WINDOW-LEVEL DIFFERENTIALLY PRIVATE FINE-TUNING OF VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting vision–language–action (VLA) models to local demonstrations creates a training-use privacy risk: generated actions can reveal whether a demonstration contributed to fine-tuning. Addressing this risk requires controlling annotation influence while preserving closed-loop behavior, where each action affects the observations and decisions that follow. We study this problem through window-level differential privacy (DP) for action annotations, with visual observations, instructions, and robot states held public and fixed. We introduce DP-VLA$_{\text{MC}}$, a fine-tuning recipe that couples tighter gradient clipping with noise-aware second-moment correction while retaining the VLA's native flow-matching objective. Clipping controls window contributions and the absolute Gaussian perturbation; moment correction adjusts AdamW's adaptive scaling for the known privacy-noise variance. Both changes preserve the privacy accounting of the baseline DP-VLA$_{\text{base}}$. To audit membership evidence beyond raw prediction accuracy, we complement Action-L1 with the Action-Reference Score (Action-Ref), which compares action errors against candidate-specific models trained without the annotation. Evaluating $\pi_{0.5}$ across 40 tasks in four LIBERO suites, with 800 rollouts per model, DP-VLA$_{\text{MC}}$ achieves higher overall task success than DP-VLA$_{\text{base}}$ at all seven tested privacy budgets. At $(\epsilon, \delta) = (10^9, 10^{-5})$, success increases from 9.25% to 46.25%, a gain of 37 percentage points and a fivefold improvement. Fixed-clipping comparisons reproduce the benefits of moment correction across two training seeds at $\epsilon = 10^8$ and 10^9. These results identify clipping and adaptive update scaling as consequential design choices for VLA fine-tuning under matched privacy accounting. They establish substantial optimization gains in the evaluated large-ϵ regime, while useful control with strong privacy remains an open challenge.

1 INTRODUCTION

Local demonstration fine-tuning allows VLA models to turn pretrained visual and language knowledge into robot skills tailored to a particular task or environment (Zitkovich et al., 2023; Kim et al., 2025; Physical Intelligence et al., 2025). Shared robot datasets and generalist policies provide reusable foundations for this adaptation (Open X-Embodiment Collaboration et al., 2023; Ghosh et al., 2024), while in-the-wild data collection and low-cost teleoperation make local demonstrations increasingly accessible (Khazatsky et al., 2024; Fu et al., 2024). The same adaptation process creates a privacy risk: the resulting model's action outputs can reveal whether a particular demonstration contributed to training (Peng et al., 2026). Thus, deploying an adapted policy can disclose a training-use relationship that the data owner intended to keep private.

Consider a hypothetical rehabilitation clinic that adapts an assistive robot from therapist-guided demonstrations. An observer who already knows a short action sequence associated with a patient may still seek to determine whether the clinic used that sequence for fine-tuning. This is a membership inference attack (MIA) (Shokri et al., 2017): the sensitive information is the use of the annotation, even when the candidate demonstration itself is known. We formalize this question as *window-level action-annotation membership*, protecting the inclusion of one short action segment while holding its visual observations, instruction, and robot state public and fixed. This unit isolates local annotation use; a guarantee for an entire patient's participation would require separate accounting.

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.

Prior work establishes that models can expose training information (Carlini et al., 2021), including membership signals in VLA action errors, temporal behavior, and internal attention (Peng et al., 2026; Luan et al., 2026). The corresponding defense problem is more demanding: the model must limit annotation influence and still act successfully. DP bounds how the distribution of the released model changes when an annotation is included or omitted (Dwork & Roth, 2014). Differentially private stochastic gradient descent (DP-SGD) implements this control by clipping individual gradients and adding Gaussian noise (Abadi et al., 2016), making optimization important even when fine-tuning pretrained models (Li et al., 2022). For a robot, prediction errors alter the states encountered next (Ross et al., 2011), so the relevant outcome is task completion through repeated observation and action. Our central question is: *How can VLA fine-tuning retain successful closed-loop control under a fixed action-window privacy budget?*

We address this question by examining the interaction between clipping and adaptive optimization. Clipping changes the contribution of each action window, while Gaussian privacy noise also introduces a positive bias into the second-moment estimate that AdamW uses to scale updates (Loshchilov & Hutter, 2019; Tang et al., 2024). Correcting this bias improves private Adam in prior work, but its extension to AdamW has also degraded classification performance (Chooi et al., 2025); its value for VLA control therefore requires direct evaluation. We develop DP-VLA$_{\text{MC}}$, which combines tighter clipping with moment correction (MC) and a positive variance floor. We adapt the correction from DP-AdamBC (Tang et al., 2024) to $\pi_{0.5}$ fine-tuning, retaining the recorded action supervision, native flow-matching (FM) objective, and action-learning randomness (Chi et al., 2023; Physical Intelligence et al., 2025). Tighter clipping changes window contributions and reduces absolute noise proportionally to sensitivity; MC adjusts how the privatized gradient is used. These changes preserve matched privacy accounting, allowing controlled comparisons to separate the optimizer's contribution from the choice of clipping bound.

Evaluating this adaptation also raises a measurement question: *How can action outputs reveal training-use information when prediction error also reflects a window's inherent predictability?* Our Action-L1 baseline builds on action-error inference (Peng et al., 2026), assigning higher membership scores to smaller normalized mean absolute action errors. Yet an easy non-member may have lower error than a difficult member. We therefore adapt candidate-specific difficulty calibration (Watson et al., 2022): Action-Ref subtracts the mean score of auxiliary *shadow models* trained without the candidate annotation from the target model's score. The resulting score measures relative fit against a non-member reference. This audit uses only action outputs from the target model and complements formal accounting with an empirical view of membership disclosure.

We evaluate $\pi_{0.5}$ on all 40 tasks in LIBERO Spatial, Object, Goal, and Long (Liu et al., 2023), with 800 rollouts per model and seven privacy budgets. The main DP-VLA$_{\text{MC}}$ recipe achieves higher overall task success at every tested budget than DP-VLA$_{\text{base}}$, which combines the DP-SGD mechanism with ordinary AdamW and a larger clipping bound. At cumulative budget $\epsilon = 10^9$ and additive slack $\delta = 10^{-5}$, success rises from 9.25% to 46.25%, a gain of 37 percentage points and a fivefold improvement. Holding clipping fixed at the same budget, MC adds 34.63 and 26.25 percentage points across two training seeds. The complete clipping–optimizer comparison at 10^8 also favors tighter clipping with MC in both seeds. Together, these controls show that the optimizer contributes substantially beyond the clipping change in the observed utility transition region.

These findings establish a substantial opportunity to improve VLA optimization under shared privacy accounting. The cumulative budgets tested, $\epsilon = 10^3$ to 10^{12} at $\delta = 10^{-5}$, provide weak formal protection, and improved task success does not uniformly reduce empirical membership leakage. The study identifies useful design choices in this regime and motivates the remaining challenge of achieving comparable control capability at small privacy budgets.

Our contributions are threefold: (1) **Window-level private VLA adaptation.** We formulate annotation-inclusion privacy with public contexts and align cumulative accounting, closed-loop task evaluation, and membership auditing around the same protected unit. (2) **Clipping and noise-aware optimization for control.** We develop DP-VLA$_{\text{MC}}$ by adapting moment correction to native VLA fine-tuning, and establish its utility contribution through fixed-clipping comparisons and repeated component ablations. (3) **Candidate-calibrated action auditing.** We adapt reference-based calibration to action-window membership through Action-Ref, separating relative fit from raw action error and evaluating both scores with matched target-model queries.

2 Related Work

Vision–language–action models. OpenVLA establishes an open pretrained model and efficient adaptation to new robot tasks (Kim et al., 2025), while $\pi_{0.5}$ learns continuous action chunks through flow matching (Physical Intelligence et al., 2025). LIBERO evaluates manipulation across distinct transfer settings (Liu et al., 2023). We retain native action learning and measure privacy interventions through closed-loop success on Spatial, Object, Goal, and Long.

Membership inference and VLA auditing. Auxiliary shadow models provide reference behavior for inferring whether a record contributed to training (Shokri et al., 2017). Difficulty calibration (Watson et al., 2022) and likelihood-ratio attacks (Carlini et al., 2022) incorporate record-dependent predictability into membership scoring. For VLAs, Peng et al. (2026) study likelihood, action-error, and temporal signals; VLALeaks uses attention discrepancies (Luan et al., 2026). Our Action-L1 and Action-Ref scores adapt action-error and difficulty-calibration ideas to window-level action-annotation membership with output-only access to the target model. The latter subtracts the candidate's mean reference score from the model's own score, combining output-only access with auxiliary training. Our setting isolates local annotation membership and measures empirical risk with a fixed attack protocol.

Differentially private fine-tuning and optimization. DP-SGD bounds individual gradients before adding Gaussian noise (Abadi et al., 2016); sampled-Gaussian accounting accumulates privacy cost across fine-tuning steps (Mironov et al., 2019). For language models, pretraining, aligned objectives, and DP-specific hyperparameters can substantially improve private fine-tuning (Li et al., 2022). Automatic Clipping addresses sensitivity to the clipping threshold (Bu et al., 2023). DP-AdamBC corrects the known privacy-noise contribution to Adam's second moment (Tang et al., 2024), while DP-AdamW studies this correction with decoupled weight decay and reports degradation on its classification benchmarks (Chooi et al., 2025). Together, these findings motivate examining clipping–optimizer interactions in VLA control. DP-VLA$_{\text{MC}}$ combines a smaller clipping bound with noise-aware second-moment correction for VLA fine-tuning; matched small-clip and large-clip controls separate their effects. Our privacy object is related to label DP (Ghazi et al., 2021): input contexts are public, and adjacency changes the presence of an action annotation.

Our position. Existing VLA membership studies characterize leakage through model behavior (Peng et al., 2026; Luan et al., 2026), while private fine-tuning research develops record-level optimization tools (Li et al., 2022; Tang et al., 2024). The key gap is understanding how these tools affect task execution when privacy is defined at the action-segment level. We address this gap through action-window privacy accounting, matched clipping–optimizer comparisons, closed-loop task evaluation, and fixed membership auditing.

3 Problem Formulation

3.1 Window-level annotation membership

We study how to preserve effective action learning while protecting the training-use information of individual *windows*, each a short continuous chunk of a demonstration. This information is defined by *window-level action-annotation membership*: whether a window's action annotation was included in fine-tuning. For candidate window i, let x_i contain its public visual observations, instruction, and robot state, and let a_i be the corresponding ground-truth action chunk. The entire chunk a_i forms one protected annotation. A VLA model π_θ, with parameters θ, predicts actions from x_i. Fine-tuning adapts a publicly available pretrained model using the available annotations.

To isolate the inclusion of an action annotation, we fix N candidate windows and their public contexts. A binary membership vector $m = (m_1, \ldots, m_N) \in \{0,1\}^N$ specifies the supervised dataset:

$$D_m = \{(x_i, y_i(m_i))\}_{i=1}^N, \qquad y_i(m_i) = \begin{cases} a_i, & m_i = 1, \\ \perp, & m_i = 0. \end{cases} \tag{1}$$

Here $m_i = 1$ makes the annotation available for supervision, whereas $m_i = 0$ marks it as absent, denoted by $\perp$. These membership labels describe whether each action annotation is available throughout fine-tuning.

3.2 PRIVACY–UTILITY OBJECTIVE

Protecting this membership requires controlling how the released model changes when one annotation is included. We call D_m and $D_{m'}$ adjacent when one annotation switches between a_i and $\perp$, with all contexts and other annotations fixed. A randomized training mechanism $\mathcal{M}$ satisfies (ϵ, δ)-DP when every adjacent pair and measurable output event E obey, in both directions,

$$\mathbb{P}[\mathcal{M}(D_m) \in E] \leq e^{\epsilon}\mathbb{P}[\mathcal{M}(D_{m'}) \in E] + \delta. \tag{2}$$

The output is the fine-tuned model; ϵ bounds distinguishability, and δ is additive slack (Dwork & Roth, 2014). The window-level privacy budget accounts for all updates in a single fine-tuning run. At fixed δ, a smaller ϵ provides a stronger privacy guarantee.

A useful private model must also complete tasks through interaction. We therefore measure utility by closed-loop task success rate (SR): the model repeatedly observes and acts, and a rollout succeeds when it achieves the task goal. Our objective is higher task success at the same privacy budget, accompanied by an empirical membership audit. The next section defines this audit; Section 5 develops the private training mechanism.

4 MEMBERSHIP AUDITING WITH ACTION ERRORS

To measure how annotation membership appears in a model's behavior, we examine its generated actions. We call the fine-tuned model π_θ under audit the target model. The attacker has output-only access to this model and knows a candidate (x_i, a_i), the pretrained model used for initialization, and the training recipe (Shokri et al., 2017; Peng et al., 2026). The context x_i is the model input; a_i is used afterward to score the prediction. We first measure agreement with the ground-truth actions (Action-L1), then introduce an auxiliary reference that accounts for candidate predictability (Action-Ref).

4.1 ACTION-L1: AGREEMENT WITH GROUND-TRUTH ACTIONS

Training on an annotation can improve the model's prediction of its actions, making prediction error a natural membership signal. We build on the negative action-error attack of Peng et al. (2026), who study both sample-level scores and trajectory-level aggregation, assigning each complete demonstration to either the member or the non-member set. Our adaptation scores the inclusion of a window's action annotation with its context fixed.

Let $\widehat{a}_{\theta,i}$ collect the repeated action predictions for candidate i, and let b be the vector of fixed public coordinate scales. For each prediction, we take the absolute difference from a_i and divide by b elementwise; the mean absolute error $e_i(\theta)$ averages these errors across valid window positions, physical action coordinates, and predictions. Action-L1 assigns the score

$$s_i^{\mathrm{L1}} = -e_i(\theta) = -\operatorname{mean}\left(\frac{|\widehat{a}_{\theta,i} - a_i|}{b}\right). \tag{3}$$

Larger scores mean closer agreement with the ground-truth actions and are interpreted as stronger membership evidence. Appendix B gives the exact error calculation and a complete example.

4.2 ACTION-REF: A CANDIDATE-SPECIFIC REFERENCE

Action error—the mean absolute difference between predicted and ground-truth actions—also reflects how predictable a window is. For example, consider a member window containing a precise gripper adjustment and a non-member window containing smooth arm motion. Training may improve prediction of the gripper adjustment, yet its error can remain higher than that of the simpler arm motion. We therefore compare each candidate with a reference built from models trained with its annotation absent, following difficulty-calibrated and shadow-model inference (Watson et al., 2022; Shokri et al., 2017; Carlini et al., 2022).

This reference comes from *shadow models*: non-private auxiliary models fine-tuned from the same pretrained model under attacker-chosen annotation memberships. For candidate i, let $\mathcal{R}_i$ index the shadow models trained without its annotation, and let $\theta^{(j)}$ denote shadow model j's parameters.

The attacker knows these auxiliary memberships, so the reference is available while the candidate annotation's membership in the target model's fine-tuning set remains unknown. The Action-Ref score is

$$s_i^{\text{ref}} = \frac{1}{|\mathcal{R}_i|} \sum_{j \in \mathcal{R}_i} e_i(\theta^{(j)}) - e_i(\theta). \tag{4}$$

It measures the model's improvement over the candidate's mean reference error.

Centering changes the question from how accurately the model predicts a window to how much better it predicts that window than its reference models. A small prediction error supplies weak evidence when reference models predict equally well; a larger improvement over the reference strengthens the signal. This comparison adjusts for candidate-dependent predictability through the auxiliary models. Appendix B.3 explains this separation, and Algorithm 1 gives the complete procedure.

Figure 1 shows receiver operating characteristic (ROC) curves: the member-detection rate (true positive rate, TPR) versus the non-member false-alarm rate (false positive rate, FPR). For the NP model on LIBERO-Long, Action-Ref attains an area under the curve (AUROC) of 0.720 compared with Action-L1's 0.541.

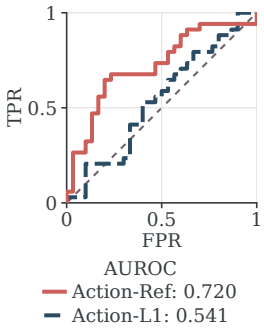


Figure 1: Non-private attack ROC curves on LIBERO-Long.

5 Differentially Private VLA Fine-Tuning

Membership auditing measures empirical leakage, while differentially private fine-tuning provides a formal privacy bound. We retain the native FM objective and its action-learning noise, and add privacy noise to gradients while preserving the recorded image–action pairs (Appendix C.1).

5.1 DP-VLA$_{\text{base}}$: DP-SGD for action windows

Bounding one annotation's influence begins with its gradient. We index training updates by t, writing θ_{t-1} and θ_t for the parameters before and after update t, respectively. For window i, the native loss ℓ_i gives the full trainable-parameter gradient vector $g_{t,i} = \nabla_\theta \ell_i(\theta_{t-1})$. For a public clipping bound $C > 0$, joint clipping across all trainable coordinates gives $\bar{g}_{t,i} = g_{t,i} \min\{1, C/\|g_{t,i}\|_2\}$. This caps each window's gradient norm at C while preserving its direction; zero gradients remain zero.

We next privatize the aggregate before it reaches the optimizer. Hidden Poisson sampling selects each of the N candidate windows independently with probability q, producing S_t. With fixed denominator $B = qN$, each step computes

$$\bar{g}_t = \frac{1}{B} \sum_{i \in S_t : m_i = 1} \bar{g}_{t,i}, \quad \widetilde{g}_t = \bar{g}_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \tau^2 I), \quad \tau = \frac{\sigma C}{B}. \tag{5}$$

Here $\bar{g}_t$ is the clipped aggregate and $\widetilde{g}_t$ the optimizer input after adding privacy noise η_t; σ is the noise multiplier, τ the per-coordinate noise standard deviation, and I the identity matrix. An absent annotation contributes zero; the fixed denominator keeps the influence bound at C/B. Fresh noise is added at every step, including empty-active batches. This is the DP-SGD mechanism (Abadi et al., 2016); the DP-VLA$_{\text{base}}$ baseline feeds $\widetilde{g}_t$ to AdamW (Loshchilov & Hutter, 2019). The resulting privacy perturbation also changes the quantities that AdamW uses to scale learning updates.

5.2 DP-VLA$_{\text{MC}}$: clipping and noise-aware moment correction

AdamW scales each coordinate by an estimate of its gradient's second moment, so privacy noise affects the update denominator as well as the gradient. For parameter coordinate c, write $[\bar{g}_t]_c$ for the corresponding component of the clipped aggregate. Averaging over fresh Gaussian noise while holding this aggregate fixed yields

$$\mathbb{E}_{\eta_t}\left[[\widetilde{g}_t]_c^2 \mid \bar{g}_t\right] = [\bar{g}_t]_c^2 + \tau^2. \tag{6}$$

The extra τ^2 enters the second-moment estimate and increases its denominator relative to the squared clipped signal. DP-VLA$_{\text{MC}}$ addresses this interaction through a smaller clipping bound and correction of the known noise contribution.

Smaller clipping changes the learning signal and absolute noise. A smaller C caps large window gradients more tightly. This changes the relative influence of windows and reduces the Gaussian scale $\tau = \sigma C/B$. The noise standard deviation and sensitivity bound decrease proportionally, keeping their ratio at σ and preserving matched accounting at fixed sampling, noise multiplier, and training length.

Moment correction changes coordinatewise scaling. Let $\widehat{v}_{t,c}$ be AdamW's second-moment estimate for coordinate c after correcting its zero-initialization bias. Following DP-AdamBC (Tang et al., 2024), we additionally subtract the known privacy-noise variance and impose a public positive floor:

$$d_{t,c}^{\text{base}} = \sqrt{\widehat{v}_{t,c}} + \gamma, \qquad d_{t,c}^{\text{MC}} = \sqrt{\max\{\widehat{v}_{t,c} - \tau^2, \rho\tau^2\}} + \gamma. \tag{7}$$

These are the ordinary and corrected update denominators; $\gamma > 0$ is AdamW's stabilizer, and $\rho > 0$ sets the variance floor. Both divide the first-moment estimate by their respective denominator. The subtraction targets the additive term in Equation (6), while the floor stabilizes coordinates near the noise level. The first moment retains its stochastic gradient input.

Together, these changes alter how windows contribute to the gradient and how the optimizer scales that privatized gradient. Ablation experiments in Section 6.4 assess the contributions of the clipping and optimizer changes. Full derivations and a worked example are provided in Appendices D and E.2; Algorithm 2 gives the complete training procedure.

5.3 Privacy guarantee

Both methods are calibrated to the same (ϵ, δ)-DP budget under the adjacency in Section 3. We fix the sampling rate, training length, and δ, and determine the noise multiplier using one shared accountant. Scaling the clipping bound and Gaussian noise together preserves the privacy bound; DP-VLA$_{\text{MC}}$'s moment correction is postprocessing of privatized gradients and public quantities. Appendices E.1 and E.3 give the assumptions, proof, and full accounting; Appendix E.4 reports the separate episode-level bound.

6 Experiments

6.1 Experiment Setup

Model and data. We jointly fine-tune the $\pi_{0.5}$ model on LIBERO Spatial, Object, Goal, and Long, covering ten tasks per suite (Physical Intelligence et al., 2025; Liu et al., 2023). Following Section 3, we organize the full dataset into non-overlapping action windows and fix their public contexts. An initial randomized membership split designates approximately half of the window action annotations as members available for supervision; the remainder are non-members. All compared experiments share this split across the complete task set.

Compared methods and privacy budgets. We compare an NP reference with DP-VLA$_{\text{base}}$ and DP-VLA$_{\text{MC}}$, the private fine-tuning methods in Section 5. All experiments start from the same pretrained model and use full-parameter fine-tuning with the native training objective. For DP-VLA$_{\text{base}}$, the larger clipping bound, denoted $C_L \simeq 2.06$, is the median gradient norm measured on public auxiliary windows at initialization (Appendix C.2), anchoring clipping to the pretrained model's gradient scale. For DP-VLA$_{\text{MC}}$, the smaller bound, denoted $C_S = 0.10$, limits individual-window influence more tightly. At a fixed training budget, we evaluate each DP method at privacy budgets $\epsilon \in \{10^3, 10^6, 10^7, 10^8, 10^9, 10^{10}, 10^{12}\}$, with $\delta = 10^{-5}$, under the shared accounting in Section 5.3. Each listed ϵ represents the cumulative window-level privacy budget across all fine-tuning steps. Appendix E.3.2 lists the associated noise multipliers and single-step privacy guarantees. We focus on closed-loop task capability over $\log_{10}\epsilon \in \{6, 7, 8, 9, 10\}$, with 10^3 and 10^{12} providing higher-noise and weak-noise diagnostic endpoints (Appendix F.2.1). Clipping–optimizer controls examine the contributions of smaller clipping and moment correction (Section 6.4).

Empirical privacy evaluation. We audit action-annotation membership with Action-L1 and Action-Ref, following Section 4. Both attacks share queries to the target model, and Action-Ref reuses a fixed set of shadow models across experiments. The audit uses 128 calibration windows and 256 test windows, each from a different episode, and covers all four suites. We report AUROC

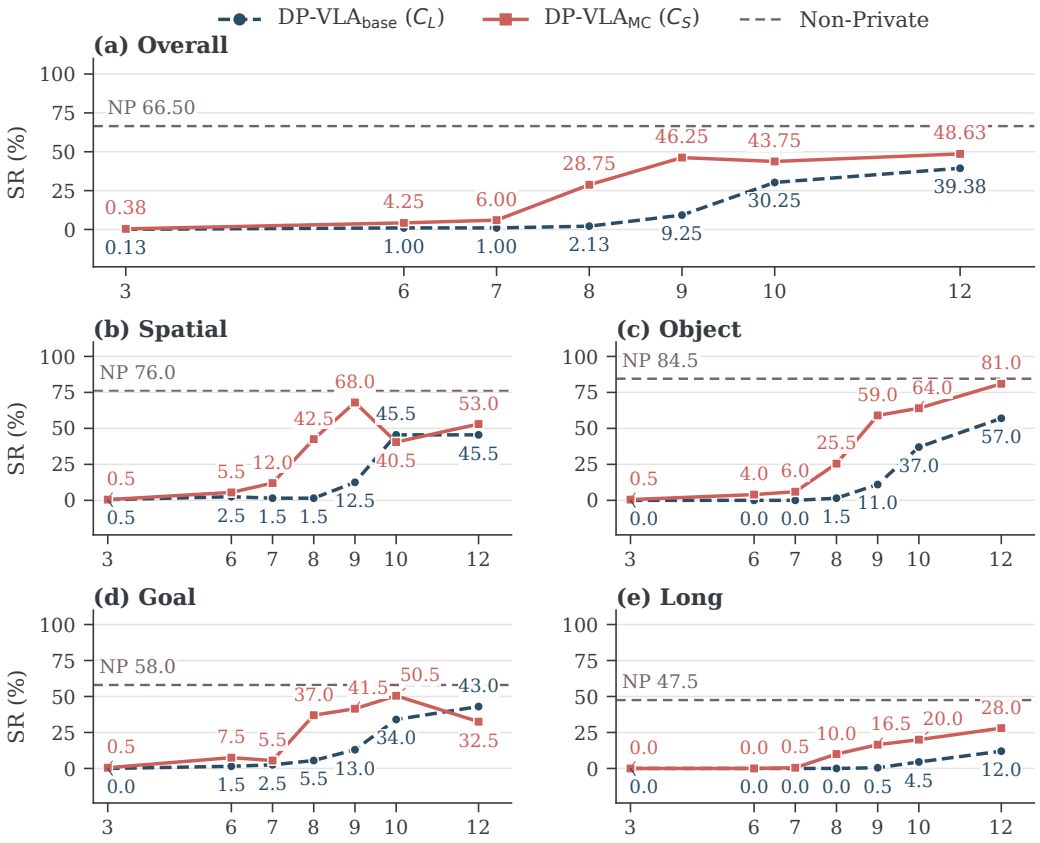


Figure 2: Task success across seven privacy budgets, with $\log_{10}(\epsilon)$ on the horizontal axis. Each point is labeled with success rate (SR) in percent: 800 rollouts overall and 200 per suite. Higher SR indicates better task performance. Dashed horizontal lines give the non-private fine-tuning reference; the two DP curves use $C_L \simeq 2.06$ and $C_S = 0.10$, respectively. Table 8 provides the complete numerical table.

for score ranking and balanced attack success rate (ASR) for equal-prior membership classification. ASR uses a threshold fixed on calibration data at a nominal 5% FPR; evaluation then uses held-out memberships.

Closed-loop evaluation. We evaluate each fine-tuned model on all 40 tasks with 20 fixed official initial states per task, totaling 800 rollouts. We report success rate (SR) for each suite and an overall equal-task average, following the utility objective in Section 3. Higher SR indicates better utility, while higher attack AUROC or ASR indicates stronger empirical membership leakage. Appendix F collects the detailed training, sampling, and evaluation configurations.

6.2 Task performance under different privacy budgets

DP-VLA$_{MC}$ achieves higher overall SR than DP-VLA$_{base}$ at all seven matched privacy budgets, with the largest gains in the utility transition region (Figure 2). At $\epsilon = 10^8$, SR rises from 2.13% to 28.75%; at 10^9, it rises from 9.25% to 46.25%, a fivefold increase and a gain of 37 percentage points (pp). The NP reference reaches 66.50% SR. These results address the objective in Section 3: preserving more task capability at the same privacy budget.

DP-VLA$_{MC}$ begins to recover task performance at tighter budgets. At 10^6 and 10^7, DP-VLA$_{base}$ remains at 1.00% SR, whereas DP-VLA$_{MC}$ reaches 4.25% and 6.00%, respectively, before rising to 28.75% at 10^8. At both 10^8 and 10^9, the gains extend across Spatial, Object, Goal, and Long. For

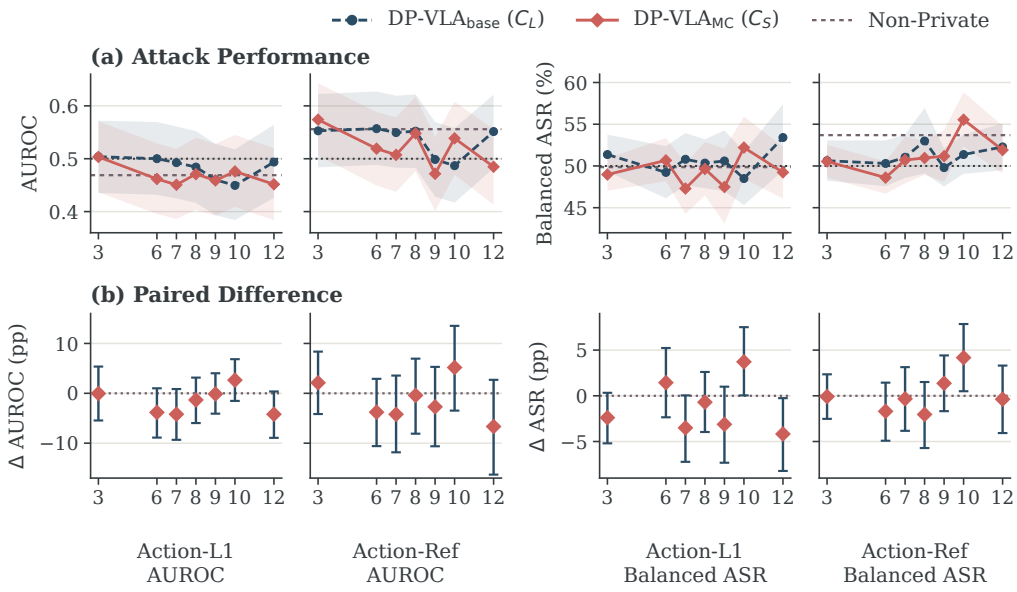


Figure 3: Membership auditing across seven privacy budgets. Both rows use $\log_{10}(\epsilon)$ on the horizontal axis. Upper panels report AUROC and balanced ASR; higher values indicate stronger attacks. Lower panels report matched-budget differences, DP-VLA$_{MC}$ (C_S) minus DP-VLA$_{base}$ (C_L), in pp. Lower differences favor DP-VLA$_{MC}$ more under same attacks. Upper dashed and dotted references mark the non-private model and chance; lower dotted lines mark zero difference. Shaded bands and error bars show pointwise 95% bootstrap intervals (Appendix F.5).

example, at 10^9, Spatial improves from 12.5% to 68.0%, while Long improves from 0.5% to 16.5%. Thus, improved control across all four suites contributes to the overall gains in the transition region.

Cross-budget comparisons within the utility transition region further demonstrate the value of these optimization choices: DP-VLA$_{MC}$ at 10^9 achieves 46.25% SR, exceeding DP-VLA$_{base}$ at 10^{10} (30.25%). Under the same window-level privacy definition and accounting (Section 5.3), the former achieves higher utility with a budget one order of magnitude smaller at these measured points. Together, smaller clipping and noise-aware optimization improve the measured privacy–utility relation. We next examine the empirical membership signals that accompany these gains.

6.3 Membership signals and corresponding relation to utility

Action-Ref's candidate-specific reference calibration improves membership discrimination on the NP test set, revealing signals that raw action errors can obscure. Action-Ref increases pooled AUROC from Action-L1's 0.469 to 0.556 and balanced ASR from 49.9% to 53.7% (Table 11). Spatial illustrates the distinction: members have larger final action errors but greater improvement relative to their reference models; reference differencing raises AUROC from 0.386 to 0.581. As expressed in Equation (4), relative fit provides membership evidence while adjusting for candidate predictability; Appendix B.3 gives the full decomposition.

Paired audits of DP models locate the relationship between utility and empirical leakage (Figure 3). At 10^8, SR increases from 2.13% to 28.75%, while both attacks' AUROC and ASR point estimates decrease; Action-Ref AUROC changes from 0.552 to 0.548 and ASR from 53.0% to 51.0%. At 10^{10}, SR improves from 30.25% to 43.75%, while Action-Ref AUROC rises from 0.487 to 0.539. These comparisons show distinct combinations of task capability and membership discrimination across budgets.

Candidate references and a fixed attack protocol provide a comparable membership audit alongside task success. The formal budget quantifies the training mechanism's privacy bound, while AUROC and ASR describe the membership evidence available to the fixed observer. Together with task SR,

Table 1: Component ablation across privacy budgets and training seeds. **Components.** ✔ marks the selected clipping bound or enabled MC; each row selects exactly one of C_L, C_S. In the MC column, $\times$ denotes DP-VLA$_{\text{base}}$ and ✔ denotes DP-VLA$_{\text{MC}}$ (moment correction with its variance floor). **Results.** Each cell is SR (%) / Action-Ref AUROC, using 800 rollouts and 256 held-out windows, respectively. Bold marks higher SR within each fixed-clipping pair and seed; higher AUROC means a stronger attack.

	Components			Task SR (%) / Action-Ref AUROC	
ϵ	$C_L \simeq 2.06$	$C_S = 0.10$	MC	Seed 17 (main)	Seed 117
10^8	✔	$\times$	$\times$	2.13 / 0.552	2.75 / 0.553
	✔	$\times$	✔	**13.38** / 0.542	**12.88** / 0.488
	$\times$	✔	$\times$	4.50 / 0.524	5.63 / 0.513
	$\times$	✔	✔	**28.75** / 0.548	**17.50** / 0.485
10^9	$\times$	✔	$\times$	11.63 / 0.497	11.00 / 0.475
	$\times$	✔	✔	**46.25** / 0.472	**37.25** / 0.512

these quantities provide complementary views of the evaluated models. We next use component ablations to identify which optimization choices contribute to the utility gains.

6.4 Clipping and moment correction

We focus on $\epsilon = 10^8$ and 10^9, where utility improves rapidly, to examine clipping and moment correction (Table 1). Gaussian noise realizations vary with the training seed, so we repeat the ablations with seeds 17 and 117 to assess sensitivity to stochastic training; seed 17 is used for the main experiments. At 10^8, reducing C_L to C_S with ordinary AdamW improves SR by 2.38 and 2.88 pp, respectively. This change adjusts relative window contributions and reduces the Gaussian standard deviation by a factor of 20.6 at the same privacy budget (Section 5.2), demonstrating a small utility benefit from tighter clipping.

At fixed clipping, MC provides further utility gains. At 10^8 with C_S, SR improves by 24.25 and 11.88 pp across the two seeds; with C_L, the gains are 11.25 and 10.13 pp. These pairs match the clipping bound and Gaussian noise scale, attributing the extra gains to the corrected optimizer recipe, including its variance floor and the coordinatewise scaling in Equation (7).

The same-clipping MC gains also repeat at 10^9: SR improves by 34.63 and 26.25 pp. Together with the complete 10^8 component comparison, where C_S plus MC achieves the highest SR in both seeds, these results support complementary roles for clipping and correction. Clipping adjusts window contributions and absolute noise, while MC improves how privatized gradients are used, helping explain the transition-region utility gains in Section 6.2.

7 Conclusion

This paper systematically studies window-level action-annotation membership privacy in VLA fine-tuning, bringing together formal privacy accounting, membership auditing, and closed-loop task evaluation. We use Action-L1 and Action-Ref to measure membership signals in model action outputs, with candidate-specific reference calibration improving discrimination on the non-private model. For private fine-tuning, DP-VLA$_{\text{MC}}$ combines a smaller clipping bound with correction of the privacy-noise contribution to the second moment, improving the use of privatized gradients. In experiments with $\pi_{0.5}$ across all four LIBERO suites, DP-VLA$_{\text{MC}}$ achieves higher overall success than DP-VLA$_{\text{base}}$ at all seven matched privacy budgets. At $\epsilon = 10^9$ and $\delta = 10^{-5}$, success increases from 9.25% to 46.25%. Component comparisons and repeated experiments in the utility transition region further support the complementary roles of smaller clipping and moment correction. These results show that jointly designing gradient clipping and noise-aware optimization can preserve more robot task capability under the same privacy definition and accounting, providing a concrete path toward improving the privacy–utility trade-off in VLA fine-tuning.

AI USE STATEMENT

We used OpenAI's ChatGPT to polish the language and improve the clarity of the manuscript. ChatGPT also assisted in drafting the step-by-step calculations for several illustrative examples in Appendix. We have not used generative AI tools for the other tasks. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016. doi: 10.1145/2976749.2978318. URL https://arxiv.org/abs/1607.00133.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8249b30d877c91611fd8c7aa6ac2b5fe-Abstract-Conference.html.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds, 2016. URL https://arxiv.org/abs/1605.02065.

Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy, 2020. URL https://arxiv.org/abs/2004.00010.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, pp. 2633–2650. USENIX Association, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy*, 2022. doi: 10.1109/SP46214.2022.9833649. URL https://arxiv.org/abs/2112.03570.

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

Jay Chooi, Kevin Cong, Russell Li, and Lillian Sun. DP-AdamW: Investigating decoupled weight decay and bias correction in private deep learning, 2025. URL https://arxiv.org/abs/2511.07843. Presented at the ICML 2025 DIG-BUGS Workshop.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042. URL https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning*, 2024. URL https://mobile-aloha.github.io/.

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/e3a54649aeec04cf1c13907bc6c5c8aa-Abstract.html.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*,

2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset, 2024. URL https://arxiv.org/abs/2403.12945.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R 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. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.05679.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Xukun Luan, Jinyan Liu, Xuesong Li, Yuanguo Bi, Renjun Wu, Zhongxiang Lei, and Di Wang. VLALeaks: Membership inference attacks against vision-language-action models, 2026. URL https://arxiv.org/abs/2606.15165.

Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium*, pp. 263–275, 2017. URL https://arxiv.org/abs/1702.07476.

Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism, 2019. URL https://arxiv.org/abs/1908.10530.

Open X-Embodiment Collaboration et al. Open X-Embodiment: Robotic learning datasets and RT-X models, 2023. URL https://arxiv.org/abs/2310.08864.

Yuefeng Peng, Mingzhe Li, Kejing Xia, Renhao Zhang, and Amir Houmansadr. Membership inference attacks on vision-language-action models, 2026. URL https://arxiv.org/abs/2605.07088.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Stephane 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*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017. URL https://arxiv.org/abs/1610.05820.

Qiaoyue Tang, Frederick Shpilevskiy, and Mathias Lécuyer. DP-AdamBC: Your DP-Adam is actually DP-SGD (unless you apply bias correction). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15276–15283, 2024. doi: 10.1609/aaai.v38i14.29451. URL https://ojs.aaai.org/index.php/AAAI/article/view/29451.

Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.08440.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A PSEUDOCODE FOR ATTACK AND DP AUDITING

Algorithm 1 specifies membership scoring and evaluation; Algorithm 2 specifies private fine-tuning and its privacy accounting. In Algorithm 1, $\mathcal{I}_{\text{cal}}$ and $\mathcal{I}_{\text{test}}$ are non-overlapping calibration and test index sets; $m^{(j)}$ is the attacker-chosen membership vector and $\theta^{(j)}$ the fitted parameters of shadow model j. NPFIT uses the fixed auxiliary training recipe in Appendix F.4; the resulting shadow models and reference errors are reused across audited models. In the training algorithm, T denotes the number of updates, p denotes the number of trainable coordinates, and I_p is the $p \times p$ identity matrix. The indices i, j, t, c denote windows, shadow models, training updates, and trainable parameter coordinates, respectively. The attack notation θ refers to the audited model's final parameters θ_T, with θ_0 denoting its pretrained initialization.

Algorithm 1 Action-L1 and Action-Ref membership auditing

Require: Output access to π_θ; candidates $\{(x_i, a_i)\}_{i=1}^N$; $\mathcal{I}_{\text{cal}}, \mathcal{I}_{\text{test}}$; calibration labels $\{m_i : i \in \mathcal{I}_{\text{cal}}\}$; pretrained model parameters θ_0, normalization, and fixed query seeds.

Ensure: Scores, frozen membership predictions, and held-out metrics for both attacks.

$J \leftarrow 16,\ K \leftarrow 4,\ \zeta \leftarrow 0.05$.
Choose $m^{(1)}, \ldots, m^{(J)} \in \{0,1\}^N$ with $\sum_{j=1}^J m_i^{(j)} = J/2$ for every i.
for $j = 1, \ldots, J$ **do**
  $\theta^{(j)} \leftarrow \text{NPFIT}(\theta_0, D_{m^{(j)}})$.
end for
for $i \in \mathcal{I}_{\text{cal}} \cup \mathcal{I}_{\text{test}}$ **do**
  Compute $e_i(\theta)$ and $\{e_i(\theta^{(j)})\}_{j=1}^J$ by Equation (9), using K matched draws.
  $\mathcal{R}_i \leftarrow \{j : m_i^{(j)} = 0\}$.
  $s_i^{\text{L1}} \leftarrow -e_i(\theta)$.
  $s_i^{\text{ref}} \leftarrow |\mathcal{R}_i|^{-1} \sum_{j \in \mathcal{R}_i} e_i(\theta^{(j)}) - e_i(\theta)$.
end for
$\mathcal{I}_{\text{cal}}^0 \leftarrow \{i \in \mathcal{I}_{\text{cal}} : m_i = 0\}$; $n_0 \leftarrow |\mathcal{I}_{\text{cal}}^0|$; $k_\zeta \leftarrow \lfloor \zeta n_0 \rfloor$.
for score family $r \in \{\text{L1}, \text{ref}\}$ **do**
  Sort $\{s_i^r : i \in \mathcal{I}_{\text{cal}}^0\}$ as $s_{(1)}^r \geq \cdots \geq s_{(n_0)}^r$.
  $\kappa_\zeta^r \leftarrow s_{(k_\zeta+1)}^r$.
  $\widehat{m}_i^r \leftarrow \mathbf{1}[s_i^r > \kappa_\zeta^r]$, for all $i \in \mathcal{I}_{\text{test}}$.
  Freeze κ_ζ^r and $\{(s_i^r, \widehat{m}_i^r) : i \in \mathcal{I}_{\text{test}}\}$.
end for
Reveal $\{m_i : i \in \mathcal{I}_{\text{test}}\}$.
For each r, report AUROC from (s_i^r, m_i) and ASR/TPR/FPR from $(\widehat{m}_i^r, m_i)$, $i \in \mathcal{I}_{\text{test}}$ (Equations (32) and (35)).

Algorithm 2 Window-private fine-tuning: DP-VLA$_{\text{base}}$ and DP-VLA$_{\text{MC}}$

Require: Fixed contexts, private annotations/mask, pretrained model parameters θ_0; public $q, C, T, \delta, \epsilon$ and optimizer parameters.

1: Solve for feasible $\sigma > 0$ with privacy accounting over all T updates.
2: Set $B = qN$, $\tau = \sigma C/B$; initialize moments to zero.
3: **for** $t = 1, \ldots, T$ **do**
4: Independently sample each candidate window with probability q; initialize the clipped-gradient sum $G = 0$.
5: **for** sampled i with $m_i = 1$ **do**
6: Compute native loss by Equation (14) and joint clipped gradient $\bar{g}_{t,i}$ as defined in Section 5.1.
7: $G \leftarrow G + \bar{g}_{t,i}$.
8: **end for**
9: Draw fresh $Z_t \sim \mathcal{N}(0, \sigma^2 C^2 I_p)$, where $Z_t = B\eta_t$; set $\widetilde{g}_t = (G + Z_t)/B$, also on empty steps.
10: Update moments by Equations (18) and (19).
11: Select the DP-VLA$_{\text{base}}$ or DP-VLA$_{\text{MC}}$ denominator in Equation (7); update parameters by Equation (20).
12: **end for**
13: Recompute accounting from the actual step count and retain the final parameters θ_T.

Ensure: Release the final model under the stated mechanism budget.

B Action-Based Membership Auditing

B.1 Action-error computation

We inherit the negative action-error signal of Peng et al. (2026): for ground-truth actions a and generated actions $\widehat{a}$, their Action-L1 score is $s = -\|\widehat{a} - a\|_1$. That study reports sample-level attacks and trajectory-level aggregation using separate member and non-member sets of complete demonstrations. Here the membership unit is a window action annotation, and the score uses public coordinate scaling, valid window positions, and repeated action predictions.

Let H be the action horizon and d_a the number of physical action coordinates. For candidate i, draw K action chunks using independent inference randomness ω_k:

$$\widehat{a}^{(k)}_{\theta,i} = \pi_\theta(x_i; \omega_k) \in \mathbb{R}^{H \times d_a}. \tag{8}$$

Let V_i contain the valid timesteps and $b_d > 0$ be the fixed public scale for coordinate d. The shorthand $\widehat{a}_{\theta,i} \in \mathbb{R}^{K \times H \times d_a}$ in Equation (3) stacks these predictions; each is compared with the same action chunk a_i, using b_d at every timestep. The public scale vector is $b = (b_1, \ldots, b_{d_a})$; the mean error in Equation (3) expands as

$$e_i(\theta) = \frac{1}{K|V_i|d_a} \sum_{k=1}^{K} \sum_{h \in V_i} \sum_{d=1}^{d_a} \frac{|\widehat{a}^{(k)}_{\theta,i,h,d} - a_{i,h,d}|}{b_d}. \tag{9}$$

The sums average over action draws, valid timesteps, and physical coordinates; the public scales make coordinate magnitudes comparable. We use $K = 4$, $d_a = 7$, and

$$b_d = \frac{Q_{.99,d} - Q_{.01,d} + 10^{-6}}{2}, \tag{10}$$

where $Q_{.01,d}$ and $Q_{.99,d}$ are the fixed public action quantiles for coordinate d; the additive constant stabilizes narrow ranges. Inference follows the matched-query protocol in Appendix F.4. The same $e_i(\cdot)$ evaluates target-model parameters θ and shadow model parameters $\theta^{(j)}$.

AUROC is invariant to a common positive rescaling of all candidate scores. Coordinate scaling and stochastic averaging define the ranking of the adapted score.

B.2 A worked membership-auditing example

A two-timestep, two-coordinate illustration makes each score calculation explicit. Suppose the public scaling factors are $w_1 = 1, w_2 = 2$, where $w_d = 1/b_d$ is the inverse scale from Equation (10).

The candidate action chunk and four outputs from the target model are

$$a = \begin{pmatrix} .20 & .50 \\ .40 & .10 \end{pmatrix}, \qquad \widehat{a}^{(1)} = \begin{pmatrix} .30 & .55 \\ .30 & .15 \end{pmatrix}, \qquad \widehat{a}^{(2)} = \begin{pmatrix} .40 & .60 \\ .20 & .20 \end{pmatrix},$$
$$\widehat{a}^{(3)} = \begin{pmatrix} .25 & .525 \\ .35 & .125 \end{pmatrix}, \qquad \widehat{a}^{(4)} = \widehat{a}^{(1)}.$$

For the first draw, the four scaled absolute errors are $1(.10), 2(.05), 1(.10), 2(.05)$. Their mean is $e^{(1)} = .10$. The corresponding means for draws two through four are $.20, .05, .10$. Thus

$$e_i(\theta) = (.10 + .20 + .05 + .10)/4 = .1125, \qquad s_i^{\mathrm{L1}} = -.1125. \tag{11}$$

The model has produced a moderately accurate action prediction. Action-L1 uses this absolute accuracy as its entire scalar evidence.

We use the fixed shadow models trained under attacker-chosen, known memberships. Whether the candidate annotation was used to fine-tune the target model remains unknown. For this candidate, suppose the eight shadow models trained without its annotation produce mean errors

$$(.25,\ .30,\ .35,\ .30,\ .20,\ .40,\ .25,\ .35).$$

Each entry is computed by the same four-draw Action-L1 procedure. The mean reference error and Action-Ref score are

$$\frac{1}{|\mathcal{R}_i|} \sum_{j \in \mathcal{R}_i} e_i(\theta^{(j)}) = 2.40/8 = .30, \qquad s_i^{\mathrm{ref}} = .30 - .1125 = .1875. \tag{12}$$

The target model predicts the candidate more accurately than its reference. Action-Ref uses these eight candidate-specific reference models.

Calibrate before looking at the candidate's true membership. For a small illustrative calibration set with $n_0 = 20$ candidates known to be non-members of the target model's fine-tuning set and nominal FPR $\zeta = .05$, at most $k_\zeta = \lfloor .05(20) \rfloor = 1$ false positive is permitted. Suppose the top two Action-L1 calibration scores are $-.04, -.08$, with all others at most $-.09$. Its decision rule predicts membership for scores strictly above $-.08$. For Action-Ref, suppose the top two calibration scores are $.22, .15$, with all others at most $.14$. Its decision rule predicts membership for scores strictly above $.15$. Both thresholds are fixed from their own scores on the same calibration candidates.

Action-L1 predicts this candidate to be a non-member because $-.1125 < -.08$, whereas Action-Ref predicts a member because $.1875 > .15$. The candidate annotation's membership is revealed only after these predictions have been fixed. If it is a member, the former is a false negative and the latter is a true positive; if it is a non-member, the latter is a false positive.

From predictions to evaluation metrics. On a separate toy held-out set containing 100 members and 100 non-members, suppose 30 members and five non-members exceed a frozen threshold. Applying Equation (35) gives

$$\mathrm{TPR} = .30, \qquad \mathrm{FPR} = .05, \qquad \mathrm{ASR} = (.30 + .95)/2 = .625.$$

Equal class priors give 62.5% balanced membership-classification success. An observer that predicts every candidate to be a non-member obtains 50% balanced ASR. The complete metric definitions and calibration rule are given in Appendix F.5.

B.3 Candidate difficulty and reference calibration

Absolute accuracy and relative fit. Candidate difficulty refers to how accurately a model can predict a window's ground-truth actions from its context. For example, steady free-space motion can be easier to predict than the timing and direction of a contact-rich adjustment. Such variation affects both members and non-members. Membership is assigned by an annotation mask; the finite evaluation sample can retain a difficulty imbalance between its two classes. Training can improve a difficult member while its final error remains larger than that of an easy non-member. Raw action error compares final errors across candidates, whereas a candidate-specific reference compares each candidate against auxiliary models. This distinction motivates difficulty calibration in membership inference (Watson et al., 2022; Carlini et al., 2022).

Write $\bar{e}_i^{\mathrm{ref}} = |\mathcal{R}_i|^{-1} \sum_{j \in \mathcal{R}_i} e_i(\theta^{(j)})$ for the mean reference error. The two scores satisfy

$$s_i^{\mathrm{L1}} = -e_i(\theta) = s_i^{\mathrm{ref}} - \bar{e}_i^{\mathrm{ref}}. \tag{13}$$

Here $\bar{e}_i^{\mathrm{ref}}$ estimates reference-relative difficulty, while s_i^{ref} measures the target model's relative fit. The identity shows how a larger baseline difficulty can mask an increase in relative fit in Action-L1. The reference captures candidate difficulty to the extent that the auxiliary models reproduce the target model's modeling conditions.

A hypothetical two-window example. Suppose a member window contains a precise gripper adjustment and a non-member window contains smooth free-space motion. The following illustrative errors use the same public action scale:

Candidate	Reference error $\bar{e}_i^{\mathrm{ref}}$	Target-model error $e_i(\theta)$	Action-L1	Action-Ref
Difficult member	0.40	0.24	-0.24	0.16
Easy non-member	0.05	0.04	-0.04	0.01

Action-L1 ranks the easy non-member above the difficult member because $-0.04 > -0.24$. Action-Ref reverses that ordering because $0.16 > 0.01$: the target model's advantage over the reference is larger for the member. These hypothetical numbers isolate the distinction between absolute error and relative fit.

B.4 Non-private results across LIBERO suites

Observed NP windows. In the Spatial test subset, members have mean reference error 0.2815, versus 0.2209 for non-members. The fine-tuned NP model has mean action errors of 0.2385 and 0.1919 for the two groups, respectively. Its mean advantage over the reference is therefore larger for members (0.0431 versus 0.0290), yet their absolute errors remain higher. This pattern is consistent with candidate difficulty masking the raw membership signal. Difficulty here is relative to the reference models: the pretrained model before fine-tuning has nearly equal Spatial class-mean errors, 0.5430 and 0.5396. The resulting AUROCs are 0.386 for Action-L1 and 0.581 for Action-Ref. Group means describe this error decomposition; AUROC uses all member–non-member score comparisons.

Table 2 gives the four-suite and pooled results behind Figure 1; Figure 4 complements the Long ROC with the pooled, Spatial, Object, and Goal curves. Figure 5 separates the pretrained model's error, reference error, NP error, and the NP model's relative fit on the same candidates. Each suite contributes 64 test windows from distinct episodes. The scores, directions, and global calibration thresholds are shared with the pooled analysis. Long was selected as an illustrative diagnostic after inspecting this four-suite breakdown.

Table 2: NP suite-level and pooled membership auditing and task success. ASR and SR are percentages; AUROC is on a 0–1 scale. ASR retains the global nominal-5%-FPR calibration threshold.

Suite	Members/non-members	Action-L1		Action-Ref		Task SR
		AUROC	ASR	AUROC	ASR	
Spatial	29/35	0.386	52.0	0.581	55.8	76.0
Object	32/32	0.486	48.4	0.473	53.1	84.5
Goal	26/38	0.401	46.7	0.442	45.3	58.0
Long	34/30	0.541	50.9	0.720	59.9	47.5
Pooled	121/135	0.469	49.9	0.556	53.7	66.5

Interpreting the Long result. For members in Long task, the fine-tuned NP model has a larger mean advantage over the reference: 0.0391, compared with 0.0124 for non-members. The Action-Ref AUROC is 0.720 with a 95% interval of $[0.585, 0.842]$, obtained from 10000 class-stratified episode-bootstrap resamples with the NP model and shadow models fixed. The paired Action-Ref-minus-Action-L1 AUROC difference is 0.178, with interval $[-0.004, 0.351]$; the corresponding balanced-ASR difference is 0.090, with interval $[-0.035, 0.214]$. These paired intervals span zero, so the

observed ranking advantage remains a suite-level point-estimate finding. At the frozen threshold, Action-Ref identifies 9 of 34 members and flags 2 of 30 non-members: TPR 26.5%, actual FPR 6.7%, and balanced ASR 59.9%. Accuracy under the observed class mixture is $(9 + 28)/64 = 57.8\%$. The intervals describe candidate-sampling uncertainty conditional on the fitted models.

Whole-task success requires a sequence of successful interactions, whereas the attack queries a short demonstrated window at its recorded context. A model can predict such a window relatively well and still fail elsewhere during a long rollout. Moreover, every non-member test episode contains other member windows under the annotation-level protocol. Generalization across these windows can improve non-member predictions. The suite-level results therefore distinguish local annotation discrimination, absolute prediction error, and closed-loop task success as three separate measurements.

Reading the complete error comparison. In Figure 5, member and non-member bars summarize the same test windows across all three models or reference constructions. The reference and NP panels share a vertical scale; their paired difference gives relative fit. Object and Goal retain weak attack discrimination after centering.

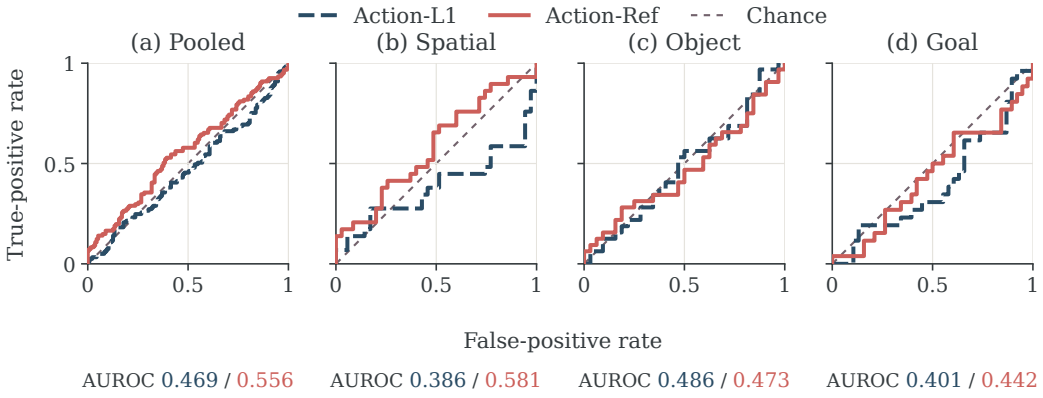


Figure 4: NP ROC curves for the pooled test set and the three suites complementing Long in Figure 1. Pooled contains all 256 windows, including Long; each individual suite contains 64. The values beneath each panel give Action-L1 and Action-Ref AUROC in legend order; the diagonal marks chance. All curves use the same frozen scores and directions as Table 2.

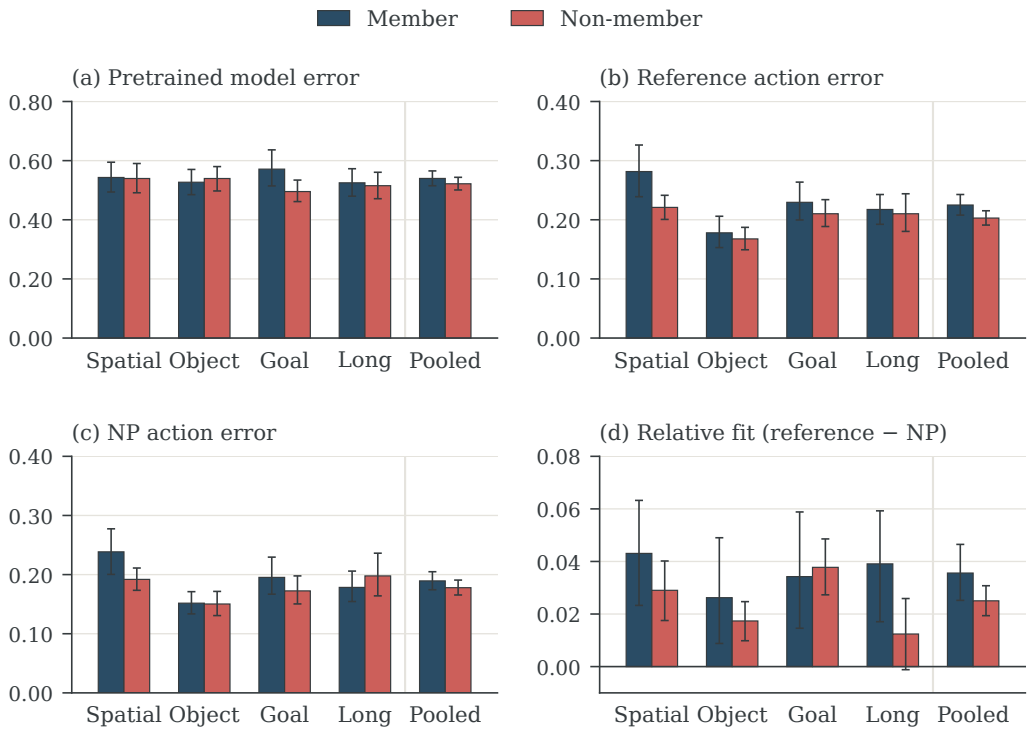


Figure 5: NP action-error decomposition on the same test candidates. Bars show class-mean public-scaled action errors for the pretrained model, candidate-specific reference, and NP model, followed by reference error minus NP error. Higher relative fit means a larger predictive advantage over the reference. Whiskers are 95% intervals from 10000 class-stratified episode resamples with the models fixed; all four measurements share the same resampled candidates. Panels (b,c) use the same vertical scale; each test episode contributes one window.

C NOISE ROLES AND CLIPPING DESIGN

C.1 ACTION-LEARNING NOISE AND GRADIENT PRIVACY NOISE

The pretrained $\pi_{0.5}$ model retains its native FM objective. Public normalization maps actions to u_i, padded to $d_m = 32$ model coordinates. Draw native flow noise $\xi \sim \mathcal{N}(0, I_{Hd_m})$ and an interpolation time r from the native distribution. Define

$$z_r = (1-r)u_i + r\xi, \qquad \ell_i(\theta;\xi,r) = \frac{1}{Hd_m}\sum_{h=1}^{H}\sum_{d=1}^{d_m}[f_\theta(x_i,z_r,r)_{h,d} - (\xi-u_i)_{h,d}]^2. \tag{14}$$

θ comprises the trainable parameters; f_θ predicts the velocity of an action interpolation; h, d index time and model coordinates. The velocity target $\xi - u_i$ points from ground-truth actions toward noise. Inference integrates in the reverse direction to generate actions. The objective learns a window-conditioned action generator.

We retain the native mean over all 10×32 coordinates. Short terminal windows repeat their final real action, and those repeated positions remain in the training loss. The attack score masks padded timesteps and uses seven physical coordinates. Native flow noise, interpolation time, and image augmentation are shared learning components across all training variants. Privatization perturbs gradients while preserving the stored actions and images.

Native ξ is sampled before differentiation, while gradient privacy noise η_t is drawn independently after clipping and aggregation. Privacy noise has one coordinate per trainable scalar parameter, with optimizer-input variance $\tau^2 = (\sigma C/B)^2$; the unnormalized sum has noise variance $(\sigma C)^2$. These are the average and sum representations of the same private update; accounting uses σ, sampling, steps, and adjacency.

C.2 WHERE THE BASELINE CLIPPING BOUND COMES FROM

The main DP-VLA$_{\text{base}}$ recipe calibrates the clipping bound using the pretrained model, with all 3,616,757,520 parameters trainable at the initial `pi05` checkpoint. The protocol uses 96 predeclared H10 probes from 24 public auxiliary Object demonstrations (task indices 4–9), one native flow-matching draw per probe, and seed seven. Each probe contributes its joint L_2 gradient norm to the calibration sample. Writing the sorted norms as $n_{(1)}, \ldots, n_{(96)}$, the rule was

$$C_L = \frac{n_{(48)} + n_{(49)}}{2} = \frac{2.0423931487518936 + 2.0707084255632937}{2} = 2.056550787157594. \tag{15}$$

The subscript in $n_{(k)}$ denotes an order statistic. The 40-task DP-VLA$_{\text{base}}$ runs use this same calibrated C_L.

The 24 calibration episodes, including their 377 public windows and all 96 probes, also belong to the 40-task candidate pool. The privacy analysis holds these public calibration assets fixed and protects the subsequent annotation-inclusion bit.

C.3 CLIPPING SCALE AND PER-WINDOW INFLUENCE

We treat $C > 0$ as a public optimization hyperparameter. For any fixed public $C > 0$, clipping bounds a present annotation's contribution by C; pairing it with Gaussian scale σC retains the same noise-to-sensitivity ratio. The main DP-VLA$_{\text{MC}}$ recipe fixes $C = C_S = 0.10$. The independently specified variance-floor ratio is $\rho = 0.1$. At fixed noise multiplier, the optimizer-input noise standard deviation is smaller than the main DP-VLA$_{\text{base}}$ recipe by a factor $C_L/C_S \simeq 20.5655$. Uniformly scaling gradients and noise makes Adam's numerator and square-root denominator scale approximately together. Here clipping is nonlinear: different windows can be shrunk by different factors, changing their relative influence and the aggregate direction. Smaller clipping bounds trade gradient magnitude information for tighter control of individual-window contributions. The component controls separate this clipping effect from the moment correction.

D A WORKED PRIVATE UPDATE: DP-VLA$_{\text{base}}$ AND DP-VLA$_{\text{MC}}$

We illustrate update $t = 1$ using a two-parameter model, retaining the update index to distinguish windows from parameter coordinates. Fix expected batch denominator $B = 4$, clipping bound $C = C_S = .1$, and noise multiplier $\sigma = .4$. Suppose one Poisson draw selects four candidate windows; the first two have present annotations. Their unbounded gradients are $g_{1,1} = (3, 4)$ and $g_{1,2} = (0, .2)$, where the second subscript is the window index i. The remaining windows contribute zero through annotation absence.

Bound one annotation's influence. The norms are $\|g_{1,1}\|_2 = 5$ and $\|g_{1,2}\|_2 = .2$, hence

$$\bar{g}_{1,1} = (.06, .08), \qquad \bar{g}_{1,2} = (0, .10), \qquad G = \bar{g}_{1,1} + \bar{g}_{1,2} = (.06, .18).$$

Removing annotation one changes the clipped sum by $\bar{g}_{1,1}$, whose norm is exactly $C_S = .1$; it changes the fixed-denominator average by a vector of norm $C_S/B = .025$. This is the sensitivity controlled by clipping.

Add Gaussian noise and keep the denominator fixed. The noise added to the sum has standard deviation $\sigma C = .04$ in each coordinate. For one illustrative realization $Z_1 = (.02, -.04)$,

$$\widetilde{g}_1 = (G + Z_1)/4 = (.020, .035), \qquad \bar{g}_1 = G/4 = (.015, .045), \qquad \tau = \sigma C/B = .01.$$

The optimizer input $\widetilde{g}_1$ combines the clipped signal and Gaussian perturbation. Both DP-VLA$_{\text{base}}$ and DP-VLA$_{\text{MC}}$ receive this same privatized-gradient *form*.

Compare the first AdamW update at the same C. Let $\widehat{m}_t^{\text{opt}}$ and $\widehat{v}_t$ denote the bias-corrected first- and second-moment vectors; opt labels an optimizer moment. At $t = 1$, $\widehat{m}_1^{\text{opt}} = \widetilde{g}_1$ and $\widehat{v}_1 = \widetilde{g}_1^{\odot 2}$, where $\odot 2$ denotes coordinatewise squaring. For coordinate $c = 1$, $[\widehat{m}_1^{\text{opt}}]_c = .020$ and $\widehat{v}_{1,c} = [\widehat{v}_1]_c = .0004$. The leading-order ordinary AdamW denominator in this example is $\sqrt{.0004} = .020$. DP-VLA$_{\text{MC}}$ knows $\tau^2 = .0001$ and uses

$$d_{1,c}^{\text{MC}} = \sqrt{\max(.0004 - .0001,\ .1(.0001))} = \sqrt{.0003} \simeq .0173205.$$

With first-step learning rate $\alpha_1 = 5 \cdot 10^{-5}$, the coordinate updates are

$$[\Delta\theta_1^{\text{base}}]_c = -.0000500, \qquad [\Delta\theta_1^{\text{MC}}]_c = -\alpha_1(.020)/.0173205 \simeq -.0000577.$$

DP-VLA$_{\text{MC}}$ changes the scale of the update while retaining the noisy first moment. Near the variance floor, smaller denominators amplify both the learning signal and the remaining noise. Component experiments measure how this rescaling affects closed-loop performance.

This comparison holds clipping fixed to isolate the optimizer. The main DP-VLA$_{\text{base}}$ recipe uses C_L, so the main comparison evaluates both clipping and the optimizer. The fixed DP-VLA$_{\text{base}}$ (C_S) ordinary-AdamW control is needed for that distinction. For completeness, the same original gradients with C_L yield clipped sum $G_1^{\text{large}} \simeq (1.23393, 1.84524)$, where the subscript denotes update one and the superscript identifies the larger clipping bound. Using the illustrative standard-normal vector $(.5, -1)$ gives sum noise $Z_1^{\text{large}} \simeq (.41131, -.82262)$, hence optimizer input $(G_1^{\text{large}} + Z_1^{\text{large}})/4 \simeq (.41131, .25566)$. The ordinary first update is approximately $(-5, -5) \times 10^{-5}$, the small-C ordinary first update is also $(-5, -5) \times 10^{-5}$, and DP-VLA$_{\text{MC}}$ gives $(-5.7735, -5.2175) \times 10^{-5}$. Their identical first updates follow from first-step Adam normalization; later updates depend on each method's evolving gradient history. Subsequent steps recompute gradients at each method's own updated parameters.

Privacy accounting. The accountant uses the public sampling rate q, noise multiplier σ, full step count T, adjacency, and chosen δ. At fixed q, σ, T, scaling C changes sensitivity and noise proportionally, preserving the accounted epsilon.

E PROOFS AND EXPLANATORY DERIVATIONS

E.1 PRIVACY-PRESERVING OPTIMIZER SUBSTITUTION

We compute the window-level (ϵ, δ)-DP guarantee through Rényi differential privacy (RDP), which supports composition of the sampled Gaussian steps (Mironov, 2017; Mironov et al., 2019). Let

$r_\alpha(q,\sigma)$ be a per-step bound at order $\alpha > 1$, valid in both annotation-presence/absence directions and incorporating amplification from hidden Poisson sampling. Adaptive composition over all T fine-tuning steps gives

$$R_\alpha = T\, r_\alpha(q,\sigma). \tag{16}$$

Here R_α is the cumulative RDP cost; Appendix E.3 gives the single-step expression, conversion to (ϵ,δ)-DP, and noise-multiplier calibration.

Theorem 1 (Privacy of the optimizer substitution)**.** *Assume public initialization/hyperparameters, fixed public T, hidden independent Poisson sampling, record-local gradients jointly bounded by C, and fresh ideal Gaussian noise, independent across coordinates and steps and independent of sampling and native randomness. Assume the optimizer receives privatized gradients and public quantities as its inputs. DP-VLA$_{\text{base}}$ and DP-VLA$_{MC}$ with the same q,σ,T admit the same R_α bound in Equation (16), and hence the same (ϵ,δ) bound under a shared RDP conversion (Appendix E.3). The moment correction and public floor are privacy-preserving postprocessing.*

Proof of Theorem 1. Condition on any preceding privatized-gradient transcript. The optimizer state and θ_{t-1} are then fixed functions of that transcript and public inputs. All unaffected windows have the same record-local contributions under the two neighboring datasets. The changed window contributes either zero or a vector of norm at most C. Dividing sensitivity and Gaussian standard deviation by the fixed $B = qN$ preserves their ratio. The hidden Poisson-sampled Gaussian bound $r_\alpha(q,\sigma)$ therefore holds uniformly for this history. Adaptive RDP composition yields Tr_α. AdamW moments, public variance subtraction, the floor, and final weights are postprocessing of this transcript. RDP-to-DP conversion yields the stated window-level guarantee. □

This argument is repeated for each optimizer's own adaptive path. **Analytical scope.** The guarantee describes a fixed-schedule ideal Gaussian mechanism whose release is the final model. All reported DP models complete the fixed 10000-update schedule; Appendix E.3.1 verifies their cumulative bounds against the specified budgets. The ideal-mechanism guarantee and its numerical accounting are distinct from end-to-end certification of mixed-precision arithmetic. Joint model releases, additional private outputs, and private-data model selection each require accounting beyond the single-model bound.

DP as a hypothesis-testing constraint. For one adjacent pair, a decision event based on the DP release and auxiliary information held fixed between the neighbors obeys

$$\mathrm{TPR} \le e^{\epsilon}\mathrm{FPR} + \delta. \tag{17}$$

This substitutes the event that the observer predicts membership into Equation (2). The formal bound applies to each adjacent pair, while empirical curves aggregate risks over a finite candidate population.

E.2 Exact cancellation of the additive second-moment term

Let $m_t^{\mathrm{opt}}, v_t^{\mathrm{opt}}$ be Adam's first- and second-moment vectors, initialized at zero, with exponential averaging factors $\beta_1,\beta_2 \in [0,1)$:

$$m_t^{\mathrm{opt}} = \beta_1 m_{t-1}^{\mathrm{opt}} + (1-\beta_1)\widetilde{g}_t, \qquad \widehat{m}_t^{\mathrm{opt}} = m_t^{\mathrm{opt}}/(1-\beta_1^t), \tag{18}$$

$$v_t^{\mathrm{opt}} = \beta_2 v_{t-1}^{\mathrm{opt}} + (1-\beta_2)\widetilde{g}_t^{\odot 2}, \qquad \widehat{v}_t = v_t^{\mathrm{opt}}/(1-\beta_2^t). \tag{19}$$

The operator $\odot 2$ denotes coordinatewise squaring; $\widehat{v}_{t,c} = [\widehat{v}_t]_c$ denotes coordinate c of the second-moment vector. The hats correct zero-initialization bias, and opt labels optimizer moments; the noise-variance correction is applied separately. For p trainable parameters, collect the chosen coordinate denominators from Equation (7) into $d_t \in \mathbb{R}^p$. With learning rate α_t and decoupled weight decay λ, the full AdamW parameter update is

$$\theta_t = (1-\alpha_t\lambda)\theta_{t-1} - \alpha_t \widehat{m}_t^{\mathrm{opt}} \oslash d_t. \tag{20}$$

Here $\oslash$ denotes coordinatewise division. Choosing $d_{t,c}^{\mathrm{base}}$ gives ordinary AdamW; choosing $d_{t,c}^{\mathrm{MC}}$ applies the noise-variance correction and public floor.

Let $\mathcal{H}_t$ contain prior noise, optimizer state, the current sample, and native flow/augmentation randomness before drawing η_t. In ideal arithmetic $\bar{g}_t$ is $\mathcal{H}_t$-measurable. With $\mathbf{1}_p$ the all-ones vector and $\odot$ coordinatewise multiplication, define the residual vector

$$\Delta_t^{\mathrm{mom}} = \widetilde{g}_t^{\odot 2} - \bar{g}_t^{\odot 2} - \tau^2 \mathbf{1}_p = 2\bar{g}_t \odot \eta_t + (\eta_t^{\odot 2} - \tau^2 \mathbf{1}_p). \tag{21}$$

Δ_t^{mom} is the random error remaining after removing the known noise variance. Fresh zero-mean Gaussian noise gives $\mathbb{E}[\Delta_t^{\mathrm{mom}} \mid \mathcal{H}_t] = 0$. For parameter coordinate c, Gaussian second/fourth moments additionally give

$$\mathbb{E}\big[[\Delta_t^{\mathrm{mom}}]_c^2 \mid \mathcal{H}_t\big] = 4\tau^2 [\bar{g}_t]_c^2 + 2\tau^4. \tag{22}$$

The two terms quantify residual uncertainty from the signal–noise cross term and from squared noise.

Unroll the second-moment recurrence and define deterministic weights

$$w_{t,s} = \frac{(1-\beta_2)\beta_2^{t-s}}{1-\beta_2^t}, \quad \sum_{s=1}^{t} w_{t,s} = 1, \qquad \widehat{v}_t - \tau^2 \mathbf{1}_p = \sum_{s=1}^{t} w_{t,s} \bar{g}_s^{\odot 2} + \sum_{s=1}^{t} w_{t,s} \Delta_s^{\mathrm{mom}}. \tag{23}$$

Here s indexes earlier optimizer steps and $w_{t,s}$ is their bias-corrected exponential weight. Taking unconditional expectations makes the final residual sum vanish, by the conditional zero-mean property. The remaining expectation is the weighted squared clipped aggregate *along the noisy adaptive path*. Fresh-noise conditional expectations account for the dependence of future $\bar{g}_s$ on earlier privacy noise.

The variance floor and nonlinear coordinate scaling determine how this moment correction translates into parameter updates.

Numerical illustration. Suppose coordinate c has $\widehat{v}_{t,c} = 10^{-3}$ and $\tau^2 = 9 \cdot 10^{-4}$. DP-VLA$_{\mathrm{MC}}$ uses $\max(10^{-4}, 9 \cdot 10^{-5}) = 10^{-4}$ before the square root. Ignoring γ, its denominator is 0.01, versus about 0.0316 for ordinary Adam. For the same first moment this permits a larger update.

E.3 Sampled-Gaussian privacy accounting

For integer Rényi order $\alpha \geq 2$, the sampled-Gaussian expression is

$$r_\alpha(q,\sigma) = \frac{1}{\alpha - 1} \log \sum_{j=0}^{\alpha} \binom{\alpha}{j} (1-q)^{\alpha-j} q^j \exp\left(\frac{j(j-1)}{2\sigma^2}\right). \tag{24}$$

The index j enumerates terms in the moment expansion. The expression incorporates privacy amplification from hidden subsampling. Fractional orders are evaluated using the numerical series of Mironov et al. (2019). The public grid is

$$\begin{aligned} \mathcal{A} = {} & \{1.0101, 1.0102, \ldots, 1.1000\} \cup \{1.101, 1.102, \ldots, 2.000\} \\ & \cup \{2.1, 2.2, \ldots, 10.0\} \cup \{11, \ldots, 256\}. \end{aligned} \tag{25}$$

It has 2126 orders. For a fixed δ, the basic RDP-to-DP conversion is

$$\epsilon_{\mathrm{basic}}(\delta) = \min_{\alpha \in \mathcal{A}} \left[R_\alpha + \frac{\log(1/\delta)}{\alpha - 1} \right]. \tag{26}$$

The first term is the composed RDP cost; the second converts that cost at the chosen δ. We use the improved RDP-to-DP conversion

$$\epsilon_{\mathrm{grid}} = \max\left\{0, \min_{\alpha \in \mathcal{A}} \left[R_\alpha + \log(1 - 1/\alpha) - \frac{\log(\delta\alpha)}{\alpha - 1} \right] \right\}. \tag{27}$$

R_α is the complete-training RDP value from Equation (16). The correction terms tighten the basic conversion for the same adjacency. This conversion follows Canonne et al. (2020).

We solve $\epsilon_{\mathrm{grid}} \leq \epsilon$ for σ with relative tolerance 10^{-6}, taking the feasible upper noise endpoint.

Table 3: Final cumulative privacy bounds for all reported DP models. Each model completes $T = 10000$ updates with $q = 4/28096$ and $\delta = 10^{-5}$. The 14 main models and nine additional component models are listed once each, retaining the recorded numerical precision. A checkmark indicates that the accounted bound is at most ϵ. NP and the non-private shadow models have no finite DP budget and are excluded.

Budget ϵ	Method	C	Seed	Steps	Accounted bound	$\leq \epsilon$
Main experiments						
10^3	DP-VLA$_{\text{base}}$	C_L	17	10000	999.9989542969109	✔
10^3	DP-VLA$_{\text{MC}}$	C_S	17	10000	999.9989542969109	✔
10^6	DP-VLA$_{\text{base}}$	C_L	17	10000	999994.7570093675	✔
10^6	DP-VLA$_{\text{MC}}$	C_S	17	10000	999994.7570093696	✔
10^7	DP-VLA$_{\text{base}}$	C_L	17	10000	9999975.813578086	✔
10^7	DP-VLA$_{\text{MC}}$	C_S	17	10000	9999975.813578086	✔
10^8	DP-VLA$_{\text{base}}$	C_L	17	10000	99999967.27242768	✔
10^8	DP-VLA$_{\text{MC}}$	C_S	17	10000	99999967.27242763	✔
10^9	DP-VLA$_{\text{base}}$	C_L	17	10000	999998783.0631965	✔
10^9	DP-VLA$_{\text{MC}}$	C_S	17	10000	999998783.0631965	✔
10^{10}	DP-VLA$_{\text{base}}$	C_L	17	10000	9999987099.421556	✔
10^{10}	DP-VLA$_{\text{MC}}$	C_S	17	10000	9999987099.421556	✔
10^{12}	DP-VLA$_{\text{base}}$	C_L	17	10000	999998561413.7336	✔
10^{12}	DP-VLA$_{\text{MC}}$	C_S	17	10000	999998561413.7336	✔
Additional component models						
10^8	DP-VLA$_{\text{MC}}$	C_L	17	10000	99999967.27242768	✔
10^8	DP-VLA$_{\text{base}}$	C_S	17	10000	99999967.27242763	✔
10^8	DP-VLA$_{\text{base}}$	C_L	117	10000	99999967.27242768	✔
10^8	DP-VLA$_{\text{MC}}$	C_L	117	10000	99999967.27242768	✔
10^8	DP-VLA$_{\text{base}}$	C_S	117	10000	99999967.27242763	✔
10^8	DP-VLA$_{\text{MC}}$	C_S	117	10000	99999967.27242763	✔
10^9	DP-VLA$_{\text{base}}$	C_S	17	10000	999998783.0631965	✔
10^9	DP-VLA$_{\text{base}}$	C_S	117	10000	999998783.0631965	✔
10^9	DP-VLA$_{\text{MC}}$	C_S	117	10000	999998783.0631965	✔

E.3.1 FINAL CUMULATIVE WINDOW-LEVEL ACCOUNTING

We evaluate Equations (16) and (27) using the actual completed updates and fixed noise parameters of every reported DP model. Table 3 gives the resulting cumulative bounds at $\delta = 10^{-5}$; all 23 models remain within their specified budgets.

E.3.2 SINGLE-STEP WINDOW-LEVEL ACCOUNTING

To complement the cumulative budgets in Section 6.1, Table 4 reports the corresponding one-update bounds. We retain each frozen noise multiplier σ, sampling probability $q = 4/28096$, annotation-presence/absence adjacency, and the same RDP grid and conversion, using `dp-accounting` version 0.6.0. Let $(\epsilon_{\text{step}}, \delta_{\text{step}})$ describe one update at a fixed preceding privatized state, including hidden Poisson sampling. We obtain this bound from Equation (27) with $R_\alpha = r_\alpha(q, \sigma)$ and $\delta = \delta_{\text{step}}$. Since DP-VLA$_{\text{base}}$ and DP-VLA$_{\text{MC}}$ share q and σ at each budget, the displayed bounds apply to both methods.

Table 4: Per-window, single-update privacy bounds from the fixed RDP accountant. The first column gives the run budget at $T = 10000$ and $\delta = 10^{-5}$; the last two columns describe one sampled update. Noise multipliers and bounds are rounded for display; calculations use the frozen full-precision values.

Run budget ϵ	Noise multiplier σ	δ_{step}	ϵ_{step}
10^3	6.919092×10^{-2}	10^{-5}	125.97
10^6	2.316980×10^{-2}	10^{-5}	1,018.03
10^7	1.636566×10^{-2}	10^{-5}	1,994.49
10^8	6.811447×10^{-3}	10^{-5}	11,101.17
10^9	2.237447×10^{-3}	10^{-5}	101,134.06
10^{10}	7.103547×10^{-4}	10^{-5}	1,001,132.89
10^{12}	7.106661×10^{-5}	10^{-5}	100,000,990.32

For the complete run, accounting first composes $R_\alpha = T\, r_\alpha(q, \sigma)$ and then converts at the chosen δ (Equations (16) and (27)). This order of operations makes ϵ_{step} generally different from ϵ/T. The main-text budgets continue to describe the cumulative guarantee for the released model, while $(\epsilon_{\text{step}}, \delta_{\text{step}})$ describes an individual update.

E.4 EPISODE ACCOUNTING IS A DIFFERENT ADJACENCY

Let $L \leq 51$ be the public maximum number of windows removed together. The triangle inequality bounds the summed gradient change by LC. A Gaussian zero-concentrated DP (zCDP) bound based on sensitivity LC provides episode accounting (Bun & Steinke, 2016):

$$\varrho_{\text{ep}} = \frac{TL^2}{2\sigma^2}, \qquad \epsilon_{\text{ep}}(\delta) = \varrho_{\text{ep}} + 2\sqrt{\varrho_{\text{ep}} \log(1/\delta)}, \qquad L = 51. \tag{28}$$

ϱ_{ep} is the composed zCDP parameter; ρ remains the variance-floor ratio of DP-VLA$_{\text{MC}}$. ϵ_{ep} is an episode-annotation bound at the specified delta, still conditioning on public contexts. The factor L^2 follows from the squared sensitivity of the summed annotation contributions.

Alternatively, a path of L single-window changes gives the generic group-privacy bound

$$\epsilon_{\text{grp}} = L\epsilon, \qquad \delta_{\text{grp}} = \min\left\{1, \delta \sum_{j=0}^{L-1} e^{j\epsilon}\right\}. \tag{29}$$

These are whole-training ϵ, δ guarantees assumed uniform along the adjacency path. Repeated application of Equation (2) proves the expression. This expression composes annotation changes within a single model release. In the large-epsilon regime, the group bound's delta typically saturates at one.

F Detailed Experiment Setup

This appendix specifies the data, fine-tuning, sampling, auxiliary models, and evaluation protocol summarized in Section 6.1.

F.1 Dataset and membership construction

The candidate pool contains all indexed demonstrations from the four-suite LIBERO dataset. Our demonstration dataset contains 1693 demonstrations and 273465 action frames. Non-overlapping windows of horizon and stride ten yield $N = 28096$ candidate action annotations. Table 5 gives the candidate-window and member/non-member counts. Every task contains both member and non-member windows. All NP/DP-VLA$_{\text{base}}$/DP-VLA$_{\text{MC}}$ main fine-tuning experiments use this same mask; the shadow models use different known masks so that each candidate has models trained with and without its annotation.

Table 5: Frozen four-suite data use. Members denote action annotations eligible for supervision during NP and DP fine-tuning. Each suite contains ten tasks.

Suite	Episodes	Action frames	Windows	Members	Non-members
Spatial	432	52970	5489	2715	2774
Object	454	66984	6908	3439	3469
Goal	428	52042	5388	2719	2669
Long	379	101469	10311	5209	5102
Total	1693	273465	28096	14082	14014

For each episode, starts are $0, 10, 20, \ldots$, and each window covers at most ten action frames. There are 1515 short terminal windows. The loader repeats the last valid action to fill a terminal chunk and records a valid-frame mask. Native training retains its mean over the full 10×32 tensor, including repeated terminal positions and padded model coordinates. Attack scoring uses valid timesteps and seven physical coordinates.

A single set of independent Bernoulli-half draws fixes membership before training. The realized supervised fraction is $14082/28096 = 50.12\%$ of window annotations. Uniform candidate-window sampling gives each annotation the same inclusion probability in both DP arms. NP uses the member-coverage schedule in Appendix F.3. The shared assignment fixes the eligible supervision set for all NP and DP experiments.

Public context and annotation use. Context comprises the two camera observations, robot state, instruction, and window metadata. For a non-member, candidate actions remain available to the attack evaluator, while their supervised training-gradient contribution is zero. Adjacency fixes public context and neighboring annotations, including their physical and temporal correlations.

Public normalization. The normalizer is the official OpenPI `pi05_libero` asset[1]; weights are initialized from `pi05`. The action-score weights in Equation (9) are the slopes of this fixed affine quantile normalization. The study conditions on public pretraining and normalization as provided, including any benchmark information they carry.

Window and episode membership. Let $\mathcal{I}_e$ be the set of window indices belonging to episode e, and $L_e = |\mathcal{I}_e|$ its window count. Our inclusion bits $\{m_i : i \in \mathcal{I}_e\}$ can differ inside the same episode. Trajectory inclusion assigns one episode-level membership bit m_e^{ep} and sets $m_i = m_e^{\text{ep}}$ for every $i \in \mathcal{I}_e$. The distinction is illustrated by the following three-window episode:

$$\begin{aligned} \text{window annotation inclusion:} \quad & (m_1, m_2, m_3) = (1, 0, 1), \\ \text{trajectory inclusion:} \quad & (m_e^{\text{ep}}, m_e^{\text{ep}}, m_e^{\text{ep}}) \in \{(1,1,1), (0,0,0)\}. \end{aligned}$$

Here one means the action annotation is eligible for supervision and zero means it is absent. In our setting, a non-member window can have member neighbors. Peng et al. use separate member and non-member sets of complete demonstrations in their experimental setup (Peng et al., 2026). Our adaptation changes the membership unit while retaining the action-error signal.

[1] OpenPI LIBERO normalization statistics

Likewise, window-level DP changes one annotation bit, whereas removing an episode's annotations changes up to L_e bits (at most 51 in this pool). Episode privacy is accounted for under the corresponding multi-annotation adjacency in the derivations appendix. Non-overlapping labels assign action coordinates to unique windows; correlated windows can carry different membership bits within one episode.

F.2 Training configurations

All fine-tuning runs start from the public `pi05_base`, retain its tokenizer and native flow-matching objective, and use the official public action normalization described in Appendix F.1. We train all model parameters with the settings in Table 6. The NP reference uses AdamW, while the two DP variants apply the updates in Section 5.

Table 6: Fine-tuning configurations. Sampling schedules are detailed in Appendix F.3.

Quantity	Setting
Trainable scope	Full model; 3,616,757,520 parameters
Native loss dimensions	10×32, including padded channels
Training budget	$T = 10000$ updates per training run
Learning rate / Adam moments	$5 \cdot 10^{-5},\ (\beta_1, \beta_2) = (0.9, 0.95)$
Adam stabilizer / weight decay	$\gamma = 10^{-8},\ \lambda = 10^{-10}$
NP batch / clipping	Four members; aggregate gradient clip one
DP sampling / denominator	$q = 4/28096,\ B = qN = 4$
Main DP-VLA$_{\text{base}}$ clipping	$C_L = 2.056550787157594$
Main DP-VLA$_{\text{MC}}$ clipping	$C_S = 0.10$
MC variance floor	$\rho\tau^2 = 0.1\tau^2$
Training repetition	Main and repeat settings in Section 6.4

The main experimental matrix contains one NP run and one run of each DP method at each of the seven budgets in Section 6.1, yielding 15 fine-tuning runs. The additional component controls preserve the same membership assignment (Appendix H). We solve Gaussian noise multipliers with the same Poisson-compatible accountant and verify the budget against the executed training steps; Appendices E.3 and E.4 give the window-level and episode-level accounting. Appendix C.2 gives the public calibration of C_L; the component comparisons also evaluate both optimizers at matched clipping bounds.

F.2.1 Privacy-budget grid and evaluation focus

We use the budget sweep to examine how much closed-loop task capability private fine-tuning retains and how optimization changes this capability. Table 7 groups the evaluated points by their role in interpreting this relationship. The five intermediate budgets, from 10^6 to 10^{10}, support the primary method comparisons and cover the observed rise in task success. At a fixed ϵ, the paired methods measure how clipping and moment correction affect utility under shared accounting. Across budgets, observed task success shows which evaluated settings achieve higher utility with a smaller cumulative budget. The component comparisons at 10^8 and 10^9 further separate the optimization effects in this transition region (Section 6.4).

Table 7: Roles of the evaluated privacy budgets.

Budget ϵ	Evaluation focus
10^3	Higher-noise diagnostic
$10^6, 10^7, 10^8, 10^9, 10^{10}$	Core utility comparisons
10^{12}	Weak-noise diagnostic

The two endpoints extend the sweep in complementary directions. At 10^3, the largest tested noise multiplier probes task learning under stronger perturbation. At 10^{12}, the smallest tested multiplier, $\sigma \simeq 7.11 \times 10^{-5}$, provides a weak-noise diagnostic while retaining gradient clipping and positive Gaussian noise. NP remains a separate reference with the sampling recipe in Appendix F.3. Every budget retains its cumulative window-level meaning; Appendices E.3.1 and E.3.2 give the final accounted values, noise multipliers, and single-step guarantees.

F.3 Sampling and supervision coverage

The fixed membership assignment defines eligible supervision, while the sampling schedule determines how often each member is visited. Finite random sampling can leave eligible members unvisited, creating supervision gaps within the designated training set. NP addresses these gaps by independently shuffling the member indices each epoch and consuming batches of four across epoch boundaries. Its 10000 updates make 40000 supervised visits, covering every member two or three times. The membership assignment remains fixed: the 14014 non-member annotations stay excluded, including the deliberately held-out intervals within episodes.

For DP, writing $M = 14082$ for the member count, independent hidden Poisson inclusion on the fixed public universe gives

$$P^{\mathrm{DP}}_{\mathrm{empty}} = (1-q)^M, \qquad P^{\mathrm{DP}}_{\mathrm{unvisited}} = (1-q)^T. \tag{30}$$

Here $P^{\mathrm{DP}}_{\mathrm{empty}}$ is the probability that a step samples zero member annotations; $P^{\mathrm{DP}}_{\mathrm{unvisited}}$ is the probability that a given member remains unvisited after T updates. For $q = 4/28096, T = 10000$, the corresponding member-coverage probability is $1 - P^{\mathrm{DP}}_{\mathrm{unvisited}} \simeq 75.92\%$. The expected number of supervised DP visits is $TqM \approx 20048$, compared with NP's 40000 visits under the same update budget.

Both DP methods retain independent Poisson sampling over the fixed candidate universe, as required by the amplification bound used in our accountant (Appendix E.3). Member-only reshuffling follows a different sampling law and would require a corresponding privacy analysis. We therefore use coverage-oriented sampling for NP and accountant-matched sampling for DP, giving different supervision exposure under the same update budget. The NP reference characterizes the coverage-oriented training recipe; the two DP methods remain matched in membership assignment, sampling, and accounting.

F.4 Shadow models and attack protocol

Auxiliary models. We use 16 NP shadow models, each fine-tuned from the same pretrained model, assigning every candidate annotation to eight models and excluding it from the other eight, so $|\mathcal{R}_i| = 8$. Each shadow model uses full-parameter training for 10000 updates, candidate-window batches of four sampled with replacement, and aggregate gradient clipping at one. The candidate pools for auxiliary training and the main fine-tuning experiments may overlap; the attacker specifies each shadow model's membership mask independently of the held-out annotations' membership labels for the target model.

The same shadow models provide a common cross-recipe reference for models fine-tuned with member-reshuffled NP or Poisson-sampled DP.

Calibration and test candidates. We form non-overlapping episode groups for calibration and testing with fixed split seed 107 before task-round-robin candidate selection. Task-round-robin selection chooses 32 calibration and 64 test windows per suite, each from a distinct episode, using public candidate metadata. The 128 calibration windows contain 68 members and 60 non-members; the 256 test windows contain 121 members and 135 non-members. Their per-suite test counts are 29/35, 32/32, 26/38, and 34/30 (member/non-member), in suite order.

Calibration and test define attack-evaluation roles; test members remain part of the fine-tuned model's annotated training set. Their identities are revealed after attack fitting, threshold selection, and held-out predictions are fixed. This setting evaluates an observer with access to the auxiliary candidate pool and known shadow memberships.

Matched action queries. Both attacks use the same four action generations from the target model per candidate, with ten flow-solver steps per generation. For Action-Ref, the same rule and candidate-specific seeds also score the shadow models (Appendix B.1). Calibration non-members fix the threshold for each target-model/attack pair at a nominal FPR of 5%, following Appendix F.5. We fix held-out scores and membership predictions before revealing test memberships, then join the labels for evaluation.

Reference-calibration contrast. Action-L1 uses $s_i^{\mathrm{L1}} = -e_i(\theta)$, whereas Action-Ref uses the candidate's mean reference error minus $e_i(\theta)$. They share the outputs of the target model; their differ-

ence tests the addition of candidate-specific references, including the auxiliary models' additional information and cost.

Costs. Each target model receives $384 \times 4 = 1536$ output queries, or 15360 solver steps. The shadow models cost 16 trainings of 10k updates and 24576 output queries (245760 solver steps), amortized across the evaluated models. Action-L1 uses queries to the target model and calibration windows with known memberships. Action-Ref reuses the same queries and amortizes auxiliary training across the evaluated models. Both attacks operate on generated model actions.

F.5 Closed-loop evaluation and metrics

Rollout protocol. Each fine-tuned model is evaluated on all 40 tasks with 20 fixed official initial states per task, yielding 800 rollouts. We use evaluation seed seven, ten initial settling steps, ten-step action chunks, replanning after five actions, and ten flow-solver steps. Interaction limits are 220/280/300/520 steps for Spatial/Object/Goal/Long. For n_{task} tasks and n_{init} initial states per task, the overall SR is

$$U(\theta) = \frac{1}{n_{\text{task}} n_{\text{init}}} \sum_{k=1}^{n_{\text{task}}} \sum_{e=1}^{n_{\text{init}}} \mathbf{1}[\pi_\theta \text{ succeeds in rollout } (k, e)]. \tag{31}$$

Here k, e index the task and initial state, and the indicator is one for a successful rollout. We report per-suite SR and this equal-task average, with $n_{\text{task}} = 40$ and $n_{\text{init}} = 20$.

Ranking. For independent draws from the member and non-member test populations,

$$\text{AUROC} = \mathbb{P}[s_{\text{member}} > s_{\text{nonmember}}] + \tfrac{1}{2}\mathbb{P}[s_{\text{member}} = s_{\text{nonmember}}]. \tag{32}$$

Here s is either attack score, with larger values indicating membership. The expression is the probability that a member ranks above a non-member, with half credit for ties. It summarizes ranking across thresholds; the following calibration fixes a specific decision rule.

Calibration. For each attack and target model, sort the scores of known calibration non-members as $s_{(1)} \geq \cdots \geq s_{(n_0)}$. For the nominal calibration FPR $\zeta = 0.05$, define

$$k_\zeta = \lfloor \zeta n_0 \rfloor, \quad \kappa_\zeta = s_{(k_\zeta + 1)}, \qquad \widehat{m}_i = \mathbf{1}[s_i > \kappa_\zeta]. \tag{33}$$

Here n_0 is the calibration non-member count, k_ζ the allowed false-positive count, and κ_ζ the score cutoff; $\widehat{m}_i = 1$ predicts membership and $\widehat{m}_i = 0$ predicts non-membership. The strict inequality assigns ties at the cutoff to non-members, keeping calibration false positives at or below k_ζ. The shared rule yields a numerical cutoff for each model/attack pair. After calibration, scores and membership predictions are fixed for held-out evaluation.

For our $n_0 = 60$, the 5% setting permits at most three calibration false positives and predicts membership for scores strictly above the fourth-largest score. These are finite-calibration operating points. We report the observed test FPR at the same calibration-derived thresholds. The test population comprises 256 candidates, with 64 per suite; tail resolution is set by the non-member count in each group.

Decision metrics and their meanings. At a frozen cutoff, true positives (TP) and false negatives (FN) are members predicted as members and non-members, respectively. False positives (FP) and true negatives (TN) are the analogous counts among true non-members. TPR, FPR, balanced ASR, and observed-mixture accuracy are

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{34}$$

$$\text{ASR} = \tfrac{1}{2}(\text{TPR} + 1 - \text{FPR}), \qquad \text{Accuracy}_{\text{obs}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{35}$$

Balanced ASR is membership-classification accuracy under equal class priors; observed accuracy weights the realized member/non-member mixture. At a cutoff with TPR $= .10$ and FPR $= .05$, balanced ASR is .525; AUROC is obtained from the complete score ranking. The supplementary summary reports these metrics at the frozen nominal-5% operating point. These operational results complement the ranking metric and the low-FPR perspective of prior MIA work (Peng et al., 2026; Carlini et al., 2022).

Table 8: Four-suite closed-loop success (%; higher is better). Each suite contains 200 rollouts; Overall averages all 800. All models are initialized from the same pretrained model and use the main experimental configuration. Bold marks the larger point estimate within each DP-VLA$_{\text{base}}$/DP-VLA$_{\text{MC}}$ budget pair; clipping bounds are shown in parentheses. NP uses the coverage-oriented sampling recipe in Appendix F.3.

ϵ	Recipe	Spatial	Object	Goal	Long	Overall
–	NP	76.0	84.5	58.0	47.5	66.50
10^3	DP-VLA$_{\text{base}}$ (C_L)	0.5	0.0	0.0	0.0	0.13
10^3	DP-VLA$_{\text{MC}}$ (C_S)	0.5	**0.5**	**0.5**	0.0	**0.38**
10^6	DP-VLA$_{\text{base}}$ (C_L)	2.5	0.0	1.5	0.0	1.00
10^6	DP-VLA$_{\text{MC}}$ (C_S)	**5.5**	**4.0**	**7.5**	0.0	**4.25**
10^7	DP-VLA$_{\text{base}}$ (C_L)	1.5	0.0	2.5	0.0	1.00
10^7	DP-VLA$_{\text{MC}}$ (C_S)	**12.0**	**6.0**	**5.5**	**0.5**	**6.00**
10^8	DP-VLA$_{\text{base}}$ (C_L)	1.5	1.5	5.5	0.0	2.13
10^8	DP-VLA$_{\text{MC}}$ (C_S)	**42.5**	**25.5**	**37.0**	**10.0**	**28.75**
10^9	DP-VLA$_{\text{base}}$ (C_L)	12.5	11.0	13.0	0.5	9.25
10^9	DP-VLA$_{\text{MC}}$ (C_S)	**68.0**	**59.0**	**41.5**	**16.5**	**46.25**
10^{10}	DP-VLA$_{\text{base}}$ (C_L)	**45.5**	37.0	34.0	4.5	30.25
10^{10}	DP-VLA$_{\text{MC}}$ (C_S)	40.5	**64.0**	**50.5**	**20.0**	**43.75**
10^{12}	DP-VLA$_{\text{base}}$ (C_L)	45.5	57.0	**43.0**	12.0	39.38
10^{12}	DP-VLA$_{\text{MC}}$ (C_S)	**53.0**	**81.0**	32.5	**28.0**	**48.63**

Table 12 reports the pooled attack decisions at the primary calibration operating point.

Attack uncertainty. For Figure 3, we compute pointwise 95% percentile intervals from 10,000 paired bootstrap resamples of the 256 held-out episodes, each represented by one window. We resample within suite-by-membership strata and apply the same sampled indices to each matched model pair. The trained models, shadow references, and calibration thresholds remain fixed throughout resampling.

G Complete task-success results

Table 8 gives the suite-level values underlying Figure 2. The same seven budget pairs and NP reference are retained.

Task-capability comparisons. The intermediate budgets in Appendix F.2.1 support the matched-budget results and component comparisons in Sections 6.2 and 6.4. Cross-budget comparisons also show improved task capability: DP-VLA$_{\text{MC}}$ reaches 46.25% SR at 10^9, compared with DP-VLA$_{\text{base}}$'s 30.25% at 10^{10}. Under the same window-level privacy definition and accounting, this is higher observed utility with a tenfold smaller cumulative budget.

Diagnostic endpoints. The 10^3 and 10^{12} rows extend the core comparisons toward higher-noise and weak-noise training, respectively. Their suite-level results show task capability at both ends of the tested noise range, while the NP row supplies a separate reference for non-private adaptation. All seven budget pairs remain available for reading the complete utility curve alongside the empirical membership results in Appendix I.

H Component analyses

This appendix expands the transition-region comparisons in Section 6.4, using the same configurations as Table 1. At $\epsilon = 10^8$, the complete 2×2 comparison combines $C \in \{C_L, C_S\}$ with ordinary or corrected AdamW. At $\epsilon = 10^9$, the comparison holds $C = C_S$ fixed and varies the optimizer. Main and Repeat refer to the two training seeds specified in Section 6.4.

Matched component comparisons. All configurations share the membership assignment, public initialization, 10000 updates, and window-level privacy accounting; each model receives 800 roll-

outs from the same evaluation initial states. Within DP-VLA$_{\text{base}}$, changing C varies clipping and absolute noise together. At fixed C, the two methods share the Gaussian noise scale, isolating the corrected optimizer recipe, including its variance floor and coordinatewise scaling (Equation (7)).

H.1 Clipping–optimizer interactions

Table 9 gives the integer success counts underlying the SR values in Table 1. At 10^8, smaller clipping with ordinary AdamW adds 19 and 23 successes in Main and Repeat, respectively. At fixed C_L, MC adds 90 and 81 successes; at fixed C_S, it adds 194 and 95. The combination of C_S and MC achieves the highest success in both complete component comparisons. At 10^9, the same-clipping MC gains are 277 and 210 successes, reproducing the optimizer benefit at the second transition-region budget. These controlled comparisons support the complementary roles of clipping and moment correction described in Section 6.4.

Table 9: Integer success counts out of 800 for the configurations in Table 1. Main and Repeat use the same training identities as that table. The clipping bounds C_L and C_S are defined in Section 6.1; corrected AdamW includes the public variance floor.

ϵ	Method (clipping bound)	Main	Repeat
10^8	DP-VLA$_{\text{base}}$ (C_L)	17	22
10^8	DP-VLA$_{\text{MC}}$ (C_L)	107	103
10^8	DP-VLA$_{\text{base}}$ (C_S)	36	45
10^8	DP-VLA$_{\text{MC}}$ (C_S)	230	140
10^9	DP-VLA$_{\text{base}}$ (C_S)	93	88
10^9	DP-VLA$_{\text{MC}}$ (C_S)	370	298

Suite-level and membership results for the repeat. Table 10 expands the 10^9, C_S Repeat pair in Table 1, reporting all four suites and both membership attacks. Both methods use C_S, the same Gaussian noise multiplier, and 10000 updates; their accounted window budgets are 9.99998783×10^8 at $\delta = 10^{-5}$. MC improves success on all four suites, with the largest gain on Object (13.5% to 57.5%). Both attacks' AUROC and balanced ASR point estimates also increase in this repeat, alongside the utility gain. The suite-level counts aggregate to the 11.00% and 37.25% overall SR reported in Table 1, alongside Action-Ref AUROCs of 0.475 and 0.512.

Table 10: Same-clipping confirmation in the repeated comparison at $\epsilon = 10^9$ and $C = C_S$. Top: task SR (%), with 200 rollouts per suite and 800 overall. Bottom: membership metrics on the same 256 test windows, using the fixed nominal-5% calibration thresholds; ASR, TPR, and actual test FPR are percentages.

Method	Spatial	Object	Goal	Long	Overall
DP-VLA$_{\text{base}}$	15.5	13.5	10.0	5.0	11.00
DP-VLA$_{\text{MC}}$	**41.0**	**57.5**	**41.5**	**9.0**	**37.25**

Method	Attack	AUROC	ASR	TPR	FPR
DP-VLA$_{\text{base}}$	Action-L1	0.447	49.1	11.6	13.3
DP-VLA$_{\text{MC}}$	Action-L1	0.467	51.5	6.6	3.7
DP-VLA$_{\text{base}}$	Action-Ref	0.475	49.1	5.0	6.7
DP-VLA$_{\text{MC}}$	Action-Ref	0.512	51.8	6.6	3.0

I Supplementary attack results

Threshold-based results. Table 12 presents overall test decisions for the two reported attacks at the calibration-determined operating points. Each uses the same 256 test windows, with 121 members and 135 non-members. Entries are percentages at nominal calibration FPR 5%, with the actual test FPR shown explicitly. Values are point estimates for the main experiments under the fixed membership assignment.

Table 11: Empirical membership auditing of the same models in the main experiments. AUROC is on a 0–1 scale; balanced ASR is a percentage at the fixed nominal-5% calibration operating point. Higher values indicate stronger membership discrimination. Action-Ref uses the same NP shadow models across fine-tuning recipes (Appendix F.4).

ϵ	Recipe	Action-L1		Action-Ref	
		AUROC	ASR (%)	AUROC	ASR (%)
–	NP	0.469	49.9	0.556	53.7
10^3	DP-VLA$_{\text{base}}$ (C_L)	0.504	51.4	0.553	50.6
10^3	DP-VLA$_{\text{MC}}$ (C_S)	0.503	49.0	0.574	50.5
10^6	DP-VLA$_{\text{base}}$ (C_L)	0.500	49.2	0.557	50.3
10^6	DP-VLA$_{\text{MC}}$ (C_S)	0.462	50.7	0.519	48.6
10^7	DP-VLA$_{\text{base}}$ (C_L)	0.493	50.8	0.549	51.0
10^7	DP-VLA$_{\text{MC}}$ (C_S)	0.451	47.3	0.507	50.7
10^8	DP-VLA$_{\text{base}}$ (C_L)	0.484	50.3	0.552	53.0
10^8	DP-VLA$_{\text{MC}}$ (C_S)	0.471	49.6	0.548	51.0
10^9	DP-VLA$_{\text{base}}$ (C_L)	0.460	50.6	0.499	49.8
10^9	DP-VLA$_{\text{MC}}$ (C_S)	0.459	47.5	0.472	51.2
10^{10}	DP-VLA$_{\text{base}}$ (C_L)	0.450	48.5	0.487	51.4
10^{10}	DP-VLA$_{\text{MC}}$ (C_S)	0.476	52.2	0.539	55.5
10^{12}	DP-VLA$_{\text{base}}$ (C_L)	0.494	53.4	0.551	52.3
10^{12}	DP-VLA$_{\text{MC}}$ (C_S)	0.452	49.2	0.485	51.9

Table 12: Frozen-threshold decision metrics (%) for Action-L1 and Action-Ref, pooled across the four suites. Balanced ASR is equal-prior membership accuracy; FPR is measured on test non-members. Clipping is C_L for DP-VLA$_{\text{base}}$ and C_S for DP-VLA$_{\text{MC}}$.

Method	ϵ	Action-L1			Action-Ref		
		ASR	TPR	FPR	ASR	TPR	FPR
NP	–	49.9	11.6	11.9	53.7	14.0	6.7
DP-VLA$_{\text{base}}$	10^3	51.4	5.0	2.2	50.6	5.0	3.7
DP-VLA$_{\text{MC}}$	10^3	49.0	1.7	3.7	50.5	3.3	2.2
DP-VLA$_{\text{base}}$	10^6	49.2	6.6	8.1	50.3	5.8	5.2
DP-VLA$_{\text{MC}}$	10^6	50.7	5.8	4.4	48.6	1.7	4.4
DP-VLA$_{\text{base}}$	10^7	50.8	8.3	6.7	51.0	5.8	3.7
DP-VLA$_{\text{MC}}$	10^7	47.3	5.0	10.4	50.7	6.6	5.2
DP-VLA$_{\text{base}}$	10^8	50.3	6.6	5.9	53.0	14.9	8.9
DP-VLA$_{\text{MC}}$	10^8	49.6	7.4	8.1	51.0	4.1	2.2
DP-VLA$_{\text{base}}$	10^9	50.6	11.6	10.4	49.8	3.3	3.7
DP-VLA$_{\text{MC}}$	10^9	47.5	15.7	20.7	51.2	8.3	5.9
DP-VLA$_{\text{base}}$	10^{10}	48.5	6.6	9.6	51.4	5.0	2.2
DP-VLA$_{\text{MC}}$	10^{10}	52.2	14.0	9.6	55.5	14.0	3.0
DP-VLA$_{\text{base}}$	10^{12}	53.4	15.7	8.9	52.3	8.3	3.7
DP-VLA$_{\text{MC}}$	10^{12}	49.2	6.6	8.1	51.9	8.3	4.4