems*, 34: 22171–22183.

Silver, D.; Wierstra, A. G. I. A. D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *DeepMind Lab. arXiv*, 1312.

Tassa, Y.; Doron, Y.; Muldal, A.; Erez, T.; and Li, Y. 2018. Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 1.

Xu, Y.; Wan, W.; Zhang, J.; Liu, H.; Shan, Z.; Shen, H.; Wang, R.; Geng, H.; Weng, Y.; Chen, J.; et al. 2023. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4737–4746.

Yarats, D.; Kostrikov, I.; and Fergus, R. 2020. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*.

Ze, Y.; Liu, Y.; Shi, R.; Qin, J.; Yuan, Z.; Wang, J.; and Xu, H. 2024. H-InDex: Visual Reinforcement Learning with Hand-Informed Representations for Dexterous Manipulation. *Advances in Neural Information Processing Systems*, 36.

Zhuang, Z.; Fu, Z.; Wang, J.; Atkeson, C.; Schwertfeger, S.; Finn, C.; and Zhao, H. 2023. Robot parkour learning. *arXiv preprint arXiv:2309.05665*.