Proceedings of the Thirty-Sixth International Conference on Automated Planning and Scheduling (ICAPS 2026)

MO-VLA: Preference Adaptation for Vision-Language-Action Models via Multi-Objective Reinforcement Learning

Yan Yang[1,2], **Yuquan Wu**[1,2], **Mingxuan Jing**[1,2]
[1]Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
Beijing, China
yangyan2025@iscas.ac.cn, yuquan@iscas.ac.cn, jingmingxuan@iscas.ac.cn

Abstract

Vision-Language-Action (VLA) models trained with large-scale behavior cloning (BC) have achieved substantial progress in producing diverse and complex robotic behaviors, and have been widely deployed on robot manipulation, human-robot cooperation, and autonomous driving tasks. However, existing VLA models focus on generating complex behaviors from generalized human instructions, at the expense of capturing the critical preference and intent information regarding speed, smoothness, and safety conveyed by users. To address this limitation, we propose MO-VLA, a two-stage framework that integrates Multi-Objective Reinforcement Learning (MORL) into VLA training. Our framework is designed to maintain the performance of the pre-trained model while accelerating convergence when modeling user preferences. It operates in two stages. First, behavioral cloning (BC) on large-scale demonstrations equips the model with general operational skills. Subsequently, a multi-objective reinforcement learning (MORL)-based fine-tuning stage is used to adapt the policy to user-specific preferences. Here, a Feature-wise Linear Modulation (FiLM) mechanism is integrated into the action head to explicitly inject preference signals into the policy generation process. Experimental results on the Meta-World benchmark demonstrate that our method achieves superior multi-objective performance while maintaining a high task success rate. These results validate its capability for preference-aware action generation for various robotic tasks.

1 Introduction

One of the central objectives in general-purpose robotics is to construct intelligent agents capable of comprehending human instructions and executing diverse tasks in complex physical environments (Brohan et al. 2023; Driess et al. 2023). With the rapid advancement of embodied intelligence, there is a growing expectation that robots will not only follow explicit commands but also infer underlying semantics and user intent, thereby making appropriate decisions in dynamic and uncertain settings (Zitkovich et al. 2023; Ichter et al. 2022).

Recently, Vision-Language-Action (VLA) models have made significant progress in this direction (Zitkovich et al. 2023; Kim et al. 2024). By unifying visual perception, language understanding, and action generation within a unified multimodal framework, VLA models can execute complex tasks across diverse environments using natural-language instructions. While these models demonstrate strong cross-task generalization and compositional reasoning (Ghosh et al. 2024; Black et al. 2024), their capabilities are typically based on a single, implicit notion of "optimality" (Brohan et al. 2023).

However, this assumption of a single notion of "optimality" rarely holds in complex, real-world applications. In practical manipulation tasks, user preferences are diverse and dynamic, often requiring trade-offs among conflicting objectives such as speed, smoothness, energy efficiency, and safety. For instance, when handling fragile objects, a user may prioritize stability over speed, whereas in time-critical scenarios, efficiency may become the primary concern. Unfortunately, existing VLA models are typically trained to produce a unimodal, stereotyped behavior pattern, leaving them ill-equipped to adapt to user-specific preferences at inference time. This limitation raises a core question: *How can we systematically introduce a preference decision mechanism while retaining the powerful visual and linguistic generalization of VLA models, thereby enabling robots to adjust their strategies in real time according to varying user trade-offs?*

Although recent works have attempted to incorporate human preferences into robot learning, their approaches often suffer from limitations in both efficiency and flexibility. One mainstream approach employs Large Language Models (LLMs) or human feedback for reward-function design and learning. Ma et al. (2024) leverages LLMs to automatically synthesize reward functions, whereas Lee, Smith, and Abbeel (2021) learns reward models from human rankings of trajectories. However, these methods typically require retraining the policy for each new preference, which is incompatible with the need for real-time adjustment of behavioral style during interaction. Another line of research attempts to integrate preference learning directly into VLA or multimodal control frameworks (Zhang et al. 2025; Hung et al. 2025). Methods such as VTLA (Zhang et al. 2025) and NORA-1.5 (Hung et al. 2025) fine-tune models using "preferred/dispreferred" trajectory pairs via Direct Preference Optimization (DPO) or world-model-based ranking

mechanisms. While these methods substantially improve task efficiency and action safety, they fundamentally collapse complex task attributes into a single scalar reward signal (Amodei et al. 2016). Consequently, the learned policies encode a fixed trade-off scheme (e.g., a preset ratio of smoothness to speed). This design prevents the model from performing flexible reasoning and control along the Pareto front (Deb 2011) when facing unseen preference combinations, which is the central goal of Multi-Objective Reinforcement Learning (MORL) (Roijers et al. 2014).

Only a limited number of studies have attempted explicit preference-conditioned policy learning via MORL. Promptable Behaviors (Hwang et al. 2024), for instance, concatenates preference vectors with state encodings to train a policy that adjusts behavior based on preference weights in navigation tasks. However, this work is restricted to specific domains with low-dimensional state inputs and is not integrated with VLA architectures, thus lacking the generalist vision-language understanding required for complex manipulation tasks. At present, a unified framework that inherits the robust task-processing capabilities of VLA models while enabling fine-grained action-style control through explicit multi-objective modeling is still lacking.

To bridge this gap, we propose a novel preference-aware Vision-Language-Action model, termed MO-VLA. We formalize the robot control problem as a Multi-Objective Markov Decision Process (MOMDP) (Roijers et al. 2014; Hayes et al. 2022) to transcend the limitations of traditional scalar rewards by explicitly optimizing vector-valued returns. Drawing inspiration from recent advances in reinforcement-learning fine-tuning for VLA models (Guo et al. 2025), we design a two-stage training pipeline that first uses BC for pre-training and then applies MORL for preference alignment. Within this pipeline, we introduce a preference-injection mechanism based on Feature-wise Linear Modulation (FiLM) (Perez et al. 2018). Unlike simple feature concatenation (Hwang et al. 2024), FiLM dynamically modulates action generation via affine transformations applied to intermediate feature channels (Perez et al. 2018). This approach endows the robot with runtime controllability, enabling it not only to comprehend semantic instructions (e.g., "open the drawer") but also to generate diverse trajectories that realize different trade-offs among speed, smoothness, and safety along the Pareto front. Given a preference weight $\mathbf{w}$ at inference time, the policy can be steered toward different points on this front. The code is available on GitHub:(https://github.com/yangyan010119/MO-VLA).

The contributions of this paper are summarized as follows:

- **Preference-aware VLA framework.** To address the challenge of preference-aware control in Vision-Language-Action models, we are, to the best of our knowledge, the first to integrate a Multi-Objective Markov Decision Process (MOMDP) into a VLA framework for robotic manipulation.
- **FiLM-based multi-objective policy learning.** We propose a two-stage pipeline that first performs BC pre-training and then MORL fine-tuning. We further introduce a FiLM-based preference-injection mechanism, enabling the training of preference-aware policies without modifying the pre-trained network backbone.
- **Pareto-efficient manipulation with semantic grounding.** On the Meta-World benchmark, MO-VLA recovers diverse Pareto-optimal behaviors and aligns more closely with user-specified preferences than existing baselines, while maintaining strong task success under natural-language instructions.

2 Preliminaries

In this section, we formalize the multi-objective control problem and introduce the fundamental concepts underlying Vision-Language-Action (VLA) models and Feature-wise Linear Modulation (FiLM), which serve as the building blocks of our framework.

2.1 Multi-Objective Markov Decision Processes

We model robotic manipulation tasks with conflicting objectives as a Multi-Objective Markov Decision Process (MOMDP) (Roijers et al. 2014). A MOMDP is defined as the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathbf{r}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ denotes the environment transition probability distribution, and $\gamma \in [0, 1)$ is the discount factor.

Unlike standard MDPs with a scalar reward, the reward function $\mathbf{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^K$ returns a K-dimensional vector of distinct objectives (Abels et al. 2019):

$$\mathbf{r}(s, a) = [r^{(1)}(s, a), \ldots, r^{(K)}(s, a)]^\top, \tag{1}$$

where each component $r^{(k)}$ corresponds to a specific task metric (e.g., efficiency or smoothness). The goal is to maximize the expected cumulative discounted return vector $\mathbf{J}(\pi)$:

$$\mathbf{J}(\pi) = \mathbb{E}_{\pi,\mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t) \right] \in \mathbb{R}^K. \tag{2}$$

To evaluate policy performance, we explicitly define the vector-valued action-value function (vector Q-function) $\mathbf{Q}^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^K$ (Yang, Sun, and Narasimhan 2019) as:

$$\mathbf{Q}^\pi(s, a) = \mathbb{E}_{\pi,\mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t) \mid s_0 = s, a_0 = a \right]. \tag{3}$$

Each component $Q^{\pi,(k)}(s, a)$ estimates the expected return associated with the k-th objective.

2.2 Preferences and Pareto Optimality

Since objectives in an MOMDP are often conflicting, there is typically no single policy that simultaneously maximizes all dimensions of $\mathbf{J}(\pi)$. Instead, optimality depends on user preferences, parameterized by a weight vector $\mathbf{w} \in \mathcal{W}$, where $\mathcal{W}$ is the $(K-1)$-dimensional simplex:

$$\mathcal{W} = \left\{ \mathbf{w} \in \mathbb{R}^K \mid \mathbf{w} \geq 0, \|\mathbf{w}\|_1 = 1 \right\}. \tag{4}$$

The weight vector $\mathbf{w}$ represents the relative importance of each objective. We adopt linear scalarization to map the vector reward to a scalar utility (Moffaert, Drugan, and Nowé

2013): $r_{\mathbf{w}}(s,a) = \mathbf{w}^\top \mathbf{r}(s,a)$. Correspondingly, the scalarized Q-value under preference $\mathbf{w}$ is given by the inner product between the preference vector and the vector Q-function:

$$Q^\pi_{\mathbf{w}}(s,a) = \mathbf{w}^\top \mathbf{Q}^\pi(s,a). \tag{5}$$

The objective is to learn a preference-conditioned policy $\pi(a|s,\mathbf{w})$ that maximizes the scalarized return $J_{\mathbf{w}}(\pi) = \mathbf{w}^\top \mathbf{J}(\pi)$ for any given $\mathbf{w}$.

When the preference vector $\mathbf{w}$ varies, we characterize a set of policies using *Pareto dominance* (Moffaert and Nowé 2014). A policy π_1 is said to Pareto-dominate π_2 (denoted $\pi_1 \succ \pi_2$) if $J^{(k)}(\pi_1) \geq J^{(k)}(\pi_2)$ for all k, and there exists at least one k such that $J^{(k)}(\pi_1) > J^{(k)}(\pi_2)$. The set of all non-dominated policies constitutes the *Pareto front*. Our goal is to learn a unified policy that approximates this front by dynamically adapting to any $\mathbf{w} \in \mathcal{W}$ (Pirotta, Parisi, and Restelli 2015).

2.3 Vision-Language-Action Models

Vision-Language-Action (VLA) models have significantly advanced general-purpose robotic control (Brohan et al. 2023; Zitkovich et al. 2023). A VLA model typically comprises a Vision-Language Model (VLM) backbone and an action head. Formally, given a high-dimensional visual observation o_t and a natural-language instruction x, the model generates an action a_t:

$$a_t = \pi(a_t \mid \mathbf{z}_t), \quad \text{where} \quad \mathbf{z}_t = f_\theta(o_t, x). \tag{6}$$

Here, $f_\theta(\cdot)$ denotes the VLM backbone parameterized by θ, which fuses visual and linguistic inputs into a semantic embedding $\mathbf{z}_t$. While standard VLA models generalize well, they typically lack mechanisms to explicitly handle the diverse trade-offs modeled by MOMDP.

2.4 Feature-wise Linear Modulation

Feature-wise Linear Modulation (FiLM) (Perez et al. 2018) is a general-purpose conditioning mechanism that modulates neural-network computations via affine transformations. Given an intermediate feature vector $\mathbf{h}$ and a conditioning input $\mathbf{c}$, a FiLM layer computes:

$$\text{FiLM}(\mathbf{h} \mid \mathbf{c}) = \boldsymbol{\delta}(\mathbf{c}) \odot \mathbf{h} + \boldsymbol{\beta}(\mathbf{c}), \tag{7}$$

where $\boldsymbol{\delta}(\cdot)$ and $\boldsymbol{\beta}(\cdot)$ are learned functions mapping $\mathbf{c}$ to scale and shift parameters, respectively, and $\odot$ denotes the element-wise product. In this work, we leverage FiLM to inject the preference vector $\mathbf{w}$ into the policy, allowing the VLA model to modulate its behavior while keeping pretrained representations intact.

3 Method

We propose MO-VLA, a unified framework that endows general-purpose VLA models with the ability to adapt to diverse user preferences. As illustrated in Figure 1, our approach introduces a preference-conditioning mechanism and trains the model using a two-stage pipeline: (1) BC pre-training to acquire robust manipulation priors, and (2) MORL fine-tuning to enable dynamic preference adaptation (Rajeswaran et al. 2018).

3.1 Setup and Overview

Based on the definitions in Sec. 2, we formalize the robotic manipulation problem as a Multi-Objective MDP (MOMDP).

- **Rewards and Preferences:** The environment provides a vector reward $\mathbf{r}_t$, and user preferences are represented by a weight vector $\mathbf{w}$ lying on the simplex $\mathcal{W}$.
- **Observations:** Under partial observability, the policy relies solely on high-dimensional visual observations o_t and natural-language instructions x.
- **Backbone:** We utilize a pretrained vision-language model (VLM) backbone f_θ (BLIP-2 (Li et al. 2023)) to map inputs (o_t, x) to a semantic embedding $\mathbf{z}_t = f_\theta(o_t, x)$. To preserve visual-linguistic generalization, f_θ is optimized via LoRA (Hu et al. 2022) during Stage I and kept **frozen** during Stage II.

3.2 Architecture: Preference-Conditioned VLA Policy

To address the multi-objective challenge, we construct a preference-conditioned policy $\pi_\phi(\mathbf{a}_t \mid \mathbf{z}_t, \mathbf{w})$. Unlike previous preference-conditioned policies (Abels et al. 2019; Hwang et al. 2024), which simply concatenate preference features with state features, we employ a cascaded FiLM architecture (Perez et al. 2018) that preserves the original MLP structure of π_{prior} and thereby avoids disrupting the pretrained network weights.

FiLM Modulation Mechanism. To effectively integrate user preference $\mathbf{w}$ into the policy, we first project the raw preference vector into a latent space. A shared preference encoder u_ξ, implemented as a two-layer MLP, transforms $\mathbf{w} \in \mathbb{R}^K$ to a higher-dimensional embedding $\mathbf{e}_{\mathbf{w}} \in \mathbb{R}^{128}$:

$$\mathbf{e}_{\mathbf{w}} = u_\xi(\mathbf{w}). \tag{8}$$

Within each FiLM modulation layer, two distinct linear heads, g_δ and g_β, project this shared embedding into channel-wise scaling ($\boldsymbol{\delta}$) and shifting ($\boldsymbol{\beta}$) parameters:

$$\boldsymbol{\delta}_k = g_{\delta_k}(\mathbf{e}_{\mathbf{w}}), \quad \boldsymbol{\beta}_k = g_{\beta_k}(\mathbf{e}_{\mathbf{w}}). \tag{9}$$

Cascaded Architecture. We integrate these modulation modules into a cascaded multi-layer perceptron (MLP) that serves as the action head. The network processes the input features progressively through hidden layers with dimensions $d_1 = 512$ and $d_2 = 256$. The data flow through the actor head $\pi_\phi(\cdot|\mathbf{z}_t, \mathbf{w})$ is formally defined as:

$$\mathbf{h}_0 = \text{Linear}_{d_1}(\mathbf{z}_t), \tag{10}$$

$$\tilde{\mathbf{h}}_0 = \boldsymbol{\delta}_0 \odot \mathbf{h}_0 + \boldsymbol{\beta}_0, \tag{11}$$

$$\mathbf{h}'_0 = \text{ReLU}(\tilde{\mathbf{h}}_0), \tag{12}$$

$$\mathbf{h}_1 = \text{Linear}_{d_2}(\mathbf{h}'_0), \tag{13}$$

$$\tilde{\mathbf{h}}_1 = \boldsymbol{\delta}_1 \odot \mathbf{h}_1 + \boldsymbol{\beta}_1, \tag{14}$$

$$\mathbf{h}'_1 = \text{ReLU}(\tilde{\mathbf{h}}_1), \tag{15}$$

$$\mathbf{a}_t = \tanh(\text{Linear}_{\text{out}}(\mathbf{h}'_1)). \tag{16}$$

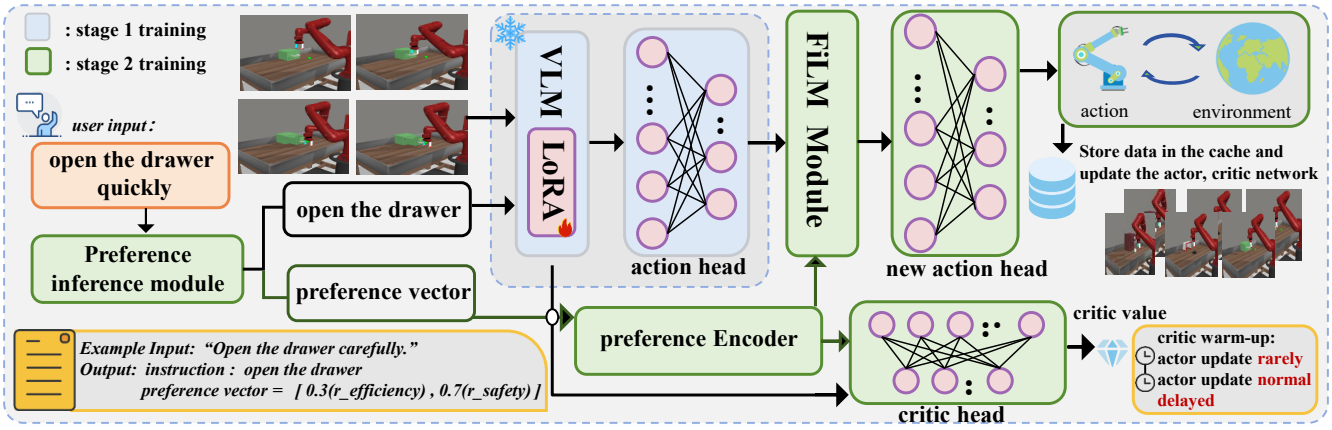

Figure 1: The two-stage training pipeline of MO-VLA. **Stage I (blue area):** A preference-agnostic VLA policy π_{prior} is pre-trained via behavior cloning on a multi-task expert demonstration dataset $\mathcal{D}_{\text{BC}}$. **Stage II (green area):** The VLA backbone is frozen, and the policy head is fine-tuned using an off-policy MORL algorithm. The agent interacts with the environment to collect preference-conditioned trajectories in a replay buffer $\mathcal{D}_{\text{RL}}$, which is used to update a FiLM-conditioned actor and multi-objective critics.

Here, $\mathbf{z}_t \in \mathbb{R}^{2560}$ denotes the VLM feature. These modulation layers apply preference-based affine transformations at multiple abstraction levels, enabling fine-grained control over the generated actions.

3.3 Training Pipeline

Stage I: Behavior Cloning Warm-up. The objective of the first stage is to acquire a preference-agnostic policy prior to π_{prior} that can perform tasks with basic competence. We utilize a multi-task demonstration dataset $\mathcal{D}_{\text{BC}}$ and jointly optimize the LoRA parameters θ and the MLP head ϕ_0 to minimize the mean squared error (MSE) between predicted and expert actions:

$$\mathcal{L}_{\text{BC}}(\theta, \phi_0) = \mathbb{E}_{(\mathbf{o}_t, x, \mathbf{a}_t)\sim\mathcal{D}_{\text{BC}}} \left[\|\pi_{\phi_0}(f_\theta(\mathbf{o}_t, x)) - \mathbf{a}_t\|_2^2 \right]. \tag{17}$$

To ensure a smooth transition to the next stage, we initialize the MLP weights of the policy network by directly copying them from the pretrained π_{prior}, and we initialize the FiLM parameters as an approximate identity mapping (i.e., $\boldsymbol{\delta} \approx \mathbf{1}, \boldsymbol{\beta} \approx \mathbf{0}$). This guarantees that the initial behavior of π_ϕ in Stage II is identical to that of π_{prior}, preventing performance collapse due to structural changes.

Stage II: Off-Policy MORL Fine-Tuning. In the second stage, we freeze the visual backbone and fine-tune the policy using an off-policy MORL algorithm to approximate the Pareto front. We employ twin vector-valued critics $\mathbf{Q}_{\psi_1}$ and $\mathbf{Q}_{\psi_2}$.

Preference sampling. At the beginning of each training episode, we sample a preference vector $\mathbf{w} \sim p(\mathbf{w})$ according to the multi-objective algorithm under consideration. The transitions $(\mathbf{z}_t, \mathbf{w}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{z}_{t+1})$ are then stored in a replay buffer.

Robust Multi-Objective Critic Learning. We employ twin vector-valued critics $\mathbf{Q}_{\psi_1}$ and $\mathbf{Q}_{\psi_2}$, to estimate the expected return vector. To mitigate overestimation bias, we use a conservative target-selection mechanism based on the current preference $\mathbf{w}_t$, similar to the clipped double Q-learning used in TD3 (Fujimoto, van Hoof, and Meger 2018). Specifically, when computing the TD target, we select the target vector from the twin critics that yields the smaller scalarized Q-value:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{r}_t + \gamma(1 - d_t)\mathbf{Q}'_{j^*}, \\ \text{where } j^* &= \arg\min_{i\in\{1,2\}} (\mathbf{w}_t^\top \mathbf{Q}'_i). \end{aligned} \tag{18}$$

The critics are updated to minimize the vector Bellman error:

$$\mathcal{L}_{\text{critic}}(\psi) = \sum_{j=1}^{2} \mathbb{E}\left[\|\mathbf{Q}_{\psi_j}(\mathbf{z}_t, \mathbf{a}_t, \mathbf{w}_t) - \mathbf{y}_t\|_2^2 \right]. \tag{19}$$

To further stabilize the fine-tuning process, we explicitly address the learning dynamics. Off-policy reinforcement learning is prone to instability during the early stages of fine-tuning, as the critics initially lack accurate value estimates. Updating the actor against erroneous Q-values can precipitate policy degradation, undermining the behaviors learned during pre-training. To mitigate this, we implement a critic warm-up schedule. During the initial N_{warmup} steps, the actor is updated at a very low frequency, allowing the critics to prioritize learning accurate value estimates. After the warm-up phase, the actor is updated at a standard delayed frequency relative to the critics to accelerate optimization. This schedule ensures that the *value landscape* stabilizes before the policy begins aggressive optimization for specific preferences.

Actor Update with Regularization. The actor aims to maximize the expected scalarized return. We minimize the following composite loss function:

$$\mathcal{L}_{\text{actor}}(\phi) = \mathbb{E}\left[-\mathbf{w}_t^\top \mathbf{Q}_{\psi_1}(\mathbf{z}_t, \pi_\phi(\mathbf{z}_t, \mathbf{w}_t), \mathbf{w}_t) \right] + \alpha \mathcal{L}_{\text{BC-reg}}. \tag{20}$$

To prevent catastrophic forgetting of the safe manipulation manifold learned in Stage I, we incorporate a dynamic regularization term $\mathcal{L}_{\text{BC-reg}}$. We regularize the outputs based on the *frozen* prior policy, which serves as a general policy incapable of handling preference information:

$$\mathcal{L}_{\text{BC-reg}} = \mathbb{E}\left[\|\pi_\phi(\mathbf{z}_t, \mathbf{w}_t) - \pi_{\text{prior}}(\mathbf{z}_t)\|_2^2\right]. \quad (21)$$

The regularization weight α is linearly annealed from 1.0 to 0.1 over the course of training. The intuition is as follows: early in training, we force the policy to adhere closely to expert demonstrations to ensure safety; as training progresses, we gradually relax this constraint, allowing the agent to explore more freely to optimize for specific preference objectives (Fujimoto and Gu 2021).

3.4 Language-to-Preference at Inference

During deployment, user intents are typically expressed as holistic natural-language commands (e.g., "*open the drawer quickly*"). To bridge the gap between high-level semantics and numerical control, we employ a lightweight LLM-based instruction parser (Yu et al. 2023). This module disentangles the raw command into two distinct components: a standard task instruction x (e.g., "*open the drawer*") that is fed into the VLM backbone for visual grounding, and a target preference vector $\mathbf{w}$ that conditions the policy via the FiLM layers. This mechanism ensures that user intent is effectively translated into both the semantic context and the specific behavioral style required by our policy. In practice, we implement this parser with a lightweight LLM-based module that maps linguistic modifiers such as "quickly" or "carefully" into normalized preference weights over efficiency and safety.

4 Experiments

In this section, we empirically evaluate MO-VLA on the Meta-World benchmark to assess its effectiveness. We aim to answer the following three research questions:

1. **Multi-Objective Efficacy:** Can MO-VLA effectively approximate the Pareto front to accommodate diverse user preferences while preserving the generalization capability of the pre-trained VLA backbone?
2. **Architectural Superiority:** Does our FiLM-based preference injection mechanism outperform standard feature concatenation strategies used in prior work?
3. **Algorithmic Robustness:** How does our off-policy MORL framework compare with standard on-policy baselines on complex manipulation tasks?

4.1 Experimental Setup

Environment and Tasks. We evaluate our framework on **Meta-World** (Yu et al. 2019), a standard benchmark for multi-task robotic manipulation. Because the original environment provides only scalar rewards based on task completion, it is insufficient for evaluating trade-offs among multiple objectives. We therefore extend the environment into a **Multi-Objective Markov Decision Process (MOMDP)** by designing a bi-objective reward structure $\mathbf{r} = [r_{\text{eff}}, r_{\text{safe}}]^\top$ for the manipulation tasks:

- **Efficiency (r_{eff}):** Encourages rapid task completion. It combines the sparse task-completion reward provided by the environment with a negative time-step penalty.
- **Safety (r_{safe}):** Encourages stable and energy-efficient motion. We formulate this objective such that higher rewards correspond to lower energy consumption (smaller action norms) and smoother trajectories (reduced action jerk).

The two-dimensional reward combines task-completion progress and a per-step penalty for the efficiency objective, while the safety objective encourages lower action energy and smoother trajectories through an energy term and a jerk-based smoothness term.

Implementation Details. We reuse the same BLIP-2 (3B) backbone and two-stage training protocol described in Sec. 3: in Stage II, the backbone (including the LoRA adapters) is frozen, and only the FiLM-conditioned action head and the multi-objective critics are optimized. Unless otherwise specified, all RL experiments use a training budget of $300{,}000$ environment steps, AdamW with a learning rate of 1×10^{-4}, batch size 128, and a replay buffer of size 4×10^5.

Evaluation Metrics. Following standard MORL evaluation protocols, we employ three complementary metrics to assess performance:

1. **Hypervolume (HV):** Measures the volume of the objective space dominated by the learned Pareto front with respect to a reference point. It serves as the primary metric for assessing both the convergence and diversity of the learned multi-objective solutions (Zitzler et al. 2003).
2. **Expected Utility (EU):** Quantifies the average scalarized return $\mathbb{E}_{\mathbf{w}}[\mathbf{w}^\top \mathbf{r}]$ under the preference distribution, reflecting the policy's ability to align with user intent.
3. **Success Rate (SR):** Measures the percentage of successfully completed episodes, ensuring that preference alignment does not come at the cost of task failure.

4.2 Comparative Analysis of Preference Injection in VLA

In this experiment, we validate the effectiveness of the MO-VLA architecture by comparing our method with Promptable Behaviors (Hwang et al. 2024), a representative framework for preference-conditioned control. Although originally designed for navigation, Promptable employs a mechanism in which preference vectors are concatenated with state embeddings to modulate behavior. In addition, to demonstrate the generality of our approach, we adopt two classical off-policy multi-objective reinforcement learning algorithms, GPI-LS (Alegre et al. 2023) and CAPQL (Lu, Herman, and Yu 2023), as the main algorithms in the reinforcement learning stage.

Baseline Adaptation. Direct comparison is challenging due to domain differences. To ensure a fair comparison that isolates the impact of the preference injection mechanism and the learning algorithm, we reimplement Promptable on

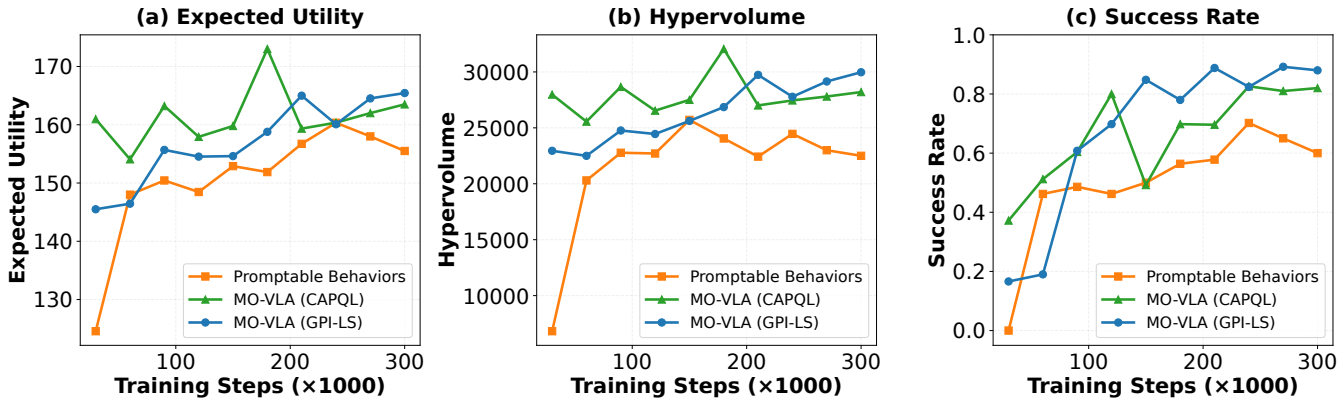

Figure 2: **Comparative Performance on `drawer-open`.** We compare MO-VLA (implemented with the GPI-LS and CAPQL algorithms) with the adapted Promptable Behaviors baseline. MO-VLA achieves substantially higher Hypervolume and Expected Utility while maintaining a stable success rate, whereas the baseline method exhibits noticeably weaker performance.

the `drawer-open` task using the same frozen BLIP-2 backbone as our method, retain its original early concatenation of the preference vector with visual features, and train the policy with PPO under the same reward structure and training horizon as our method.

Results and Analysis. As illustrated in Figure 2, MO-VLA consistently outperforms the Promptable baseline across all metrics. Across all evaluated preference settings, both variants of our method (GPI-LS and CAPQL) converge rapidly and achieve substantially higher Hypervolume and Expected Utility, while also attaining consistently higher success rates. In contrast, the Promptable baseline exhibits much slower improvement in Hypervolume and Expected Utility and converges to clearly inferior levels. Its success rate is also noticeably lower, indicating that it struggles to capture the desired preference trade-offs in this task.

In summary, these results demonstrate that, compared with naive concatenation mechanisms and standard on-policy fine-tuning, our FiLM-based MO-VLA architecture exhibits clear advantages on multi-objective control tasks.

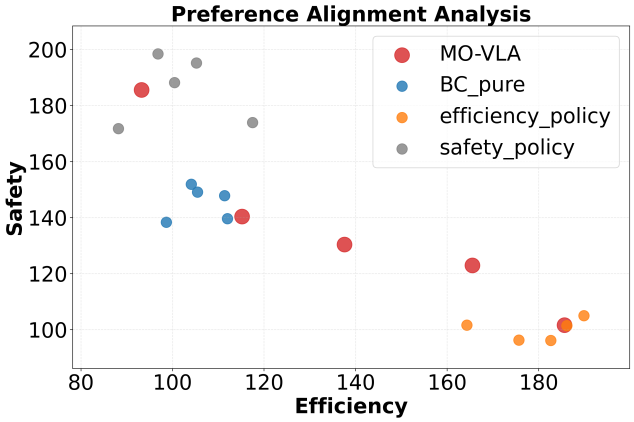

Figure 3: Reference alignment analysis on `drawer-open`. We evaluate the cumulative rewards for Efficiency and Safety under five user preferences ranging from efficiency-oriented to safety-oriented, where each point corresponds to one of the preference settings described in the text.

4.3 Multi-Objective Preference Alignment and Trade-off Analysis

In this experiment, we investigate the core capability of MO-VLA: dynamic adaptation to user preferences. We aim to verify whether our policy can seamlessly transition between conflicting behavioral modes (e.g., maximizing efficiency vs. ensuring safety) in response to user input, in contrast to single-objective baselines that are theoretically confined to fixed behaviors.

Experimental Setup. We conduct evaluations on the `drawer-open` task, using the bi-objective reward structure defined in Sec. 4.1: Efficiency (r_{eff}) and Safety (r_{safe}). We compare three paradigms:

- **Pure BC (Prior):** A static policy π_{prior} obtained from Stage I pre-training, serving as the initialization point.
- **iRe-VLA (Single-Objective Baselines) (Guo et al. 2025):** To provide a strong comparative baseline for single-objective performance, we follow the fine-tuning protocol proposed in iRe-VLA. Because iRe-VLA is inherently designed for scalar rewards, we train two separate and specialized agents under two extreme preference settings:
  - **Efficiency policy:** Fine-tuned using PPO solely on the efficiency objective, i.e., the policy trained under the *"Rapidly"* preference.
  - **Safety policy:** Fine-tuned using PPO solely on the safety objective, i.e., the policy trained under the *"Precisely"* preference.

  Both variants use the same frozen VLM backbone and action head architecture as ours, but lack the preference injection module.
- **MO-VLA (Ours):** A single unified policy conditioned on variable preferences.

Method	Button-Press		Window-Open		Drawer-Open		Hammer		Drawer-Close		Door-Open		Faucet-Close		Faucet-Open	
	SR	Rank	SR	Rank	SR	Rank	SR	Rank	SR	Rank	SR	Rank	SR	Rank	SR	Rank
DrQ-v2 (Pixel RL)	0.15	4	0.79	3	0.30	4	0.45	4	**0.93**	1	0.12	4	0.83	3	**1.00**	1
BC Policy	0.60	3	0.70	4	0.60	3	0.60	3	0.55	4	0.65	3	0.60	4	0.80	4
iRe-VLA	**1.00**	1	0.84	2	0.84	2	0.80	2	0.89	2	**0.84**	1	0.90	2	0.92	3
MO-VLA (Ours)	**1.00**	1	**0.89**	1	**0.88**	1	**0.86**	1	0.81	3	0.79	2	**0.96**	1	**1.00**	1

Table 1: Task Success Rate Comparison on Meta-World Tasks. We compare the average success rates of MO-VLA with those of the pixel-based DrQ-v2 RL baseline, the Pure BC prior, and the single-objective iRe-VLA baseline. Tasks are sorted by the success rate of our method. MO-VLA achieves Rank 1 on most tasks, demonstrating superior robustness.

For evaluation, we define five linguistic instructions to span a spectrum of preferences—"Rapidly", "Swiftly", "Balanced", "Carefully", and "Precisely"—and, for each algorithm, we separately evaluate the bi-objective rewards of the corresponding policies under these five input preferences. These linguistic preferences are converted into preference weight vectors by a lightweight language-to-preference parser that maps style-related expressions to normalized efficiency–safety trade-offs.

Results and Analysis. Figure 3 reports the efficiency and safety returns under different preference instructions. We observe that Pure BC and the two single-objective iRe-VLA baselines exhibit only minor variations across the entire preference range: Pure BC maintains moderate performance on both objectives but largely ignores changes in the input preferences; the efficiency policy consistently favors high efficiency at the expense of safety, whereas the safety policy shows the opposite trend. In contrast, MO-VLA continuously adjusts its behavior along the efficiency–safety spectrum as the instruction shifts from *"Rapidly"* to *"Precisely"*, and its performance under these two extreme preferences closely matches that of the corresponding single-objective expert models. This indicates that MO-VLA captures the continuous trade-off between efficiency and safety within a single parameterized policy.

4.4 Task Success Rate and Policy Robustness Analysis

Experimental Configuration. Under a unified setup, we consider the following methods:

- **Pure BC (Prior):** A static policy obtained via behavior cloning on the Stage I offline demonstrations, serving as a reference for pure imitation-learning performance.
- **DrQ-v2 (Pixel RL Baseline):** A strong model-free visual RL baseline trained from scratch with pixel observations on the same Meta-World tasks. We closely follow the original DrQ-v2 implementation and hyperparameter configuration, training directly from environment images with the original scalar reward (binary success plus dense shaping), without any preference conditioning or offline demonstrations (Yarats et al. 2022). This baseline represents a high-performing, purely RL-based controller optimized solely for task completion under pixel observations.
- **iRe-VLA:** Following the iRe-VLA protocol (Guo et al. 2025), we fine-tune the model with PPO using the original scalar reward of the environment (binary success plus dense shaping), without any preference conditioning. This provides a strong upper bound for single-task performance optimized solely for task completion.
- **MO-VLA (Ours):** Our multi-objective, preference-conditioned policy learning framework. To assess overall reliability, we sample a diverse set of preference vectors from $p(\mathbf{w})$ and report the average task success rate across these preferences.

To better approximate real-world robotic deployment, we adopt stricter success criteria (e.g., tighter tolerance thresholds on object placement and articulation) during Stage II fine-tuning and evaluation for all RL-based methods.

Results and Analysis. Quantitative results in Table 1 demonstrate that MO-VLA consistently achieves strong task performance across all evaluated scenarios. Across all eight tasks, MO-VLA achieves a consistently and significantly higher success rate than the Pure BC baseline, indicating its ability to refine policies to meet high-precision requirements. Moreover, MO-VLA outperforms the expert baseline iRe-VLA, which is optimized solely for scalar task rewards, and substantially surpasses the pixel-level reinforcement learning baseline DrQ-v2. Given that MO-VLA not only aligns closely with user preferences but also maintains near-expert task success rates, these findings further highlight the superiority and practical advantages of our proposed framework.

4.5 Ablation Study and Component Analysis

To rigorously validate the necessity of each algorithmic component within MO-VLA, we conduct a comprehensive ablation study on the `drawer-open` task. We focus on evaluating the impact of removing three critical design elements: the FiLM-based architecture, the behavior-cloning (BC) regularization, and the critic warm-up mechanism.

Ablation Variants. We compare the full MO-VLA framework against the following three variants:

- **w/o FiLM (Concat + RandInit):** We replace the FiLM modulation with a naive early-fusion strategy, in which the preference embedding $\mathbf{e_w}$ is directly concatenated with the VLM feature $\mathbf{z}_t$. Crucially, due to the structural change in the input dimensionality, this variant cannot inherit the pre-trained weights from Stage I. Consequently, the action head must be randomly initialized, forcing the

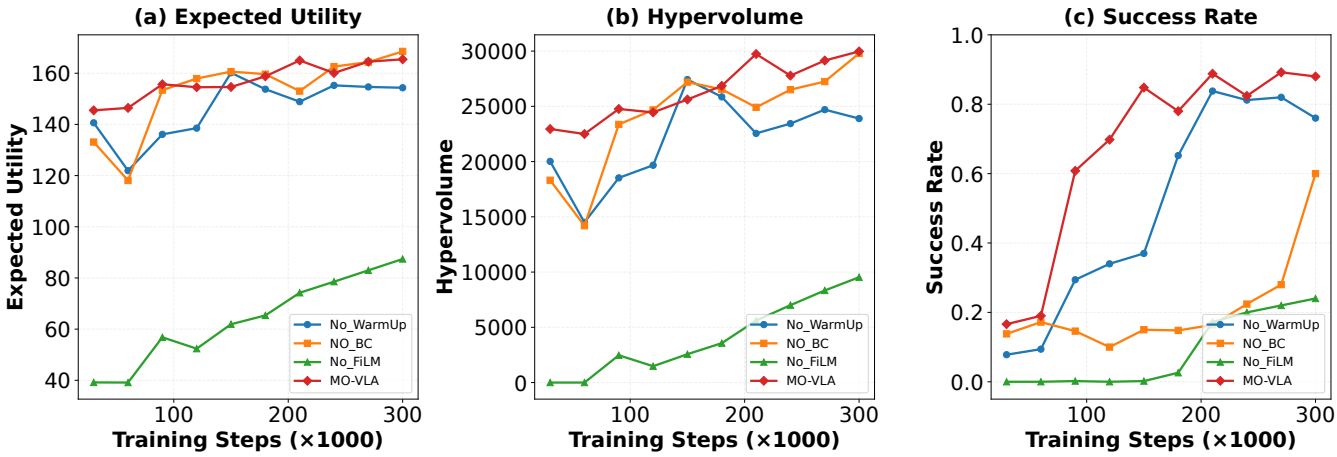

Figure 4: **Ablation study on `drawer-open`.** We plot the training curves for Hypervolume, Expected Utility, and Success Rate over 30k steps for different ablations of MO-VLA. **(1) w/o Warm-up(blue):** exhibits noticeable instability in the early phase of training. **(2) w/o FiLM (green):** suffers from slow convergence and poor overall performance due to the loss of pre-trained priors (randomly initialized action head). **(3) w/o BC Reg (orange):** shows clear instability and degraded success rates, indicating catastrophic forgetting. **(4) MO-VLA (red):** The full method maintains stable learning dynamics and achieves the best performance across all metrics.

policy to learn both manipulation skills and preference alignment from scratch.

- **w/o BC Regularization:** We set the regularization coefficient $\alpha = 0$ (Eq. 20), thereby removing the constraint that anchors the policy to the pre-trained prior. This tests the system's resistance to catastrophic forgetting.

- **w/o Critic Warm-up:** We bypass the critic warm-up phase and begin updating the actor network immediately at the start of Stage II ($N_{\text{warmup}} = 0$). This evaluates the sensitivity of the algorithm to initial value-estimation errors.

To focus on convergence dynamics and training stability, all variants are trained on the `drawer-open` task for a horizon of 30,000 steps. We track the learning curves for Hypervolume (HV), Expected Utility (EU), and Success Rate (SR).

Results and Analysis. As shown in Figure 4, all three ablated variants exhibit clear degradation, confirming the importance of each component. The *w/o FiLM* variant, which cannot inherit the pre-trained action head and must learn from random initialization, fails to acquire strong policies within 30,000 steps. The *w/o BC Reg* variant largely maintains Hypervolume and Expected Utility but exhibits highly fluctuating and poorly converged success rates, indicating catastrophic forgetting once the behavior-cloning prior is removed. The *w/o Warm-up* variant diverges early in training, suggesting that updating the actor before the critic has stabilized introduces substantial gradient noise. In contrast, the full MO-VLA configuration converges smoothly and achieves the best performance across all three metrics.

5 Conclusion and Future Work

In this work, we present MO-VLA, a unified framework that integrates general Vision-Language-Action (VLA) models with Multi-Objective Markov Decision Processes (MOMDPs) via a FiLM-based two-stage training paradigm. By preserving pretrained semantic representations and incorporating dynamic user preferences, our method achieves precise Pareto-optimal trade-offs between conflicting objectives (e.g., efficiency and safety) under natural-language instructions, significantly outperforming the unimodal behavioral patterns produced by conventional scalar-reward approaches. In future work, we plan to enhance both the dimensionality and expressiveness of objective preferences by incorporating additional task attributes—such as comfort and human–robot collaboration—into a unified multi-objective modeling framework, thereby enabling finer-grained and more interpretable preference control. In parallel, we aim to conduct systematic experiments and closed-loop optimization on real robotic platforms and within more uncertain, real-world scenarios to further validate and strengthen the robustness and generalization capabilities of MO-VLA in complex, open-world environments.

References

Abels, A.; Roijers, D. M.; Lenaerts, T.; Nowé, A.; and Steckelmacher, D. 2019. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, 11–20. PMLR.

Alegre, L. N.; Bazzan, A. L. C.; Roijers, D. M.; Nowé, A.; and da Silva, B. C. 2023. Sample-Efficient Multi-Objective Learning via Generalized Policy Improvement Prioritization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2003–2012. ACM.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P. F.; Schulman, J.; and Mané, D. 2016. Concrete Problems in AI Safety. *CoRR*, abs/1606.06565.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Shi, L. X.; Tanner, J.; Vuong, Q.; Walling, A.; Wang, H.; and Zhilinsky, U. 2024. π_0: A Vision-Language-Action Flow Model for General Robot Control. *CoRR*, abs/2410.24164.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; Ibarz, J.; Ichter, B.; Irpan, A.; Jackson, T.; Jesmonth, S.; Joshi, N. J.; Julian, R.; Kalashnikov, D.; Kuang, Y.; Leal, I.; Lee, K.; Levine, S.; Lu, Y.; Malla, U.; Manjunath, D.; Mordatch, I.; Nachum, O.; Parada, C.; Peralta, J.; Perez, E.; Pertsch, K.; Quiambao, J.; Rao, K.; Ryoo, M. S.; Salazar, G.; Sanketi, P. R.; Sayed, K.; Singh, J.; Sontakke, S.; Stone, A.; Tan, C.; Tran, H. T.; Vanhoucke, V.; Vega, S.; Vuong, Q.; Xia, F.; Xiao, T.; Xu, P.; Xu, S.; Yu, T.; and Zitkovich, B. 2023. RT-1: Robotics Transformer for Real-World Control at Scale. In *Robotics: Science and Systems XIX, Daegu*.

Deb, K. 2011. Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction. In *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, 3–34. Springer.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; Chebotar, Y.; Sermanet, P.; Duckworth, D.; Levine, S.; Vanhoucke, V.; Hausman, K.; Toussaint, M.; Greff, K.; Zeng, A.; Mordatch, I.; and Florence, P. 2023. PaLM-E: An Embodied Multimodal Language Model. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, 8469–8488. PMLR.

Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems*, 20132–20145.

Fujimoto, S.; van Hoof, H.; and Meger, D. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, 1582–1591. PMLR.

Ghosh, D.; Walke, H. R.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; Luo, J.; Tan, Y. L.; Chen, L. Y.; Vuong, Q.; Xiao, T.; Sanketi, P. R.; Sadigh, D.; Finn, C.; and Levine, S. 2024. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems XX*.

Guo, Y.; Zhang, J.; Chen, X.; Ji, X.; Wang, Y.; Hu, Y.; and Chen, J. 2025. Improving Vision-Language-Action Model with Online Reinforcement Learning. In *ICRA*, 15665–15672. IEEE.

Hayes, C. F.; Radulescu, R.; Bargiacchi, E.; Källström, J.; Macfarlane, M.; Reymond, M.; Verstraeten, T.; Zintgraf, L. M.; Dazeley, R.; Heintz, F.; Howley, E.; Irissappane, A. A.; Mannion, P.; Nowé, A.; de Oliveira Ramos, G.; Restelli, M.; Vamplew, P.; and Roijers, D. M. 2022. A practical guide to multi-objective reinforcement learning and planning. *Auton. Agents Multi Agent Syst.*, 36(1): 26.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*. OpenReview.net.

Hung, C.-Y.; Majumder, N.; Deng, H.; Liu, R.; Ang, Y.; Zadeh, A.; Li, C.; Herremans, D.; Wang, Z.; and Poria, S. 2025. NORA-1.5: A Vision-Language-Action Model Trained using World Model- and Action-Based Preference Rewards. arXiv:2511.14659.

Hwang, M.; Weihs, L.; Park, C.; Lee, K.; Kembhavi, A.; and Ehsani, K. 2024. Promptable Behaviors: Personalizing Multi-Objective Rewards from Human Preferences. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 16216–16226. IEEE.

Ichter, B.; Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; Julian, R.; Kalashnikov, D.; Levine, S.; Lu, Y.; Parada, C.; Rao, K.; Sermanet, P.; Toshev, A.; Vanhoucke, V.; Xia, F.; Xiao, T.; Xu, P.; Yan, M.; Brown, N.; Ahn, M.; Cortes, O.; Sievers, N.; Tan, C.; Xu, S.; Reyes, D.; Rettinghouse, J.; Quiambao, J.; Pastor, P.; Luu, L.; Lee, K.; Kuang, Y.; Jesmonth, S.; Joshi, N. J.; Jeffrey, K.; Ruano, R. J.; Hsu, J.; Gopalakrishnan, K.; David, B.; Zeng, A.; and Fu, C. K. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *CoRL*, volume 205 of *Proceedings of Machine Learning Research*, 287–318. PMLR.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E. P.; Sanketi, P. R.; Vuong, Q.; Kollar, T.; Burchfiel, B.; Tedrake, R.; Sadigh, D.; Levine, S.; Liang, P.; and Finn, C. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, 2679–2713. PMLR.

Lee, K.; Smith, L. M.; and Abbeel, P. 2021. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, 6152–6163. PMLR.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. C. H. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, 19730–19742. PMLR.

Lu, H.; Herman, D.; and Yu, Y. 2023. Multi-Objective Reinforcement Learning: Convexity, Stationarity and Pareto Optimality. In *ICLR*. OpenReview.net.

Ma, Y. J.; Liang, W.; Wang, G.; Huang, D.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2024. Eureka: Human-Level Reward Design via Coding Large Language Models. In *ICLR*. OpenReview.net.

Moffaert, K. V.; Drugan, M. M.; and Nowé, A. 2013. Scalarized multi-objective reinforcement learning: Novel design techniques. In *Proceedings of the 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL, IEEE Symposium Series on Computational Intelligence (SSCI)*, 191–199. IEEE.

Moffaert, K. V.; and Nowé, A. 2014. Multi-objective reinforcement learning using sets of pareto dominating policies. *J. Mach. Learn. Res.*, 15(1): 3483–3512.

Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; and Courville, A. C. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, 3942–3951. AAAI Press.

Pirotta, M.; Parisi, S.; and Restelli, M. 2015. Multi-Objective Reinforcement Learning with Continuous Pareto Frontier Approximation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2928–2934. AAAI Press.

Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; and Levine, S. 2018. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Robotics: Science and Systems XIV*.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2014. A Survey of Multi-Objective Sequential Decision-Making. *CoRR*, abs/1402.0590.

Yang, R.; Sun, X.; and Narasimhan, K. 2019. A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, 14610–14621.

Yarats, D.; Fergus, R.; Lazaric, A.; and Pinto, L. 2022. Mastering Visual Continuous Control: Improved Data-Augmented Reinforcement Learning. In *ICLR*. OpenReview.net.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2019. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *CoRL*, volume 100 of *Proceedings of Machine Learning Research*, 1094–1100. PMLR.

Yu, W.; Gileadi, N.; Fu, C.; Kirmani, S.; Lee, K.; Arenas, M. G.; Chiang, H. L.; Erez, T.; Hasenclever, L.; Humplik, J.; Ichter, B.; Xiao, T.; Xu, P.; Zeng, A.; Zhang, T.; Heess, N.; Sadigh, D.; Tan, J.; Tassa, Y.; and Xia, F. 2023. Language to Rewards for Robotic Skill Synthesis. In *CoRL*, volume 229 of *Proceedings of Machine Learning Research*, 374–404. PMLR.

Zhang, C.; Hao, P.; Cao, X.; Hao, X.; Cui, S.; and Wang, S. 2025. VTLA: Vision-Tactile-Language-Action Model with Preference Learning for Insertion Manipulation. *CoRR*, abs/2505.09577.

Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; Vuong, Q.; Vanhoucke, V.; Tran, H. T.; Soricut, R.; Singh, A.; Singh, J.; Sermanet, P.; Sanketi, P. R.; Salazar, G.; Ryoo, M. S.; Reymann, K.; Rao, K.; Pertsch, K.; Mordatch, I.; Michalewski, H.; Lu, Y.; Levine, S.; Lee, L.; Lee, T. E.; Leal, I.; Kuang, Y.; Kalashnikov, D.; Julian, R.; Joshi, N. J.; Irpan, A.; Ichter, B.; Hsu, J.; Herzog, A.; Hausman, K.; Gopalakrishnan, K.; Fu, C.; Florence, P.; Finn, C.; Dubey, K. A.; Driess, D.; Ding, T.; Choromanski, K. M.; Chen, X.; Chebotar, Y.; Carbajal, J.; Brown, N.; Brohan, A.; Arenas, M. G.; and Han, K. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *CoRL*, volume 229 of *Proceedings of Machine Learning Research*, 2165–2183. PMLR.

Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C. M.; and da Fonseca, V. G. 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.*, 7(2): 117–132.