RFTF: REINFORCEMENT FINE-TUNING FOR VISION-LANGUAGE-ACTION MODELS WITH TEMPORAL FEEDBACK

Junyang Shu* **Zhiwei Lin*** **Yongtao Wang†**
Wangxuan Institute of Computer Technology, Peking University, China
jyshu25@stu.pku.edu.cn {zwlin, wyt}@pku.edu.cn

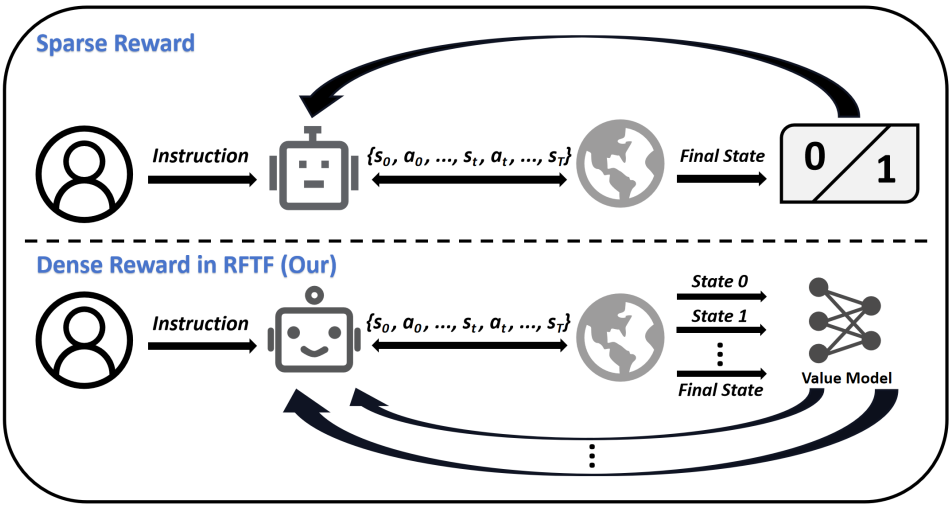


Figure 1: **Comparison between sparse reward and dense reward.** In typical reinforcement fine-tuning methods for VLAs, only sparse, outcome-based rewards are provided, which can confuse VLAs when encountering partially correct or incorrect episodes. In contrast, RFTF leverages a value model trained with temporal information to predict the value of each state within an episode, providing VLAs with higher-granularity dense rewards.

ABSTRACT

Vision-Language-Action (VLA) models have demonstrated significant potential in the field of embodied intelligence, enabling models to follow human instructions to complete complex tasks in physical environments. Existing VLAs are often trained through behavior cloning, which requires expensive data and computational resources and is constrained by human demonstrations. To address this issue, many researchers explore the application of reinforcement fine-tuning to VLAs. However, typical reinforcement fine-tuning methods for VLAs usually rely on sparse, outcome-based rewards, which struggle to provide fine-grained feedback for specific actions within an episode, thus limiting the model's manipulation capabilities and generalization performance. In this paper, we propose RFTF, a novel reinforcement fine-tuning method that leverages a value model to generate dense rewards in embodied scenarios. Specifically, our value model is trained using temporal information, eliminating the need for costly robot action labels. In addition, RFTF incorporates a range of techniques, such as GAE and sample balance to enhance the effectiveness of the fine-tuning process. By addressing the sparse reward problem in reinforcement fine-tuning, our method significantly improves the performance of VLAs, delivering superior generalization

*Equal contribution.
†Corresponding author.

and adaptation capabilities across diverse embodied tasks. Experimental results show that VLAs fine-tuned with RFTF achieve new state-of-the-art performance on the challenging CALVIN ABC-D with an average success length of 4.296. Moreover, RFTF enables rapid adaptation to new environments. After fine-tuning in the D environment of CALVIN for a few episodes, RFTF achieved an average success length of 4.301 in this new environment.

1 INTRODUCTION

The field of embodied intelligence has made remarkable progress in recent years [23]. Vision-Language-Action models, by integrating visual perception, language understanding, and action execution, empower models to perform a variety of tasks in the physical world following human instructions [25]. To enhance the generalization and reasoning abilities of VLAs while reducing reliance on extensive data and computational resources, increasing research efforts have focused on reinforcement fine-tuning for VLAs [8; 32; 13; 11]. Reinforcement learning (RL) trains models through trial-and-error, allowing them to learn from past experiences without depending on human-provided data. In addition, RL allows models to adapt to new environments by trying out and updating the parameters without human intervention. However, a significant challenge in reinforcement fine-tuning for VLAs is the reliance on sparse, outcome-based rewards. Such reward mechanisms fail to capture the nuanced correctness of individual actions within an episode. For instance, when there is partial correctness or incorrectness, it can lead to erroneous encouragement or suppression of certain actions, ultimately compromising the efficiency and stability of reinforcement fine-tuning [24].

In this paper, we propose RFTF, a reinforcement fine-tuning method with dense-reward from temporal feedback for VLAs. RFTF contains two stages. Specifically, in the first stage, we present a value model tailored for embodied scenarios. This value model is trained using temporal information to predict the value of the current state based on the input state and human instruction. In the second stage, we integrate this value model into an RL fine-tuning framework for VLAs based on Proximal Policy Optimization (PPO) [34]. By combining reward shaping and generalized advantage estimation (GAE) [33], we ensure an efficient reinforcement fine-tuning process. Notably, the entire training process for the value model and VLAs does not require robot action labels, showing the proposed method's potential for efficient data utilization.

The contributions of this paper are as follows:

- We introduce a dense-reward reinforcement fine-tuning method for VLAs without any robot action labels from humans.
- We present a value model trained with temporal information to generate dense rewards and combine reward shaping and GAE strategy to facilitate the RL fine-tuning process.
- Experimental results on the CALVIN benchmark [28] show that the proposed method achieves new state-of-the-art performance under the ABC-D setting. Moreover, RFTF exhibits superior adaptation capabilities in a new environment.

2 RELATED WORK

2.1 FOUNDATION MODELS FOR EMBODIED INTELLIGENCE

In recent years, large language models (LLMs) [1; 43; 39; 10] and vision-language models (VLMs) [17; 22; 9; 37; 2] trained on web-scale data have not only demonstrated the ability to engage in human-like dialogue but also exhibited remarkable reasoning capabilities and understanding of the physical world [15; 36; 41; 21; 27]. Leveraging these strengths, many researchers explore how to apply foundation models to interact with the physical world, *i.e.*, for embodied intelligence.

Broadly, physical environment interaction with foundation models is achieved via high-level planning or low-level manipulation. For high-level planning, LLMs or VLMs translate complex human instructions into simple robot skills (e.g., move to, grasp) by leveraging their robust comprehension and reasoning abilities [5; 14; 16; 26; 42], but cannot directly produce control signals and require additional models to execute these skills.In contrast, low-level manipulation modifies

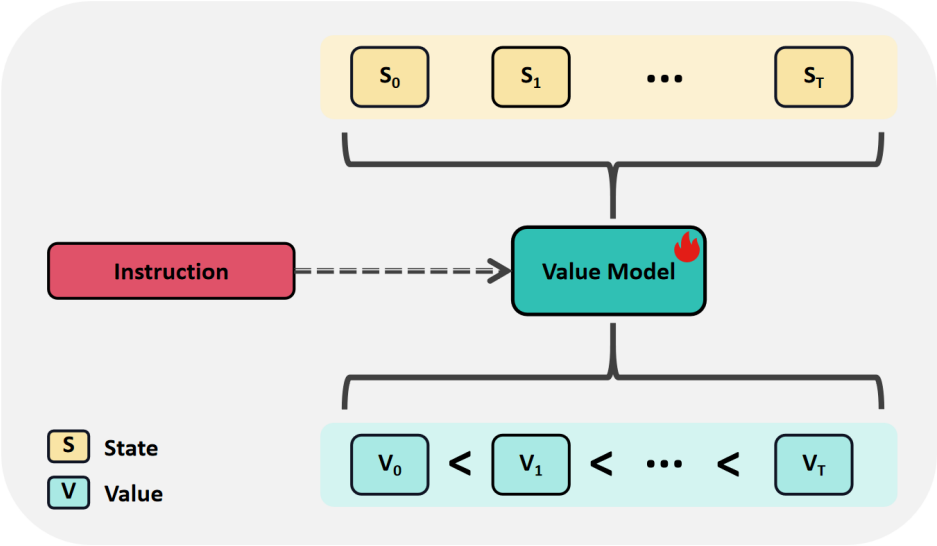


Figure 2: **Training procedure of the value model**. We assume that during an episode of a human-demonstrated successful embodied task, the state value increases monotonically over time.

pretrained VLMs to output actions directly, constructing vision-language-action models (VLAs). Specifically, RT-2 [45] fine-tunes PaLI-X [7] on vision-language data and robot demonstrations; OpenVLA [19] fine-tunes Prismatic on the Open X-Embodiment dataset [31], which includes 22 robotic embodiments from 21 institutions; and π_0 [4] introduces a flow matching architecture built on PaliGemma [3] to inherit Internet-scale semantic knowledge.

However, these models are trained with human action labels in a behavior cloning way. In this paper, we train VLAs through reinforcement fine-tuning to improve their generalization and help them adapt to novel environments with any action labels.

2.2 Reinforcement fine-tuning for large models

Reinforcement learning is a technique that enables learning from a model's past experiences. Unlike supervised learning, RL does not require large amounts of manually annotated data, nor is it constrained by expert demonstrations. Recently, using RL to fine-tune pretrained large language models has become a trend [30; 10; 35; 40; 44]. However, reinforcement fine-tuning in the field of embodied intelligence differs from that in LLMs, as it necessitates extensive interaction with the environment to collect data. FLaRe [13] is a large scale reinforcement fine-tuning framework that introduces a series of design choices that help stabilize the RL training process. iRe-VLA [11] iterates between reinforcement learning and supervised learning to address the instability issues of reinforcement learning in large-scale VLAs. DPPO [32] improves diffusion-based policies by leveraging the sequential nature of the diffusion denoising process and fine-tuning the entire chain of diffusion MDPs.

Nonetheless, due to the high precision requirements for output actions, directly applying sparse rewards in the reinforcement fine-tuning of VLA models often yields suboptimal results and may even lead to performance degradation. In contrast, we incorporate dense rewards into the reinforcement fine-tuning of VLAs through a value model trained with temporal information.

3 Method

3.1 Notation and preliminary

We model each robot task as a Partially Observable Markov Decision Process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, \mathcal{L}, \gamma)$. Here, $\mathcal{S}$ and $\mathcal{A}$ represent the state space and action space, respectively. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denotes the state transition probability function.

$\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{L} \times \mathcal{S} \to \mathbb{R}$ is the reward function. $\mathcal{O}$ is the observation space and $\mathcal{L}$ is the set of human instructions guiding the robot to complete tasks. $\gamma \in [0, 1]$ is the discount factor.

We denote the current robot policy as π_θ, *i.e.*. The policy π_θ takes the state as input and outputs the corresponding action. The overall optimization objective is to maximize the expected return of π_θ with the discount factor γ, *i.e.*, $J(\theta) = \mathbb{E}_{(s_t,a_t)\sim P} \sum_t \gamma^t R(s_t, a_t)$.

3.2 VALUE MODEL

To automatically label the correctness of each action in an episode and provide dense reward for training, we employ a value model parameterized by ϕ to predict the value of the state at each time step. Specifically, the value model takes the state and human instruction as input and outputs the value of the current state, *i.e.*, $v_t = V_\phi(s_t, l)$. Notably, both the inference and training of the value model do not require expensive robot action labels, resulting in low data dependency.

However, there are no explicit value labels available for training the value model. Inspired by RLHF [30], we collect data pairs and train the value model with contrastive learning. Specifically, as shown in Fig. 2, $(s_t, s_{t+1}, ..., s_{t+n-1} \mid l)$ is an expert-demonstrated trajectory without action labels. As the state progresses toward task completion, we assume that the state value increases with each time step, *i.e.*, $v_t < v_{t+1} < ... < v_{t+n-1}$. Following RLHF, we adopt the contrastive loss function as the optimization objective:

$$loss(\phi) = -\frac{1}{C_n^2}\mathbb{E}_{(s_t,a_t)\sim P}[log(\sigma(V_\phi(s_{t+\Delta t}, l) - V_\phi(s_t, l))], \tag{1}$$

where C_n^2 is the number of combinations, σ is the sigmoid function and Δt is a positive integer belonging to $[1, n - t)$. This implies that we aim for the value at later time steps to be as large as possible compared to earlier time steps.

The architecture of our value model is based on the VLA model, with only the action tokens of the VLA model replaced by value tokens. Given that the VLA model is already capable of processing mixed inputs of states and human instructions, we initialize the training of the value model using the weights of the VLA model.

3.3 RL FINE-TUNING PIPELINE

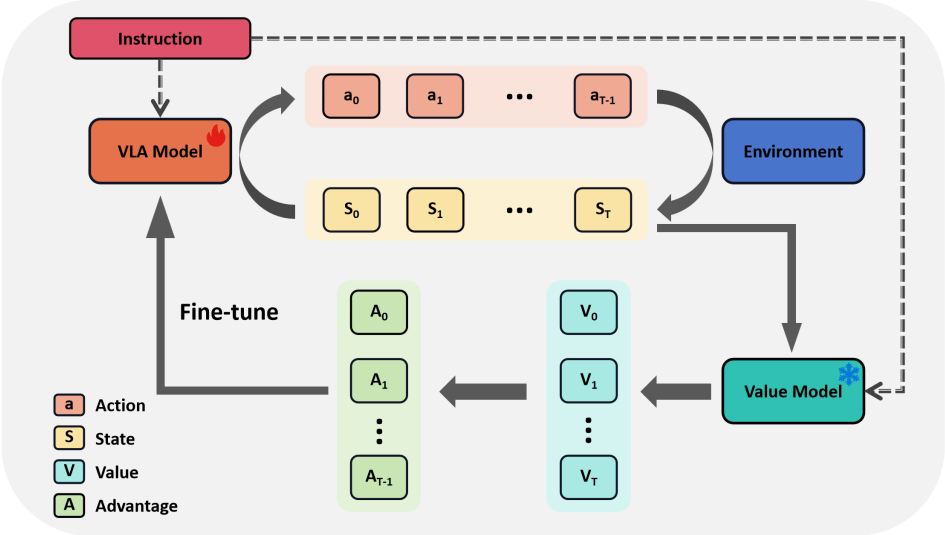


Figure 3: **Illustration of RL fine-tuning pipeline.** RFTF utilizes a value model trained with temporal information to predict the value of each state in episodes of interaction between the VLA model and the environment, thereby providing guidance for each action in episodes to fine-tune the VLA model.

In this stage, as shown in Fig. 3, we utilize the trained value model to guide the RL fine-tuning process, aiming to enhance the performance of VLAs. To achieve effective fine-tuning, we adopt Proximal Policy Optimization (PPO) [34] as our reinforcement learning algorithm framework.

Specifically, after obtaining the raw value of each state in an episode from the trained value model, we normalize all state values within the episode, as the output range of the value model varies significantly across different tasks. To make progress toward the final goal, we employ a reward shaping term [29] as the reward function:

$$R_t = \begin{cases} \gamma V(s_{t+1}, l) - V(s_t, l) & \text{if } t \text{ is not the end step} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Notably, if we directly use the discounted sum of rewards with the discount factor γ as the advantage function in PPO, the simplified form of the advantage function will contain only the current and last states, thereby neglecting the influence of intermediate states. To address this, we adopt Generalized Advantage Estimation (GAE) [33], introducing an additional hyperparameter λ to incorporate the state values at all intermediate time steps into the advantage function. Additionally, to leverage information about task completion, instead of solely adding +1 or -1 to the reward at the final time step, we incorporate feedback on task success or failure into the advantage function at each time step. This ensures that early decisions in long episodes receive relevant feedback. Furthermore, we introduce a balancing coefficient to address the imbalance between successful and failed samples. The final form of our advantage function is as follows:

$$A_t = \eta \left[I(success) + \sum_{n=t}^{T} (\gamma\lambda)^{n-t} R_t \right], \tag{3}$$

where η is the coefficient for balancing positive and negative samples, set to 0.25 when the task succeeds and 1 when the task fails. λ is the hyperparameter in GAE. I is an indicator function that returns +1 for task success and -1 for task failure.

Previous works [13; 11; 8] have highlighted that, due to the high precision requirements for VLA model outputs, directly applying reinforcement fine-tuning to VLAs often results in performance degradation. To mitigate this, inspired by [35], we incorporate both a surrogate objective clipping term and adaptive KL divergence into the optimization objective of reinforcement fine-tuning. Specifically, for an VLA model π parameterized by θ, the loss function for reinforcement fine-tuning is as follows:

$$\begin{aligned} loss(\theta) = -\mathbb{E}_{(s_t, a_t)\sim P}\Big\{ min\Big[& \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t, \\ & clip\Big(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \Big) A_t \Big] \\ & -\beta D_{KL}\left[\pi_\theta(a_t|s_t) || \pi_{\theta_{ref}}(a_t|s_t) \right] \Big\}, \end{aligned} \tag{4}$$

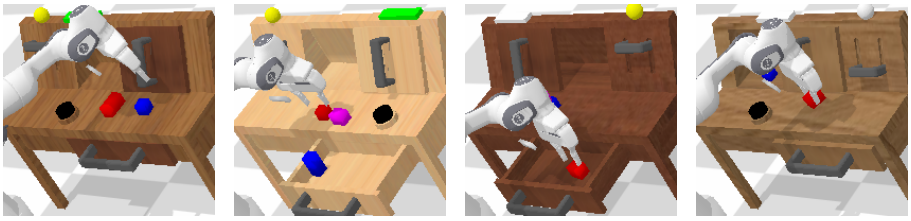
where ϵ and β are hyper-parameters that prevent excessive divergence between the new and old policies.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

4.1.1 BENCHMARK

We evaluated our approach on the challenging CALVIN benchmark, which focuses on long-horizon language-conditioned robotic tasks across 34 tasks in four distinct simulated environments (Env A–D). As shown in Fig. 4, each environment features a Franka Emika Panda robot with a parallel-jaw gripper and a table for manipulation tasks. The diversity of these environments enables both generalization and adaptation experiments to validate our method.



(a) Env A (b) Env B (c) Env C (d) Env D

Figure 4: **Visualization of the CALVIN benchmark**. The CALVIN benchmark includes four distinct environments, differing in the positions of the LED, light bulb, slider, drawer, switch, and button, as well as the material of the table.

Table 1: **CALVIN ABC-D results.** We present the average success rates of top-3 checkpoints computed over 1000 rollouts for each task and the average number of completed tasks to solve 5 instructions consecutively (Avg. Len.). The model being fine-tuned is specified in parentheses after RFTF.

Method	Task completed in a row					
	L1	L2	L3	L4	L5	Avg. Len. ↑
3D Diffuser Actor [18]	92.2	78.7	63.9	51.2	41.2	3.272
CLOVER [6]	96.0	83.5	70.8	57.5	45.4	3.532
Diffusion Transformer Policy [12]	94.5	82.5	72.8	61.3	50.0	3.611
Seer [38]	94.4	87.2	79.9	72.2	64.3	3.980
GR-MG [20]	96.8	89.3	81.5	72.7	64.4	4.047
RFTF(GR-MG)	**96.9**	**88.8**	**82.1**	**74.9**	**65.4**	**4.081**
Seer-Large [38]	96.3	91.6	86.1	80.3	74.0	4.283
RFTF(Seer-Large)	**96.4**	**91.7**	**86.7**	**80.7**	**74.1**	**4.296**

4.1.2 BASELINES

In RFTF, we fine-tune the top two models on the CALVIN ABC-D benchmark under two distinct settings. In the generalization setting, we fine-tune the models on the CALVIN ABC environments, which the models have already encountered, to evaluate whether RFTF enhances the models' generalization performance. In the adaptation setting, we fine-tune the models on the CALVIN D environment, which the models have not previously seen, to assess whether RFTF helps the models adapt to new environments.

4.1.3 METRICS

The CALVIN benchmark evaluation requires VLAs to execute 1000 sequences in the D simulated environment, with each sequence comprising 5 tasks. The VLA model performs these 5 tasks sequentially, exiting the sequence upon failing any task. Ln denotes the proportion of n tasks completed out of 5. We use the average number of tasks completed per sequence as the evaluation metric.

4.1.4 IMPLEMENTATION DETAILS

For the value model, we train it with a batch size of 4×8 and a learning rate of 1e-5. As described in Value model Section3.2, the structure of the value model involves replacing the original VLA's action tokens and action decoder with value tokens and a value decoder.

For the RL fine-tuning, we apply the same implementation details across both the generalization and adaptation settings to ensure the method's universality. For the VLA model, we first discretize the model's output with 1000 bins to obtain the probability term in the PPO optimization objective. To enhance training stability, we freeze the model's encoders and transformer backbone, and only update the action head. During RL fine-tuning, we train the model with a learning rate of 1e-7,

covering roughly 1000 episodes. The RL fine-tuning process is done with four NVIDIA A40 GPUs within 10 hours for Seer-Large and 14 hours for GR-MG. To prevent overfitting to task instructions, we deliberately used different instructions during the fine-tuning phase than those used during the testing phase.

4.2 Main results

4.2.1 Generalization

In this experiment, we show the better generalization ability of VLAs fine-tuned with RFTF. Specifically, RFTF fine-tunes models on CALVIN's A, B, and C environments and tests them in the D environment. As shown in Tab. 1, GR-MG fine-tuned by RFTF achieved a score of 4.081, surpassing the baseline of 4.043; Seer-Large fine-tuned by RFTF achieved a score of 4.296, surpassing the baseline of 4.283, which also achieves new state-of-the-art performance.

4.2.2 Adaptation

Table 2: **Adaptation Experiments.** VLA refers to models to be fine-tuned. Env denotes the environments we use to fine-tune the model, where "–" indicates no fine-tuning.

VLA	Env	Task completed in a row					
		L1	L2	L3	L4	L5	Avg. Len. ↑
GR-MG	-	96.8	89.3	81.5	72.7	64.4	4.047
GR-MG	ABC	96.9	88.8	82.1	74.9	65.4	4.081
GR-MG	**D**	**96.1**	**90.5**	**83.9**	**75.0**	**65.8**	**4.113**
Seer-Large	-	96.3	91.6	86.1	80.3	74.0	4.283
Seer-Large	ABC	96.4	91.7	86.7	80.7	74.1	4.296
Seer-Large	**D**	**97.0**	**92.0**	**86.0**	**80.6**	**74.5**	**4.301**

In this experiment, we demonstrate that VLAs can be adapted to new environments that they have never seen before with the proposed RFTF. Specifically, we fine-tune models with RFTF in the unseen D environment of CALVIN. As indicated in Tab. 2, GR-MG fine-tuned by RFTF achieved a score of 4.113, and Seer-Large fine-tuned by RFTF achieved a score of 4.301, significantly outperforming the model's original performance in CALVIN's D environment. Since other methods are exclusively trained on the A, B, and C environments in CALVIN and have no exposure to the D environment, we refrain from comparing them with the results in our adaptation setting.

4.3 Analysis of value model

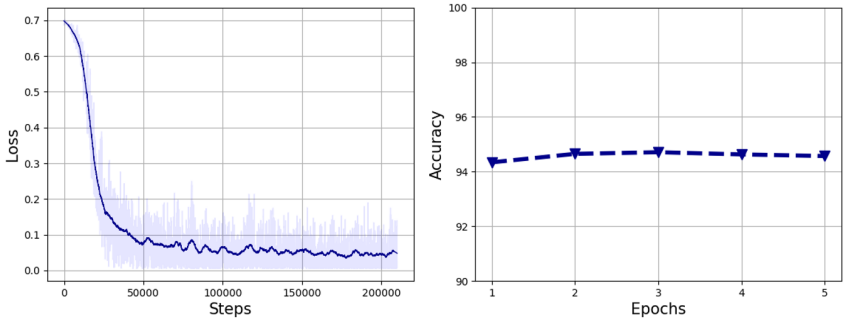


Figure 5: **Train curves of the value model.** We show the loss curve of the value model during training and evaluation results of the value model across different epochs.

To verify whether the value model can accurately predict state values, we tested its accuracy on the CALVIN ABC validation set. Specifically, we selected two frames from expert-demonstrated

Table 3: **Ablation studies.** VLA refers to models to be fine-tuned. SR refers to sparse reward. Replacing the dense rewards provided by RFTF with sparse rewards leads to performance drops.

VLA	Type	Reward	Task completed in a row					
			1	2	3	4	5	Avg. Len. ↑
GR-MG	Generalization	**RFTF**	**96.9**	**88.8**	**82.1**	**74.9**	**65.4**	**4.081**
GR-MG	Generalization	SR	95.3	88.3	80.8	72.1	62.8	3.993
GR-MG	Adaptation	**RFTF**	**96.1**	**90.5**	**83.9**	**75.0**	**65.8**	**4.113**
GR-MG	Adaptation	SR	95.9	88.3	79.8	72.4	64.5	4.009
Seer-Large	Generalization	**RFTF**	**96.4**	**91.7**	**86.7**	**80.7**	**74.1**	**4.296**
Seer-Large	Generalization	SR	95.2	89.9	85.1	79.4	72.9	4.225
Seer-Large	Adaptation	**RFTF**	**97.0**	**92.0**	**86.0**	**80.6**	**74.5**	**4.301**
Seer-Large	Adaptation	SR	95.3	90.7	85.8	79.8	73.3	4.249

trajectories, and if the value model assigned a higher value to the later frame compared to the earlier one, we considered the value prediction correct.

As shown in Fig. 5, the value model achieved an accuracy of more than 94% after the first epoch, with subsequent training nearly yielding no further improvements in accuracy. This observation aligns with findings in RLHF. To mitigate potential overfitting from prolonged training, we selected the value model from the first epoch for use in the reinforcement fine-tuning process. Notably, as shown in Fig. 6, we found that in episodes sampled by the VLA model itself, the state values produced by the value model did not exhibit a monotonic increase over time. This ensures that the optimization objective of our reinforcement fine-tuning does not merely encourage the VLA model to reinforce its original actions.

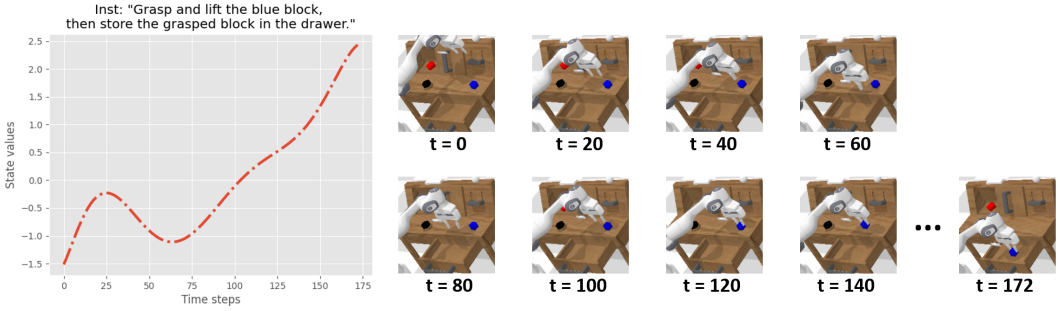


Figure 6: **An example of a state value curve.** As depicted, the curve exhibits a decline midway due to an incorrect grasping action by the VLA model.

4.4 Ablation Study

We conducted ablation experiments to evaluate the effectiveness of the dense rewards in RFTF. To ensure a fair and controlled comparison, all experimental conditions were kept identical except for the rewards used for fine-tuning. The sparse rewards for fine-tuning are derived solely from whether the model successfully completed the given task or not, which is a common method used in standard PPO algorithms.

As shown in Tab. 3, unlike models fine-tuned with RFTF, the models using only sparse rewards exhibited varying degrees of performance drop. This finding is consistent with the observations in [13] and [11].

5 Conclusion and Limitation

In this paper, we propose RFTF, an online reinforcement fine-tuning method for vision-language-action models. To obtain dense rewards, we first train a value model using temporal information while maintaining low data dependency. Then, we integrate the value model into the reinforcement

fine-tuning process for VLAs, providing reward signals for intermediate decision steps, addressing the prevalent issue of sparse rewards, and enhancing the effectiveness of fine-tuning. Experimental results demonstrate that VLAs fine-tuned with RFTF exhibit superior generalization and overall performance. Additionally, RFTF enables rapid adaptation to new environments. The primary limitation of RFTF is that it has only been verified on the simulated benchmark. In the future, we will apply RFTF to real-world robots.

Acknowledgments. This work was supported by National Natural Science Foundation of China under Grant 62176007.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2023.

[6] Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. *arXiv preprint arXiv:2409.09016*, 2024.

[7] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[8] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.

[9] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[10] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[11] Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. *arXiv preprint arXiv:2501.16664*, 2025.

[12] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Hengjun Pu, Chengyang Zhao, Ronglei Tong, Yu Qiao, Jifeng Dai, and Yuntao Chen. Diffusion transformer policy. *arXiv preprint arXiv:2410.15959*, 2024.

[13] Jiaheng Hu, Rose Hendrix, Ali Farhadi, Aniruddha Kembhavi, Roberto Martín-Martín, Peter Stone, Kuo-Hao Zeng, and Kiana Ehsani. Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning. *arXiv preprint arXiv:2409.16578*, 2024.

[14] Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*, 2023.

[15] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[16] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025.

[17] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *International Conference on Machine Learning (ICML)*, 2024.

[18] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[19] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[20] Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. *IEEE Robotics and Automation Letters*, 2025.

[21] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Neural Information Processing Systems (NeurIPS)*, 2023.

[23] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *arXiv preprint arXiv:2407.06886*, 2024.

[24] Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, et al. Exploring the limit of outcome reward for learning mathematical reasoning. *arXiv preprint arXiv:2502.06781*, 2025.

[25] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

[26] Weixin Mao, Weiheng Zhong, Zhou Jiang, Dong Fang, Zhongyue Zhang, Zihan Lan, Haosheng Li, Fan Jia, Tiancai Wang, Haoqiang Fan, et al. Robomatrix: A skill-centric hierarchical framework for scalable robot task planning and execution in open-world. *arXiv preprint arXiv:2412.00171*, 2024.

[27] Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. Genrl: Multimodal-foundation world models for generalization in embodied agents. *Neural Information Processing Systems (NeurIPS)*, 2024.

[28] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022.

[29] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, 1999.

[30] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Neural Information Processing Systems (NeurIPS)*, 2022.

[31] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[32] Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

[33] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[35] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[36] Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In *International Conference on Learning Representations (ICLR)*, 2023.

[37] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[38] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024.

[39] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[40] Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[41] Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023.

[42] Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023.

[43] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[44] Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2024.

[45] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.