TRAC: TRAJECTORY-RELIABLE ALIGNMENT TRAINING FOR ACTION CHUNKS IN VLA MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action chunking predicts a sequence of future actions for execution between policy queries. Small action errors can accumulate and alter the intended motion, making relations across the sequence important for reliable manipulation. However, standard point-wise prediction losses sum penalties at individual action positions without explicitly evaluating cumulative deviations or coordinated variation. We introduce TRAC, Trajectory-Reliable Alignment Training for Action Chunks in VLA Models, which treats action chunks as structured action-space trajectories. TRAC complements flow matching with two objectives: weighted prefix consistency aligns accumulated translations with demonstrations throughout the chunk, while a local residual-structure prior encourages coordinated variation among neighboring actions. Both objectives operate on clean-action estimates, preserving the backbone's architecture and inference procedure. On LIBERO, TRAC achieves 96.55% success with SmolVLA, improving the original policy by 10.05 percentage points and a clean-action baseline by 3.40 points. Further evaluations show lower cumulative prediction errors and gains across chunk lengths, with additional improvements on LIBERO-Plus, in an adapted GR00T policy, and on four real-robot tasks.

1 INTRODUCTION

Vision-language-action (VLA) models connect visual and language representations to robot control. Recent systems, including OpenVLA (Kim et al., 2024), π_0 (Black et al., 2024), GR00T (NVIDIA et al., 2025), and SmolVLA (Shukor et al., 2025), generate actions conditioned on observations and instructions. Many continuous-action policies use action chunking: a single query predicts multiple future commands, and a sequence of these commands is executed before the next policy query. This reduces query frequency while placing part of the responsibility for the intervening motion on the quality of the predicted chunk.

An action chunk specifies an ordered segment of motion. Approaching an object, entering a slot, and completing a placement require commands that remain appropriate as that motion unfolds. For relative translations, errors in a common direction can accumulate across commands; opposing errors can conceal intermediate deviations when only the final displacement is considered. Neighboring actions also describe successive parts of the same movement, so their coordinated variation matters alongside their agreement with individual targets. These observations motivate treating a chunk as a structured action-space trajectory, whose quality depends on relations among its commands.

How should an action chunk be learned as a whole, beyond per-action accuracy?

Flow matching trains an action generator through squared velocity errors summed over positions and coordinates (Lipman et al., 2022; Black et al., 2024; Shukor et al., 2025). Although the action expert jointly models the chunk, a point-wise objective does not explicitly distinguish how prediction errors combine. Let $\mathbf{e}_h^{(A)}$ and $\mathbf{e}_h^{(B)}$ denote translation errors in two predicted chunks. Then

$$\sum_h \|\mathbf{e}_h^{(A)}\|_2^2 = \sum_h \|\mathbf{e}_h^{(B)}\|_2^2 \quad \not\Rightarrow \quad \sum_h \Big\| \sum_{i\le h} \mathbf{e}_i^{(A)} \Big\|_2^2 = \sum_h \Big\| \sum_{i\le h} \mathbf{e}_i^{(B)} \Big\|_2^2.$$

For example, $(\mathbf{u}, \mathbf{u})$ and $(\mathbf{u}, -\mathbf{u})$ have the same per-action squared error, but different accumulated errors. A training criterion that evaluates the chunk as a whole can distinguish these arrangements

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.

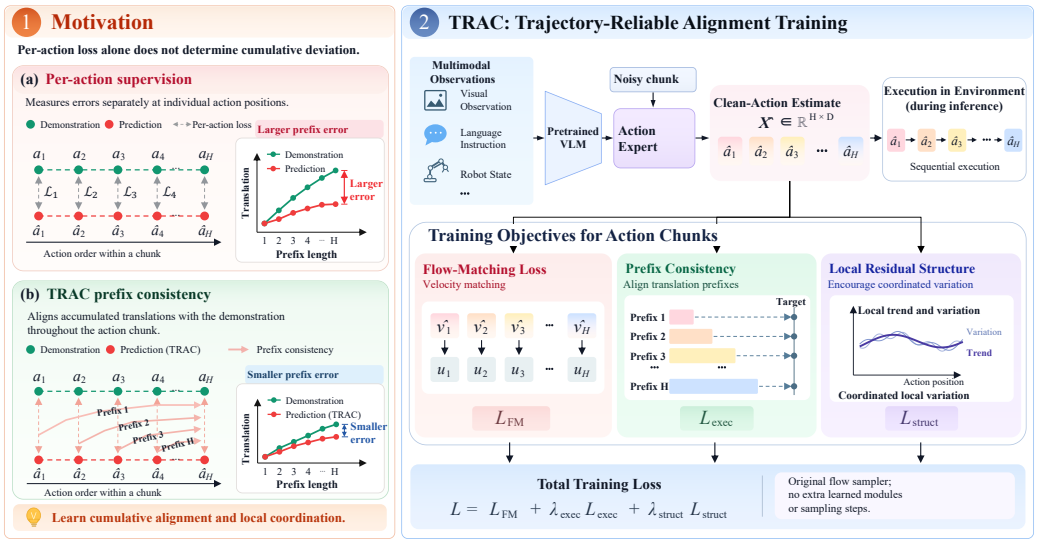


Figure 1: **Learning action chunks as structured trajectories.** Equal per-action errors can yield different cumulative discrepancies. TRAC supervises translation prefixes and coordinated local residual variation through two objectives on clean-action estimates. Both act only during training, retaining flow matching and the backbone's architecture and sampler.

and account for intermediate deviations, even when the final displacement agrees. Figure 1 illustrates this distinction and the complementary role of local coordination.

This perspective shifts attention from the error at each action position to the relations that give the sequence its motion-level meaning. Accumulated translations describe how relative commands compose over progressively longer parts of the chunk. Local variation describes how neighboring commands change together over shorter windows. Matching the accumulated quantities to a demonstration and expressing a prior on local coordination provide complementary training signals. The first specifies where the commanded translation should lead at each prefix; the second shapes how continuous action channels vary along the way.

We introduce TRAC, *Trajectory-Reliable Alignment Training for Action Chunks in VLA Models*, to align action-chunk learning with the motion formed by the predicted actions. A weighted prefix objective supervises accumulated translations throughout the chunk, so each action receives a training signal from every prefix to which it contributes. A complementary local residual-structure prior encourages coordinated variation among neighboring continuous actions around their underlying motion trends.

These objectives distinguish chunks with equal per-action loss through their cumulative discrepancies and coordinated local variation. Clean-action estimates provide a common domain for both objectives, while the backbone's architecture and inference procedure remain unchanged.

We evaluate TRAC through task execution and action-chunk prediction quality. On LIBERO, it improves SmolVLA success by 3.40 percentage points over a clean-action baseline and reduces held-out cumulative translation errors. The prefix objective benefits both output parameterizations, and the full method improves success across the tested chunk lengths. LIBERO-Plus, an adapted GR00T implementation, and four physical manipulation tasks further evaluate the practical value of this training formulation. All added computations are confined to training, preserving the model size and inference-step budget. Our contributions are:

- We formulate weighted prefix consistency for action-chunk learning, supervising how relative translation errors combine throughout a chunk and making intermediate discrepancies visible during training.
- We complement cumulative alignment with a local residual-structure prior that encourages coordinated variation among neighboring actions around their underlying motion trends.

- We evaluate this formulation through task success, cumulative prediction error, and execution horizons across two VLA implementations, with gains over a clean-action baseline and no added inference steps.

2 Related Work

VLA policies and action chunking. VLA models learn instruction-conditioned control from visual inputs and robot data (Kim et al., 2024; Black et al., 2024; NVIDIA et al., 2025; Shukor et al., 2025). Diffusion Policy (Chi et al., 2025) and flow matching (Lipman et al., 2022) support continuous action-sequence generation. Real-time chunking and continuation methods coordinate successive chunks (Black et al., 2026; 2025; Liu et al., 2026b;a). DAgger (Ross et al., 2011) addresses the state distribution encountered by a learned policy. TRAC focuses on the relations within a predicted action-space trajectory: the composition of relative commands and the coordinated variation of neighboring actions.

Action prediction and structure. Prediction space and loss space can be chosen separately in generative models (Li & He, 2026; Jin & Wang, 2026). Direct clean-action prediction has also been explored in VLA policies (Yang et al., 2026). Structured motion representations (Li et al., 2026; Nikulin & Tani, 2022) motivate modeling coordinated variation across action coordinates and time. We use clean-action estimates as a common domain for supervising accumulated translations and local action structure. Our local prior expresses this structure through soft constraints on residual variation around local action trends.

Trajectory-level supervision. Structured prediction has been used to compare demonstrated and reproduced probabilistic trajectories (Duan et al., 2024). Streaming Flow Policy treats action trajectories as flow trajectories and streams actions during generation (Jiang et al., 2025). Trajectory-Consistent Flow Matching supervises integration over segments of the generative flow (Ahmed et al., 2026). TRAC evaluates relations along action positions within a clean-action estimate. Its prefix objective compares accumulated translation commands with demonstrations, while its local prior acts on residual variation within predicted windows. The trajectory axis in these objectives is command order within the output, distinct from flow time during its generation.

3 Method

TRAC trains an action chunk as a structured action-space trajectory. We make two relations explicit: accumulated translations should agree with the demonstration throughout the chunk, and neighboring continuous actions should exhibit coordinated local variation. A clean-action estimate provides the common domain for evaluating these relations. The resulting training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{exec}}\mathcal{L}_{\mathrm{exec}} + \lambda_{\mathrm{struct}}\mathcal{L}_{\mathrm{struct}}. \tag{1}$$

Flow matching supervises the generative prediction, prefix consistency evaluates cumulative agreement, and the structural prior shapes local coordination. The nonnegative coefficients set the strength of the added objectives. All three terms use the same network evaluation during training; inference follows the backbone's original sampler.

3.1 Action-Space Formulation

Evaluating how commands compose and vary together requires an estimate in action space. Its translation channels define accumulated prefixes, and its neighboring continuous actions define local windows. We first establish this representation, then specify the two relational objectives. Let $\mathbf{X} = [\mathbf{a}_1^\top; \ldots; \mathbf{a}_H^\top] \in \mathbb{R}^{H\times D}$ be an expert action chunk in normalized coordinates, conditioned on multimodal observations $\mathbf{c}$. The index h denotes an action position and t denotes flow time. For $t \sim p_t$ with $0 \le t < 1$, define

$$\mathbf{Z}_t = (1-t)\boldsymbol{\epsilon} + t\mathbf{X}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \widehat{\mathbf{X}} = f_\theta(\mathbf{Z}_t, t, \mathbf{c}). \tag{2}$$

The network directly predicts the clean-action estimate. We retain flow matching by converting this estimate to velocity and applying velocity-space supervision (Li & He, 2026):

$$\hat{\mathbf{V}}_\theta = \tfrac{\hat{\mathbf{X}}-\mathbf{Z}_t}{1-t}, \quad \mathcal{L}_{\mathrm{FM}} = \mathbb{E}\tfrac{\|\hat{\mathbf{V}}_\theta - (\mathbf{X}-\boldsymbol{\epsilon})\|_F^2}{HD}. \tag{3}$$

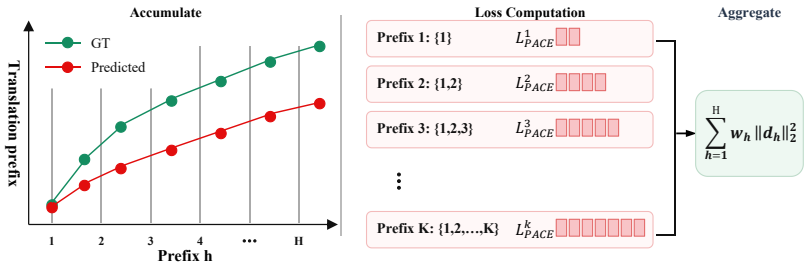


Figure 2: **Prefix consistency.** Predicted and demonstrated translations are accumulated at each prefix and compared in normalized action units (Eq. (4)). Equation (5) combines their squared discrepancies with prefix weights and a flow-time gate.

The induced velocity is used by the backbone's sampler. The estimate $\widehat{\mathbf{X}}$ also makes action relations directly available during training. A velocity head provides the same interface through $\widehat{\mathbf{X}} = \mathbf{Z}_t + (1-t)\mathbf{V}_\theta$; Section 4.6 evaluates both choices. We use the direct action head for the full method, with the same action expert and output width as the velocity baseline. Both added objectives are evaluated on $\widehat{\mathbf{X}}$ at the sampled flow time. Their gradients therefore shape the action estimate during denoising, while the induced velocity continues to define flow matching and sampling. Appendix A gives the loss identity and sampling details.

3.2 Prefix Consistency

A relative translation contributes to every subsequent prefix of the chunk. Its effect on cumulative agreement therefore depends on the other commands with which it is composed. Let $\mathcal{P}_{\mathrm{tr}}$ select the three translation coordinates. The translation prefix and its prediction discrepancy are

$$\mathbf{C}_h(\mathbf{X}) = \textstyle\sum_{i=1}^{h} \mathcal{P}_{\mathrm{tr}}(\mathbf{a}_i), \quad \mathbf{d}_h = \mathbf{C}_h(\widehat{\mathbf{X}}) - \mathbf{C}_h(\mathbf{X}). \tag{4}$$

We compare accumulated commands at every prefix, as illustrated in Figure 2, and minimize

$$\mathcal{L}_{\mathrm{exec}} = \mathbb{E}\left[c_{\mathrm{pre}} g(t) \textstyle\sum_{h=1}^{H} w_h \|\mathbf{d}_h\|_2^2\right], \qquad w_h > 0,\ \textstyle\sum_h w_h = 1. \tag{5}$$

Here $c_{\mathrm{pre}} > 0$ is a fixed normalization factor and $g(t)$ controls the supervision strength over flow time. We use increasing prefix weights to moderate the repeated inclusion of early commands; the experimental choice is specified in Appendix B.1. Rotation and gripper channels remain covered by flow matching. The prefix operator measures normalized translation-command discrepancy.

Interpretation of the prefix objective. Write $\mathbf{e}_i = \mathcal{P}_{\mathrm{tr}}(\widehat{\mathbf{a}}_i - \mathbf{a}_i)$. Expanding the squared accumulated errors in Eq. (5) gives

$$\textstyle\sum_h w_h \left\| \sum_{i \le h} \mathbf{e}_i \right\|_2^2 = \sum_{i,j} q_{ij} \mathbf{e}_i^\top \mathbf{e}_j, \quad q_{ij} = \sum_{h \ge \max(i,j)} w_h. \tag{6}$$

The cross terms arise directly from accumulation before squaring; they make the contribution of one error depend on the other errors in the same prefixes. Directionally aligned errors reinforce one another, while opposing errors may cancel. For the two-command example in the introduction, the prefix penalties are $(w_1 + 4w_2)\|\mathbf{u}\|^2$ and $w_1\|\mathbf{u}\|^2$, respectively. The second remains nonzero for $\mathbf{u} \neq \mathbf{0}$ because the intermediate prefix still deviates, despite zero final discrepancy. Each action consequently receives a training signal from all prefixes that contain it. This is the sense in which the objective evaluates the chunk jointly: learning responds to accumulated discrepancies across positions, in addition to the error at each position. Appendix B gives the gradient and quadratic-metric interpretation.

3.3 Local Residual Structure

Prefix consistency anchors accumulated translations to the demonstration. A complementary aspect of trajectory quality is how neighboring continuous actions vary together. Successive commands

describe a shared movement, and their channels can exhibit coordinated changes (Li et al., 2026; Nikulin & Tani, 2022). We express this structure through a soft local prior: a fitted trend accommodates the main movement, while a small number of residual modes capture coordinated departures from it.

This distinction matters because useful motion need not have small differences between adjacent commands. A steadily increasing translation command, for example, can change appreciably while following a simple local trend. Removing that trend lets the prior focus on departures from the main change. Likewise, translation and rotation channels may depart from their trends together. A residual mode represents such a shared pattern along the window and across channels, motivating more freedom for dominant modes than for the remaining variation.

For a predicted window $\widehat{\mathbf{X}}_j \in \mathbb{R}^{W \times D_c}$, let $\mathbf{P}$ be a local trend-fitting operator and $\mathbf{T}$ a temporal normalization operator. Define

$$\mathbf{R}_j = \mathbf{T}(\mathbf{I} - \mathbf{P})\widehat{\mathbf{X}}_j, \quad e_{\parallel,j} = \left(\frac{\sum_{i \le r} \sigma_i^2}{W D_c}\right)^{1/2}, \quad e_{\perp,j} = \left(\frac{\sum_{i > r} \sigma_i^2}{W D_c}\right)^{1/2}. \tag{7}$$

The singular values $\sigma_i = \sigma_i(\mathbf{R}_j)$ are ordered from largest to smallest. The two quantities measure residual magnitude in the dominant r modes and the remaining modes, respectively. Temporal normalization sets the scale at which departures from the trend are assessed. Concentrating residual variation in a few modes gives a compact description of how positions and channels change together within the window.

We impose a soft penalty on excess modal magnitude:

$$\ell_{\mathrm{str},j} = \omega(t) \sum_{k \in \{\parallel, \perp\}} \beta_k \rho([e_{k,j} - \delta_{k,j}(t)]_+), \quad \mathcal{L}_{\mathrm{struct}} = \mathbb{E}_{\mathcal{V}}[\ell_{\mathrm{str},j}]. \tag{8}$$

where $\mathcal{V}$ denotes valid windows, ρ is a robust penalty, and $\beta_\perp > \beta_\parallel \ge 0$ gives the dominant modes greater freedom. Nonnegative thresholds $\delta_{k,j}(t)$ adapt to stopped-gradient statistics of the predicted window. The weight $\omega(t)$ controls the prior's strength over flow time.

Variations within the thresholds incur no penalty; larger departures are constrained according to their modal magnitude. The thresholds adapt to the current prediction, allowing the admissible variation to change as the clean-action estimate develops during denoising. This expresses a preference for coordinated residual changes while retaining room for local motion variation. Flow matching and prefix consistency continue to anchor the prediction to the demonstrated commands.

The two added objectives consequently act on distinct relations in the same action-space trajectory. Prefix consistency compares the composition of predicted translations with that of the demonstration; the local prior shapes coordination within the prediction's continuous channels. Their combination links cumulative agreement with local structure without specifying a target curve for each window. Our instantiation uses $W = 9$, $D_c = 6$, and $r = 1$. Appendix C specifies the trend and temporal normalization operators, adaptive thresholds, and numerical settings. All these computations are confined to training.

4 Experiments

4.1 Experimental Setup

We evaluate task success on LIBERO (Liu et al., 2023), its perturbed extension LIBERO-Plus (Fei et al., 2025), and four real-robot tasks. Simulation results cover the Long, Goal, Object, and Spatial suites. SmolVLA in LeRobot v0.3.2 is the primary backbone. Its architecture, demonstrations, observation processing, optimization budget, and evaluation protocol are shared within each controlled comparison. Training follows the backbone's data processing and loss handling. Unless specified otherwise, the policy predicts and executes $H = 10$ actions before the next query, using ten flow integration steps.

The experiments address three questions: whether training on action relations improves task execution, how the resulting chunks compare in cumulative prediction quality and across execution horizons, and how the two objectives contribute. A clean-action baseline provides the common prediction and velocity-matching formulation for evaluating the added objectives. Velocity-head experiments further test prefix supervision on a different source of the action estimate. For GR00T,

the adapted clean-action baseline, prefix variant, and full method share their conditioning interface and flow-matching formulation. Published reference models provide benchmark context alongside these controlled comparisons.

The default training seed is 1000. Repeated configurations use seeds 1000, 2000, and 3000. Tables identify the number of seeds, and reported standard deviations use the population convention. GR00T experiments use StarVLA with a shared adapted conditioning interface, described in Section 4.7. Appendix D lists the software and model settings.

Table 1: **Benchmark success rates (%).** Reference rows retain published scores, including Avg.; LIBERO-Plus values follow Fei et al. (2025). Our Avg. averages the four suites; LIBERO SmolVLA rows use seed 1000. "Adapted clean" denotes the shared GR00T conditioning adaptation and clean-action head. Reference models use their respective training protocols.

Method	Long	Goal	Object	Spatial	Avg.
LIBERO: published references					
π_0 (Black et al., 2024)	73.0	95.0	86.0	90.0	86.0
GraspVLA (Deng et al., 2025)	82.0	91.2	94.1	89.1	89.1
NORA (Hung et al., 2025)	74.6	89.4	95.4	92.2	87.9
LIBERO: SmolVLA experiments					
Velocity baseline	72.00	89.00	98.00	87.00	86.50
Clean-action baseline	81.80	93.60	100.00	97.20	93.15
+ prefix	89.20	98.00	99.80	95.00	95.50
TRAC	91.00	97.80	99.80	97.60	**96.55**
LIBERO-Plus: published references					
NORA (Hung et al., 2025)	36.3	38.8	34.4	47.6	39.0
UniVLA (Bu et al., 2025)	39.9	40.7	36.7	55.5	42.9
π_0 (Black et al., 2024)	48.4	44.9	61.4	60.7	53.6
OpenVLA-OFT (Kim et al., 2025)	66.4	63.0	66.5	84.0	69.6
LIBERO-Plus: backbone experiments					
SmolVLA	46.53	35.89	66.20	32.85	45.37
$\mathrm{TRAC}_{\mathrm{SmolVLA}}$	49.27	55.27	72.36	54.04	**55.24**
GR00T, original	62.21	68.54	84.55	85.64	75.23
GR00T, adapted clean	69.00	82.36	88.36	88.09	81.95
+ prefix	77.61	77.27	89.56	88.34	83.20
$\mathrm{TRAC}_{\mathrm{GR00T}}$	79.75	83.17	89.48	90.30	**85.68**

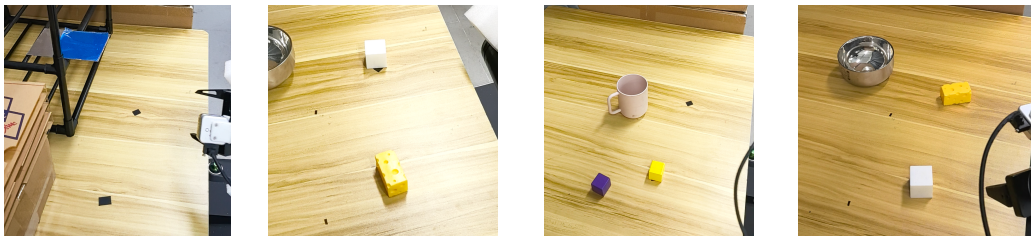
4.2 Benchmark Results

Table 1 evaluates the practical effect of training action chunks with cumulative alignment and local structure. On LIBERO, TRAC reaches 96.55% average success with SmolVLA, compared with 93.15% for the clean-action baseline. The largest suite-level gain is on LIBERO-Long, from 81.80% to 91.00%, with additional gains on Goal and Spatial. Repeated full-model runs yield $96.60 \pm 0.15\%$ over three training seeds.

The full training formulation improves success by 10.05 percentage points over the original velocity baseline while retaining the same architecture, demonstration data, and inference-step count. The clean-action comparison fixes the prediction parameterization and shows the additional value of supervising relations among commands. The prefix and component results are analyzed in Section 4.6. Published reference policies in the table place these results in the broader benchmark context.

4.3 Real-World Robot Experiments

We evaluate four manipulation tasks: plate insertion into a grid-box slot, placing a yellow cheese object on a white block, placing a small yellow block into a cup, and placing both the white block and yellow cheese into a bowl. Figure 3 shows their initial configurations. Each task has 50 teleoperated demonstrations shared by both policies. Evaluation uses the same robot control interface and ten trials per task under matched initial configurations.



(a) Plate insertion (b) Object-to-block (c) Block-to-cup (d) Two-object bowl

Figure 3: **Initial configurations of the four real-robot tasks.** See Section 4.3 and Appendix E.3 for evaluation settings and task instructions.

Table 2: **Real-robot task success.** Entries report successful trials / total trials, with SR (%) in parentheses. Each policy is evaluated in 40 trials.

Task	SmolVLA	TRAC
Plate insertion	6/10 (60)	8/10 (80)
Object-to-block placement	7/10 (70)	8/10 (80)
Block-to-cup placement	7/10 (70)	8/10 (80)
Two-object bowl placement	6/10 (60)	8/10 (80)
Overall	26/40 (65)	32/40 (80)

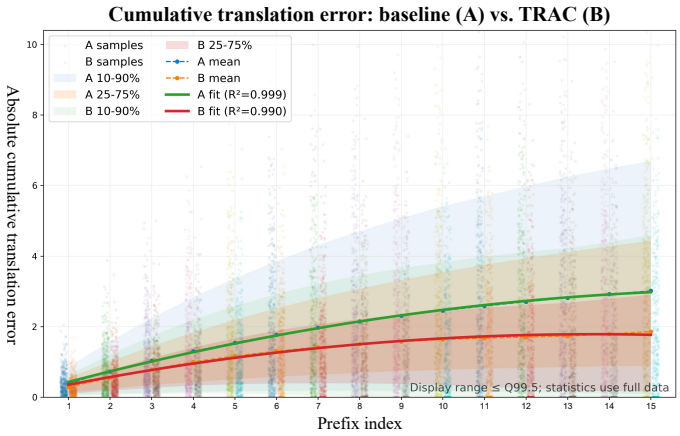


Figure 4: **Cumulative translation error across action prefixes.** A: velocity baseline; B: full TRAC. Absolute cumulative error is shown for the first translation channel on held-out chunks ($H = 15$), in normalized action units. Zero-based indices denote $h - 1$. Points: samples; bands: empirical percentile intervals; dashed curves: means; solid curves: fitted trends. Table 3 summarizes all three channels.

TRAC completes $32/40$ trials, compared with $26/40$ for SmolVLA (Table 2). Success improves on each task, covering insertion, placement, and a two-object manipulation sequence. These settings give the action-chunk objective a concrete role: the policy must produce commands that support successive approach and placement movements through the shared control interface. The results evaluate the practical effect of the training formulation through completed physical tasks. Appendix E.3 provides the task instructions.

4.4 Action-Chunk Prediction Quality

Task success evaluates the usefulness of the learned policy. We next examine the cumulative translation discrepancies that motivate training on action relations. We compare full TRAC with the original velocity baseline on 40,000 held-out LIBERO action chunks, using models trained with $H = 15$. Predictions are sampled through each policy's inference procedure and evaluated against demonstrated actions in normalized coordinates. Figure 4 shows the distribution of cumulative errors for the first translation channel across prefixes; Table 3 summarizes all three channels.

Table 3: **Prefix error in normalized action units.** For 40,000 held-out chunks with $H = 15$, each entry is the mean absolute value of one coordinate of $\mathbf{d}_h$ in Eq. (4). Indices count commands from one; smaller values are better.

	Translation 1		Translation 2		Translation 3	
h	Baseline	TRAC	Baseline	TRAC	Baseline	TRAC
1	0.393	0.287	0.272	0.204	0.325	0.237
6	1.768	1.306	1.230	0.925	1.461	1.081
11	2.585	1.674	1.804	1.197	2.154	1.368
15	3.014	1.860	2.111	1.331	2.542	1.543

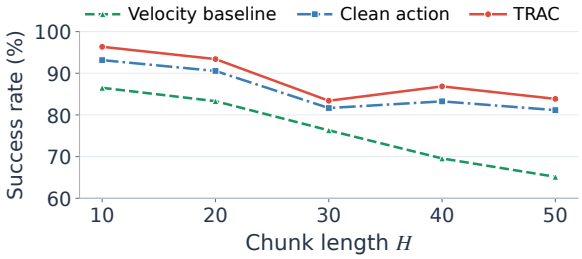


Figure 5: **Success across training and execution horizons.** Each configuration is trained and executed with the same horizon H. TRAC has the highest observed SR at every tested horizon.

TRAC has lower mean cumulative errors at every reported prefix and in each translation channel. At $h = 15$, the reductions are approximately 38%, 37%, and 39%, respectively. These measurements evaluate the translation obtained by composing successive predicted commands. Together with the task results, they show that the full training formulation improves both execution success and agreement with demonstrated accumulated translations. Appendix E.1 defines the prediction metrics.

4.5 Training and Execution Horizon

The chunk horizon sets how many predicted commands are executed before the next policy query. We train separate configurations for $H \in \{10, 20, 30, 40, 50\}$ and execute the corresponding number of commands, evaluating whether the formulation remains useful across different intervals between policy updates. Figure 5 compares velocity prediction, clean-action prediction, and full TRAC. TRAC achieves the highest observed success at every tested length, with gains of 1.75–3.60 points over clean-action prediction. At $H = 50$, it reaches 83.85% against 81.15% for clean-action prediction and 65.15% for the velocity baseline. Appendix E.2 provides all values.

4.6 Objective Analysis

Prefix supervision under two parameterizations. Table 4 evaluates the same prefix objective on clean-action estimates supplied by either output head. All rows retain velocity matching. Prefix consistency raises average success from 86.50% to 88.75% for the velocity head and from 93.15% to 95.50% for the clean-action head. Direct action prediction provides the stronger baseline in this setting, while both heads benefit from supervision of cumulative translation errors. The benefit in both cases is consistent with the distinction between choosing a prediction space and evaluating how predicted commands compose. A clean-action estimate makes this relation accessible from either head; the prefix objective determines how cumulative discrepancies contribute to training.

Local structure and prefix coverage. Table 5 reports the clean-action component results. The structure-only configuration has a three-seed mean of 93.50%, and full TRAC reaches 96.60%. In the matched seed-1000 comparison, adding the local prior to prefix supervision raises success from 95.50% to 96.55%. For seed 1000, the added structural prior improves Long and Spatial, while Goal and Object remain close to the prefix configuration. The structural prior alone gives a smaller

Table 4: **Parameterization and prefix-supervision ablation.** LIBERO SR (%), seed 1000. All rows retain velocity matching; the first column specifies the network output. Structure regularization is absent.

Output	Prefix	Long	Goal	Object	Spatial	Avg.
Velocity	No	72.0	89.0	98.0	87.0	86.50
Velocity	Yes	73.6	90.6	99.0	91.8	88.75
Clean action	No	81.8	93.6	100.0	97.2	93.15
Clean action	Yes	89.2	98.0	99.8	95.0	95.50

change in average success in this setting, whereas its addition to cumulative supervision improves the combined policy. This pattern supports using the local prior to complement target-based prefix alignment. The GR00T comparison in Section 4.7 also evaluates this combination.

Table 5: **Objective components under clean-action prediction.** LIBERO SR (%), averaged over n training seeds. Individual full-model runs and population standard deviations appear in Appendix E.5.

Objectives added	n	Long	Goal	Object	Spatial	Avg.
None	1	81.80	93.60	100.00	97.20	93.15
Prefix	1	89.20	98.00	99.80	95.00	95.50
Structure	3	83.73	95.40	99.87	95.00	93.50
Prefix + structure	3	91.00	97.73	99.87	97.80	**96.60**

A final-prefix training variant achieves 94.50%, compared with 95.50% for weighted prefix supervision and 93.15% without either addition. The ordering is consistent with the motivation for evaluating the course of accumulated translation throughout a chunk. Intermediate prefixes retain information about deviations that can cancel at the last command. Appendix E.4 gives the variant definitions and suite-level results; Appendix E.5 lists repeated runs.

4.7 Generalization and Backbone Transfer

On LIBERO-Plus, TRAC improves SmolVLA from 45.37% to 55.24% average success, with gains in all four suites. This evaluates the trained policy under variations in visual appearance, language, robot initialization, and object layout.

For StarVLA-GR00T, we use an adapted clean-action baseline with multi-layer vision-language conditioning. This interface is shared by the prefix and full-model variants in Table 1. The adapted baseline reaches 81.95%; adding prefix consistency gives 83.20%, and full TRAC reaches 85.68%. The combined gain is approximately 3.72 points using unrounded suite averages, with a 10.75-point gain on Long. The structural term adds 2.48 points to the prefix configuration. These results extend the objectives to a second VLA implementation under perturbed evaluation conditions.

5 Conclusion

An action chunk carries an ordered segment of motion, making the relations among its commands part of the learning problem. TRAC expresses this view through cumulative translation alignment and a local residual-structure prior. The prefix objective evaluates how translation errors combine throughout the chunk; the local prior shapes coordinated variation among neighboring continuous actions. Clean-action estimates provide their common domain while retaining flow matching and the backbone's original sampling procedure.

Experiments show gains over a clean-action baseline, lower cumulative prediction errors, and improvements across chunk lengths, benchmarks, and physical tasks. The benefit of prefix supervision with both output heads further supports evaluating action relations alongside individual predictions. These results show the practical value of training for the motion represented by a chunk within an existing policy architecture. Extending composition to other action representations and coordinating successive chunks are natural directions for this perspective.

REFERENCES

Riad Ahmed, Sujosh Nag, Moniruzzaman Akash, Mostafa Hussein, and Momotaz Begum. Trajectory-consistent flow matching for robust visuomotor policy learning, 2026. URL https://arxiv.org/abs/2605.08511.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. *arXiv preprint arXiv:2512.05964*, 2025.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

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, 2025. URL https://arxiv.org/abs/2505.06111.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Wenhao Zhang, Heming Cui, Zhizheng Zhang, and He Wang. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data, 2025. URL https://arxiv.org/abs/2505.03233.

Anqing Duan, Iason Batzianoulis, Raffaello Camoriano, Lorenzo Rosasco, Daniele Pucci, and Aude Billard. A structured prediction approach for robot imitation learning. *The International Journal of Robotics Research*, 43(2):113–133, 2024. doi: 10.1177/02783649231204656. URL https://doi.org/10.1177/02783649231204656.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. Nora: A small open-sourced generalist vision language action model for embodied tasks, 2025. URL https://arxiv.org/abs/2504.19854.

Sunshine Jiang, Xiaolin Fang, Nicholas Roy, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Siddharth Ancha. Streaming flow policy: Simplifying diffusion/flow-matching policies by treating action trajectories as flow trajectories. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 238–257. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/jiang25a.html.

Qing Jin and Chaoyang Wang. Revisiting diffusion model predictions through dimensionality. *ArXiv*, abs/2601.21419, 2026. URL https://api.semanticscholar.org/CorpusID:285140502.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36115–36125, June 2026.

Yiming Li, Nael Darwiche, Amirreza Razmjoo, Sichao Liu, Yilun Du, Auke Ijspeert, and Sylvain Calinon. Geometry-aware policy imitation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 117714–117737, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/bf235a1d6780afd979f2f81676f43413-Paper-Conference.pdf.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The eleventh international conference on learning representations*, 2022.

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*, 36:44776–44791, 2023.

Dongyang Liu, Zhaowen Zheng, Yu Sun, Longxu Zhang, Yixuan Liu, and Hao Wan. Action-prior denoising for smooth real-time chunking. *arXiv preprint arXiv:2605.25537*, 2026a.

Yufeng Liu, Hang Yu, Juntu Zhao, Bocheng Li, Di Zhang, Mingzhu Li, Wenxuan Wu, Yingdong Hu, Junyuan Xie, Junliang Guo, et al. Learning native continuation for action chunking flow policies. *arXiv preprint arXiv:2602.12978*, 2026b.

Vsevolod Nikulin and Jun Tani. Initialization of latent space coordinates via random linear projections for learning robotic sensory-motor sequences. *Frontiers in Neurorobotics*, Volume 16 - 2022, 2022. ISSN 1662-5218. doi: 10.3389/fnbot.2022.891031. URL https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2022.891031.

NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning, 2026. URL https://arxiv.org/abs/2602.11236.