LogicVLA: Neural Logic Critics for Efficient Vision-Language-Action Fine-Tuning

Anonymous authors
Paper under double-blind review

Abstract

Vision-language-action (VLA) models achieve strong performance in robotic manipulation, yet downstream adaptation remains computationally expensive. A fundamental limitation of standard VLA fine-tuning is its reliance on pointwise behavioral cloning, which penalizes numerical deviations from expert actions but provides no explicit supervision on whether predicted actions are compatible with the task instruction and current state. To solve this challenge, this paper develops LogicVLA, which is a training-time logical regularization framework that complements imitation learning with task-conditioned action consistency. Particularly, a lightweight *neural logic consistency* critic maps language instructions, visual observations, proprioceptive states, and actions into a shared latent space, composes their interactions through differentiable logic operators, and produces a consistency score that measures the compatibility between the multimodal task context and the predicted action. The critic contains 737K parameters and accounts for only 0.01% of the VLA backbone. It is trained directly from the original demonstrations by contrasting expert tuples with in-batch mismatched tuples, without requiring additional annotations or environment interaction. During fine-tuning, the learned consistency score provides an auxiliary regularization signal for predicted actions, while a dedicated stability mechanism prevents unstable critic gradients from affecting policy optimization. Extensive simulation results on LIBERO-Spatial show that, compared with OpenVLA, LogicVLA improves closed-loop success by 2.6% at 15K training steps and by 26% on the complex tasks, while introducing only 0.0096% additional trainable parameters during fine-tuning and no additional deployment-time computation. Code is available at https://anonymous.4open.science/r/LogicVLA-379E.

1 Introduction

Vision-language-action (VLA) models provide a unified framework for robotic manipulation by mapping language instructions and multimodal observations directly to robot actions (Kim et al., 2024; Brohan et al., 2023; Octo Model Team et al., 2024; Black et al., 2024). Recent advances in model architecture, action representation, and fine-tuning have substantially improved downstream policy performance (Kim et al., 2025), yet task-specific adaptation of large VLA models remains costly and continues to rely primarily on behavioral cloning from expert demonstrations. A fundamental limitation of this paradigm lies in the granularity of its supervision. Expert demonstrations specify successful actions along observed trajectories, but do not explicitly characterize the set of actions that remain valid under a given task context. This ambiguity is intrinsic to robotic manipulation, where action distributions are state dependent and frequently multimodal, making pointwise regression insufficient to represent multi-valued behaviors (Florence et al., 2022; Chi et al., 2023). Consequently, numerical proximity to a demonstrated action does not fully characterize whether a policy decision is appropriate for the current task. Existing VLA methods improve pretrained representations, action parameterizations, data utilization, and fine-tuning strategies (Kim et al., 2024; Octo Model Team et al., 2024; Black et al., 2024; Kim et al., 2025; Li et al., 2025a; Ye et al., 2025), but imitation-based action prediction

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

remains the principal source of downstream supervision. Representation-centric objectives improve features for control (Nair et al., 2023; Li et al., 2025a) without directly evaluating policy-generated actions, while predictive approaches assess actions through modeled future consequences (Hansen et al., 2024; Wang et al., 2025a) and require additional dynamics structure. Neither paradigm directly provides structured supervision on whether a policy action is consistent with the current multimodal task context. These limitations expose three challenges. First, VLA adaptation requires an action-level supervisory signal that captures task-relevant structure beyond pointwise imitation error. Second, this signal must be learned from the original demonstrations without additional annotation or environment interaction. Third, it must be integrated with a large pretrained policy without destabilizing optimization or increasing deployment-time computation.

To address these challenges, in this paper, we propose LogicVLA, which is a training-time framework that augments behavioral cloning with task-conditioned logical consistency. Particularly, we develop a lightweight neural logic consistency critic that maps language, visual observations, proprioceptive states, and actions into a shared latent logic space and learns structured relations through differentiable logical operators. The developed logical inference evaluates the consistency between the current task context and each policy-generated action, providing direct action-level supervision in addition to pointwise imitation. The critic is trained exclusively from the original demonstrations by contrasting matched expert context-action pairs with context-mismatched actions, and its learned consistency signal is propagated through predicted actions to optimize the VLA policy. Because joint critic-policy optimization introduces a direct pathway through which critic instability can affect the pretrained policy, LogicVLA further incorporates a stabilization mechanism that suppresses abnormal critic updates and prevents unstable consistency gradients from propagating into the VLA. The contributions of this work are threefold. First, we introduce task-conditioned logical consistency as structured action-level supervision for VLA adaptation, complementing behavioral cloning with information that is not expressed by pointwise action regression. Second, we develop a differentiable neural logic framework that learns this supervision directly from existing demonstrations and transfers it to policy optimization without additional labels, environment interaction, future-state prediction, or inference-time reasoning. Third, LogicVLA introduces only 0.0096% additional trainable parameters during fine-tuning and incurs no additional deployment-time computation. Extensive simulation results on LIBERO-Spatial show that, compared to OpneVLA, LogicVLA improves closed-loop success by 2.6% at 15K training steps and 26% on the complex tasks.

2 Related Work

Adaptation of VLA models. VLA models extend pretrained multimodal representations to robotic control by mapping language and visual observations to actions. RT-2 (Brohan et al., 2023) represents robot actions as language tokens, OpenVLA (Kim et al., 2024) scales open-source VLA pretraining with large robot datasets, and Octo (Octo Model Team et al., 2024) and π_0 (Black et al., 2024) develop generalist policies with expressive action-generation mechanisms. Recent work further improves downstream adaptation through action representation, fine-tuning, and data utilization. OpenVLA-OFT (Kim et al., 2025) introduces continuous action prediction and optimized fine-tuning, LLaRA (Li et al., 2025a) derives auxiliary instruction-tuning objectives from robot data, and LAPA (Ye et al., 2025) learns latent actions from unlabeled videos. These methods substantially improve representation quality and adaptation efficiency, but downstream policy learning remains primarily supervised by demonstrated actions. Consequently, they improve the effectiveness of imitation without explicitly modeling whether a policy-generated action is consistent with the current task context. LogicVLA addresses this complementary supervision problem by introducing structured action-level feedback during VLA fine-tuning without modifying the deployed policy architecture.

Supervision beyond pointwise imitation. Prior imitation-learning methods augment behavioral cloning with additional supervision derived from temporal structure, learned representations, expert feedback, or pretrained behavioral models. Temporal and contrastive objectives improve transferable representations by exploiting relations across ob-

servations (Sermanet et al., 2017; Nair et al., 2023). Interactive imitation introduces corrective expert feedback, while pretrained behavioral models reduce the amount of task-specific optimization required for policy adaptation. Fast Imitation (Pirotta et al., 2024) leverages behavior foundation models to accelerate imitation from limited demonstrations, and LLaRA (Li et al., 2025a) constructs auxiliary supervision from existing robot datasets. Although these approaches provide learning signals beyond direct action regression, their objectives primarily improve representation transfer, policy initialization, or corrective learning. They do not explicitly evaluate the structured relation between the current multimodal task context and the action generated by the policy. LogicVLA instead learns task-conditioned logical consistency directly from the original demonstrations and applies this supervision to policy actions without additional expert feedback, manually specified rewards, or environment interaction.

Neural logic and structured reasoning. Neural-symbolic methods introduce relational and compositional structure into learned models through differentiable abstractions and logical operators (Yang et al., 2020; Dong et al., 2019). In robotics, neural logic has been primarily used for predictive modeling, abstraction, and planning. For instance, DMWM (Wang et al., 2025a) incorporates neural logical reasoning into a world model to evaluate consistency over imagined dynamics, while VisualPredicator (Liang et al., 2025) learns neuro-symbolic predicates and abstract world models for robot planning. These approaches demonstrate the value of logical structure for reasoning about future states and long-horizon behavior, but their influence on policy learning is mediated through predictive models or planning procedures. They therefore do not directly address the need for structured action-level supervision during VLA adaptation. LogicVLA addresses this limitation by using neural logic as a training-time supervisory mechanism that learns structured relations among language, visual observations, proprioceptive states, and policy actions directly from demonstrations. The resulting logical consistency signal regularizes the actions produced by the VLA itself, providing task-aware supervision without future-state prediction, explicit planning, additional environment interaction, or inference-time computation.

2.1 Problem Formulation

We consider a vision-language-action policy π_θ parameterized by θ. Given a language instruction ℓ, a visual observation $\mathbf{i}$, and a proprioceptive state $\mathbf{p}$, the policy predicts an action chunk $\hat{\mathbf{A}} = \pi_\theta(\ell, \mathbf{i}, \mathbf{p}) \in \mathbb{R}^{C\times d_a}$, where $\hat{\mathbf{A}} = [\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_C]$ is the predicted action chunk, $\hat{\mathbf{a}}_c \in \mathbb{R}^{d_a}$ is the c-th predicted action, C is the action-chunk length, and d_a is the dimensionality of each action.

Let $\mathcal{D}$ be the demonstration dataset composed of expert tuples $(\ell, \mathbf{i}, \mathbf{p}, \mathbf{A}^*)$, where $\mathbf{A}^* \in \mathbb{R}^{C\times d_a}$ is the expert action chunk associated with the current task context. Standard VLA fine-tuning minimizes the pointwise discrepancy between the predicted and expert action chunks through the imitation objective

$$\mathcal{L}_{\text{imit}}(\theta) = \mathbb{E}_{(\ell,\mathbf{i},\mathbf{p},\mathbf{A}^*)\sim\mathcal{D}} \left[\|\pi_\theta(\ell, \mathbf{i}, \mathbf{p}) - \mathbf{A}^*\|_1\right], \tag{1}$$

where $\mathcal{L}_{\text{imit}}$ is the behavioral-cloning objective and $\|\cdot\|_1$ is the element-wise ℓ_1 norm.

Although $\mathcal{L}_{\text{imit}}$ provides dense supervision for action prediction, it constrains the policy only through numerical agreement with a demonstrated trajectory and does not explicitly encode the structured relation between the task context and the predicted action. To address this limitation, we introduce task-conditioned logical consistency as complementary supervision for VLA fine-tuning. Specifically, we learn a differentiable consistency function

$$C_\omega : (\ell, \mathbf{i}, \mathbf{p}, \mathbf{a}) \mapsto [0, 1], \tag{2}$$

where C_ω is parameterized by ω, $\mathbf{a} \in \mathbb{R}^{d_a}$ is a candidate action, and $C_\omega(\ell, \mathbf{i}, \mathbf{p}, \mathbf{a})$ measures the logical consistency between the action and the current multimodal task context. A larger consistency score indicates stronger support for the action under the structured logical relations learned from the demonstrations. The objective of LogicVLA is to learn these logical relations jointly across language, visual observations, proprioceptive states, and actions, and to use the resulting consistency signal to complement pointwise behavioral cloning. For an

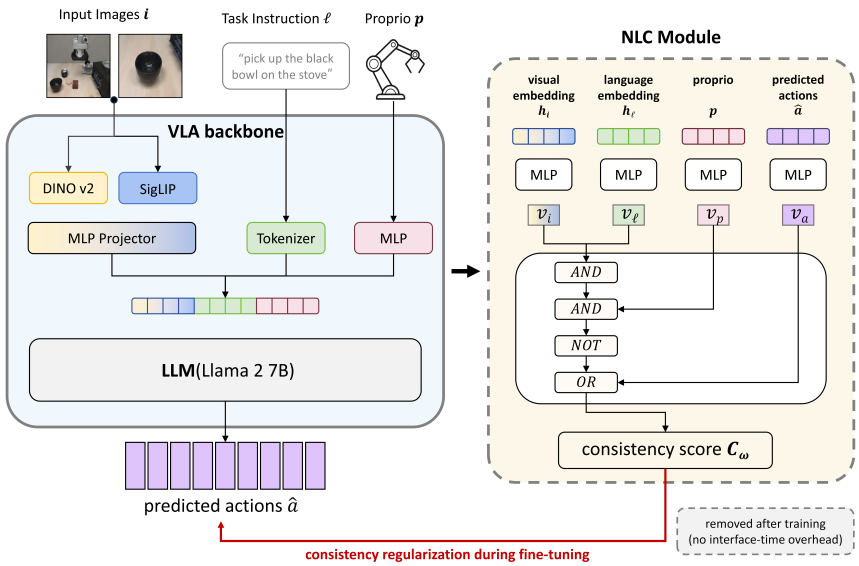


Figure 1: **LogicVLA architecture.** The VLA backbone predicts actions from multimodal task inputs, while the training-only NLC module evaluates task-conditioned logical consistency and uses the resulting score C_ω to regularize policy fine-tuning without introducing deployment overhead.

action chunk, the consistency function is evaluated on each constituent action $\hat{\mathbf{a}}_c$, which enables logical supervision to act directly on the policy prediction while preserving the original VLA output space. The following section introduces the proposed neural logic consistency module used to represent and reason over these structured relations.

2.2 Neural Logic Consistency Module

Multimodal logic representation. The neural logic consistency module provides a differentiable reasoning mechanism for evaluating whether a policy action is logically consistent with the current task context. As illustrated in Figure 1, the module first maps heterogeneous VLA representations and actions into a shared latent space in which structured logical relations can be learned. Let $\mathbf{h}_\ell \in \mathbb{R}^{d_{\text{llm}}}$ and $\mathbf{h}_i \in \mathbb{R}^{d_{\text{llm}}}$ be the pooled language and visual representations extracted from the VLA backbone, respecitvely. Let $\mathbf{p} \in \mathbb{R}^{d_p}$ denote the proprioceptive state and $\mathbf{a} \in \mathbb{R}^{d_a}$ be a candidate action. The language representation $\mathbf{h}_\ell$, visual representation $\mathbf{h}_i$, proprioceptive state $\mathbf{p}$, and candidate action $\mathbf{a}$ are projected into a shared d_v-dimensional logic space by f_ℓ, f_i, f_p, and f_a, respectively. Their latent logic representations are $\mathbf{v}_\ell = f_\ell(\mathbf{h}_\ell)$, $\mathbf{v}_i = f_i(\mathbf{h}_i)$, $\mathbf{v}_p = f_p(\mathbf{p})$, and $\mathbf{v}_a = f_a(\mathbf{a})$, where $d_v = 64$ is the logic-space dimensionality and each $f_\bullet$ is a three-layer ReLU multilayer perceptron. The parameters of the projection functions and neural logic operators are ω.

Learned logical operators. A shared latent space does not by itself impose logical structure on multimodal representations. To enable differentiable logical reasoning, we introduce learnable conjunction, disjunction, and negation operators following Wang et al. (2025a). For two logic vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_v}$, the operators are defined as

$$\text{AND}(\mathbf{u}, \mathbf{v}) = g_{\text{c}}^{\text{and}}([\mathbf{u}, \mathbf{v}]) + g_{\text{m}}^{\text{and}}(\mathbf{u} \odot \mathbf{v}), \tag{3}$$

$$\text{OR}(\mathbf{u}, \mathbf{v}) = g_{\text{c}}^{\text{or}}([\mathbf{u}, \mathbf{v}]) + g_{\text{m}}^{\text{or}}(\mathbf{u} \odot \mathbf{v}), \tag{4}$$

$$\text{NOT}(\mathbf{u}) = g^{\text{not}}(\mathbf{u}), \tag{5}$$

where $\mathbf{u}$ and $\mathbf{v}$ are two operands in the latent logic space, $[\mathbf{u}, \mathbf{v}]$ is vector concatenation, and $\mathbf{u} \odot \mathbf{v}$ is element-wise multiplication. The functions $g_{\text{c}}^{\text{and}}$ and g_{c}^{or} denote the concatenation branches for conjunction and disjunction, respectively, while $g_{\text{m}}^{\text{and}}$ and g_{m}^{or} denote their multiplicative branches. The function g^{not} parameterizes negation. Each g is implemented as a

three-layer multilayer perceptron, with the output linear transformation included in the corresponding branch network. The concatenation branch captures joint information between the operands, while the multiplicative branch captures their element-wise interactions. Together, these operators provide a differentiable parameterization of logical composition in the shared logic space.

Task-conditioned logical inference. The language, visual, and proprioceptive logic representations jointly characterize the current task context. We compose them through conjunction as

$$\mathbf{v}_{\text{ctx}} = \text{AND}\left(\text{AND}(\mathbf{v}_\ell, \mathbf{v}_i), \mathbf{v}_p\right), \tag{6}$$

where $\mathbf{v}_{\text{ctx}} \in \mathbb{R}^{d_v}$ is the logic representation of the multimodal task context. During training, the three operands are randomly permuted before conjunction to reduce dependence on a fixed composition order and encourage commutative behavior. To evaluate the logical consistency of a candidate action with the task context, we formulate their relation as an implication from the task context to the action. Following the classical equivalence $P \to Q \equiv \neg P \vee Q$, the implication representation is

$$\mathbf{v}_{\text{imp}} = \text{OR}\left(\text{NOT}(\mathbf{v}_{\text{ctx}}), \mathbf{v}_a\right), \tag{7}$$

where $\mathbf{v}_{\text{imp}} \in \mathbb{R}^{d_v}$ is the latent representation of the logical implication between the task context and the candidate action. We measure its consistency with logical truth as

$$C_\omega(\ell, \mathbf{i}, \mathbf{p}, \mathbf{a}) = \sigma\left(\cos\left(\mathbf{v}_{\text{imp}}, \mathbf{t}_{\text{true}}\right)\right), \tag{8}$$

where $\cos(\cdot,\cdot)$ is cosine similarity, $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{t}_{\text{true}} \in \mathbb{R}^{d_v}$ is a fixed non-trainable truth vector initialized element-wise from $\mathcal{U}(0,1)$. A larger C_ω indicates stronger logical consistency between the candidate action and the current task context.

Logic-algebra regularization. The learned operators in Eqs. (3)–(5) are neural function approximators and therefore do not inherently satisfy the algebraic properties required for consistent logical reasoning. To impose explicit logical structure on the shared latent space, we regularize these operators with 15 logical identities covering commutativity, associativity, De Morgan's laws, idempotence, complement, double negation, identity, and annihilation. The resulting objective is

$$\mathcal{L}_{\text{logic}}(\omega) = \alpha \sum_{k=1}^{15} r_k(\omega) + \beta \|\mathbf{z}\|_2, \tag{9}$$

where $r_k(\omega)$ is the violation of the k-th logical identity, $\mathbf{z} \in \mathbb{R}^{d_v}$ is the latent logic representation used to instantiate the corresponding constraint, $\alpha = 0.1$ controls the strength of the logical-identity regularization, and $\beta = 0.001$ controls the magnitude regularization on the logic representation. For example, conjunction idempotence requires $\text{AND}(\mathbf{z}, \mathbf{z})$ to remain consistent with $\mathbf{z}$ and penalizes their discrepancy. These constraints encourage the learned operators to form a coherent logic algebra rather than degenerate into unconstrained nonlinear transformations.

Action-pathway initialization. Reliable consistency evaluation requires the critic to remain sensitive to the candidate action when distinguishing expert-consistent actions from mismatched actions under the same task context. However, the action pathway exhibits a substantial scale imbalance at initialization because the action input has only $d_a = 7$ dimensions, whereas the task-context representation is derived from substantially higher-dimensional VLA features. Under standard Xavier initialization, the measured gradient norm through the action pathway is approximately 5×10^{-4}, compared with approximately 0.26 through the task-context pathway. This imbalance suppresses the contribution of $\mathbf{v}_a$ to the implication representation $\mathbf{v}_{\text{imp}}$ and causes the critic to depend predominantly on $\mathbf{v}_{\text{ctx}}$. To preserve action sensitivity from the beginning of training, we initialize the weight matrix $\mathbf{W}_a$ of the action projection f_a with an amplified Xavier gain,

$$\mathbf{W}_a \sim \text{Xavier}(\text{gain} = 5.0), \tag{10}$$

where $\mathbf{W}_a$ is the learnable weight matrix of f_a. This initialization increases the initial contribution of the action pathway and prevents the learned logical implication from being dominated by the task-context representation.

Parameter overhead. The complete neural logic consistency critic contains 736,833 trainable parameters, corresponding to approximately 0.0096% of the 7.65B-parameter VLA backbone. The critic is introduced exclusively during fine-tuning and discarded after training, leaving the architecture, parameter count, and inference procedure of the deployed VLA unchanged.

2.3 Two-Stage Training and Stable Co-training

To jointly optimize the VLA policy and the neural logic consistency critic while avoiding interference between their objectives, each training iteration consists of two sequential stages on the same minibatch. In the first stage, the critic parameters ω are fixed and the policy parameters θ are updated using imitation and logical consistency. Given the predicted action chunk $\hat{\mathbf{A}} = [\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_C]$, we define the consistency regularization loss as

$$\mathcal{L}_{\text{cons}}(\theta; \omega) = 1 - \frac{1}{C} \sum_{c=1}^{C} C_\omega \left(\ell, \mathbf{i}, \mathbf{p}, \hat{\mathbf{a}}_c\right). \tag{11}$$

The policy objective is

$$\mathcal{L}_{\text{pol}}(\theta) = \mathcal{L}_{\text{imit}}(\theta) + \lambda \mathcal{L}_{\text{cons}}(\theta; \omega), \tag{12}$$

where $\lambda = 0.1$ controls the contribution of logical consistency. Although ω is fixed in this stage, gradients propagate through C_ω to the predicted actions and VLA representations, allowing the learned logical structure to directly influence policy optimization. Early in training, policy predictions can lie far from the expert-action distribution used to train the critic, making the resulting consistency gradients unreliable. To avoid such supervision, we use a warmup period of $T_w = 1500$ gradient steps. During this period, the policy is optimized only with $\mathcal{L}_{\text{imit}}$, after which the consistency term is activated.

In the second stage, the VLA parameters θ are fixed and all VLA-derived representations are detached so that gradients update only the critic parameters ω. The critic learns to distinguish matched expert context-action pairs from context-mismatched pairs constructed within the same minibatch. For a minibatch of size B, let $b \in \{1, \ldots, B\}$ denote the sample index. We define the cyclic permutation

$$\rho(b) = 1 + \left(b - 1 + \left\lfloor \frac{B}{2} \right\rfloor\right) \bmod B. \tag{13}$$

The positive pair associates the task context $(\ell_b, \mathbf{i}_b, \mathbf{p}_b)$ with its expert action chunk $\mathbf{A}^*_b$, while the negative pair associates the same context with the expert action chunk $\mathbf{A}^*_{\rho(b)}$ from another sample. Their chunk-level consistency scores are

$$s_b^+ = \frac{1}{C} \sum_{c=1}^{C} C_\omega \left(\ell_b, \mathbf{i}_b, \mathbf{p}_b, \mathbf{a}^*_{b,c}\right), \tag{14}$$

$$s_b^- = \frac{1}{C} \sum_{c=1}^{C} C_\omega \left(\ell_b, \mathbf{i}_b, \mathbf{p}_b, \mathbf{a}^*_{\rho(b),c}\right), \tag{15}$$

where s_b^+ and s_b^- denote the consistency scores of the matched and mismatched pairs, respectively. The critic is optimized with the margin-based ranking objective

$$\mathcal{L}_{\text{crit}}(\omega) = \frac{1}{B} \sum_{b=1}^{B} \max\left(0, m - s_b^+ + s_b^-\right) + \mathcal{L}_{\text{logic}}(\omega) + \mathcal{L}_{\text{L2}}(\omega), \tag{16}$$

where $m = 0.25$ is the ranking margin and $\mathcal{L}_{\text{L2}}$ is the additional ℓ_2 regularization applied to the critic parameters. The critic is optimized using SGD with its gradient norm clipped to 1.0. We construct negative examples from expert actions within the minibatch rather than from artificial perturbations. This preserves the marginal structure of valid robot actions while breaking their correspondence with the current task context, requiring the critic to learn task-conditioned action consistency rather than distinguish expert actions from synthetic noise.

The differentiable coupling between the critic and policy enables logical consistency to shape the VLA directly, but it also creates a pathway through which critic instability can affect

policy optimization. We therefore stabilize co-training at three levels. A loss-spike filter discards an iteration when the monitored training loss exceeds ten times its sliding-window mean, with the decision synchronized across distributed workers. The gradient norm of all trainable VLA parameters is clipped to 1.0 before the policy optimizer step. We further monitor the critic gradient norm at its optimizer boundary and define $\tau = 10.0$ as the instability threshold. Let $q_t \in \{0, 1\}$ indicate whether logical consistency is enabled at iteration t. After the critic update at iteration t, the state for the following policy update is determined by

$$q_{t+1} = \mathbb{I}\left[\left\|\nabla_\omega \mathcal{L}_{\text{crit}}^{(t)}\right\|_2 \leq \tau\right], \tag{17}$$

where $\mathbb{I}[\cdot]$ is the indicator function. After the warmup period, the policy objective is therefore written as

$$\widetilde{\mathcal{L}}_{\text{pol}}^{(t)} = \mathcal{L}_{\text{imit}}^{(t)} + q_t \lambda \mathcal{L}_{\text{cons}}^{(t)}. \tag{18}$$

When the critic gradient norm exceeds τ, its optimizer update is skipped and $q_{t+1} = 0$. The following policy update then uses only $\mathcal{L}_{\text{imit}}$. Logical consistency is reactivated once the critic gradient returns below the threshold. Under distributed data parallel training, critic gradients are synchronized before the norm is evaluated so that all workers make the same update and suppression decisions.

Overall, our training strategy combines two-stage optimization with a dedicated stability scheme to enable reliable joint learning of the VLA policy and neural logic critic. The two-stage procedure separates policy optimization from critic learning by freezing the critic during policy updates and detaching the VLA during critic updates, which prevents the two objectives from interfering within the same optimization step. Building on this separation, VLA gradient clipping bounds policy updates, critic update skipping prevents abnormal critic gradients from modifying the critic, and one-step suppression blocks unstable logical supervision from entering the subsequent policy update. These mechanisms coordinate policy learning and critic adaptation while isolating the VLA from transient critic instability, allowing logical consistency to provide effective supervision throughout training. The complete training procedure is summarized in Algorithm 1 in Appendix **??**.

3 Experiments

3.1 Setup

We evaluate LogicVLA on LIBERO (Liu et al., 2024), using LIBERO-Spatial as the primary benchmark and LIBERO-Long to examine the stability of critic–policy co-training. LIBERO-Spatial contains 10 manipulation tasks with 50 fixed initial states per task, resulting in 500 closed-loop rollouts per evaluation. We use OpenVLA-7B (Kim et al., 2024) with the OFT fine-tuning recipe (Kim et al., 2025). Our primary baseline uses the same VLA backbone, demonstrations, hardware, and optimization configuration as LogicVLA but removes the NLC critic and its associated consistency objective. We additionally report OpenVLA-OFT results to contextualize training-step efficiency. The NLC critic introduces only 0.0096% additional trainable parameters during fine-tuning and is removed entirely after training. Complete implementation hyperparameters are provided in Appendix C, while additional training diagnostics and statistical analyses are presented in Appendix E. The NLC critic introduces only 0.0096% additional trainable parameters during fine-tuning and is removed entirely after training, introducing no additional deployment-time parameters or computation. Complete implementation and parameter-overhead details are provided in Appendix D.

3.2 Main Results on LIBERO-Spatial

Table 1 reports the closed-loop performance on LIBERO-Spatial. Relative to the otherwise identical unregularized baseline, LogicVLA increases the success rate from 84.6% to 91.6% at 4K training steps and from 94.4% to 97.0% at 15K steps. The larger improvement at the earlier checkpoint is consistent with the hypothesis that task-conditioned logical consistency provides complementary supervision during the early stage of policy adaptation, while the retained advantage at 15K indicates that its effect persists as behavioral cloning

Table 1: Closed-loop success on LIBERO-Spatial over three training seeds. Δ denotes the absolute improvement of LogicVLA over the otherwise identical unregularized baseline.

Method	**4K**	**10K**	**15K**	**150K**
LogicVLA (ours)	$\mathbf{91.6 \pm 0.4}$	$\mathbf{93.8 \pm 0.2}$	$\mathbf{97.0 \pm 0.2}$	99.0 ± 0.2
w/o NLC	84.6 ± 0.4	93.2 ± 0.2	94.4 ± 0.4	–
Δ over w/o NLC (pp)	+7.0	+0.6	+2.6	–
OpenVLA-OFT (Kim et al., 2025)	–	–	–	97.6[†]
OpenVLA-OFT, reproduced	–	–	–	97.8[‡]

[†]Paper-reported result averaged over three training seeds. [‡]Official checkpoint evaluated under our local protocol. Paired statistical tests are reported in Appendix E.3.

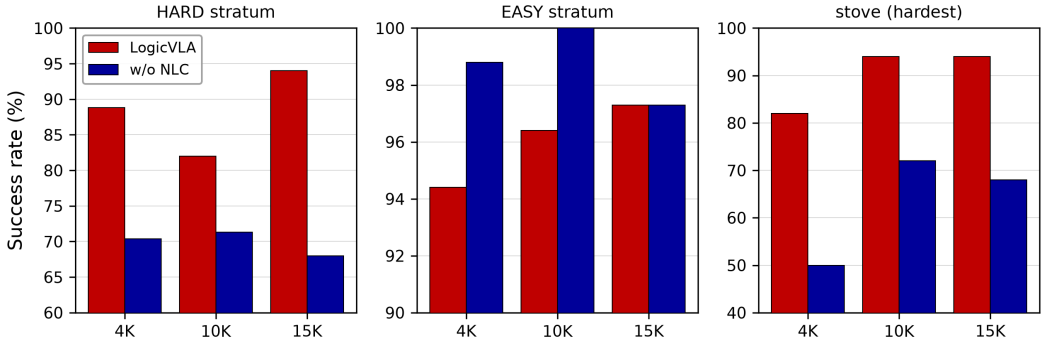


Figure 2: Difficulty-stratified closed-loop success on LIBERO-Spatial. Hard and easy strata are defined according to the unregularized baseline at each checkpoint. The fixed stove task is shown separately to provide a checkpoint-consistent comparison.

approaches convergence. At 15K steps, LogicVLA achieves 97.0% success, compared with the 97.6% reported by OpenVLA-OFT at 150K steps, showing that comparable closed-loop performance can be reached with one-tenth of the reported training steps. The 15K checkpoint also lies within the performance range of representative VLA methods reported on LIBERO-Spatial, while the 150K checkpoint reaches 99.0% success, with the broader comparison provided in Appendix E.4. The aggregate results further reveal that the benefit of logical consistency is not uniform across tasks. As examined in Section 3.3, the improvement is concentrated on more difficult tasks, which is consistent with the role of the NLC critic in providing task-conditioned supervision when pointwise imitation alone does not fully capture the compatibility between the multimodal task context and the generated action. The complete training trajectory is reported in Figure 3 in Appendix E.

3.3 Task-Difficulty Analysis

Figure 2 analyzes the performance difference according to task difficulty. The benefit of LogicVLA is concentrated on the hard stratum, whereas the difference on easy tasks progressively disappears as behavioral cloning approaches saturation. At 15K steps, LogicVLA achieves a 26% improvement on the fixed hardest stove task compared with the unregularized baseline. The same task exhibits improvements of 32% and 22% at 4K and 10K steps, respectively, showing that the advantage persists throughout adaptation rather than emerging only near convergence.

The difficulty dependence provides insight into the role of logical consistency. Easy tasks can already be solved reliably through behavioral cloning, leaving limited room for an additional structured objective. In contrast, difficult behaviors place stronger demands on the relation among language instructions, visual observations, proprioceptive states, and generated actions. The larger improvement in this regime is therefore consistent with the intended role of the NLC critic as task-conditioned action supervision beyond pointwise regression.

Table 2: Ablation results on LIBERO-Spatial at 15K training steps. Stove denotes the fixed hardest task.

Configuration	**Overall**	**Stove**	**Δ Overall**
Full	97.0%	94%	–
w/o logical consistency	94.4%	68%	−2.6 pp
w/o policy warmup	1.8%	0%	−95.2 pp
$\lambda = 0.01$	96.8%	92%	−0.2 pp
$\lambda = 1.0$	0.0%	0%	−97.0 pp

The distinction is further supported by open-loop evaluation. At 15K steps, LogicVLA achieves a 2.6% improvement in closed-loop success despite a 34% higher open-loop ℓ_1 action error than the unregularized baseline. The improvement therefore cannot be explained by closer numerical reproduction of the demonstrated actions. Instead, the result indicates that the learned consistency objective captures task-relevant action structure that is not fully reflected by pointwise imitation error.

3.4 Ablation Studies

Table 2 evaluates the ablation studies of LogicVLA at 15K training steps. Removing logical consistency decreases overall closed-loop success by 2.6% and reduces success on the hardest task by 26%, showing that the improvement is attributable to the proposed task-conditioned action supervision rather than to the underlying OFT fine-tuning recipe. The substantially larger degradation on the hardest task further indicates that logical consistency is particularly beneficial when pointwise behavioral cloning alone is insufficient to capture the relation between the multimodal task context and the generated action. The policy warmup is also critical. Removing it decreases overall success by 95.2% because early policy predictions can lie far from the expert-action distribution used to train the critic, which makes the resulting consistency gradients unreliable for policy optimization. The warmup allows behavioral cloning to first adapt the policy toward the demonstrated action distribution before logical supervision is introduced. The consistency weight exhibits asymmetric sensitivity. Reducing λ by a factor of ten decreases overall success by only 0.2%, whereas increasing it by a factor of ten causes optimization to collapse. This result indicates that the learned consistency signal remains effective even with a relatively small contribution, while excessive logical regularization can overwhelm the behavioral-cloning objective and destabilize policy learning. Together, these ablations show that the effectiveness of LogicVLA depends on both the quality and the timing of logical supervision. Task-conditioned consistency provides structured information beyond pointwise imitation, while warmup and moderate regularization ensure that this information remains aligned with the evolving policy distribution. Additional training diagnostics, statistical tests, and comparisons with prior methods are provided in Appendix E.

4 Conclusion

In this paper, we presented LogicVLA, a training-time framework that enhances behavioral cloning with task-conditioned logical consistency for efficient VLA fine-tuning. A lightweight neural logic consistency critic learns structured relations among language instructions, visual observations, proprioceptive states, and actions directly from the original demonstrations, and provides action-level supervision on policy predictions beyond pointwise imitation. A two-stage optimization procedure together with stable critic–policy co-training enables this consistency signal to guide the VLA without affecting the deployed policy. On LIBERO-Spatial, LogicVLA improves closed-loop success over an otherwise identical unregularized baseline by 2.6% at 15K training steps and by 26% on the hardest task. It further reaches 97.0% success at 15K steps, comparable to the 97.6% reported by OpenVLA-OFT at 150K steps, while using one-tenth of the training steps. With only 0.0096% additional trainable parameters during fine-tuning and no additional deployment-time computation, LogicVLA demonstrates that task-conditioned logical consistency can serve as a lightweight structured supervisory signal for efficient VLA adaptation.

References

Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In *International Conference on Learning Representations*, volume 2025, pp. 56729–56763, 2025.

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems*, 2025.

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: a dual-system vla model unifying fast manipulation within slow reasoning. *Advances in Neural Information Processing Systems*, 38:98049–98083, 2026.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023.

Honghua Dong, Jiayuan Mao, Tian Lin, Chong Zhang, Lihong Li, and Denny Zhou. Neural logic machines. In *International Conference on Learning Representations*, 2019.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *Advances in Neural Information Processing Systems*, 38:102867–102888, 2026.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning*, 2022.

Asher Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. In *International Conference on Learning Representations*, volume 2026, pp. 75333–75360, 2026.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Hongyang Li, et al. Wholebodyvla: Towards unified latent vla for whole-body loco-manipulation control. In *International Conference on Learning Representations*, volume 2026, pp. 157438–157461, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2501.16764*, 2025.

Hao Li, Zi-han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, Si Liu, et al. Robointer: A holistic intermediate representation suite towards robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 113314–113381, 2026a.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Yang Zhaohui, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 125846–125869, 2026b.

Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuang Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, and Michael S. Ryoo. LLaRA: Supercharging robot learning data for vision-language policy. In *International Conference on Learning Representations*, 2025a.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Caelan Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, et al. Hamster: Hierarchical action models for open-world robot manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 24040–24068, 2025b.

Yichao Liang, Nishanth Kumar, Hao Tang, Adrian Weller, Joshua B. Tenenbaum, Tom Silver, João F. Henriques, and Kevin Ellis. VisualPredicator: Learning abstract world models with neuro-symbolic predicates for robot planning. In *International Conference on Learning Representations*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 2024.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Proceedings of the 6th Conference on Robot Learning*, 2023.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 54853–54876, 2025.

Karl Pertsch, Kevin Black, et al. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Matteo Pirotta, Andrea Tirinzoni, Ahmed Touati, Alessandro Lazaric, and Yann Ollivier. Fast imitation via behavior foundation models. In *International Conference on Learning Representations*, 2024.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, Jiayuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems*, 2025.

Pierre Sermanet, Corey Lynch, Jasmine Hsu, and Sergey Levine. Time-contrastive networks and self-supervised learning. In *IEEE International Conference on Robotics and Automation*, 2017.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026.

Lingyi Wang, Rashed Shelim, Walid Saad, and Naren Ramakrishnan. DMWM: Dual-mind world model with long-term imagination. *Advances in Neural Information Processing Systems*, 2025a.

Lingyi Wang, Rashed Shelim, Walid Saad, and Naren Ramakrishnan. Dual-mind world models: A general framework for learning in dynamic wireless networks. *arXiv preprint arXiv:2510.24546*, 2025b.

Lingyi Wang, Rashed Shelim, Walid Saad, and Naren Ramakrishnan. World model-based learning for long-term age of information minimization in vehicular networks. In *GLOBECOM 2025-2025 IEEE Global Communications Conference*, pp. 6202–6207. IEEE, 2025c.

Lingyi Wang, Tingyu Shui, Walid Saad, and Pascal Adjakple. World model-enabled causal digital twins for semantic communications in physical ai systems. *arXiv preprint arXiv:2605.16547*, 2026.

Zhun Yang, Joohyung Lee, and Chitta Park. Deep learning with logical constraints. In *International Joint Conference on Artificial Intelligence*, 2020.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Lee Kimin, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38:24195–24228, 2026a.

Wenyao Zhang, Bozhou Zhang, Zekun Qi, Wenjun Zeng, Xin Jin, and Li Zhang. Disentangled robot learning via separate forward and inverse dynamics pretraining. *arXiv preprint arXiv:2604.16391*, 2026b.

Wei Zhao, Pengxiang Ding, Zhang Min, Zhefei Gong, Shuanghao Bai, Han Zhao, and Donglin Wang. Vlas: Vision-language-action model with speech instructions for customized robot manipulation. In *International conference on learning representations*, volume 2025, pp. 51676–51693, 2025.

Contents of Appendix

A Analysis and Discussion **14**

A.1 Limitations and Future Work 14

A.2 Why Does Logical Consistency Improve VLA Adaptation? 14

A.3 Relationship and Difference to DMWM 15

B Algorithm **16**

C Implementation Hyperparameters **16**

D Parameter and Deployment Overhead **17**

E Additional Simulation Results **17**

E.1 Complete Training Trajectory 18

E.2 Critic–Policy Co-training Stability 18

E.3 Paired Statistical Tests 19

E.4 Comparison with Prior VLA Methods 19

A Analysis and Discussion

A.1 Limitations and Future Work

Our current study has several limitations. First, the evaluation is restricted to the LIBERO benchmark family and a single VLA backbone based on OpenVLA-7B with the OFT fine-tuning recipe, so the generality of task-conditioned logical consistency across different VLA architectures, action representations, and manipulation domains remains to be established. Second, LIBERO-Spatial approaches saturation at longer training horizons, which limits its ability to distinguish strong policies after convergence. The LIBERO-Long experiments therefore focus on the training stability of critic–policy co-training rather than a complete closed-loop comparison of long-horizon task performance. Finally, while the main LIBERO-Spatial results are averaged over three training seeds, the ablation studies use a single seed for each configuration, so training variability is characterized more thoroughly for the main comparison than for the component analysis.

Future work should evaluate LogicVLA across additional VLA architectures, manipulation benchmarks, and real robotic systems to assess its robustness under broader perception and control conditions. A more comprehensive closed-loop evaluation on long-horizon tasks would also help determine whether logical consistency provides persistent benefits as task complexity increases. In addition, repeated ablation runs and broader sensitivity studies could further characterize the effects of the consistency weight, critic optimization, and stabilization strategy. Beyond these empirical extensions, an important direction is to investigate whether task-conditioned logical consistency can support richer forms of structured supervision for large robot policies without requiring additional annotations, environment interaction, future-state prediction, or inference-time reasoning.

A.2 Why Does Logical Consistency Improve VLA Adaptation?

The benefit of logical consistency arises from the difference between pointwise imitation supervision and task-conditioned action supervision. Behavioral cloning optimizes each predicted action according to its numerical deviation from the demonstrated action. For a predicted action $\hat{\mathbf{a}}_c$, the corresponding policy gradient is determined by

$$\nabla_\theta \mathcal{L}_{\text{imit}} = \mathbf{J}_{\hat{\mathbf{A}}}(\theta)^\top \nabla_{\hat{\mathbf{A}}} \left\| \hat{\mathbf{A}} - \mathbf{A}^* \right\|_1 , \tag{19}$$

where $\mathbf{J}_{\hat{\mathbf{A}}}(\theta)$ denotes the Jacobian of the predicted action chunk with respect to the policy parameters. This gradient provides dense supervision toward the demonstrated trajectory, but its direction is determined entirely by action-space discrepancy. It does not explicitly account for whether a deviation remains compatible with the current language instruction, visual observation, and proprioceptive state.

Logical consistency introduces an additional gradient whose direction depends jointly on the predicted action and the multimodal task context. From Eq. (11), the corresponding gradient is

$$\nabla_\theta \mathcal{L}_{\text{cons}} = -\frac{1}{C} \sum_{c=1}^{C} \mathbf{J}_{\hat{\mathbf{a}}_c}(\theta)^\top \nabla_{\hat{\mathbf{a}}_c} C_\omega \left(\ell, \mathbf{i}, \mathbf{p}, \hat{\mathbf{a}}_c\right), \tag{20}$$

where $\mathbf{J}_{\hat{\mathbf{a}}_c}(\theta)$ denotes the Jacobian of the c-th predicted action with respect to θ. The resulting policy update therefore combines pointwise imitation with a context-dependent direction

$$\nabla_\theta \mathcal{L}_{\text{pol}} = \nabla_\theta \mathcal{L}_{\text{imit}} + \lambda \nabla_\theta \mathcal{L}_{\text{cons}}. \tag{21}$$

This additional direction allows two actions with similar imitation errors to receive different supervision when their relations to the task context differ. In particular,

$$\|\mathbf{a}_1 - \mathbf{a}^*\|_1 \approx \|\mathbf{a}_2 - \mathbf{a}^*\|_1 \;\not\Rightarrow\; C_\omega \left(\ell, \mathbf{i}, \mathbf{p}, \mathbf{a}_1\right) \approx C_\omega \left(\ell, \mathbf{i}, \mathbf{p}, \mathbf{a}_2\right). \tag{22}$$

The critic can therefore distinguish actions that are numerically similar with respect to a single demonstration but differ in their learned compatibility with the current task context.

The experimental results are consistent with this interpretation. The largest improvement appears on the most challenging task, where LogicVLA improves closed-loop success by 26%

over the unregularized baseline. At the same time, LogicVLA achieves higher closed-loop success despite a higher open-loop ℓ_1 error. This dissociation indicates that improved task performance cannot be explained solely by closer reproduction of the demonstrated actions. Instead, the auxiliary consistency objective changes the policy optimization direction toward action representations that better preserve task-relevant structure. We therefore interpret the observed learning acceleration as the result of complementary supervision rather than a stronger form of pointwise imitation.

A.3 Relationship and Difference to DMWM

LogicVLA adopts differentiable logical operators inspired by DMWM (Wang et al., 2025a), but the two methods use neural logic to solve fundamentally different learning problems. DMWM is a model-based framework designed to improve the logical consistency of long-horizon imagination. Its RSSM-based System 1 predicts latent environment dynamics, while its logic-based System 2 evaluates logical relations among states and actions over imagined trajectories. A local dynamics relation in DMWM can be represented schematically as

$$\Phi_t^{\mathrm{DMWM}} = (\mathbf{z}_{t-1} \wedge \mathbf{a}_{t-1}) \rightarrow \mathbf{z}_t, \tag{23}$$

where $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$ denote consecutive latent states and $\mathbf{a}_{t-1}$ denotes the intervening action. DMWM recursively composes such relations across time so that logical reasoning constrains the consistency of imagined state transitions. Its logical signal therefore operates primarily on the learned dynamics model and is incorporated into the world-model objective to improve future-state prediction and long-horizon imagination.

The reasoning target of LogicVLA is different. We do not model the transition from a current state to a future state. Instead, the language instruction, current visual observation, and proprioceptive state define the task context, and the predicted policy action is evaluated directly under this context. The relation modeled by LogicVLA is

$$\Phi^{\mathrm{LogicVLA}} = (\ell \wedge \mathbf{i} \wedge \mathbf{p}) \rightarrow \mathbf{a}. \tag{24}$$

This formulation removes the requirement to predict $\mathbf{z}_{t+1}$ or construct an imagined trajectory before logical consistency can affect the policy. Neural logic therefore changes from a constraint on model-based imagination to a direct source of supervision for VLA adaptation. This distinction also changes how logical information reaches the policy. In DMWM, the logical objective improves the dynamics representation used to generate more coherent imagined trajectories. The effect on control is consequently mediated through the learned world model and the subsequent decision-making procedure.

In LogicVLA, the consistency score is evaluated on the action generated by the VLA itself, and its gradient is propagated directly to the policy parameters

$$C_\omega\left(\ell, \mathbf{i}, \mathbf{p}, \pi_\theta(\ell, \mathbf{i}, \mathbf{p})\right) \longrightarrow \nabla_\theta \mathcal{L}_{\mathrm{cons}}. \tag{25}$$

The supervision path is therefore substantially shorter and does not depend on learning an environment transition model.

The two approaches also differ in their temporal requirements. DMWM explicitly reasons over sequential state transitions and extends logical consistency across imagined horizons. This design is appropriate for its objective of reducing accumulated prediction errors during long-term imagination. LogicVLA instead operates on individual demonstration tuples and evaluates each predicted action under the current multimodal task context. As a result, the proposed critic can be trained directly from standard VLA demonstrations without introducing future-state prediction, recursive imagination, or an additional planning process. The critic is further removed after fine-tuning, so the learned logical structure influences the deployed policy through its training trajectory rather than through an additional reasoning component at inference.

These differences define the role of neural logic in LogicVLA. DMWM demonstrates that differentiable logic can improve the temporal coherence of a learned world model. LogicVLA extends this principle to a different supervision problem by using logical structure to evaluate policy-generated actions directly. This formulation complements behavioral cloning with task-conditioned action supervision while preserving the original VLA architecture and requiring no additional annotations, environment interaction, future-state model, or deployment-time reasoning.

B Algorithm

Algorithm 1 summarizes the complete training procedure of LogicVLA. Each iteration performs policy shaping and critic learning sequentially on the same minibatch, while the proposed stability mechanism monitors critic behavior and suppresses unreliable logical supervision when instability is detected.

Algorithm 1 LogicVLA Training with Stable Critic-Policy Co-training

Set hyper-parameters N, B, C, λ, T_w, m, τ, η_θ, and η_ω.
Initialize the pretrained VLA policy π_θ and the neural logic consistency critic C_ω.
Initialize the action projection $\mathbf{W}_a$ with Xavier gain 5.0 and fix $\mathbf{t}_{\text{true}} \sim \mathcal{U}(0,1)$.
Set the consistency state $q_1 \leftarrow 1$.
for Training step $n = 1 \rightarrow N$ **do**
  Sample a minibatch $\{(\ell_b, \mathbf{i}_b, \mathbf{p}_b, \mathbf{A}_b^*)\}_{b=1}^B \sim \mathcal{D}$.
  // Stage 1: Policy Shaping
  Predict $\hat{\mathbf{A}}_b = \pi_\theta(\ell_b, \mathbf{i}_b, \mathbf{p}_b)$.
  Compute the imitation loss $\mathcal{L}_{\text{imit}}$.
  if $n > T_w$ and $q_n = 1$ **then**
    Evaluate $C_\omega(\ell_b, \mathbf{i}_b, \mathbf{p}_b, \hat{\mathbf{a}}_{b,c})$ for each predicted action.
    Compute $\mathcal{L}_{\text{cons}} = 1 - \frac{1}{BC}\sum_{b=1}^B \sum_{c=1}^C C_\omega(\ell_b, \mathbf{i}_b, \mathbf{p}_b, \hat{\mathbf{a}}_{b,c})$.
    Set $\mathcal{L}_{\text{pol}} = \mathcal{L}_{\text{imit}} + \lambda \mathcal{L}_{\text{cons}}$.
  else
    Set $\mathcal{L}_{\text{pol}} = \mathcal{L}_{\text{imit}}$.
  end if
  Freeze ω, clip the VLA gradient norm to 1.0, and update $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_{\text{pol}}$.
  // Stage 2: Neural Logic Critic Learning
  Detach all VLA-derived representations from the policy computation graph.
  Construct the cyclic permutation $\rho(b) = 1 + (b - 1 + \lfloor B/2 \rfloor) \bmod B$.
  Compute matched context-action scores $s_b^+ = \frac{1}{C}\sum_{c=1}^C C_\omega(\ell_b, \mathbf{i}_b, \mathbf{p}_b, \mathbf{a}_{b,c}^*)$.
  Compute mismatched context-action scores $s_b^- = \frac{1}{C}\sum_{c=1}^C C_\omega(\ell_b, \mathbf{i}_b, \mathbf{p}_b, \mathbf{a}_{\rho(b),c}^*)$.
  Compute $\mathcal{L}_{\text{crit}} = \frac{1}{B}\sum_{b=1}^B \max\left(0, m - s_b^+ + s_b^-\right) + \mathcal{L}_{\text{logic}} + \mathcal{L}_{\text{L2}}$.
  // Stable Critic–Policy Co-training
  Synchronize critic gradients across distributed workers and compute $G_n = \|\nabla_\omega \mathcal{L}_{\text{crit}}\|_2$.
  if $G_n > \tau$ **then**
    Skip the critic update and set $q_{n+1} \leftarrow 0$.
  else
    Clip the critic gradient norm to 1.0.
    Update $\omega \leftarrow \omega - \eta_\omega \nabla_\omega \mathcal{L}_{\text{crit}}$.
    Set $q_{n+1} \leftarrow 1$.
  end if
  if the monitored loss exceeds 10 times its sliding-window mean **then**
    Synchronize the decision across workers and discard the unstable iteration.
  end if
end for
Discard the critic C_ω and apply the optimized VLA policy π_θ.

C Implementation Hyperparameters

Table 3 summarizes the hyperparameters used throughout our experiments. Unless otherwise specified, the same VLA training configuration is used for LogicVLA and the unregularized baseline, with the latter removing only the neural logic consistency critic and its associated training objectives.

Table 3: Hyperparameters used for LogicVLA training.

Component	Hyperparameter	Value
VLA	Backbone	OpenVLA-7B
	Fine-tuning recipe	OpenVLA-OFT
	Action dimension d_a	7
	Proprioceptive dimension d_p	8
	Image observations	Agent view + wrist view
	LoRA rank	32
	Action chunk size C	8
	Effective batch size B	64
	Optimizer	AdamW
	Learning rate	5×10^{-4}
	Learning-rate decay	10× at 100K steps
	Maximum training steps	150K
	Gradient-norm clipping	1.0
NLC critic	Logic-space dimension d_v	64
	Projection networks	3-layer ReLU MLPs
	Logical operator networks	3-layer MLPs
	Action projection gain	5.0
	Logic identity weight α	0.1
	Magnitude regularization β	0.001
	Optimizer	SGD
	Learning rate	3×10^{-4}
	Gradient-norm clipping	1.0
Two-stage training	Consistency weight λ	0.1
	Policy warmup T_w	1500 steps
	Ranking margin m	0.2
	Negative construction	In-batch cyclic permutation
Stability	Critic instability threshold τ	10.0
	Loss-spike threshold	10× sliding-window mean
	Consistency suppression	One subsequent policy step
	Distributed training	DDP
Hardware	GPUs	4× A800-80GB

Table 4: Parameter and deployment overhead of LogicVLA. The NLC critic is used only during fine-tuning and is removed for deployment.

Configuration	Parameters	Additional	Deployment
VLA backbone	7.65B	–	✓
NLC critic	736,833	0.0096%	–
LogicVLA fine-tuning	7.65B + 736,833	0.0096%	–
LogicVLA deployment	7.65B	0%	✓

D Parameter and Deployment Overhead

The NLC critic is introduced only during fine-tuning and is removed after training. Table 4 reports its parameter overhead and the resulting deployment configuration.

E Additional Simulation Results

This section provides experimental evidence complementary to the main results in Section 3. We report the complete LIBERO-Spatial training trajectory, critic–policy co-training diagnostics on LIBERO-Long, paired statistical tests, and a broader comparison with prior VLA methods.

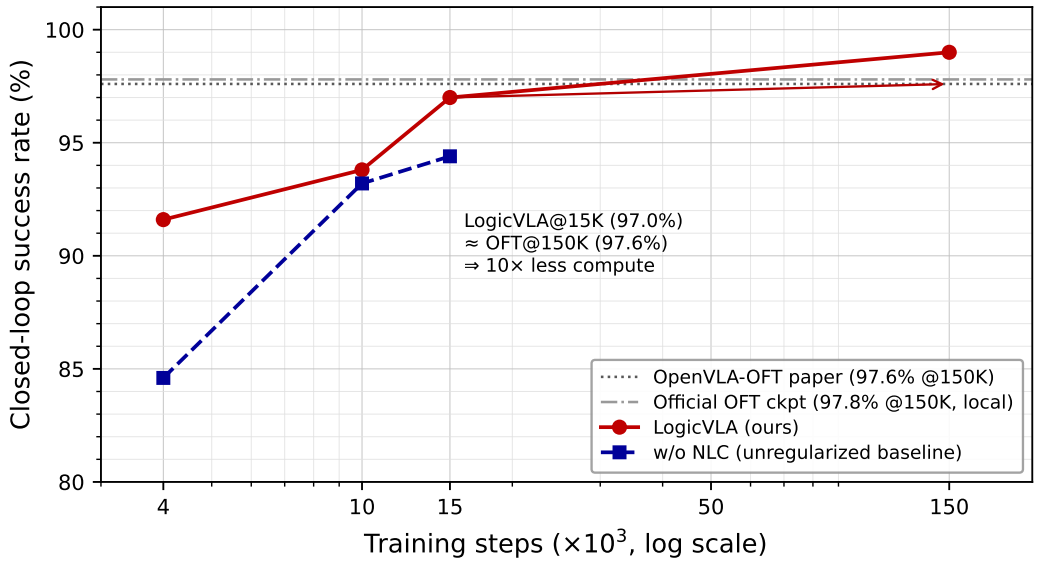


Figure 3: Closed-loop success on LIBERO-Spatial as a function of training steps. Horizontal lines denote the paper-reported and locally evaluated OpenVLA-OFT results at 150K steps.

E.1 Complete Training Trajectory

Figure 3 complements the checkpoint results in Table 1 by visualizing closed-loop performance throughout training. The purpose of this analysis is to characterize how the performance gap evolves between the reported checkpoints rather than to repeat the quantitative comparisons in the main paper.

As shown in Figure 3, the effect of logical consistency varies over optimization rather than producing a constant offset from the unregularized baseline. Nevertheless, LogicVLA enters the performance range of the 150K-step OpenVLA-OFT references substantially earlier. This trajectory complements the reduced-budget comparison in the main paper and shows that the benefit of logical supervision is retained as policy optimization progresses.

We describe this result in terms of training steps rather than computational cost because wall-clock time and total floating-point computation are not measured directly.

E.2 Critic–Policy Co-training Stability

We use LIBERO-Long as a stress test for the differentiable coupling between the NLC critic and the VLA policy. This experiment evaluates optimization stability rather than closed-loop superiority on LIBERO-Long.

As shown in Figure 4, the VLA validation loss decreases from 0.172 to 0.082, corresponding to a 52.3% reduction from the first to the final reported checkpoint. During the same interval, critic AUC improves from 0.754 to 0.902, corresponding to a 19.6% relative improvement, while the positive-negative margin increases from 0.218 to 0.314, corresponding to a 44.0% relative improvement. Pairwise discrimination accuracy additionally increases by 11.9%. These trends show that critic discrimination improves while VLA optimization continues to progress.

The training run contains 515 detected critic-gradient instability events. The largest observed critic gradient norm reaches 120.28 times the configured instability threshold. The subsequently measured VLA validation loss does not exhibit a persistent increase around this event. This behavior is consistent with the intended role of the proposed suppression mechanism, which prevents a detected unstable critic state from contributing logical supervision to the following policy update. We therefore interpret the experiment as evidence

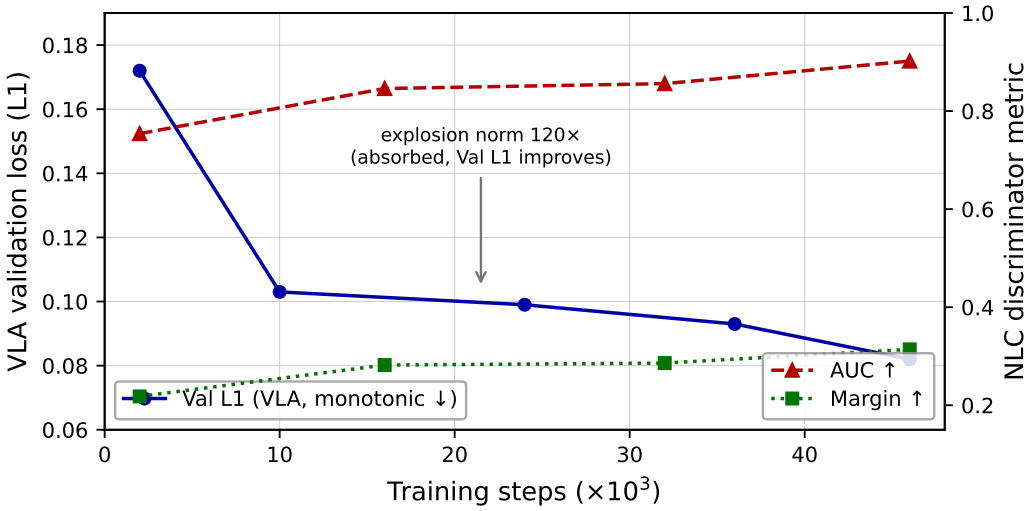


Figure 4: Critic–policy co-training diagnostics on LIBERO-Long over 47.5K training steps. VLA validation loss is shown together with critic AUC and positive-negative margin. The annotation marks the largest detected critic-gradient instability event.

Table 5: Exact McNemar p-values for paired LIBERO-Spatial evaluations.

Evaluation set	**4K**	**10K**	**15K**
Overall	4.9×10^{-4}	0.79	0.0533
Hard stratum	2.3×10^{-7}	0.011	2.35×10^{-3}
Easy stratum	0.013	0.004	1.00

that the proposed training strategy maintains stable critic–policy co-training under repeated transient instability.

E.3 Paired Statistical Tests

Because LogicVLA and the unregularized baseline are evaluated from the same fixed initial states, we use McNemar's exact binomial test (McNemar, 1947) to evaluate paired closed-loop outcomes. Let b denote episodes solved only by LogicVLA and c denote episodes solved only by the unregularized baseline. Under the null hypothesis, the number of discordant outcomes follows a binomial distribution with probability 0.5.

Table 5 shows that the aggregate improvement at 4K is statistically significant, whereas the 15K aggregate comparison is marginally above the conventional 0.05 threshold. The strongest statistical evidence appears on difficult tasks. At 15K, the hard-task comparison remains significant while the easy-task comparison shows no measurable difference. This pattern is consistent with Figure 2, where the performance advantage is concentrated on difficult behaviors rather than on already-saturated easy tasks.

E.4 Comparison with Prior VLA Methods

Table 6 places LogicVLA alongside representative VLA methods on LIBERO-Spatial. Paper-reported results and locally evaluated checkpoints are distinguished because differences in training configuration and evaluation procedure can affect absolute success rates. We therefore use the table for contextual positioning rather than direct cross-paper claims of statistical superiority.

The controlled comparison with the otherwise identical unregularized baseline in Table 1 remains the primary evidence for the effectiveness of LogicVLA. The broader comparison

Table 6: Comparison with representative methods on LIBERO-Spatial. Reported results may use different evaluation protocols and are included for contextual positioning.

Method	**Evaluation**	**Steps**	**Success**
OpenVLA (Kim et al., 2024)	Reported	–	84.7%
SpatialVLA (Qu et al., 2025)	Reported	–	88.2%
π_0-FAST (Pertsch et al., 2025)	Reported	–	96.4%
OpenVLA-OFT (Kim et al., 2025)	Reported	150K	97.6%
UniVLA-LAM (Bu et al., 2025)	Reported	–	96.5%
OpenVLA-OFT	Reproduced	150K	97.8%
UniVLA-LAM	Reproduced	–	97.0%
LogicVLA (ours)	Ours	15K	97.0%
LogicVLA (ours)	Ours	150K	99.0%

in Table 6 shows that the 15K checkpoint already lies within the performance range of strong VLA methods, while the 150K model reaches the saturated upper range of LIBERO-Spatial. Because the benchmark approaches saturation for strongly tuned policies, reduced-budget performance and difficult-task behavior provide more informative evidence than small absolute differences among full-budget checkpoints.