FT-NCFM: An Influence-Aware Data Distillation Framework for Efficient VLA Models

Kewei Chen[1, 2], **Yayu Long**[1, 2], **Shuai Li**[3], **Mingsheng Shang**[1, 2*]

[1]Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences
[2]Chongqing School, University of Chinese Academy of Sciences
[3]Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland
{chenkewei24, longyayu24}@mails.ucas.ac.cn, shuai.li@oulu.fi, msshang@cigit.ac.cn

Abstract

The powerful generalization of Vision-Language-Action (VLA) models is bottlenecked by their heavy reliance on massive, redundant, and unevenly valued datasets, hindering their widespread application. Existing model-centric optimization paths, such as model compression (which often leads to performance degradation) or policy distillation (whose products are model-dependent and lack generality), fail to fundamentally address this data-level challenge. To this end, this paper introduces FT-NCFM, a fundamentally different, data-centric generative data distillation framework. Our framework employs a self-contained Fact-Tracing (FT) engine that combines causal attribution with programmatic contrastive verification to assess the intrinsic value of samples. Guided by these assessments, an adversarial NCFM process synthesizes a model-agnostic, information-dense, and reusable data asset. Experimental results on several mainstream VLA benchmarks show that models trained on just 5% of our distilled coreset achieve a success rate of 85-90% compared with training on the full dataset, while reducing training time by over 80%. Our work demonstrates that intelligent data distillation is a highly promising new path for building efficient, high-performance VLA models.

Introduction

With the rise of deep learning, particularly the Transformer architecture (Vaswani et al. 2017), the field of Embodied AI has experienced rapid development. Vision-Language-Action (VLA) models, as a prominent example, achieve end-to-end learning from "perception" to "action" by jointly processing visual perception, natural language instructions, and robot action outputs. Large VLA models like Google's RT series (Brohan et al. 2023; Zitkovich et al. 2023) and OpenVLA (Kim et al. 2024), after absorbing massive internet and robot operation data, have demonstrated astonishing zero-shot or few-shot generalization capabilities (Li et al. 2025; Long et al. 2025; Phan et al. 2024; Chen et al. 2023; Han et al. 2025; Babazaki, Shibata, and Takahashi 2024; Nag et al. 2022), enabling them to complete complex, unseen tasks.

However, this exceptional performance comes at a great resource cost. The success of these models heavily relies on large-scale, diverse training datasets, such as the Open X-Embodiment dataset (Vuong et al. 2023), which contains over one million real robot trajectories. Furthermore, model architectures with billions or even tens of billions of parameters require large GPU clusters for weeks or even months of training, imposing a heavy financial burden on most research institutions and companies. The large model size and high inference latency also make deploying these advanced models on resource-constrained physical robot platforms a severe challenge.

To alleviate this problem, mainstream research has focused on model-centric optimization paths. These paths are mainly divided into two categories. The first is model architecture lightweighting. For example, SmolVLA (Shukor et al. 2025) and TinyVLA (Wen et al. 2025) improve efficiency by simplifying the network structure, but their performance on complex tasks drops significantly. The second category is policy distillation, represented by DROC (Zha et al. 2024) and RLDG (Xu et al. 2024), which aims to transfer knowledge from a large "teacher" model to a small "student" model. Although policy distillation (Wang et al. 2025b; Li et al. 2024; Zhang et al. 2025a; Wang et al. 2025c; Prasad et al. 2024) performs well in maintaining performance, its knowledge carrier is always the model parameters. This prevents the knowledge itself from being directly analyzed or reused as an independent, transferable, and composable asset. Moreover, the success of this process is highly dependent on a pre-trained, expensive teacher model. In summary, all these model-centric methods share a common limitation: they fail to address the efficiency and quality bottlenecks at the data level. These data-level issues—namely the widespread redundancy, noise, and uneven value in datasets (Fang et al. 2024; Khazatsky et al. 2024; Cheng et al. 2024; Liu et al. 2024b; Ji et al. 2025; Hang et al. 2024; Nasiriany et al. 2024; Zhang et al. 2024)—are precisely what limits the further improvement of current VLA models.

For this reason, this paper shifts its perspective to a neglected yet more promising direction: data-level efficiency optimization. Our core insight is that training directly on the full, undifferentiated dataset is not only inefficient but may also hinder the model from focusing on critical task features. We therefore propose a pioneering generative data distillation framework, FT-NCFM. This framework intelligently synthesizes a small but causally-enriched knowledge

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

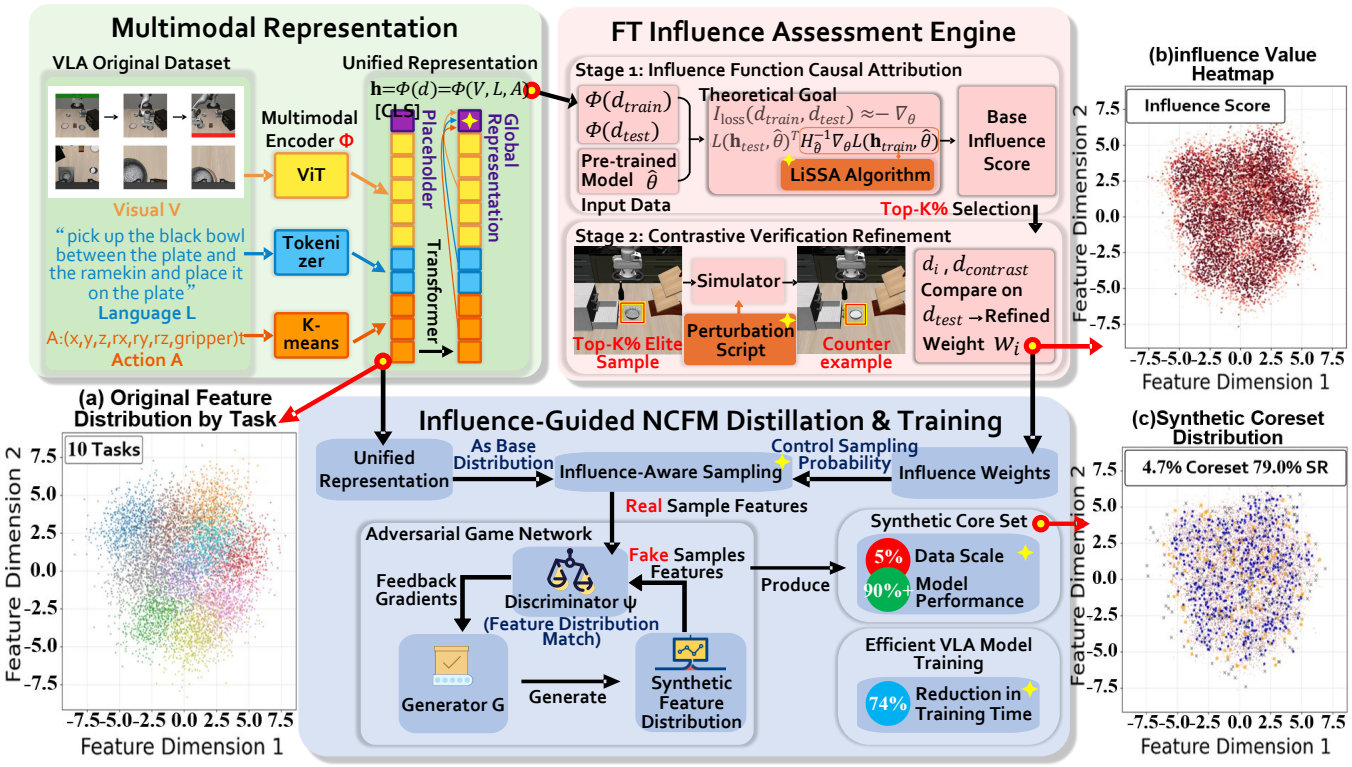

Figure 1: Overview of the FT-NCFM framework. (Top Left) The Multimodal Representation Module fuses raw VLA streams into global features h. (Top Right) The FT Assessment Engine calculates base influence scores and refines weights W via contrastive verification with generated counterexamples. (Bottom) Influence-Guided NCFM Distillation uses weights W to guide the adversarial training of generator G and discriminator Ψ to produce a synthetic coreset. t-SNE plots show: (a) original task distribution; (b) influence heatmap; and (c) the resulting synthetic coreset covering high-value feature regions.

coreset from massive raw data. The goal is to fundamentally improve training efficiency without sacrificing, and potentially even enhancing, model performance.

Based on this philosophy, our main contributions are as follows:

1. **A new generative data distillation paradigm for VLAs**: Unlike existing work focusing on model compression and policy distillation, our FT-NCFM framework provides a novel path to building efficient VLA models by optimizing at the data level.
2. **A self-contained intrinsic value assessment engine, FT**: This engine evaluates sample value through a two-stage process. It first uses causal attribution for initial screening and then refines the elite samples through a novel programmatic contrastive verification module. We designed a set of reusable perturbation templates that can automatically generate high-quality "minimal counterexamples" for elite samples through semantic parsing and simulator instantiation. This allows FT to robustly quantify a sample's causal contribution and generalization potential based on the data itself.
3. **Demonstrated superior performance and efficiency on multiple mainstream VLA benchmarks**: We conducted systematic evaluations on public benchmarks like CALVIN and Meta-World. The results show that models trained with our FT-NCFM framework, using only 5% of the synthetic data, achieve over 85-90% of the task success rate of models trained on the full dataset, while significantly reducing training time and computational resource consumption (training time reduced by over 80%). This fully demonstrates the great potential of our method in building efficient, high-performance, and easily deployable VLA systems.

Preliminaries

Our method builds on two key concepts. **Influence Functions (IF)** are a classic technique to quantify the effect of a training sample on a model's loss (Yan et al. 2024; Klochkov and Liu 2024), which we adapt to approximate sample value. **Neural Characteristic Function Matching (NCFM)** is a generative method that synthesizes data by matching the distributions of real and synthetic data in a feature space, typically via an adversarial game (Wang et al. 2025a). Our framework, FT-NCFM, creates an influence-aware NCFM process.

Related Work

VLA Models and Model-Centric Optimization. State-of-the-art VLA models like the RT series (Brohan et al. 2023;

Zitkovich et al. 2023) and OpenVLA (Kim et al. 2024) show great generalization but demand enormous resources (e.g., 15k-20k GPU-hours) (Brohan et al. 2023). Mainstream *model-centric* solutions are insufficient: model compression (e.g., SmolVLA (Shukor et al. 2025)) suffers performance drops, while policy distillation (e.g., RLDG (Xu et al. 2024), DROC (Zha et al. 2024)) transfers knowledge but makes it parameter-bound, non-reusable, and dependent on an expensive teacher model.

Data-Centric Optimization. A different path, *data-centric* optimization, improves efficiency at the source. The most relevant work is Coreset Selection (Dass et al. 2025; Dharmasiri et al. 2025; Griffin, Marks, and Corso 2024), which selects representative subsets using techniques like influence functions. However, this "selection" paradigm is fundamentally limited by the information density of existing samples. This exposes a critical research gap: can we transcend "selecting" and directly "synthesize" datasets that capture the essence of visuomotor knowledge? Our FT-NCFM framework is designed to address this gap through generative data distillation.

Methodology

The proposed FT-NCFM framework (Figure 1) employs a three-stage pipeline to synthesize a data coreset. First, a multimodal representation module transforms raw VLA data into unified features. Then, an FT influence assessment engine calculates an influence weight for each sample. Finally, these weights guide an influence-aware NCFM process to synthesize the final coreset.

VLA Multimodal Representation Learning

To evaluate each sample's causal effect, we adopt the core idea of influence functions (Yan et al. 2024) to approximate the change in model loss upon its removal from the training data. The influence of a training sample d_{train} on the loss of a test sample d_{test} is approximated by the following core formula:

$$\mathbf{h} = \Phi(d) = \Phi(V, L, A) \in \mathbb{R}^{d_{model}} \tag{1}$$

where $\mathbf{h}$ is a d_{model}-dimensional feature vector. This vector serves as the object for all subsequent influence analysis and distribution matching operations.

FT: The Influence Assessment Engine

To accurately quantify the value of each raw sample $d_i = (V_i, L_i, A_i)$, this paper designs a two-stage FT engine to compute its influence weight w_i.

Stage 1: Causal Attribution Pre-screening based on Influence Functions We first adopt the core idea of influence functions (Yan et al. 2024) to evaluate the causal effect of each sample. This method aims to efficiently approximate the potential impact of removing a single training sample on the model's loss. Specifically, the influence of a training sample d_{train} (with representation $\mathbf{h}_{train} = \Phi(d_{train})$) on the loss of a test sample d_{test} (with representation $\mathbf{h}_{test}$) can be approximated by the following core formula:

$$I_{\text{loss}}(d_{train}, d_{test}) \approx -\nabla_\theta L(\mathbf{h}_{test}, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_\theta L(\mathbf{h}_{train}, \hat{\theta}) \tag{2}$$

where $L(h, \theta)$ is the standard loss function of the downstream policy model (MSE loss in our implementation), $\nabla_\theta L$ is the loss gradient, and $H_{\hat{\theta}}^{-1}$ is the inverse of the Hessian matrix. Here, $\hat{\theta}$ are the parameters of a "guide model". Its purpose is not to provide perfect expert decisions, but merely to offer a stable and reasonable gradient field for the influence function calculation. Therefore, it does not need to be a fully converged optimal model. In our implementation, this guide model uses the exact same network architecture as the downstream policy model and is obtained by light, insufficient training on the original dataset (only 10%-20% of the total time required for the standard procedure). Its cost has been accounted for as part of our framework's total overhead.

In our implementation, we use the LiSSA algorithm (Klochkov and Liu 2024) to efficiently approximate the Hessian-inverse-vector product (see Appendix for hyperparameter discussion). The influence score calculated by this method constitutes the base influence score of this engine, $Score_{base}(d_i)$.

Stage 2: Contrastive Verification Refinement To further ensure that samples with a high $Score_{base}$ are indeed positively valued and contribute to generalization, we perform contrastive verification on the top-K% elite samples.

For each elite sample d_i, we generate a "minimal counterexample" d_{contrast} using a **Cross-Modal Mismatch** strategy common in robot robustness evaluation. Specifically, while keeping the language (L) and action (A) sequences fixed, we programmatically modify the initial visual scene (V) in a simulator to create a new scene (V') with semantic or physical contradictions, as illustrated in Figure 2.

Programmatic Counterexample Generation. This strategy's automation and generalizability are rooted in the structured VLA task space. Since task categories are limited and classifiable (e.g., 40 in the LIBERO dataset), we can pre-design a small set of **Reusable Perturbation Templates**. As detailed in Figure 2, for any given elite sample d_i, our script generates a counterexample d_{contrast} through a fully automated "Template Matching and Instantiation" process. This process, which ensures efficiency and consistency, involves three stages: (1) Instruction Semantic Parsing, (2) Perturbation Template Selection, and (3) Scene Instantiation via simulator APIs to create a semantically contradictory scene. The effects of our core templates (e.g., object substitution, changing spatial relations) are detailed in Figure 3.

After obtaining the counterexample, our goal is to quickly evaluate whether d_i is more helpful than $d_{contrast}$ for a relevant standard test case d_{test}, without retraining the model. To this end, we use gradient dot products to approximate influence, calculating the influence scores of the elite sample

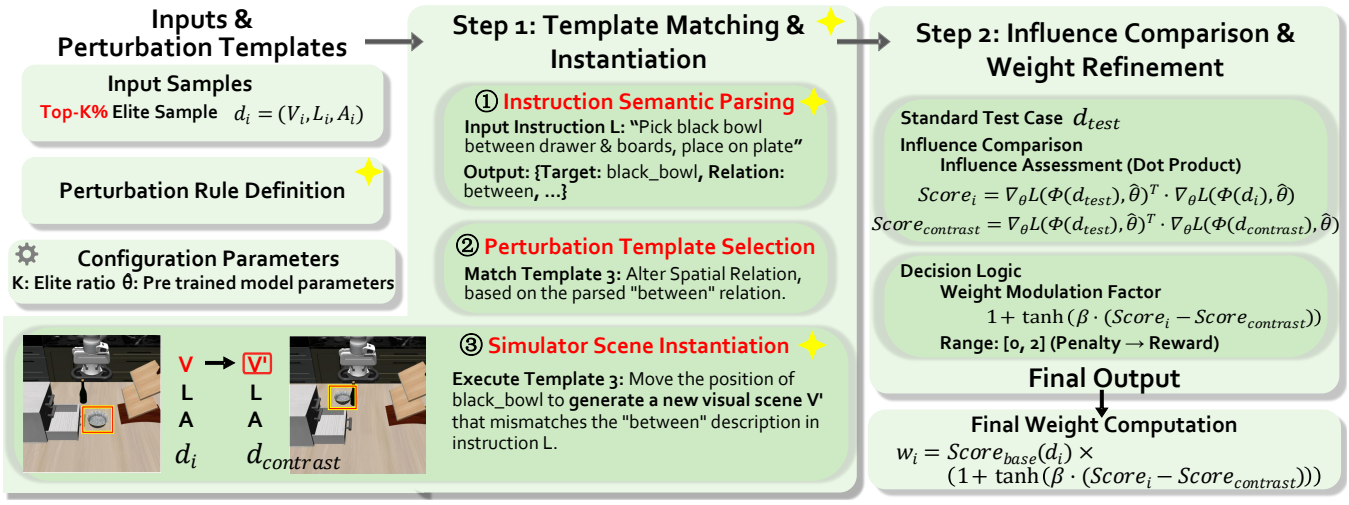

Figure 2: Detailed workflow of the "Contrastive Verification Refinement" stage in the FT engine. This figure illustrates how we refine the value of the top-K% elite samples through a two-stage process. Step 1 explains how we programmatically generate a corresponding "minimal counterexample" $d_{contrast}$ for each elite sample d_i in the simulator through an automated "Template Matching and Instantiation" process. Step 2 shows how we quantify the value difference between d_i and its counterexample $d_{contrast}$ into the final, refined influence weight w_i through influence comparison and a weight modulation function.

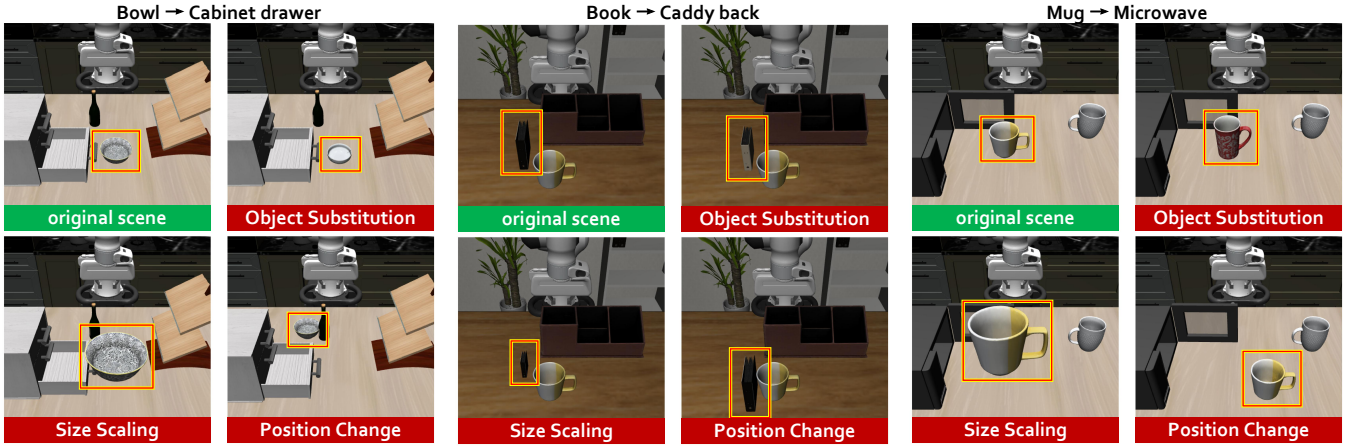

Figure 3: Application of our perturbation templates on three independent VLA tasks. The figure visualizes the generation of "minimal counterexamples" through Object Substitution, Size Scaling, and Position Change.

and its counterexample on the test case d_{test} respectively:

$$Score_i = \nabla_\theta L(\Phi(d_{test}), \hat{\theta})^T \cdot \nabla_\theta L(\Phi(d_i), \hat{\theta}) \quad (3)$$

$$Score_{contrast} = \nabla_\theta L(\Phi(d_{test}), \hat{\theta})^T \cdot \nabla_\theta L(\Phi(d_{contrast}), \hat{\theta}) \quad (4)$$

An ideal elite sample d_i should exhibit high positive influence, while its counterexample $d_{contrast}$ should show negative or significantly lower influence. To convert this contrastive verification result from a discrete binary decision (pass/fail) into a continuous weight modulation mechanism, we designed the following formula for the final influence weight w_i:

$$w_i = Score_{base}(d_i) \times (1 + \tanh(\beta \cdot (Score_i - Score_{contrast}))) \quad (5)$$

In this equation, the hyperparameter β controls the sensitivity of the modulation function. The core difference term $(Score_i - Score_{contrast})$ quantifies the relative effectiveness of the original sample d_i compared to its counterexample. We use the hyperbolic tangent function (tanh) to smoothly map this difference to the $[-1, 1]$ interval, thereby constructing a continuous Weight Modulation Factor '$(1 + \tanh(...))$' ranging from $[0, 2]$. This factor dynamically and smoothly adjusts the base influence score $Score_{base}(d_i)$ either positively or negatively based on the strength of the verification result. This design makes the weight refinement

process more precise and robust.

Influence-Guided NCFM

After obtaining the influence weights $W = \{w_i\}_{i=1}^N$ for all original samples, we use them to guide the NCFM distillation process. The core idea of NCFM is to match the feature functions of real and synthetic data through a minimax game. The original NCFM optimization objective can be simplified as:

$$\min_{D_{synth}} \max_{\psi} \left\| \mathbb{E}_{d \sim D_{real}}[\psi(\Phi(d))] - \mathbb{E}_{d' \sim D_{synth}}[\psi(\Phi(d'))] \right\|^2 \tag{6}$$

where ψ is an auxiliary network used to maximize the distribution difference. We point out that the expectation $\mathbb{E}_{d \sim D_{real}}$ in the above equation assumes uniform sampling, which ignores the differences in sample value.

Our core improvement is to transform this expectation into a weighted expectation based on influence weights. The final optimization objective of our FT-NCFM becomes:

$$\min_{D_{synth}} \max_{\psi} \left\| \sum_{i=1}^{N} \frac{w_i}{\sum_j w_j} \psi(\Phi(d_i)) - \mathbb{E}_{d' \sim D_{synth}}[\psi(\Phi(d'))] \right\|^2 \tag{7}$$

Through this weighted objective, the discriminator ψ is forced to focus on the more valuable real samples with higher weights w_i. Consequently, the generator (i.e., the optimization over D_{synth}) must also prioritize learning to imitate the distribution features of these high-value samples, thereby synthesizing a coreset D_{synth} that is rich in critical causal knowledge and has extremely high information density. This synthetic coreset is identical in data format to D_{real} and can be directly used for training any downstream VLA model. (Detailed pseudocode for our framework is provided in Appendix, Algorithm 1.)

Experiments

We conducted extensive experiments on several mainstream robotics manipulation benchmarks to systematically evaluate our proposed FT-NCFM framework. Our experiments aim to answer the following core questions:

1. **Effectiveness and Efficiency**: Compared to SOTA models (e.g., OpenVLA) trained on the full dataset, can FT-NCFM achieve competitive performance using only a very small fraction (e.g., 1%, 5%, 10%) of synthetic data, while significantly reducing training costs?
2. **Paradigm Comparison**: How does our data-centric FT-NCFM framework perform in terms of performance and resource consumption compared to mainstream model-centric optimization paths, especially SOTA policy distillation methods (e.g., RLDG, DROC)?
3. **Component Contribution**: How crucial are the key designs in the FT-NCFM framework—especially the FT influence assessment engine—to its final success?

Experimental Setup

Benchmarks and Environments. Our main experiments were conducted on three widely used VLA benchmarks:

- **CALVIN** (Mees et al. 2022): A large-scale robotics manipulation benchmark that requires the model to follow language instructions in long-horizon tasks. We follow the standard D→A,B,C,D evaluation protocol, focusing on the model's generalization ability.
- **Meta-World** (Yu et al. 2020): A benchmark containing 50 different tabletop manipulation tasks, used to evaluate the model's skill acquisition in a multi-task learning environment.
- **LIBERO** (Liu et al. 2023): A benchmark suite designed to promote lifelong learning. We use its Spatial, Object, Goal, and Long subsets to evaluate the model's performance across different generalization dimensions.

Base VLA Model and Evaluation Metrics. To ensure fairness, all experiments (including our method and baselines) use a unified, standard VLA model architecture based on a pre-trained ViT-B/16 visual backbone and a 6-layer Transformer decoder, consistent with the settings in works like OpenVLA (Kim et al. 2024) and RT-2 (Zitkovich et al. 2023). We focus on two core metrics: **Success Rate (SR %)** or **Average Task Completion Length (Avg. Len)** to measure performance, and **Total Training Time (GPU-hours)** to measure efficiency. All time-related metrics were measured on a single NVIDIA A100 80GB GPU. All training times refer to the total time required for the model to converge from random initialization. We used the same optimizer (e.g., AdamW), learning rate, and batch size settings as in the original papers of the respective baselines to ensure a fair comparison.

Comparison Baselines. We compare FT-NCFM with the following categories of methods:

- **SOTA VLA Models (Full Data)**: Including advanced models trained on the full dataset such as OpenVLA (Kim et al. 2024), RT-2 (Zitkovich et al. 2023), SpatialVLA (Qu et al. 2025), RoboUniview (Liu et al. 2024a), and GR-1 (Wu et al. 2024), which serve as the gold standard for performance.
- **Model-Centric Baselines**: Including SOTA policy distillation methods like RLDG (CoRL 2024) (Xu et al. 2024), DROC (ICLR 2023) (Zha et al. 2024), and Mole-VLA (ICLR 2024) (Zhang et al. 2025b). Their total training time includes the time for both training the teacher model and performing policy distillation.
- **Coreset Selection Baselines**: Including Random Sampling and Influence Function Coreset selection.

Main Results and Analysis

Performance on CALVIN and Meta-World We systematically evaluated the performance of FT-NCFM at different data compression rates on the CALVIN and Meta-World benchmarks.

As shown in Tables 1 and 2, FT-NCFM demonstrates a clear and efficient performance scaling curve. On the CALVIN benchmark, using only 1% of the synthetic data,

Method	Data Ratio	i^{th} Task Success Rate					
		1	2	3	4	5	Avg. Len↑
RT-1 (Brohan et al. 2023)	100 %	0.533	0.222	0.094	0.038	0.013	0.90
GR-1 (Wu et al. 2024)	100 %	0.854	0.712	0.596	0.497	0.401	3.06
Vidman (Wen et al. 2024)	100 %	0.915	0.764	0.682	0.592	0.467	3.42
RoboUniview (Liu et al. 2024a)	100 %	0.942	0.842	0.734	0.622	0.507	3.65
FT-NCFM (Ours)	**1 %**	0.755	0.531	0.402	0.298	0.204	2.19
FT-NCFM (Ours)	**5 %**	0.895	0.733	0.612	0.501	0.373	3.11
FT-NCFM (Ours)	**10 %**	**0.925**	**0.791**	**0.688**	**0.590**	**0.476**	**3.47**

Table 1: Zero-shot long-horizon evaluation on the Calvin ABC $\rightarrow$ D benchmark. The results show the performance scalability of FT-NCFM at different data ratios and compare it with various baselines.

Method	Data Ratio	Avg. SR(%) ↑
RT-1 (Brohan et al. 2023)	100%	34.6
Susie (Black et al. 2024)	100%	41.0
GR-1 (Wu et al. 2024)	100%	57.4
FT-NCFM (Ours)	**1%**	**34.4**
FT-NCFM (Ours)	**5%**	**50.5**
FT-NCFM (Ours)	**10%**	**54.5**

Table 2: Multi-task success rate on 50 Meta-World tasks.

our method achieves about 60% of the performance of the SOTA method RoboUniview (Avg. Len 3.65) and already surpasses many earlier baselines. When the data amount is increased to 10%, our method achieves an average task completion length of 3.47, reaching 95% of the SOTA performance and almost matching Vidman (3.42) trained on 100% data. A similar trend is observed on Meta-World, where using 10% of the data recovers about 95% of the performance of the SOTA method GR-1 (57.4%). These results strongly prove that our framework can maintain highly competitive performance with extremely high data efficiency, while significantly reducing data dependency.

Generalization Ability Evaluation on LIBERO To further test the capabilities of FT-NCFM in more complex generalization scenarios, we conducted evaluations on the LIBERO benchmark.

Method	**Data(%)**	**Spatial**	**Object**	**Goal**	**Long**	**Avg.**
OpenVLA (Kim et al. 2024)	100	84.7	88.4	79.2	53.7	76.5
SpatialVLA (Qu et al. 2025)	100	88.2	89.9	78.6	55.5	78.1
FT-NCFM (Ours)	**1**	52.1	55.3	45.8	34.4	46.9
	5	78.3	80.1	68.5	54.3	70.3
	10	82.8	84.5	72.9	56.6	74.2

Table 3: **Success Rate Evaluation on LIBERO.** The SOTA baseline is SpatialVLA. FT-NCFM shows strong performance scalability.

As shown in Table 3, FT-NCFM also performs exceptionally well on the LIBERO benchmark. With the best-performing baseline SpatialVLA (average success rate 78.1%) as a reference, our method achieves 95% of its performance (74.2%) using only 10% of the data. It is particularly noteworthy that on the LIBERO-LONG task set, which has the highest demands on temporal reasoning and multi-step planning, FT-NCFM with just 10% of the data (56.6%) surpasses all baseline methods that use 100% of the data. This strongly suggests that our generative data distillation framework can not only preserve but even enhance the causal and generalization knowledge crucial for learning complex, long-horizon tasks from the original data.

To more intuitively demonstrate the effectiveness of the coreset synthesized by our method in real-world tasks, we conducted a series of qualitative evaluations. Figure 4 shows typical results on the complex, long-horizon task of "stacking six bowls".

Paradigm	**Method**	**Data Ratio (%)**	**Total Training Time (GPU-h) ↓**	**Avg. Len ↑**
Model-Centric	DROC (Zha et al. 2024)	100	128 + 65 = 193	3.05
	Mole-VLA (Zhang et al. 2025b)	100	128 + 50 = 178	3.20
	RLDG (Xu et al. 2024)	100	128 + 70 = 198	3.15
Data-Centric (Coreset)	Random Sampling	5	6.5	1.88
	Influence Function Coreset	5	18	2.45
Data-Centric (Ours)	**FT-NCFM**	**1**	**20.0**	**2.19**
	FT-NCFM	**5**	**25.0**	**3.11**
	FT-NCFM	**10**	**31.5**	**3.47**

Table 4: Comparison of FT-NCFM with model-centric (policy distillation) and data-centric (coreset selection) methods on the CALVIN benchmark.

Paradigm Comparison with SOTA Methods The results in Table 4 strongly answer our second research question (Q2). Compared to SOTA policy distillation methods (e.g., RLDG, Mole-VLA), our FT-NCFM, using only 5% of the data and 25 hours of total time, achieves comparable performance (3.11 vs 3.15-3.20) with about one-seventh of their resource consumption. When using 10% of the data, our performance (3.47) far exceeds all policy distillation methods, while the total time (31.5h) is still less than one-sixth of theirs. This highlights the fundamental efficiency bottleneck brought by the reliance of policy distillation methods on expensive teacher models. Furthermore, compared to traditional "selection"-based coreset methods, our "synthesis"-based FT-NCFM demonstrates enormous superiority.

Cost-Benefit Analysis

A potential concern is the additional overhead introduced by the data preprocessing stage of FT-NCFM. In this section, we quantify this cost using the large-scale LIBERO benchmark as an example.

As shown in Table 5, the preprocessing stage of FT-NCFM (including FT engine assessment and NCFM synthesis) requires about 24 GPU-hours on LIBERO, which is a **One-Time Investment**. Even when this overhead is included, the total time for FT-NCFM with 10% data (39.0h)

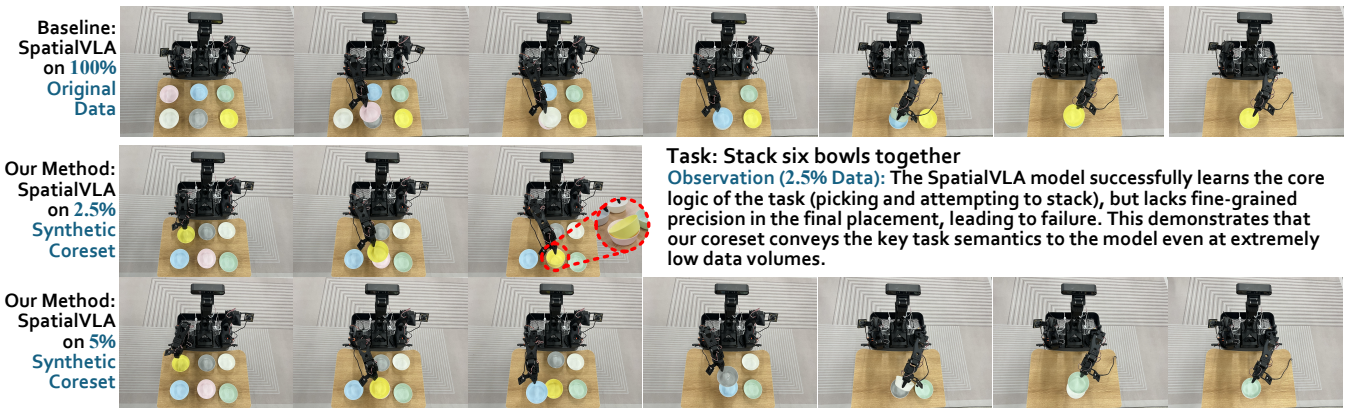

Figure 4: Qualitative results on the real-world "stack six bowls" task. (Row 1) Baseline Model (100% original data) successfully completes the task. (Row 2) Our Method (2.5% coreset) captures the core logic but exhibits minor placement precision errors. (Row 3) Our Method (5% coreset) successfully completes the task, achieving performance comparable to the baseline.

Stage	**Description**	**Time Cost (GPU-h) for X% Data**		
		1%	**5%**	**10%**
SpatialVLA (100% data)	Train SOTA model on full data	192		
FT-NCFM Framework				
FT Engine + NCFM Preprocessing	One-time investment for value assessment	24.0		
Policy Training on D_{synth}	Policy learning on synthetic data	1.5	7.0	15.0
FT-NCFM (Total)	**Total Time**	**25.5**	**31.0**	**39.0**

Table 5: Cost-benefit analysis of each stage of FT-NCFM on the LIBERO dataset.

is only 20.3% of the standard OpenVLA model training time (192h). This "invest first, benefit later" model has significant "amortization benefits" in development cycles that require multiple model iterations.

Ablation Study

To verify the necessity of each component in our FT engine, we conducted a series of ablation experiments.

Method Variant	**Avg. Len ↑**
FT-NCFM (Full Method)	**3.11**
Ablating FT Engine Components:	
w/o Contrastive Verification (use $Score_{base}$ only)	2.81
w/o FT Engine (weighted NCFM with random weights)	2.15

Table 6: Ablation study of key components of FT-NCFM on the CALVIN benchmark (Avg. Len, at 5% data).

The results of this experiment clearly answer our third research question (Q3). At the 5% data setting, when the contrastive verification refinement stage is removed, performance drops from 3.11 to 2.81, proving that the contrastive verification module can effectively filter out high-influence but potentially harmful samples. When the FT engine is completely removed, performance drops sharply further to 2.15, which strongly demonstrates that our FT influence assessment engine is the cornerstone of the entire framework's success. We conducted more detailed ablation studies (including a sensitivity analysis for β (Eq. 5)) and observed trends consistent with CALVIN; detailed results can be found in the appendix.

Conclusion

To address VLA training costs, we propose FT-NCFM, a data-centric generative distillation framework demonstrating that data optimization is more fundamental than model compression. Using only 5% synthetic data, models achieve 85-90% SOTA performance while reducing training time by over 80% and outperforming policy distillation and coreset selection methods. This proves that enhancing data-source information density offers a promising path for efficient, high-performance VLA models.

Limitation

Although FT-NCFM achieves encouraging results, we acknowledge current limitations that point to future research directions. First, our perturbation template library covers core dimensions like object substitution and size scaling, but does not encompass all failure scenarios such as physical property changes (mass, friction). However, the key advantage of this library lies in its extensibility, allowing convenient template additions to better align with real-world scenarios. Second, our automated counterexample generation effectiveness relies on simulator-originated datasets that can be programmatically modified, which limits applicability to real-world data that cannot be directly edited. Transferring these core ideas to non-editable real data remains an important open question. Future research could explore generative models (GANs, Diffusion Models) for semantic editing to create counterexamples.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62372427, in part by Chongqing Natural Science Foundation Innovation and Development Joint Fund (No. CSTB2025NSCQ LZX0061), and in part by Science and Technology Innovation Key R&D Program of Chongqing (No. CSTB2025TIAD-STX0023).

References

Babazaki, Y.; Shibata, T.; and Takahashi, T. 2024. Zero-Shot Spatio-Temporal Action Detection by Enhancing Context-Relation Capability of Vision-Language Models. In *International Conference on Pattern Recognition*, 229–244. Springer.

Black, K.; Nakamoto, M.; Atreya, P.; Walke, H. R.; Finn, C.; Kumar, A.; and Levine, S. 2024. Zero-Shot Robotic Manipulation with Pre-Trained Image-Editing Diffusion Models. In *The Twelfth International Conference on Learning Representations*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2023. RT-1: Robotics Transformer for Real-World Control at Scale. *Robotics: Science and Systems XIX*.

Chen, P.; Sun, X.; Zhi, H.; Zeng, R.; Li, T. H.; Liu, G.; Tan, M.; and Gan, C. 2023. A^2 Nav: Action-Aware Zero-Shot Robot Navigation by Exploiting Vision-and-Language Ability of Foundation Models. *arXiv preprint arXiv:2308.07997*.

Cheng, D.; Cladera, F.; Prabhu, A.; Liu, X.; Zhu, A.; Green, P. C.; Ehsani, R.; Chaudhari, P.; and Kumar, V. 2024. Treescope: An agricultural robotics dataset for lidar-based mapping of trees in forests and orchards. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 14860–14866. IEEE.

Dass, S.; Khaddaj, A.; Engstrom, L.; Madry, A.; Ilyas, A.; and Martín-Martín, R. 2025. DataMIL: Selecting Data for Robot Imitation Learning with Datamodels. *arXiv preprint arXiv:2505.09603*.

Dharmasiri, A.; Yang, W.; Kirichenko, P.; Liu, L. T.; and Russakovsky, O. 2025. The Impact of Coreset Selection on Spurious Correlations and Group Robustness. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Fang, H.-S.; Fang, H.; Tang, Z.; Liu, J.; Wang, C.; Wang, J.; Zhu, H.; and Lu, C. 2024. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 653–660. IEEE.

Griffin, B. A.; Marks, J.; and Corso, J. J. 2024. Zero-shot coreset selection: Efficient pruning for unlabeled data. *arXiv preprint arXiv:2411.15349*.

Han, C.; Wang, H.; Kuang, J.; Zhang, L.; and Gui, J. 2025. Training-Free Zero-Shot Temporal Action Detection with Vision-Language Models. *arXiv preprint arXiv:2501.13795*.

Hang, J.; Lin, X.; Zhu, T.; Li, X.; Wu, R.; Ma, X.; and Sun, Y. 2024. Dexfuncgrasp: A robotic dexterous functional grasp dataset constructed from a cost-effective real-simulation annotation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 10306–10313.

Ji, Y.; Tan, H.; Shi, J.; Hao, X.; Zhang, Y.; Zhang, H.; Wang, P.; Zhao, M.; Mu, Y.; An, P.; et al. 2025. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 1724–1734.

Khazatsky, A.; Pertsch, K.; Nair, S.; Balakrishna, A.; Dasari, S.; Karamcheti, S.; Nasiriany, S.; Srirama, M. K.; Chen, L. Y.; Ellis, K.; et al. 2024. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *RSS 2024 Workshop: Data Generation for Robotics*.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E. P.; Sanketi, P. R.; Vuong, Q.; et al. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. In *8th Annual Conference on Robot Learning*.

Klochkov, Y.; and Liu, Y. 2024. Revisiting inverse hessian vector products for calculating influence functions. *arXiv preprint arXiv:2409.17357*.

Li, L.; Donato, E.; Lomonaco, V.; and Falotico, E. 2024. Continual policy distillation of reinforcement learning-based controllers for soft robotic in-hand manipulation. In *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft)*, 1026–1033. IEEE.

Li, P.; Wu, Y.; Xi, Z.; Li, W.; Huang, Y.; Zhang, Z.; Chen, Y.; Wang, J.; Zhu, S.-C.; Liu, T.; et al. 2025. ControlVLA: Few-shot Object-centric Adaptation for Pre-trained Vision-Language-Action Models. *arXiv preprint arXiv:2506.16211*.

Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36: 44776–44791.

Liu, F.; Yan, F.; Zheng, L.; Feng, C.; Huang, Y.; and Ma, L. 2024a. Robouniview: Visual-language model with unified view representation for robotic manipulation. *arXiv preprint arXiv:2406.18977*.

Liu, Y.; Fu, Y.; Qin, M.; Xu, Y.; Xu, B.; Chen, F.; Goossens, B.; Sun, P. Z.; Yu, H.; Liu, C.; et al. 2024b. Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments. *IEEE Robotics and Automation Letters*, 9(3): 2798–2805.

Long, Y.; Chen, K.; Jin, L.; and Shang, M. 2025. DRAE: Dynamic Retrieval-Augmented Expert Networks for Lifelong Learning and Task Adaptation in Robotics. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 23098–23141.

Mees, O.; Hermann, L.; Rosete-Beas, E.; and Burgard, W. 2022. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334.

Nag, S.; Zhu, X.; Song, Y.-Z.; and Xiang, T. 2022. Zero-shot temporal action detection via vision-language prompting. In *European conference on computer vision*, 681–697. Springer.

Nasiriany, S.; Maddukuri, A.; Zhang, L.; Parikh, A.; Lo, A.; Joshi, A.; Mandlekar, A.; and Zhu, Y. 2024. RoboCasa: Large-Scale Simulation of Everyday Tasks for Generalist Robots. In *RSS 2024 Workshop: Data Generation for Robotics*.

Phan, T.; Vo, K.; Le, D.; Doretto, G.; Adjeroh, D.; and Le, N. 2024. Zeetad: Adapting pretrained vision-language model for zero-shot end-to-end temporal action detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 7046–7055.

Prasad, A.; Lin, K.; Wu, J.; Zhou, L.; and Bohg, J. 2024. Consistency Policy: Accelerated Visuomotor Policies via Consistency Distillation. In *Robotics: Science and Systems*.

Qu, D.; Song, H.; Chen, Q.; Yao, Y.; Ye, X.; Ding, Y.; Wang, Z.; Gu, J.; Zhao, B.; Wang, D.; et al. 2025. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*.

Shukor, M.; Aubakirova, D.; Capuano, F.; Kooijmans, P.; Palma, S.; Zouitine, A.; Aractingi, M.; Pascal, C.; Russi, M.; Marafioti, A.; et al. 2025. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vuong, Q.; Levine, S.; Walke, H. R.; Pertsch, K.; Singh, A.; Doshi, R.; Xu, C.; Luo, J.; Tan, L.; Shah, D.; et al. 2023. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*.

Wang, S.; Yang, Y.; Liu, Z.; Sun, C.; Hu, X.; He, C.; and Zhang, L. 2025a. Dataset distillation with neural characteristic function: A minmax perspective. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 25570–25580.

Wang, W.; Wei, F.; Zhou, L.; Chen, X.; Luo, L.; Yi, X.; Zhang, Y.; Liang, Y.; Xu, C.; Lu, Y.; et al. 2025b. Unigrasptransformer: Simplified policy distillation for scalable dexterous robotic grasping. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 12199–12208.

Wang, Z.; Li, M.; Mandlekar, A.; Xu, Z.; Fan, J.; Narang, Y.; Fan, L.; Zhu, Y.; Balaji, Y.; Zhou, M.; et al. 2025c. One-Step Diffusion Policy: Fast Visuomotor Policies via Diffusion Distillation. In *Forty-second International Conference on Machine Learning*.

Wen, J.; Zhu, Y.; Li, J.; Zhu, M.; Tang, Z.; Wu, K.; Xu, Z.; Liu, N.; Cheng, R.; Shen, C.; et al. 2025. TinyVLA: Towards Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation. *IEEE Robotics and Automation Letters*.

Wen, Y.; Lin, J.; Zhu, Y.; Han, J.; Xu, H.; Zhao, S.; and Liang, X. 2024. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. *Advances in Neural Information Processing Systems*, 37: 41051–41075.

Wu, H.; Jing, Y.; Cheang, C.; Chen, G.; Xu, J.; Li, X.; Liu, M.; Li, H.; and Kong, T. 2024. Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation. In *International Conference on Learning Representations*.

Xu, C.; Li, Q.; Luo, J.; and Levine, S. 2024. RLDG: Robotic Generalist Policy Distillation via Reinforcement Learning. *arXiv preprint arXiv:2412.09858*.

Yan, K.; Li, L.; Cheng, R.; Liu, X.; Li, X.; Bai, Y.; and Zhang, X. 2024. Mapping model of ribbon contour and tool influence function based on distributed parallel neural networks in magneto-rheological finishing. *Optics Express*, 32(16): 27099–27111.

Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; and Levine, S. 2020. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, 1094–1100. PMLR.

Zha, L.; Cui, Y.; Lin, L.-H.; Kwon, M.; Arenas, M. G.; Zeng, A.; Xia, F.; and Sadigh, D. 2024. Distilling and retrieving generalizable knowledge for robot manipulation via language corrections. In *2024 IEEE international conference on robotics and automation (ICRA)*, 15172–15179. IEEE.

Zhang, Q.; Han, G.; Sun, J.; Zhao, W.; Sun, C.; Cao, J.; Wang, J.; Guo, Y.; and Xu, R. 2025a. Distillation-ppo: A novel two-stage reinforcement learning framework for humanoid robot perceptive locomotion. *arXiv preprint arXiv:2503.08299*.

Zhang, R.; Dong, M.; Zhang, Y.; Heng, L.; Chi, X.; Dai, G.; Du, L.; Du, Y.; and Zhang, S. 2025b. Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. *arXiv preprint arXiv:2503.20384*.

Zhang, T.; Li, D.; Li, Y.; Zeng, Z.; Zhao, L.; Sun, L.; Chen, Y.; Wei, X.; Zhan, Y.; Li, L.; et al. 2024. Empowering embodied manipulation: A bimanual-mobile robot manipulation dataset for household tasks. *arXiv preprint arXiv:2405.18860*.

Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *7th Annual Conference on Robot Learning*.