Balancing Signal and Variance: Adaptive Offline RL Post-Training for VLA Flow Models

Hongyin Zhang[1,2], **Shiyuan Zhang**[3], **Junxi Jin**[2], **Qixin Zeng**[2], **Yifan Qiao**[4], **Hongchao Lu**[2], **Donglin Wang**[2*]

[1] Zhejiang University, Hangzhou, China
[2] Westlake University, Hangzhou, China
[3] University of California, Los Angeles, USA
[4] National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
wangdonglin@westlake.edu.cn

Abstract

Vision-Language-Action (VLA) models based on flow matching have shown excellent performance in general-purpose robotic manipulation tasks. However, the action accuracy of these models on complex downstream tasks is unsatisfactory. One important reason is that these models rely solely on the post-training paradigm of imitation learning, which makes it difficult to have a deeper understanding of the distribution properties of data quality, which is exactly what Reinforcement Learning (RL) excels at. In this paper, we theoretically propose an offline RL post-training objective for VLA flow models and induce an efficient and feasible offline RL fine-tuning algorithm —— **A**daptive **R**einforced **F**low **M**atching (**ARFM**). By introducing an adaptively adjusted scaling factor in the VLA flow model loss, we construct a principled bias-variance trade-off objective function to optimally control the impact of RL signal on flow loss. ARFM adaptively balances RL advantage preservation and flow loss gradient variance control, resulting in a more stable and efficient fine-tuning process. Extensive simulation and real-world experimental results show that ARFM exhibits excellent generalization, robustness, few-shot learning, and continuous learning performance.

Introduction

In recent years, the rapid development of vision-language-action (VLA) models has enabled robots to perform a variety of manipulation tasks based on multi-modal perception. Large-scale pre-training systems have demonstrated the feasibility of learning general policies from internet-level multi-modal data and have performed well in real-world robotic manipulation tasks, such as RT-1 (Brohan et al. 2022), RT-2 (Brohan et al. 2023), PaLM-E (Driess et al. 2023), Octo (Team et al. 2024), OpenVLA (Kim et al. 2024), π_0 (Black et al. 2024). In particular, π_0 is a VLA flow policy model based on trajectory vector fields, which utilizes trajectory-level denoising modeling mechanism to achieve parallel and efficient policy learning and achieves state-of-the-art performance in realistic manipulation tasks.

Although pre-trained VLA models show good generalizability, fine-tuning on downstream tasks based on the imitation learning paradigm still faces great challenges in terms of action accuracy (Brohan et al. 2023; Black et al. 2023; Li et al. 2024; Zhang et al. 2025a). In this scenario, VLA models that rely solely on behavior cloning or flow matching may not be able to effectively exploit the quality distribution structure in the training data. Some works (Mark et al. 2024; Zhai et al. 2024; Zhao et al. 2025; Guo et al. 2025b; Zhang et al. 2025b) resort to offline Reinforcement Learning (RL) in the hope of mining deeper and fine-grained data quality features to achieve more efficient VLA model fine-tuning. Recently, an algorithm called ReinboT (Zhang et al. 2025b) was proposed, which attempts to introduce RL return-to-go as a fine-grained goal to guide the fine-tuning of the VLA model. However, we observed that this approach has limited performance in VLA flow models (such as π_0), see the experimental section for details. We infer that this is because VLA flow models the entire action trajectory distribution through a vector field. During the inference phase of the VLA flow model, the maximized return-to-go can only control the generation of the trajectory vector field, thereby indirectly and inefficiently guiding the final action prediction. Therefore, how to effectively perform offline RL fine-tuning of VLA flow models remains underexplored.

To this end, we propose **A**daptive **R**einforced **F**low **M**atching (**ARFM**), a novel adaptive offline RL fine-tuning method for VLA flow models. We control the strength of the RL signal introduced into the sample data by introducing an adaptively adjusted scaling factor in the VLA flow loss function. Specifically, we theoretically construct a variance-bias trade-off optimization objective with respect to the scaling factor. We aim to dynamically select the scaling factor to preserve the RL signal, i.e., samples with higher RL advantage are still amplified in the data distribution during flow model fine-tuning. Meanwhile, we hope to control the flow matching loss gradient variance, i.e., avoid some weights from exponentially exploding, resulting in excessive gradient variance and training crash.

Therefore, we focus on adaptively balancing "retaining enough RL advantage signal" and "controlling loss gradient variance" in the scaling factor as the distribution of the

[*]Corresponding author.

current batch data changes during the post-training process of the VLA flow model. We theoretically construct an optimization objective for the scaling factor and construct an approximately analyzable objective function through some reasonable assumptions. A bisection iterative algorithm for real-time updating of the scaling factor is further induced, so that the loss of the VLA flow model is adaptively adjusted and corrected according to the data distribution. Our principled contributions include three aspects:

- We propose ARFM, a novel offline RL post-training method for VLA flow models that can adaptively adjust the data quality distribution.
- We theoretically establish the optimization objective of adaptively adjusting the scaling factor and induce a bisection iterative algorithm to update the factor in real time, thereby achieving efficient VLA flow model fine-tuning.
- Extensive experiments in simulation and real-world robot manipulation tasks show that our proposed ARFM exhibits state-of-the-art generalization ability, robustness to dynamic perturbations, and excellent performance in few-shot and continuous learning scenarios.

Related Work

Flow matching models in RL. Flow matching (Lipman et al. 2023) is a more general counterpart to the diffusion model (Ho, Jain, and Abbeel 2020). Both diffusion and flow matching tend to have good results in offline RL. Some previous work explored modeling behavior policies based on diffusion models (Janner et al. 2022; Wang, Hunt, and Zhou 2022). Following these studies, some researchers modeled offline RL objectives as energy-guided diffusion processes (Chen et al. 2022; Lu et al. 2023), while others adopted the same policy optimization approach utilizing diffusion and flow matching models without classifiers (Ajay et al. 2022; Zheng et al. 2023). Recently, an energy-weighted flow matching method (Zhang, Zhang, and Gu 2025) for learning energy-guided distributions was proposed, which can directly utilize RL rewards or other metrics as energy functions without learning intermediate energy models. Compared with previous studies, our work further extends the energy-weighted flow matching method to the VLA model post-training setting and proposes a new adaptive energy-weighted algorithm.

RL for VLA models. RL has recently emerged as a pivotal technique for post-training VLA models, overcoming the limitations of vanilla imitation learning (Zhai et al. 2024; Mark et al. 2024) on infrastructure such as RT-1 (Brohan et al. 2022). These efforts are mainly divided into online and offline RL fine-tuning. Online methods exploit direct environment interaction, adopt algorithms such as PPO (Schulman et al. 2017) or develop more data-efficient interaction frameworks (Lu et al. 2025; Tan et al. 2025; Guo et al. 2025b). On the other hand, offline methods learn from static datasets by introducing various signals, such as human preferences (Chen et al. 2025; Zhang et al. 2024) and value guidance (Nakamoto et al. 2024). Recent work, ReinboT (Zhang et al. 2025b), has effectively implemented the core idea of RL in predicting the future to maximize return when examining offline RL post-training of VLA models. The idea of utilizing RL returns as fine-grained goals to guide the policy comes from the previous new paradigms of sequence modeling, Decision Transformer (Chen et al. 2021) and Reinformer (Zhuang et al. 2024). Different from these studies, our work considers the implementation of offline RL post-training in the energy-weighted VLA flow model to achieve efficient and stable policy optimization.

Preliminaries

In this section, we first list the energy-weighted flow matching and give two equivalent loss functions to train a neural network policy to approximate the energy-guided flow. We then list the RL advantage evaluation method for post-training data and the expressive VLA flow model.

Energy-weighted flow matching

Definition 1. Both flow matching (Lipman et al. 2023) and Energy-Weighted Flow Matching (EWFM) (Zhang, Zhang, and Gu 2025) learn the *vector field* (Chen et al. 2019). Consider a probability density path: $p : [0, 1] \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, $\mathbf{x} \mapsto p_t(\mathbf{x})$. Define the *vector field* $\mathbf{v} : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ which *generates* the probability density path p, satisfying:

$$\frac{d}{dt}p_t(\mathbf{x}) + \operatorname{div}(\mathbf{v}_t(\mathbf{x})p_t(\mathbf{x})) = 0. \tag{1}$$

For the conditional distribution $p_{t0}(\mathbf{x}|\mathbf{x}_0)$, the conditional vector field is defined in the same way. Consider a probability distribution $p_0(\mathbf{x}_0)$ and an energy function $\mathcal{E}(\cdot)$, then the energy-guided distribution satisfies:

$$q_0(\mathbf{x}_0) \propto p_0(\mathbf{x}_0)\exp(-\beta\mathcal{E}(\mathbf{x}_0)). \tag{2}$$

Then consider adding Gaussian noises as conditional distributions to the two distributions: $p_{t0}(\mathbf{x}|\mathbf{x}_0) = q_{t0}(\mathbf{x}|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t|\alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$. Define $p_t(\mathbf{x}) := \int_{\mathbf{x}_0} p_{t0}(\mathbf{x}|\mathbf{x}_0)p_0(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0$, $q_t(\mathbf{x}) := \int_{\mathbf{x}_0} q_{t0}(\mathbf{x}|\mathbf{x}_0)q_0(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0$ as the marginal distributions at time t, then energy-weighted flow matching aims to efficiently learn the vector field $\hat{\mathbf{u}}_t(\mathbf{x})$ which generates $q_t(\mathbf{x})$. **Theorem** 1 calculates the vector field $\hat{\mathbf{u}}_t(\mathbf{x})$ which generates q, and **Theorem** 2 finds the equivalent loss between learning the conditional vector field and the marginal vector field. The proof process is in the Appendix.

Theorem 1. *Suppose $p_0(\mathbf{x}_0)$, $q_0(\mathbf{x}_0)$ are defined in Eq. (2), and the conditional distributions $p_{t0}(\mathbf{x}|\mathbf{x}_0), q_{t0}(\mathbf{x}|\mathbf{x}_0)$, marginal distributions $p_t(\mathbf{x}), q_t(\mathbf{x})$ are also defined above. Consider the conditional vector field $\mathbf{u}_{t0}(\mathbf{x}|\mathbf{x}_0)$ which generates $p_{t0}(\mathbf{x}|\mathbf{x}_0)$, and the vector field $\hat{\mathbf{u}}_t(\mathbf{x})$ which generates $q_t(\mathbf{x})$, then we have:*

$$\hat{\mathbf{u}}_t(\mathbf{x}) = \int_{\mathbf{x}_0} p_{0t}(\mathbf{x}_0|\mathbf{x})\mathbf{u}_{t0}(\mathbf{x}|\mathbf{x}_0)\frac{\exp(-\beta\mathcal{E}(\mathbf{x}_0))}{\exp(-\mathcal{E}_t(\mathbf{x}))}\mathrm{d}\mathbf{x}_0, \tag{3}$$

where $\mathcal{E}_t(\mathbf{x})$ is an intermediate energy function which is defined as:

$$\mathcal{E}_t(\mathbf{x}) = -\log\mathbb{E}_{p_{0t}(\mathbf{x}_0|\mathbf{x})}[\exp(-\beta\mathcal{E}(\mathbf{x}_0))].$$

Theorem 1 finds out a closed form of the vector field $\mathbf{u}_t(\mathbf{x})$, but it still remains challenging to learn $\mathbf{u}_t(\mathbf{x})$, since the intermediate energy function is unknown during the process, therefore **Theorem** 2 gives the method to simplify the training process.

Theorem 2. $p_0(\mathbf{x}_0), q_0(\mathbf{x}_0), p_t(\mathbf{x}), q_t(\mathbf{x}), \mathcal{E}(\mathbf{x}_0), \mathcal{E}_t(\mathbf{x}_t)$, $\hat{\mathbf{u}}_t(\mathbf{x}), \mathbf{u}_{t0}(\mathbf{x}|\mathbf{x}_0)$ *are all defined above. Consider learning a model* $\mathbf{v}_\theta(\mathbf{x})$ *with the learnable parameter* θ*, then we define the Energy-weighted Flow Matching (EFM) loss* $\mathcal{L}_{EFM}$ *as:*

$$\mathcal{L}_{EFM}(\theta) = \mathbb{E}_{\mathbf{x},t}[\frac{\exp(-\mathcal{E}_t(\mathbf{x}))}{\mathbb{E}_{\tilde{x}\sim p_t(\tilde{\mathbf{x}})}[\exp(-\mathcal{E}_t(\tilde{\mathbf{x}}))]}||\mathbf{v}_\theta(\mathbf{x}) - \hat{\mathbf{u}}_t(\mathbf{x})||^2],$$

and the Conditional Energy-weighted Flow Matching (CEFM) loss $\mathcal{L}_{CEFM}$ *as:*

$$\mathcal{L}_{CEFM}(\theta) =$$
$$\mathbb{E}_{\mathbf{x},t}[\frac{\exp(-\beta\mathcal{E}(\mathbf{x}_0))}{\mathbb{E}_{\tilde{x}\sim p_0(\tilde{\mathbf{x}}_0)}[\exp(-\beta\mathcal{E}(\tilde{\mathbf{x}}_0))]}||\mathbf{v}_\theta(\mathbf{x}) - \mathbf{u}_{t0}(\mathbf{x}|\mathbf{x}_0)||^2],$$

where the expectation on t *is taken over some predefined distribution* $\lambda(t)$*,* $\mathbf{x}_0$ *is sampled from the data distribution* $p_0(\cdot)$ *and* $\mathbf{x}$ *at time* t *is sampled by* $p_t(\mathbf{x})$ *with conditional distribution* $p_{t0}(\mathbf{x}|\mathbf{x}_0)$ *generated by the flow* $\mathbf{u}_{t0}(\mathbf{x}|\mathbf{x}_0)$*. Then the two losses are equal up to a constant factor to* θ*. Hence,*

$$\nabla_\theta \mathcal{L}_{EFM}(\theta) = \nabla_\theta \mathcal{L}_{CEFM}(\theta).$$

Theorem 2 proves that the training process of the vector field of the marginal energy-guided distribution could be simplified to learning the conditional vector field, without the calculation of the intermediate energy function. In the VLA offline RL post-training process, given the observation and action actor $\mathbf{o}_t, \mathbf{A}_t$, and compared with the vanilla imitation learning action distribution $p(\mathbf{A}_t|\mathbf{o}_t)$, we consider designing a new energy function $\mathcal{E}(\mathbf{A}_t, \mathbf{o}_t)$ and further extend the EWFM method to fine-tune the energy-guided VLA model distribution $\pi(\mathbf{A}_t|\mathbf{o}_t) \propto p(\mathbf{A}_t|\mathbf{o}_t)\exp(\beta\mathcal{E}(\mathbf{A}_t, \mathbf{o}_t))$.

RL advantage signal estimation

Leave-one-out (Kool, van Hoof, and Welling 2019) estimates the RL advantage using sampling. The REINFORCE Leave-One-Out generates K independent samples $x_1, \dots, x_K \sim p_\theta(\cdot|c)$ and utilizes all other samples to compute a baseline for the current return: $R^*(c, x_k) = R(c, x_k) - \frac{1}{K-1}\sum_{i=1, i\neq k}^{K} R(c, x_i)$. The equivalent form of this objective is:

$$R^*(c, x_k) = \frac{K}{K-1}\left(R(c, x_k) - \frac{1}{K}\sum_{i=1}^{K} R(c, x_i)\right). \quad (4)$$

This is a simple, unbiased, and low-variance advantage estimate, so we are inspired by it to design a critic-free offline RL advantage signal for the VLA multi-task setting.

VLA flow model

The π_0 model (Black et al. 2024) is a recently proposed VLA flow model with excellent performance, which mainly consists of a language model transformer backbone. Following the standard late fusion VLM recipe, image encoders embed the robot's image observations into the same embedding

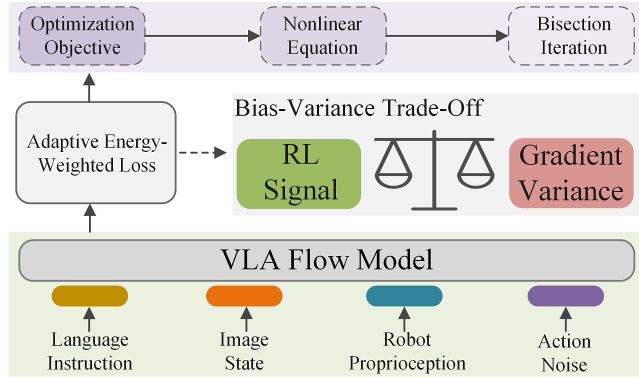

Figure 1: The proposed ARFM method. We construct an energy-weighted flow matching loss with adaptive scaling factor, aiming to balance the RL signal and gradient variance on the data samples. Then, we establish a solvable optimization objective and obtain a nonlinear equation about the scaling factor, which is solved by bisection iteration.

space as language tokens. The observation $\mathbf{o}_t$ consists of multiple RGB images, a language command, and the robot's proprioceptive state. Formally, $\mathbf{o}_t = [I_1^t, \dots, I_n^t, \ell^t, q^t]$, where I_i^t is the i-th image, ℓ^t is a sequence of language tokens, and q^t is a vector of joint angles. The action chunk $\mathbf{A}_t = [\mathbf{a}_t, \mathbf{a}_{t+1}, \dots, \mathbf{a}_{t+H-1}]$ corresponds to a sequence of future actions, where H is the action horizon. The Flow Matching (FM) loss is given by:

$$L_{FM}(\theta) = \mathbb{E}_{p(\mathbf{A}_t|\mathbf{o}_t), q(\mathbf{A}_\tau^t|\mathbf{A}_t)}\left[\left\|\mathbf{v}_\theta(\mathbf{A}_\tau^t, \mathbf{o}_t) - \mathbf{u}(\mathbf{A}_\tau^t|\mathbf{A}_t)\right\|^2\right], \quad (5)$$

where $\mathbf{v}_\theta(\mathbf{A}_\tau^t, \mathbf{o}_t)$ is the model's output, $\mathbf{u}(\mathbf{A}_\tau^t|\mathbf{A}_t)$ is the denoising vector field, and $q(\mathbf{A}_\tau^t|\mathbf{A}_t) = \mathcal{N}(\tau\mathbf{A}_t, (1-\tau)^2 I)$ is the noisy distribution.

The noisy actions $\mathbf{A}_\tau^t$ are computed as $\mathbf{A}_\tau^t = \tau\mathbf{A}_t + (1-\tau)\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. During inference, actions are generated by integrating the learned vector field from $\tau = 0$ to $\tau = 1$, starting with random noise $\mathbf{A}_0^t \sim \mathcal{N}(0, I)$. The integration rule is given by:

$$\mathbf{A}_{\tau+\delta}^t = \mathbf{A}_\tau^t + \delta\mathbf{v}_\theta(\mathbf{A}_\tau^t, \mathbf{o}_t), \quad (6)$$

where δ is the integration step size.

Methodology

This work aims to construct a novel efficient post-training method for the VLA Flow model and induce a practical offline RL fine-tuning algorithm (Fig. 1). First, we construct an energy-weighted VLA model and the corresponding energy-weighted flow matching loss. Then, we explain in detail how to construct the optimization objective of the adaptive scaling factor in the loss function, which aims to weigh the RL signal and gradient variance of the data samples. Finally, we give the solution equations of the scaling factor, as well as the corresponding bisection iteration and the fine-tuning algorithm of the VLA flow model.

Energy-weighted VLA flow model

Consider a data distribution $p(\mathbf{A}_t|\mathbf{o}_t)$, where $\mathbf{A}_t = [\mathbf{a}_t, \dots \mathbf{a}_{t+H}]$ is the action chunk, and $\mathbf{o}_t$ is the observation. Following ReinboT (Zhang et al. 2025b), we utilize a dense reward (App. Tab. 8) and undiscounted Return-To-Go (RTG). Action chunks in a trajectory share the same RTG (corresponding to the first action), divided by the remaining timesteps. The RTG advantage $R^*(\mathbf{A}_t, \mathbf{o}_t)$ is used as the negative energy function $-\mathcal{E}(\cdot)$, and α is used to represent the guidance scale β. Therefore, the policy distribution π based on EWFM is:

$$\pi(\mathbf{A}_t|\mathbf{o}_t) \propto p(\mathbf{A}_t|\mathbf{o}_t)\exp(\alpha R^*(\mathbf{A}_t, \mathbf{o}_t)).$$

We could learn the vector field of the policy distribution π by optimizing CEFM loss:

$$L^\tau(\theta) = \mathbb{E}[\mathcal{E}^*(\mathbf{A}_t, \mathbf{o}_t)||\mathbf{v}_\theta(\mathbf{A}_t^\tau, \mathbf{o}_t) - \mathbf{u}(\mathbf{A}_t^\tau|\mathbf{A}_t)||^2],$$

$$\text{where } \mathcal{E}^*(\mathbf{A}_t, \mathbf{o}_t) = \frac{\exp(\alpha R^*(\mathbf{A}_t, \mathbf{o}_t))}{\mathbb{E}_{\mathbf{A}_t^* \sim p(\cdot|\mathbf{o}_t)} \exp(\alpha R^*(\mathbf{A}_t^*, \mathbf{o}_t))}.$$

Here in the expectation in $L^\tau(\theta)$, the action $\mathbf{A}_t$ is sampled from the distribution $p(\cdot|\mathbf{o}_t)$, while $\mathbf{A}_t^\tau$ is to add a simple linear Gaussian noise (Optimal Transport) on $\mathbf{A}_t$, which is $\mathbf{A}_t^\tau = \tau\mathbf{A}_t + (1-\tau)\epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$ is a standard Gaussian variable. $\mathbf{u}(\mathbf{A}_t^\tau|\mathbf{A}_t)$ is the conditional vector field, which could be formulated by: $\mathbf{u}(\mathbf{A}_t^\tau|\mathbf{A}_t) = \mathbf{A}_t - \epsilon$.

In practical post-training, we utilize the sampling assumption instead of directly calculating the expectation. In each step, we sample a batch of B data pairs of $(\mathbf{o}_t, \mathbf{A}_t)$, and calculate the softmax of the energy function as the energy weight $\mathcal{E}^*$, and then calculate the weighted loss $L^\tau(\theta)$ within this batch, so the practical loss function should be:

$$L_1^\tau(\theta) = \sum_{i=1}^{B} w_i(\alpha)||\mathbf{v}_\theta(\{\mathbf{A}_t^i\}^\tau, \mathbf{o}_t) - \mathbf{u}(\{\mathbf{A}_t^i\}^\tau|\mathbf{A}_t^i)||^2,$$

$$\text{where } w_i(\alpha) = \frac{\exp(\alpha R^*(\mathbf{A}_t^i, \mathbf{o}_t))}{\sum_{j=1}^{B} \exp(\alpha R^*(\mathbf{A}_t^j, \mathbf{o}_t))}.$$

Here $\{\mathbf{A}_t^i\}$ are a batch sampled from $p(\cdot|\mathbf{o}_t)$, while $\{\mathbf{A}_t^i\}^\tau$ is the noisy action from $\mathbf{A}_t^i$. We utilize the standardized RL advantage $R^*(\cdot)$ as the energy weight in the loss function $L_1^\tau(\theta)$. The RL advantage is computed per task category (Eq. 4), where K depends on the number of tasks, not a hyperparameter. This enables trajectory-level quality comparison beyond single-step group-relative schemes, e.g., GRPO (Guo et al. 2025a). Moreover, α in the loss function $L_1^\tau(\theta)$ is an important adaptive parameter, and we adaptively adjust its value by constructing an optimization objective that trades off between RL signal and gradient variance.

Adaptive adjustment of the scaling factor α

The scaling factor α plays an important role in post-training of the VLA flow model. Intuitively, if α is too small, the advantage of the data sample may not be fully reflected, so the improvement on the original flow matching is not significant. When $\alpha = 0$, energy-weighted fine-tuning degenerates into vanilla flow matching. When α is large, fine-tuning tends to focus only on those data samples with higher energy. This will cause the sample data variance to be too large, resulting in gradient explosion and thus destroying the stability of fine-tuning. Therefore, the α is adaptively adjusted in each fine-tuning step, and its adjustment direction is to minimize $J(\alpha)$:

$$J(\alpha) = \text{Var}(\hat{g}(\alpha)) - \lambda S(\alpha),$$

where $\hat{g}(\alpha) = \nabla_\theta L_1^\tau(\theta) = \sum_{i=1}^{B} \hat{w}_i(\alpha)\nabla_\theta||\mathbf{v}_\theta(\{\mathbf{A}_t^i\}^\tau, \mathbf{o}_t) - \mathbf{u}(\{\mathbf{A}_t^i\}^\tau|\mathbf{A}_t^i)||^2$ is the gradient of the loss $L_1^\tau(\theta)$, where $\hat{w}_i = \exp(\alpha R^*(\mathbf{A}_t^i, \mathbf{o}_t))$ is the energy weight, so α tends to minimize the variation of the gradient. Then we define $S(\alpha) = \sum_{i=1}^{B} \hat{w}_i(\alpha)R^*(\mathbf{A}_t^i, \mathbf{o}_t)/\sum_{i=1}^{B} \hat{w}_i(\alpha)$ is a score function which shows the effect of the RL advantage as the energy weight, and α tends to maximize the score. λ is a hyperparameter, to adjust the ratio between the RL signal and the gradient variance.

Intuitively, minimizing $J(\alpha)$ is to balance the trade-off to emphasize the energy weight and prevent the post-training from gradient explosion. In order to make this optimization objective solvable, three assumptions need to be considered:

Algorithm 1: Bisection Iteration of Scaling Factor α

Input: RL advantages R_i^*, FM losses L_{FM}^i, batch size B, hyperparameter λ, value range of α $[\alpha_{min}, \alpha_{max}]$, number of iterations M and tolerance ϵ.

Calculate $\sigma_R^2 = \sum_i R_i^2/B$, $\mu_L = \sum_i L_{FM}^i/B$, $\sigma_L^2 = \sum_i(L_{FM}^i - \mu_L)^2/B$, $x_{low} = \sigma_R^2\alpha_{min}$, $x_{high} = \sigma_R^2\alpha_{max}$
Define $F(x) = 4\sqrt{x}e^{2x} - 2\sqrt{x}e^{x} - \frac{\lambda\sigma_R}{\sigma_L^2}$
for $m = 1$ to M **do**
  $x_{mid} = 0.5(x_{low} + x_{high})$
  if $|F(x_{mid})| < \epsilon$: **Break**
  if $F(x_{mid}) > 0$: $x_{high} = x_{mid}$
  else: $x_{low} = x_{mid}$
end for
$\alpha^* = \sqrt{0.5(x_{low} + x_{high})}/\sigma_A$
Return clip $(\alpha^*, \alpha_{max}, \alpha_{min})$

Algorithm 2: ARFM: Post-Training of VLA Flow Model

Input: Post-traing data $\{\mathbf{A}_t, \mathbf{o}_t\}$, batch size B, VLA flow model $\mathbf{v}_\theta$.

for a batch of data $\{\mathbf{A}_t^i, \mathbf{o}_t^i\}$ **do**
  for i in $[B]$ **do**
    Sample $\epsilon_i \sim \mathcal{N}(0, \mathbf{I})$, $\tau \sim$ Uniform$(0,1)$
    $\{\mathbf{A}_t^i\}^\tau = \tau(\mathbf{A}_t^i) + (1-\tau)\epsilon_i$
    $R^i = R^*(\mathbf{A}_t^i, \mathbf{o}_t^i)$, $g_i = \exp(R^*(\mathbf{A}_t^i, \mathbf{o}_t^i))$
    $L_{FM}^i = ||\mathbf{v}_\theta(\{\mathbf{A}_t^i\}^\tau, \mathbf{o}_t) - (\mathbf{A}_t^i - \epsilon)||^2$
  end for
  Calculate the optimal α^* from Alg. 1
  $w_i(\alpha^*) = \exp(\alpha^* g_i)/\sum_j \exp(\alpha^* g_j)$
  $L_1^\tau(\theta) = \sum_i w_i(\alpha^*)L_{FM}^i$
  Take a gradient step of $L_1^\tau(\theta)$
end for

Assumption 1. *The standardized RL advantage signal $R^*(\mathbf{A}_t, \mathbf{o}_t)$ is assumed to be Gaussian variable $\mathcal{N}(0, \sigma_R^2)$.*

Assumption 2. *The Conditional Flow Matching (CFM) loss $L_{CFM}^i = ||\mathbf{v}_\theta(\mathbf{A}_t^i, \mathbf{o}_t^i) - \mathbf{u}(\{\mathbf{A}_t^i\}^\tau | \mathbf{A}_t^i)||^2$ is also assumed to be Gaussian variable $\mathcal{N}(\mu_L, \sigma_L^2)$.*

Assumption 3. *When the batch size B is large enough, the sample-based expectation and variance could be utilized to approximate $\alpha_L, \sigma_A, \sigma_L$.*

In the post-training stage of the VLA flow model, the RL advantage is standardized, and the CFM loss value will quickly tend to have a lower variance, so the Gaussian distribution assumption here is mild and reasonable. Based on these assumptions, we can obtain two important corollaries on the solution of the scaling factor α. The derivation details are in the Appendix.

Corollary 1. *With the assumptions and functions defined above, the objective $J(\alpha)$ could be represented by:*

$$J(\alpha) = \sigma_L^2[e^{2\alpha^2\sigma_R^2} - e^{\alpha^2\sigma_R^2}] - \lambda\alpha\sigma_R^2. \tag{7}$$

Corollary 2. *The α^* which minimizes $J(\alpha)$ is solved by:*

$$4\sqrt{x}e^{2x} - 2\sqrt{x}e^x - \lambda\sigma_R/\sigma_L^2 = 0, \alpha^* = \sqrt{x}/\sigma_R. \tag{8}$$

Corollary 1 gives a specific solvable optimization objective for α. **Corollary** 2 gives the corresponding nonlinear equation for α, which can be quickly solved utilizing bisection iteration (Alg. 1). Therefore, we can utilize the Alg. 1 to obtain the optimal α^*, thereby obtaining the adaptive loss, which is utilized in the post-training process of the VLA flow model (Alg. 2).

Experiments

In this section, we conduct extensive experiments in various scenarios to evaluate the effectiveness of the proposed ARFM method. Specifically, we aim to examine the following five questions: **1)** Does ARFM show superior generalization ability compared to previous SOTA baseline methods? **2)** How resistant is ARFM to action noise in robot manipulation tasks with action noise? **3)** How does ARFM perform in data-scarce scenarios and lifelong learning, especially in terms of few-shot learning and continuous learning capabilities? **4)** To what extent do the key hyper-parameters in the ARFM method affect the performance of the VLA flow model? **5)** How does ARFM perform in real-world robot manipulation tasks, especially manipulating disturbed objects?

Experimental setup. We evaluate the performance of the proposed ARFM method in LIBERO (Liu et al. 2023) simulation and the real-world UR5 platform (Fig. 2). The LIBERO is a comprehensive lifelong learning benchmark that encompasses multiple task suites. These suites are designed to assess specific aspects of general robot manipulation, with tasks defined through language-guided instructions. Specifically, LIBERO includes four main suites: Object, Long, Spatial and Goal. Each suite is tailored to test different object manipulation capabilities and comprises 10 distinct tasks. For real-world evaluation, we utilize a UR5 robotic arm to assess the ARFM's performance. We set up three distinct pick-and-place tasks to evaluate the model's robustness under physical perturbations. Hyperparameters and reward configurations can be found in Appendix Tab. 7 and Tab. 8 respectively.

Figure 2: We evaluate performance on four categories of the LIBERO benchmark suite (Object, Long, Spatial and Goal) and three categories of real-world UR5 grasping and placing tasks. In realistic experiments, we introduce multiple external perturbations to the objects to be manipulated.

Baselines. We mainly consider two types of baseline models —— non-flow matching type and flow matching type. In the non-flow matching baseline model, we include the general auto-regressive models Octo (Team et al. 2024) and OpenVLA (Kim et al. 2024). We also consider diffusion-based models such as Diffusion Policy (Chi et al. 2023), MDT (Reuss et al. 2024) and Dita (Hou et al. 2025). In addition, we include QueST (Mete et al. 2024), which discretizes the continuous action space into a skill codebook through VQ-VAE and then predicts the skills auto-regressively. On the other hand, for the flow matching type of baseline methods, we first consider π_0 (Black et al. 2024), which is a VLA flow model that utilizes trajectory-level flow matching to achieve efficient policy learning. We also consider the offline RL fine-tuning baselines ReinboT (Zhang et al. 2025b) and RWR (Peters and Schaal 2007). ReinboT guides VLA action generation by predicting densely maximized future return, while RWR optimizes the VLA model by performing reward-weighted regression on samples. For a more fair comparison, we implemented the flow model versions of ReinboT and RWR based on the π_0 model. The baseline reproduction details are in the Appendix.

Multi-task learning setting. Tab. 1 compares ARFM with previous state-of-the-art baseline algorithms. The results show that the performance of flow matching models is higher than that of non-flow matching models overall, which may be mainly attributed to the powerful trajectory modeling ability of flow matching models. Moreover, among flow matching models, ARFM achieves the highest success rate 92.1%, which is 4.5% higher than the baseline π_0. It is followed by ReinboT (91.2) and RWR (90.8) baseline algorithms, which are 3.5% and 3.3% higher than the baseline π_0, respectively. The results preliminarily prove that our ARFM method can more efficiently fine-tune the VLA flow model through the adaptive energy-weighted scaling mechanism, thereby obtaining better generalization performance.

Model Type	Models	LIBERO Multi-task Learning				
		Goal	Spatial	Object	Long	Average
Non-Flow Matching	Octo	84.6	78.9	85.7	51.1	75.1
	OpenVLA	79.2	84.7	88.4	53.7	76.5
	Dita	85.4	84.2	**96.3**	63.8	82.4
	Diffusion Policy	68.3	78.3	92.5	50.5	72.4
	MDT	73.5	78.5	87.5	64.8	76.1
	QueST	80.8	87.4	93.6	68.8	82.7
Flow Matching	π_0	93.8	91.2	93.2	74.2	88.1
	ReinboT	94.0	95.6	93.8	81.4	91.2$_{(+3.5\%)}$
	RWR	94.4	94.0	94.3	80.4	90.8$_{(+3.1\%)}$
	ARFM (Ours)	**94.9**	**95.8**	95.0	**82.6**	**92.1**$_{(+\mathbf{4.5\%})}$

Table 1: Multi-task Success Rate (SR) on the LIBERO benchmark, with the best results highlighted in bold.

Models	LIBERO Action Perturbation				Avg.
	Goal	Spatial	Object	Long	
π_0	47.5	50.6	44.9	30.0	43.3
ReinboT	**51.4**	59.6	44.8	29.3	46.3$_{(+6.9\%)}$
RWR	49.5	60.1	46.9	29.1	46.4$_{(+7.2\%)}$
ARFM (Ours)	49.7	**61.1**	**48.9**	**33.0**	**48.2**$_{(+\mathbf{11.4\%})}$

Table 2: Average SR(%) under action perturbations (noise levels = 0.1, 0.15, 0.2, 0.25, 0.3) across four LIBERO suites.

Models	LIBERO-Long Few-Shot			Avg.
	30-shot	20-shot	10-shot	
π_0	41.7	33.8	22.1	32.5
ReinboT	39.5	37.5	24.6	33.9$_{(+4.1\%)}$
RWR	39.5	37.7	26.7	34.6$_{(+6.5\%)}$
ARFM (Ours)	**42.9**	**38.9**	**27.7**	**36.5**$_{(+\mathbf{12.2\%})}$

Table 3: Average SR(%) of few-shot learning settings on LIBERO-Long task (noise levels = 0.1, 0.15, 0.20).

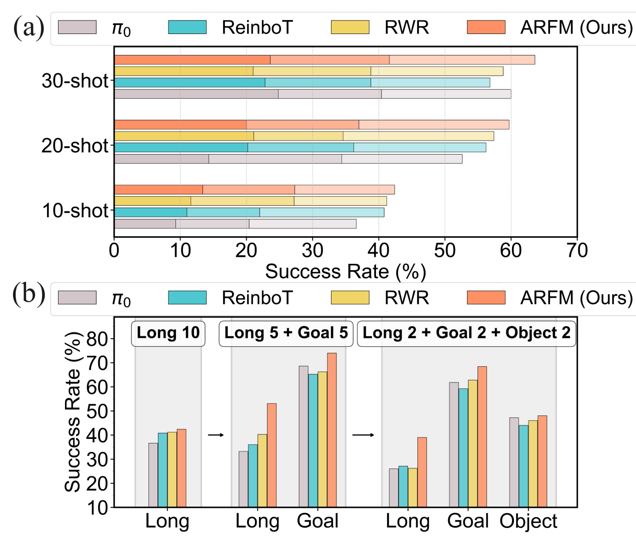

Figure 3: (a) Average SR(%) of different few-shot learning settings on LIBERO-Long under action perturbations (noise levels = 0.1, 0.15, 0.2). The three color shades from light to dark correspond to increasing levels of action noise (0.1 $\rightarrow$ 0.2). (b) Continuous learning on LIBERO benchmark (noise levels=0.1). Training sequence: Long 10 $\rightarrow$ Long 5 + Goal 5 $\rightarrow$ Long 2 + Goal 2 + Object 2. "Long 10" means that the model is trained on 10 trajectories per task on the LIBERO-Long suite, and other abbreviations are similar.

Action perturbation setting. To evaluate the model's resistance to perturbations, we added different levels of Gaussian noise ($0.1 \sim 0.3$) to the actions inferred by the model during evaluation. The experimental results are shown in Tab. 2. The results show that compared with the ReinboT (46.3) and RWR (46.4), ARFM has the highest average success rate (48.2), which is 11.4% higher than the baseline π_0 (43.3). This indicates that by balancing the RL signal and gradient variance, ARFM learns a more robust VLA model that is effectively resistant to action perturbations.

Few-shot learning setting. We examine the few-shot learning capabilities of the model in the LIBERO-Long suite, as shown in Tab. 3, Fig. 3 (a) and Appendix Tab. 5. The experiment shows that ARFM performs best (36.5), followed by RWR (34.6) and ReinboT (33.9), and finally π_0 (32.5). This shows that compared with the offline RL fine-tuning method introduced by the baseline ReinboT and RWR, ARFM's adaptive energy-weighted RL fine-tuning has better data utilization efficiency, thus reflecting better small sample learning and continuous learning performance.

Continual learning setting. In the continual learning setting, we evaluate the model's ability to learn tasks sequentially without forgetting previously acquired skills. We utilize the Negative Backward Transfer (NBT) metric to measure how much the model's performance degrades on task i after learning all tasks: $NBT = \frac{1}{T-1}\sum_i^{T-1}(max(0, (SR)_i - (SR)_i^T))$. $(SR)_i$ indicates the success rate of the model on task i after learning it, and $(SR)_i^T$ indicates the success rate of the model on task i after learning all tasks. The experiment result are in Tab. 4, Fig. 3 (b) and Appendix Tab. 6. Compared with the base-

Models	LIBERO Continual Learning							Avg. NBT ↓	Avg. SR ↑
	L 30 → L 15 + G 15		L 20 → L 10 + G 10		L 10 → L 5 + G 5 → L 2 + G 2 + O 2				
	L NBT ↓	Avg. SR ↑	L NBT ↓	Avg. SR ↑	L NBT ↓	G NBT ↓	Avg. SR ↑		
π_0	3.2	64.3	9.4	55.8	10.6	6.8	45.6	7.5	55.2
ReinboT	**0**	63.1	**6.7**	59.2	13.7	6.0	45.4	$6.6_{(-12.0\%)}$	$55.9_{(+1.2\%)}$
RWR	**0**	60.3	11.4	58.6	14.5	**3.4**	47.1	$7.3_{(-2.3\%)}$	$55.3_{(+0.2\%)}$
ARFM (Ours)	1.1	**67.6**	8.5	**61.1**	**3.4**	5.6	**54.2**	$\mathbf{4.7}_{(-38.0\%)}$	$\mathbf{61.0}_{(+10.5\%)}$

Table 4: Negative Backward Transfer (NBT) and Success Rate (SR) under continual learning on LIBERO-Long (L), Goal (G), and Object (O) suites (noise levels=0.1).

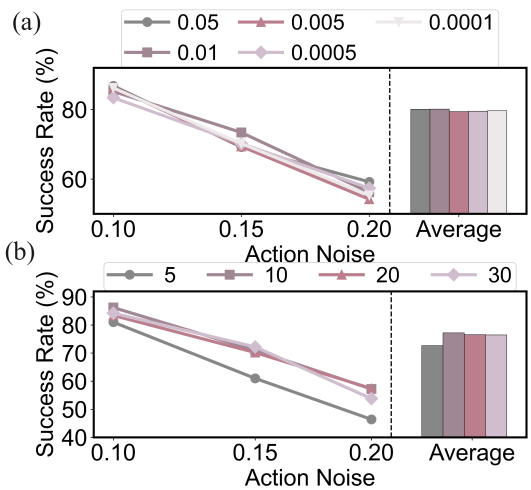

Figure 4: Ablation study in the LIBERO-Goal suite on the optimization objection hyperparameter λ (a) and the number of bisection iterations M (b).

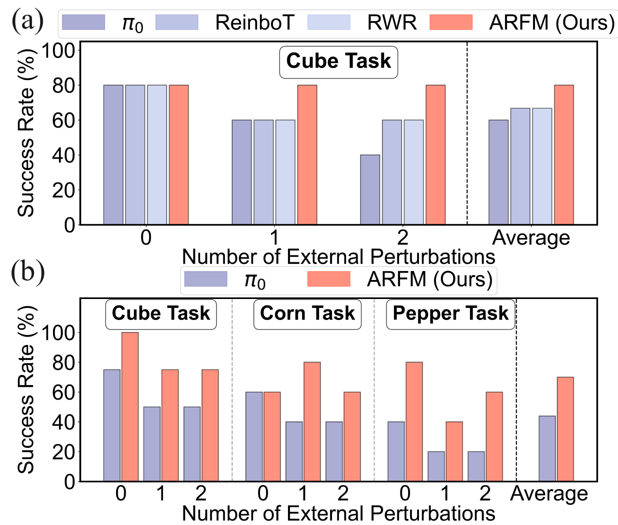

Figure 5: Performance comparison of real-world pick-and-place tasks under external perturbations.

line π_0 (SR of 55.2 and NBT of 7.5), ARFM (SR of 61.0 and NBT of 4.7) not only improves the final average success rate by 10.5%, but also reduces the NBT by 38.0%. This shows that ARFM is not only able to learn new tasks more effectively, but also significantly alleviates catastrophic forgetting, which is crucial for lifelong learning agents.

Ablation study

In the implementation of the proposed ARFM method, the optimization objective hyperparameter λ and the total number of bisection iterations M play an important role. The λ is utilized to balance the relative magnitude between the RL signal and the gradient variance, while the M directly affects the accuracy of the α^*. The results in Fig. 4 show that the performance of ARFM is insensitive to the λ, that is, different λ values have little effect on the performance of the VLA flow model. This may be mainly due to the adaptive adjustment ability of the loss weight of the ARFM method itself. In terms of the number of iterations M, the performance of the VLA flow model begins to stabilize when the M value is 10 or higher, indicating that ARFM can find an approximate optimal solution within a small number of steps.

Real-world experiments

We compared the model's performance on real-world tasks subject to external perturbations (Fig. 5). The results show that ARFM achieves the best performance, followed by ReinboT and RWR, with π_0 being the worst. Moreover, the ability of ARFM to resist perturbation is significantly better than that of baseline π_0. These results confirm that our adaptive offline RL fine-tuning method can enable the robot to perform more robust decision-making actions to cope with real-world complex scenarios.

Conclusion

In this work, we propose an adaptive offline RL post-training method for VLA flow models. We consider the balance between retaining RL advantage signals and controlling the loss gradient variance, so that the loss can change adaptively according to the quality distribution of the current batch of post-training data. Extensive experiments verify that ARFM has excellent generalization ability, robustness to dynamic perturbations, and few-shot learning and continuous learning capabilities. A promising research direction is to explore online RL post-training of VLA flow models.

Acknowledgments

This work was supported by the National Science and Technology Innovation 2030 - Major Project (Grant No. 2022ZD0208800), and NSFC General Program (Grant No. 62176215).

References

Ajay, A.; Du, Y.; Gupta, A.; Tenenbaum, J. B.; Jaakkola, T. S.; and Agrawal, P. 2022. Is Conditional Generative Modeling all you need for Decision Making? In *The Eleventh International Conference on Learning Representations*.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*.

Black, K.; Nakamoto, M.; Atreya, P.; Walke, H.; Finn, C.; Kumar, A.; and Levine, S. 2023. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Cadene, R.; Alibert, S.; Soare, A.; Gallouedec, Q.; Zouitine, A.; Palma, S.; Kooijmans, P.; Aractingi, M.; Shukor, M.; Aubakirova, D.; Russi, M.; Capuano, F.; Pascale, C.; Choghari, J.; Moss, J.; and Wolf, T. 2024. LeRobot: State-of-the-art Machine Learning for Real-World Robotics in Pytorch. https://github.com/huggingface/lerobot.

Chen, H.; Lu, C.; Ying, C.; Su, H.; and Zhu, J. 2022. Offline Reinforcement Learning via High-Fidelity Generative Behavior Modeling. In *The Eleventh International Conference on Learning Representations*.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2019. Neural Ordinary Differential Equations. arXiv:1806.07366.

Chen, Y.; Jha, D. K.; Tomizuka, M.; and Romeres, D. 2025. FDPP: Fine-tune Diffusion Policy with Human Preference. *arXiv preprint arXiv:2501.08259*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 02783649241273668.

Driess, D.; Xia, F.; Sajjadi, M. S.; Lynch, C.; Chowdhery, A.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; et al. 2023. Palm-e: An embodied multimodal language model.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Guo, Y.; Zhang, J.; Chen, X.; Ji, X.; Wang, Y.-J.; Hu, Y.; and Chen, J. 2025b. Improving Vision-Language-Action Model with Online Reinforcement Learning. *arXiv preprint arXiv:2501.16664*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. *CoRR*, abs/2006.11239.

Hou, Z.; Zhang, T.; Xiong, Y.; Duan, H.; Pu, H.; Tong, R.; Zhao, C.; Zhu, X.; Qiao, Y.; Dai, J.; et al. 2025. Dita: Scaling diffusion transformer for generalist vision-language-action policy. *arXiv preprint arXiv:2503.19757*.

Janner, M.; Du, Y.; Tenenbaum, J.; and Levine, S. 2022. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning*, 9902–9915. PMLR.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Kool, W.; van Hoof, H.; and Welling, M. 2019. Buy 4 reinforce samples, get a baseline for free!

Li, X.; Hsu, K.; Gu, J.; Pertsch, K.; Mees, O.; Walke, H. R.; Fu, C.; Lunawat, I.; Sieh, I.; Kirmani, S.; et al. 2024. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. arXiv:2210.02747.

Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. *arXiv preprint arXiv:2306.03310*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lu, C.; Chen, H.; Chen, J.; Su, H.; Li, C.; and Zhu, J. 2023. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, 22825–22855. PMLR.

Lu, G.; Guo, W.; Zhang, C.; Zhou, Y.; Jiang, H.; Gao, Z.; Tang, Y.; and Wang, Z. 2025. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*.

Mark, M. S.; Gao, T.; Sampaio, G. G.; Srirama, M. K.; Sharma, A.; Finn, C.; and Kumar, A. 2024. Policy Agnostic

RL: Offline RL and Online RL Fine-Tuning of Any Class and Backbone. *arXiv preprint arXiv:2412.06685*.

Mete, A.; Xue, H.; Wilcox, A.; Chen, Y.; and Garg, A. 2024. Quest: Self-supervised skill abstractions for learning continuous control. *Advances in Neural Information Processing Systems*, 37: 4062–4089.

Nakamoto, M.; Mees, O.; Kumar, A.; and Levine, S. 2024. Steering your generalists: Improving robotic foundation models via value guidance. *arXiv preprint arXiv:2410.13816*.

Peters, J.; and Schaal, S. 2007. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, 745–750. New York, NY, USA: Association for Computing Machinery. ISBN 9781595937933.

Reuss, M.; Yağmurlu, Ö. E.; Wenzel, F.; and Lioutikov, R. 2024. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. *arXiv preprint arXiv:2407.05996*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Tan, S.; Dou, K.; Zhao, Y.; and Krähenbühl, P. 2025. Interactive Post-Training for Vision-Language-Action Models. *arXiv preprint arXiv:2505.17016*.

Team, O. M.; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. 2024. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2022. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.

Zhai, S.; Bai, H.; Lin, Z.; Pan, J.; Tong, P.; Zhou, Y.; Suhr, A.; Xie, S.; LeCun, Y.; Ma, Y.; et al. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37: 110935–110971.

Zhang, H.; Ding, P.; Lyu, S.; Peng, Y.; and Wang, D. 2025a. Gevrm: Goal-expressive video generation model for robust visual manipulation. *arXiv preprint arXiv:2502.09268*.

Zhang, H.; Zhuang, Z.; Zhao, H.; Ding, P.; Lu, H.; and Wang, D. 2025b. ReinboT: Amplifying Robot Visual-Language Manipulation with Reinforcement Learning. *arXiv preprint arXiv:2505.07395*.

Zhang, S.; Zhang, W.; and Gu, Q. 2025. Energy-Weighted Flow Matching for Offline Reinforcement Learning. arXiv:2503.04975.

Zhang, Z.; Zheng, K.; Chen, Z.; Jang, J.; Li, Y.; Wang, C.; Ding, M.; Fox, D.; and Yao, H. 2024. Grape: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*.

Zhao, H.; Song, W.; Wang, D.; Tong, X.; Ding, P.; Cheng, X.; and Ge, Z. 2025. MoRE: Unlocking Scalability in Reinforcement Learning for Quadruped Vision-Language-Action Models. *arXiv preprint arXiv:2503.08007*.

Zheng, Q.; Le, M.; Shaul, N.; Lipman, Y.; Grover, A.; and Chen, R. T. 2023. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*.

Zhuang, Z.; Peng, D.; Liu, J.; Zhang, Z.; and Wang, D. 2024. Reinformer: Max-return sequence modeling for offline rl. *arXiv preprint arXiv:2405.08740*.