GAUSSIAN DISTILLATION OF FLOW POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow policies are among the strongest robot controllers and are well suited to modelling multimodal action distributions. For robot manipulation tasks, we show empirically that this capability is not always needed and propose the Gaussian Action Readout (GAR), a Gaussian model trained with 0.02% of eligible states on frozen readout features. It samples with one action-network pass, gives closed-form likelihoods, and achieves near-teacher task success at up to twice the inference speed.

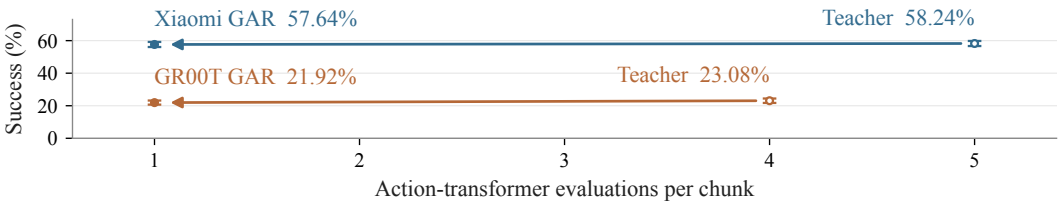


Figure 1: **Near-teacher success with one action-transformer evaluation.** Diagonal GAR uses 6,000 fitting contexts (0.02% of eligible training states), with teacher networks frozen. Each policy is evaluated on 50 RoboCasa365 (Nasiriany et al., 2026) tasks with 50 episodes per task. Bars show pointwise 95% within-task episode-bootstrap intervals (Appendix F.2).

1 INTRODUCTION

Diffusion and flow-matching policies achieve strong performance in robot manipulation (Chi et al., 2023; Black et al., 2025). An appealing feature is their ability to model multimodal action distributions, capturing the different valid ways to act in human demonstrations (Chi et al., 2023). Generating actions, however, requires iterative sampling with repeated network evaluations, making inference computationally costly. Their action likelihoods are also unavailable in closed form. This raises the question: is this distributional flexibility needed for task performance, or could a simpler policy perform just as well while allowing cheaper direct sampling and closed-form likelihoods?

To explore this possibility, we fit separate Gaussians to the translation and rotation action-chunk distributions of Xiaomi-Robotics-1 (Xiaomi Robotics Team et al., 2026) and GR00T (NVIDIA et al., 2025) at fixed inputs. We find small approximation errors for both models (Section 3). These results motivate us to look for a simple, inexpensive way to convert these models into Gaussian policies while retaining their task performance.

A Gaussian is fully specified by its mean and covariance. We derive relations between these parameters and the flow-matching velocity field (Section 4.1). These relations motivate our *Gaussian Action Readout* (GAR), which predicts the Gaussian parameters and binary-command probabilities through small readouts of frozen readout features (Section 4.2). Training only these readouts limits the scope for overfitting, allowing us to fit them using just 6,000 contexts across 300 tasks (0.02% of eligible training states).

The readouts predict action moments with small median errors (Section 5.1). With GAR, one action-transformer evaluation supplies the parameters for direct Gaussian sampling and closed-form likelihoods. On 50 RoboCasa365 tasks (Nasiriany et al., 2026), the resulting Gaussian policies achieve similar task performance to Xiaomi and GR00T (Section 5.2). Inference is faster, with a single action-transformer evaluation replacing five flow steps for Xiaomi and four for GR00T.

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.

Even with a Gaussian action distribution at each decision, a policy can produce different behaviors over a full execution. Empirically, we find that our Gaussian policy produces different object-movement orders in successful RoboCasa executions from the same initial context (Section 5.3).

Our contributions are:

- **Analysis:** We measure Gaussian approximation errors for Xiaomi and GR00T's translation and rotation distributions across sixteen-action chunks, comparing full and diagonal covariance (Section 3).
- **Method:** Gaussian Action Readout (GAR) predicts action moments and command probabilities through small readouts of frozen readout features. One action-transformer evaluation enables direct sampling and closed-form likelihoods (Section 4).
- **Experiments:** Gaussian policies retain near-teacher success on 50 RoboCasa365 tasks, with inference speedups of $1.91\times$ for Xiaomi and $1.57\times$ for GR00T (Sections 5.1–5.2). A Gaussian policy can produce diverse behaviors despite its unimodal action distribution (Section 5.3).

2 Related Work

Flow-based vision-language-action models Xiaomi-Robotics-1 (Xiaomi Robotics Team et al., 2026) combines a Qwen3-VL backbone (Bai et al., 2025) with a diffusion transformer in a Mixture-of-Transformers architecture. GR00T N1.5 (NVIDIA et al., 2025; NVIDIA, 2025) uses an Eagle 2.5 backbone (Chen et al., 2025) and a diffusion transformer that alternates self-attention over robot-state and action tokens with cross-attention to vision-language features. Both generate action chunks through iterative flow sampling, requiring repeated action-transformer evaluations.

Simpler action prediction OpenVLA-OFT (Kim et al., 2025) predicts continuous action chunks directly, using an L1 loss against demonstration actions during training. Its task performance is comparable to that of a diffusion head. Chen et al. (2026) improve one-step sampling from flow-matching policies by drawing training inputs more frequently near the pure-noise endpoint. AdaFlow (Hu et al., 2024) uses a learned variance estimate to adapt the number of flow-sampling steps. Sheebaelhamd et al. (2025) train a quantization-free transformer (Q-FAT) on demonstrations to predict Gaussian-mixture action distributions with explicit likelihoods. They report one active density mode at approximately 70% of rollout steps (Sheebaelhamd et al., 2025) in Kitchen (Gupta et al., 2020) and UR3 BlockPush (Kim et al., 2022). Pan et al. (2026) argue that multimodality does not fully explain the observed advantage of flow policies and introduce Minimal Iterative Policy (MIP), a lightweight two-step regression policy that approximately matches flow-policy performance.

Direct regression and MIP produce point predictions, while Q-FAT learns Gaussian-mixture action distributions from demonstrations. We instead ask how close pretrained flow policies are to Gaussian, and whether we can easily convert them into Gaussian policies enabling one-step sampling and closed-form likelihoods.

Policy distillation Distillation reduces sampling cost by training a student to reproduce a teacher's outputs. Consistency Policy (Prasad et al., 2024), One-Step Diffusion Policy (Wang et al., 2025) and SnapFlow (Luan et al., 2026) use this approach to generate actions with one or a few network evaluations. Refined Policy Distillation (Jülg et al., 2025) trains a separate student both to match teacher actions and to maximize task rewards. Its Proximal Policy Distillation (Spigler, 2025) baseline fits a Gaussian to ten teacher actions at each input and uses a KL penalty toward this distribution within PPO. This provides a precedent for Gaussian approximation of a teacher policy, with an input-dependent student mean and a diagonal covariance shared across inputs in the released implementation.

These methods train a complete student action network to imitate the teacher. GAR reuses the teacher's frozen networks and trains only small readouts to predict a Gaussian action distribution in one pass.

3 GAUSSIANITY OF PRETRAINED POLICIES

A Gaussian approximation would allow continuous actions to be sampled directly from their mean and covariance. We first test how well a single Gaussian approximates the continuous-action distributions of pretrained flow policies at fixed observations.

Approximation error. At each of 300 fixed observations per teacher, we fit separate Gaussians to translation and rotation commands and evaluate them on held-out teacher samples. We analyze translation and rotation separately because they have different physical units: combining millimetres and degrees in one distance would require choosing their relative weighting. Each Gaussian models the three translation or rotation coordinates across all sixteen actions, giving $16 \times 3 = 48$ coordinates. We compare the teacher and Gaussian distributions along many one-dimensional projections. For each projection, we measure the minimum average displacement needed to match the two distributions. Averaging these distances gives the sliced 1-Wasserstein distance (Villani, 2009; Bonneel et al., 2015). Small shifts in command values therefore incur small errors, whereas KL divergence can be large even for small shifts between narrowly concentrated distributions (Appendix D).

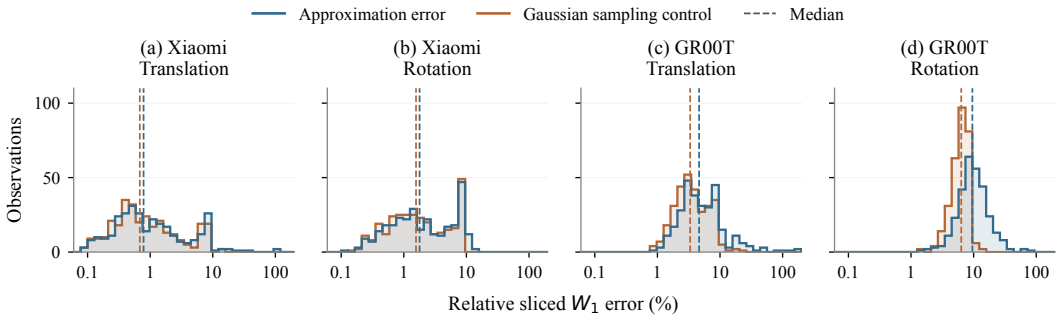


Figure 2: Full-covariance Gaussian approximation across sixteen-action chunks at 300 fixed observations per teacher. Histograms show relative sliced 1-Wasserstein errors (%) for the 48-dimensional translation and rotation groups, normalized by the teacher's projected command magnitude. Gaussian sampling controls measure finite-sample error. Dashed lines mark medians; logarithmic horizontal axes retain all observations, and lower errors are better.

Finite samples control. Even when the underlying action distribution is Gaussian, fitting and evaluating a Gaussian on finite samples can give a nonzero approximation error. We estimate this baseline error by drawing samples from each fitted Gaussian, fitting a new Gaussian to one subset, and calculating its distance from held-out samples. Figure 2 compares the approximation errors with this baseline.

With full covariance, Xiaomi's median approximation errors are close to the corresponding controls, whereas GR00T's are somewhat larger (Figure 2). Diagonal covariance gives larger median approximation errors for both models (Table 5 and Figure 8 in Appendix D.2). These results suggest that the continuous-action distributions of these flow policies are already close to Gaussian.

4 GAUSSIAN DISTILLATION

We study a simple method for distilling a pretrained flow policy (the teacher) into a Gaussian policy. Our Gaussian Action Readout (GAR) predicts the Gaussian mean and covariance from the teacher's frozen readout features. Let c denote the policy context: the task instruction and the image and robot-state observations supplied to the policy, including any observation history. Let $a = (m, b)$ denote an action chunk, with continuous coordinates m and binary commands b. Let $\pi_T^m(\cdot \mid c)$ denote the teacher's continuous-action marginal, with mean $\mu(c)$ and covariance $\Sigma(c)$. We approximate this distribution by

$$\pi_T^m(m \mid c) \approx \mathcal{N}(m; \mu(c), \Sigma(c)) \,. \tag{1}$$

4.1 Moment Information in Flow Fields

A Gaussian is determined by its mean and covariance, so we ask what the flow-matching velocity field tells us about these parameters.

Mean. Fix c and draw $m \sim \pi_T^m(\cdot \mid c)$ independently of $\epsilon \sim \mathcal{N}(0, I)$. The teacher policies are trained with flow matching using a linear interpolation from noise to actions (Lipman et al., 2023). The interpolation and its associated velocity field are

$$m_t = (1-t)\epsilon + tm, \qquad v(m_t, t \mid c) = \mathbb{E}[m - \epsilon \mid m_t, c]. \tag{2}$$

At $t = 0$, the input is independent of the target, giving

$$\epsilon + v(\epsilon, 0 \mid c) = \mathbb{E}[m \mid c] = \mu(c). \tag{3}$$

Thus, a one-unit Euler step recovers the mean, without a Gaussian assumption (see also Mei et al., 2026).

Diagonal variance. Let i index the continuous coordinates of the action chunk. For a Gaussian target with diagonal covariance $\Sigma(c) = \mathrm{diag}(\sigma_i^2(c))$, $m - \epsilon$ and m_t are jointly Gaussian. Evaluating the conditional expectation in Equation (2) gives (Appendix A)

$$\begin{aligned} v_i(m_t, t \mid c) &= \mu_i(c) + \alpha_i(t, c)\big(m_{t,i} - t\mu_i(c)\big), \\ \alpha_i(t, c) &= \frac{t\sigma_i^2(c) - (1-t)}{(1-t)^2 + t^2\sigma_i^2(c)}. \end{aligned} \tag{4}$$

For $0 < t < 1$, a known mean and one velocity query determine α_i whenever $m_{t,i} \neq t\mu_i(c)$, and hence

$$\sigma_i^2(c) = \frac{(1-t)\,[1 + (1-t)\alpha_i(t, c)]}{t\,[1 - t\alpha_i(t, c)]}. \tag{5}$$

At $t = 0$, $\alpha_i = -1$, so variance is absent from the velocity.

4.2 Gaussian Action Readouts

At $t = 0$, the velocity determines the mean but contains no information about the variance. We ask whether the teacher's internal features still contain enough information to predict both from a single evaluation at $t = 0$. Let $u(c, x, t)$ denote the teacher's *readout features*: the frozen features immediately before its final velocity projection, where x is the noisy action input. For a fixed noise input $\epsilon_\star$, GAR predicts the distribution parameters from $u(c, \epsilon_\star, 0)$ (Figure 3).

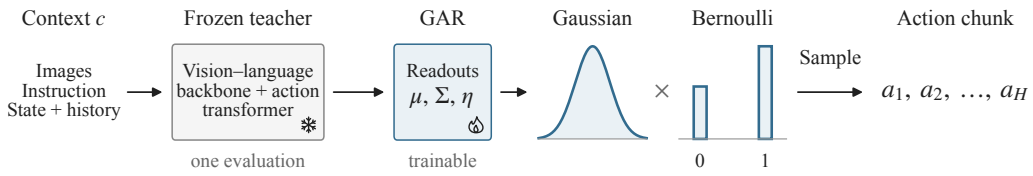


Figure 3: GAR predicts Gaussian parameters and Bernoulli probabilities from one evaluation of the frozen action transformer. Only small readouts are trained; sampling continuous actions and binary commands produces an action chunk.

Predicting the mean. The teacher reads out velocity through a linear projection,

$$v(x, t \mid c) = W_v u(c, x, t) + b_v. \tag{6}$$

Combining this projection with the mean identity in Equation (3) gives

$$\begin{aligned} \mu(c) &= \epsilon_\star + v(\epsilon_\star, 0 \mid c) \\ &= W_v u(c, \epsilon_\star, 0) + (b_v + \epsilon_\star). \end{aligned} \tag{7}$$

At fixed $\epsilon_\star$, the mean is thus an affine function of the features. This motivates fitting an affine readout of $u(c, \epsilon_\star, 0)$ to the mean of teacher samples.

Predicting the variance. Under the diagonal Gaussian model, the mean and an intermediate-time velocity query determine each variance through

$$\begin{aligned} \alpha_i(t,c) &= \frac{[W_v u(c,x,t) + b_v]_i - \mu_i(c)}{x_i - t\mu_i(c)}, \\ \sigma_i^2(c) &= \frac{(1-t)\,[1+(1-t)\alpha_i(t,c)]}{t\,[1-t\alpha_i(t,c)]}, \end{aligned} \tag{8}$$

for $0 < t < 1$ and $x_i \neq t\mu_i(c)$. These relations connect variance to the intermediate-time features once the mean and query input are known. Although variance is absent from the time-zero velocity, we ask whether it can still be recovered from the time-zero features. In both teachers, these features are nearly parallel to those at an intermediate time $s \in (0,1)$, for example $s = 0.1$, when c and x are fixed (Appendix A.2). This motivates learning a variance readout from the time-zero features:

$$\sigma_\theta^2(c) = g_\theta(u(c,\epsilon_\star,0))\,, \qquad \Sigma_\theta(c) = \mathrm{diag}\big(\sigma_\theta^2(c)\big)\,, \tag{9}$$

where g_θ is an MLP with one hidden layer of 128 units and positive outputs. We evaluate the accuracy of the predicted moments in Section 5.1. In Section 5.2, we compare the task performance of GAR policies using diagonal and full covariance with that of their teachers.

Predicting binary commands. Each binary command has two possible outcomes, but its Bernoulli distribution is fully specified by one parameter: its mean $p_j(c) = \Pr(b_j = 1 \mid c)$. We represent this distribution by thresholding a Gaussian latent at zero, a construction that also allows correlated latents (Appendix C.2). In GAR, we use independent Gaussian latents with variance fixed to one. For each command j, an affine readout of the same readout features predicts the latent mean $\eta_{\theta,j}(c)$, giving activation probability $p_{\theta,j}(c) = \Phi(\eta_{\theta,j}(c))$, where Φ is the standard normal cumulative distribution function.

Fitting. The identities above motivate predicting action moments through small readouts of the teacher's readout features. Training only these readouts limits the scope for overfitting the small fitting set. To learn these readouts, we estimate teacher means, covariances and command frequencies from samples at each fitting context. We fit the mean by least squares, the covariance by normalized squared error, and command probabilities by binary cross-entropy. Covariance targets are centered on the empirical teacher mean, so mean-prediction errors do not inflate the target covariance. All fitting uses cached features and teacher statistics (Appendix C.2).

Execution. At deployment, one action-transformer evaluation supplies the readouts, and we independently sample

$$\begin{aligned} m &\sim \mathcal{N}(\mu_\theta(c), \Sigma_\theta(c))\,, \\ z_j &\sim \mathcal{N}(\eta_{\theta,j}(c), 1)\,, \qquad b_j = \mathbf{1}\{z_j \geq 0\}. \end{aligned} \tag{10}$$

The resulting action chunk $a = (m, b)$ requires no velocity-field integration.

Likelihood. Unless otherwise stated, GAR uses diagonal covariance for continuous actions, motivated by the argument above. With binary commands independent of one another and of continuous actions given c, the model has the following closed-form likelihood:

$$p_\theta(m, b \mid c) = \mathcal{N}(m; \mu_\theta(c), \Sigma_\theta(c)) \prod_j p_{\theta,j}(c)^{b_j}\,[1 - p_{\theta,j}(c)]^{1-b_j}\,, \tag{11}$$

where $p_{\theta,j}(c) = \Phi(\eta_{\theta,j}(c))$ and Φ is the standard normal cumulative distribution function. Once the parameters have been computed, this expression scores candidate action chunks without further network evaluations or flow integration.

5 EXPERIMENTS

We evaluate the accuracy of GAR's predicted action moments (Section 5.1) and test whether its Gaussian policies retain the teachers' task performance while reducing inference cost (Section 5.2). Repeated executions from the same initial context then test whether Gaussian sampling can produce different successful behaviors (Section 5.3).

Our main experiments use RoboCasa365, a simulation benchmark with 365 household manipulation tasks across 2,500 kitchen environments (Nasiriany et al., 2026). Its tasks range from basic skills, such as opening drawers, to multistep activities, such as preparing coffee. The teachers, Xiaomi-Robotics-1 and GR00T N1.5, were trained on Human300 demonstrations covering 300 of these tasks. Figure 5 shows successful GAR executions on several of these tasks.

Unless stated otherwise, we fit the readouts on 6,000 contexts, taking four evenly spaced interior states from each of five randomly selected human demonstrations per task (Appendix C.1). Control evaluation then compares the policies on the 50-task target suite, which includes seen tasks and unseen composite tasks, with 50 episodes per policy per task.

We also evaluate GAR with the Xiaomi-Robotics-0 teacher (Cai et al., 2026) on the ten LIBERO-Long tasks (Liu et al., 2023). These are multi-step manipulation tasks, such as placing a bowl in a drawer and closing it.

5.1 MOMENT RECOVERY

We first test how accurately small readouts predict the teacher's mean and marginal standard deviations (SDs). With 6,000 fitting contexts, median errors in the predicted mean and SDs are below 10% of the norm of the teacher's mean action chunk for both teachers (Table 1). For GR00T, median variance predictions are close to the teacher estimates, but individual predictions still show substantial scatter. For Xiaomi, median predictions broadly track the teacher estimates, but the largest variances are systematically underestimated. Individual predictions also show greater scatter (Figure 4).

Table 1: Moment prediction on 600 held-out contexts per teacher, using estimates from teacher samples as references. Entries give median [25th, 75th percentile] errors, with lower values better. Mean and SD errors are $100\|\mu_\theta(c) - \widehat{\mu}_c\|_2 / \|\widehat{\mu}_c\|_2$ and $100\|\sigma_\theta(c) - \widehat{\sigma}_c\|_2 / \|\widehat{\mu}_c\|_2$, respectively, where σ denotes the vector of marginal SDs. Diagonal and full variants share the mean readout. Errors use normalized coordinates for Xiaomi and decoded coordinates for GR00T, so comparisons are within each teacher. More fitting contexts reduce median diagonal-SD error, with little change in mean error.

Teacher	Fitting contexts	Mean error (%)	SD error (%)	
			Diagonal	Full
Xiaomi	6,000	4.63 [3.09, 7.73]	5.58 [3.61, 9.03]	2.91 [1.72, 5.95]
Xiaomi	24,000	4.54 [3.00, 7.37]	4.35 [2.69, 7.37]	2.92 [1.56, 5.57]
GR00T	6,000	8.43 [5.98, 12.91]	9.76 [6.51, 16.42]	8.09 [5.36, 13.42]
GR00T	24,000	8.29 [5.86, 12.55]	8.47 [5.74, 14.65]	7.61 [5.08, 12.21]

Data requirements. To test whether more supervision improves moment recovery, we increase the fitting set from 6,000 to 24,000 contexts using additional demonstrations. Median diagonal SD error drops by 1.23 percentage points for Xiaomi and 1.29 for GR00T, while median mean error changes by less than 0.2 percentage points.

Fitting cost. Fitting the readouts is fast once readout features and targets are cached. For Xiaomi's 6,000-context diagonal model, the recorded fitting times total approximately 115 seconds, including the initial joint fit that supplies the command readout but excluding teacher sampling, feature extraction and untimed setup (Appendix C.4).

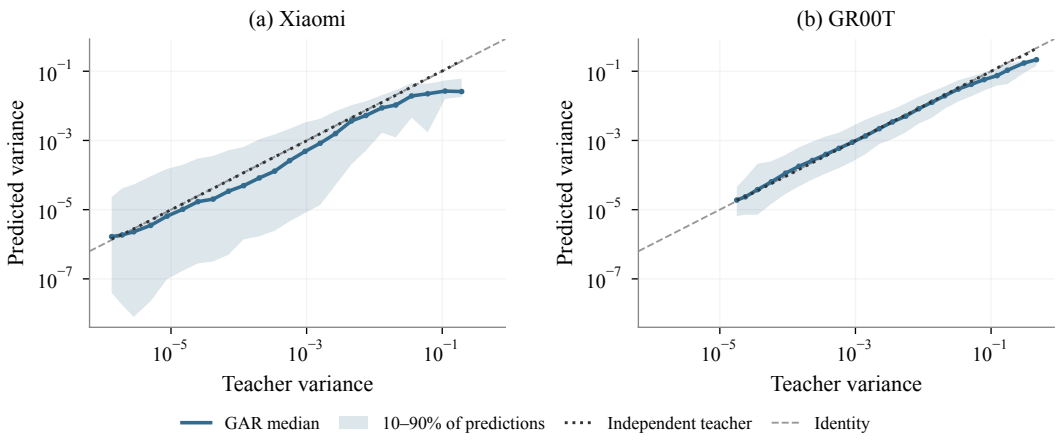


Figure 4: Variance predictions from diagonal GAR fitted on 6,000 contexts, evaluated on 600 held-out contexts per teacher. Predictions track teacher variances but underestimate the largest values, especially for Xiaomi. Curves show medians within logarithmic teacher-variance bins; shading spans the 10th–90th percentiles of predictions, not confidence intervals. The dotted curve compares variance estimates from 128 independent teacher samples against 384-sample targets, providing a finite-sampling reference. Appendix E.1 gives all predictions and upper-tail errors.

5.2 Task Performance

We now test whether a Gaussian policy can retain the task performance of iterative flow sampling. With 6,000 fitting contexts, diagonal GAR achieves success rates close to those of both teachers using one action-transformer evaluation (Table 2). It also achieves higher observed success than raw one-step execution for both teachers. Xiaomi remains close to its teacher across all three task groups, while GR00T's deficit is concentrated in seen composite tasks. GR00T's small gap on unseen composite tasks is less informative because teacher success is only 5%.

Table 2: RoboCasa365 success (%) across 50 tasks and 2,500 unpaired episodes per policy; higher is better. Groups contain 18 seen atomic, 16 seen composite and 16 unseen composite tasks, with 50 episodes per task. Raw one-step uses the teacher without fitted readouts. Mean-only GAR retains stochastic binary commands; Gaussian variants use predicted variance. Context counts refer to mean and covariance fitting; command readouts use 6,000 contexts. Execution, fitting and uncertainty are detailed in Appendices B, C and F.2, respectively.

Teacher	Fitting contexts	Task group	Teacher	Raw one-step	Mean only	Diagonal	Full
Xiaomi	6,000	**Overall**	**58.24**	**55.76**	**56.64**	**57.64**	**57.28**
		Atomic seen	81.11	81.22	78.44	80.56	81.67
		Composite seen	58.75	52.63	58.13	58.00	57.38
		Composite unseen	32.00	30.25	30.63	31.50	29.75
	24,000	**Overall**	**58.24**	**55.76**	**56.44**	**57.72**	**56.44**
		Atomic seen	81.11	81.22	80.11	81.00	81.11
		Composite seen	58.75	52.63	54.50	56.75	55.25
		Composite unseen	32.00	30.25	31.75	32.50	29.88
GR00T	6,000	**Overall**	**23.08**	**19.28**	**19.72**	**21.92**	**20.32**
		Atomic seen	46.00	43.00	44.33	45.78	43.89
		Composite seen	15.38	8.13	7.75	11.75	11.50
		Composite unseen	5.00	3.75	4.00	5.25	2.63
	24,000	**Overall**	**23.08**	**19.28**	**20.20**	**21.44**	**21.16**
		Atomic seen	46.00	43.00	43.22	44.11	44.67
		Composite seen	15.38	8.13	10.50	12.63	12.63
		Composite unseen	5.00	3.75	4.00	4.75	3.25



Scrub a cutting board | Rinse a sink | Pack lunches | Store leftovers

Figure 5: **One-pass Gaussian control on RoboCasa365.** Selected successful executions using GR00T full-covariance readouts (left two) and Xiaomi diagonal readouts (right two). Each action chunk uses one action-transformer evaluation.

Mean-only and full-covariance ablations. Executing the mean replaces the continuous-action Gaussian with a point mass while keeping command sampling unchanged. Diagonal sampling has higher observed success in all four comparisons, supporting a benefit from retaining continuous-action variance. Full covariance does not improve observed success over diagonal sampling. Its lower marginal-SD error does not establish a better covariance fit, since this metric omits correlation errors. Within the diagonal variant, increasing the fitting set reduces marginal-SD error for both teachers but does not consistently improve success.

Moment-prediction errors. Some GR00T failures may reflect inaccurate moment predictions rather than a limitation of the diagonal Gaussian policy family. On eight matched ScrubCuttingBoard scenes, replacing the learned continuous-action moments with estimates from repeated teacher samples raises observed success from 1/8 to 5/8, with the command predictor unchanged. This improvement suggests that better moment prediction can recover some failures in these scenes while retaining a diagonal Gaussian policy (Appendix F.4).

Inference cost. GAR replaces five action-transformer evaluations for Xiaomi and four for GR00T with one. For the 6,000-context diagonal policies, batch-one latency falls from 220.25 to 115.36 ms for Xiaomi ($1.91\times$) and from 68.01 to 43.30 ms for GR00T ($1.57\times$). These timings include vision-language processing, which limits the total speedup (Appendix F.1).

LIBERO-Long. To test whether GAR retains near-teacher performance beyond RoboCasa365, we evaluate it on LIBERO-Long using the Xiaomi-Robotics-0 teacher. Across all ten tasks, diagonal GAR fitted on 6,000 contexts achieves 95.4% success, compared with 96.6% for the teacher and 96.0% for raw one-step execution. These results extend the near-teacher performance of Gaussian control to a second benchmark, where raw one-step execution also performs similarly. Appendices B.2, C.3 and F.5 give the evaluation protocol, fitting procedure and per-task results, respectively.

5.3 Multimodal behavior

Diffusion policies generate actions by iterating in denoising time (Chi et al., 2023). A single Gaussian at each decision can produce multimodal behavior through diffusion in physical time.

Diffusion in physical time. For direct position updates, Gaussian action sampling gives

$$s_{k+1} = s_k + \mu(s_k) + \Sigma^{1/2}\xi_k, \qquad \xi_k \overset{\text{iid}}{\sim} \mathcal{N}(0, I). \tag{12}$$

This update is a discretized diffusion in physical time (Appendix G.1).

Each transition is Gaussian conditional on the current state, but the distribution of later states can be multimodal. A nonlinear mean μ can amplify small random deviations into distinct routes. In our constructed example, trajectories from the same start separate around an obstacle before returning toward a common goal (Figure 6).

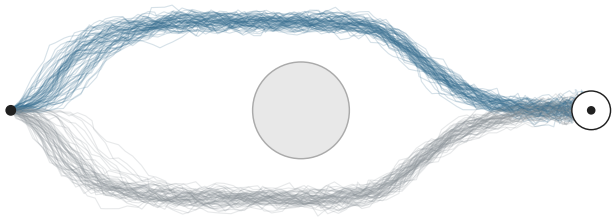


Figure 6: Gaussian actions, two routes: constructed example. All 128 rollouts start at the left dot, pass on either side of the obstacle, and approach the goal marked by the right circle. Actions follow $a_k \sim \mathcal{N}(\mu(s_k), \Sigma)$, with $s_{k+1} = s_k + a_k$; the state-dependent mean allows trajectories to follow different routes. Appendix G.1 specifies the policy.

Different successful movement orders. Xiaomi's Diagonal policy also produces distinct successful movement orders from the same initial context. In PackIdenticalLunches, the robot must place one garlic and one chicken drumstick into each container. Three of 32 executions succeed, with two different object-movement orders among these successes (Figure 7; Appendix G.2).

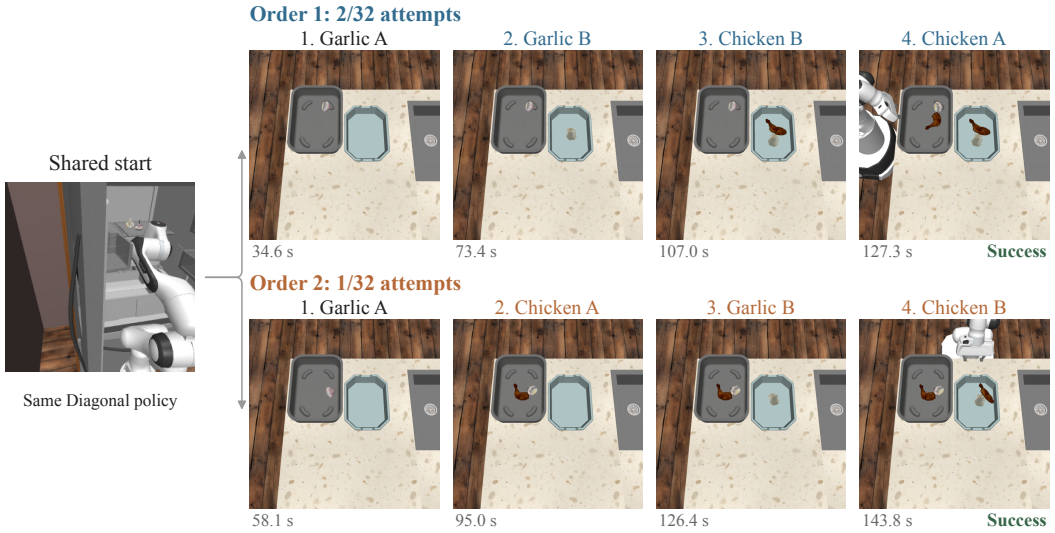


Figure 7: Xiaomi's diagonal GAR produces two successful movement orders from the same start. The task is to pack one garlic and one chicken drumstick into each container; A and B distinguish objects of each type. Left: shared initial policy-camera view. Right: recorded simulator states after successive transfers, rendered from a fixed external camera and labeled with elapsed physical time. The two orders account for all three successes among 32 attempts: two follow the upper order and one follows the lower order. (Appendix G.2).

6 Conclusion

Gaussian Distillation replaces iterative flow sampling with a Gaussian Action Readout trained on frozen readout features. Small readouts fitted on 0.02% of eligible training states predict action means, covariances and binary-command probabilities, enabling direct sampling and closed-form likelihoods from one action-transformer evaluation. On 50 RoboCasa365 tasks, GAR retains near-teacher success for Xiaomi and GR00T while reducing inference latency. Retaining the predicted continuous-action variance gives higher observed success than executing the mean in all four tested comparisons.

A Gaussian action distribution at each decision can still produce distinct behaviors over a full execution through state feedback. The diagonal policy demonstrates this through distinct object-movement orders in successful executions from the same initial context.

Reproducibility Statement

Code and readout checkpoints are available at https://anonymous.4open.science/r/gaussian-action-readout-E4BA/. Appendix B specifies the checkpoints and evaluation protocols. Appendix C gives data splits, teacher sampling and readout fitting. Appendices D and E describe the Gaussian-approximation and moment-prediction measurements. Appendices F and G give the task-performance and diversity protocols, including uncertainty calculations.

AI Use Statement

We used generative AI tools to assist with hypothesis development, experiment design, mathematical derivations, code, analysis, and manuscript preparation. The authors directed the research and take responsibility for the final text, claims, and results.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631. 2

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://roboticsproceedings.org/rss21/p010.html. 1

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. doi: 10.1007/s10851-014-0506-3. URL https://doi.org/10.1007/s10851-014-0506-3. 3, 19

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-Robotics-0: An Open-Sourced Vision-Language-Action Model with Real-Time Execution. *arXiv preprint arXiv:2602.12684*, 2026. URL https://arxiv.org/abs/2602.12684. 6

Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Max Ehrlich, Tong Lu, Limin Wang, Bryan Catanzaro, Jan Kautz, Andrew Tao, Zhiding Yu, and Guilin Liu. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 91077–91100, 2025. doi: 10.52202/085713-3047. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/833295cb9278a3ba973842a94ea68e3c-Abstract-Conference.html. 2

Yitong Chen, Shiduo Zhang, Jingjing Gong, and Xipeng Qiu. Let it be simple: One-step action generation for vision-language-action models. *arXiv preprint arXiv:2606.05737*, 2026. URL https://arxiv.org/abs/2606.05737. 2

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://roboticsproceedings.org/rss19/p026.html. 1, 8

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1025–1037. PMLR, 2020. URL https://proceedings.mlr.press/v100/gupta20a.html. 2

Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. AdaFlow: Imitation learning with variance-adaptive flow-based policies. In *Advances in Neural Information Processing Systems*, volume 37, pp. 138836–138858, 2024. doi: 10.52202/079017-4405. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fa9cfdb49f7de3dee54007f84c0745b1-Abstract-Conference.html. 2

Tobias Jülg, Wolfram Burgard, and Florian Walter. Refined policy distillation: From VLA generalists to RL experts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 11677–11684, 2025. doi: 10.1109/IROS60139.2025.11246761. URL https://doi.org/10.1109/IROS60139.2025.11246761. 2

Jigang Kim, J. hyeon Park, Daesol Cho, and H. Jin Kim. Automating reinforcement learning with example-based resets. *IEEE Robotics and Automation Letters*, 7(3):6606–6613, 2022. doi: 10.1109/LRA.2022.3173039. URL https://doi.org/10.1109/LRA.2022.3173039. 2

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.017. URL https://roboticsproceedings.org/rss21/p017.html. 2

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t. 4

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html. 6

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. 17, 18

Wuyang Luan, Junhui Li, Weiguang Zhao, Wenjian Zhang, Tieru Wu, and Rui Ma. SnapFlow: One-step action generation for flow-matching VLAs via progressive self-distillation. *arXiv preprint arXiv:2604.05656*, 2026. URL https://arxiv.org/abs/2604.05656. 2

Jiří Matoušek. On the L_2-discrepancy for anchored boxes. *Journal of Complexity*, 14(4):527–556, 1998. doi: 10.1006/jcom.1998.0489. URL https://doi.org/10.1006/jcom.1998.0489. 19

Yunpeng Mei, Jiakai He, Hongjie Cao, Chenyu Wang, Xiaowen Zhu, Yihan Zhou, Jiamin Wang, Chenbo Xin, Peng Cheng, Yuxuan Yang, Yijie Wang, Xinhu Zheng, Gao Huang, Jie Chen, and Gang Wang. Potential-guided flow matching for vision-language-action policy improvement. *arXiv preprint arXiv:2606.04968*, 2026. URL https://arxiv.org/abs/2606.04968. 4

Soroush Nasiriany, Sep Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A Large-Scale Simulation Framework for Training and Benchmarking Generalist Robots. In *International Conference on Learning Representations*, volume 2026, pp. 98643–98667, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05003fdb1e9562ab0c0a9719ea4de10-Abstract-Conference.html. 1, 6, 15

NVIDIA. GR00T-N1.5-3B: Model card, 2025. URL https://huggingface.co/nvidia/GR00T-N1.5-3B. 2

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734. 1, 2, 15

Chaoyi Pan, Giridharan Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, and Max Simchowitz. Much ado about noising: Dispelling the myths of generative robotic control. In *International Conference on Learning Representations*, volume 2026, pp. 90575–90614, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9299ca9ed58731945e934adb5b71728c-Abstract-Conference.html. 2

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.071. URL https://roboticsproceedings.org/rss20/p071.html. 2

Ziyad Sheebaelhamd, Michael Tschannen, Michael Muehlebach, and Claire Vernade. Quantization-free autoregressive action transformer. In *Advances in Neural Information Processing Systems*, volume 38, pp. 81187–81212, 2025. doi: 10.52202/085713-2712. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7499cd7bf0b08d810807c191bc177545-Abstract-Conference.html. 2

I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. doi: 10.1016/0041-5553(67)90144-9. URL https://doi.org/10.1016/0041-5553(67)90144-9. 19

Giacomo Spigler. Proximal policy distillation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=WfVXe88oMh. 2

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009. doi: 10.1007/978-3-540-71050-9. URL https://doi.org/10.1007/978-3-540-71050-9. 3, 19

Zhendong Wang, Max Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, Ming-Yu Liu, and Yu Zeng. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63399–63416. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25ba.html. 2

Xiaomi Robotics. Xiaomi-Robotics-1-RoboCasa365: checkpoint and evaluation instructions, 2026a. URL https://huggingface.co/XiaomiRobotics/Xiaomi-Robotics-1-RoboCasa365. 15

Xiaomi Robotics. Xiaomi-Robotics-0-LIBERO: checkpoint and model card, 2026b. URL https://huggingface.co/XiaomiRobotics/Xiaomi-Robotics-0-LIBERO. 15

Xiaomi Robotics Team, Jun Guo, Piaopiao Jin, Jason Li, Peiyan Li, Yingyan Li, Futeng Liu, Wanli Peng, Optimus Qin, Yifei Su, et al. Xiaomi-Robotics-1: Scaling Vision-Language-Action Models with over 100K Hours of Real-World Trajectories. *arXiv preprint arXiv:2607.15330*, 2026. URL https://arxiv.org/abs/2607.15330. 1, 2, 15

A GAUSSIAN FLOW THEORY AND FEATURE ANALYSIS

For a Gaussian target, the flow-matching velocity is affine in its input. Fix the context c, write $\boldsymbol{\mu} = \boldsymbol{\mu}(c)$ and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(c)$, and assume $\boldsymbol{\Sigma}$ is positive definite. Let $\mathbf{M} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be independent, with $\mathbf{X}_t = (1-t)\mathbf{Z} + t\mathbf{M}$. The velocity field is

$$\mathbf{v}(\mathbf{x}, t, c) = \mathbb{E}[\mathbf{M} - \mathbf{Z} \mid \mathbf{X}_t = \mathbf{x}, c]. \tag{13}$$

The interpolated action and target velocity are jointly Gaussian, with

$$\begin{aligned} \mathbb{E}[\mathbf{X}_t \mid c] &= t\boldsymbol{\mu}, & \mathrm{Cov}(\mathbf{X}_t \mid c) &= \mathbf{C}_t = (1-t)^2\mathbf{I} + t^2\boldsymbol{\Sigma}, \\ \mathbb{E}[\mathbf{M} - \mathbf{Z} \mid c] &= \boldsymbol{\mu}, & \mathrm{Cov}(\mathbf{M} - \mathbf{Z}, \mathbf{X}_t \mid c) &= t\boldsymbol{\Sigma} - (1-t)\mathbf{I}. \end{aligned} \tag{14}$$

For jointly Gaussian vectors $\mathbf{Y}$ and $\mathbf{X}$, with $\mathrm{Cov}(\mathbf{X})$ invertible, the conditional mean is

$$\mathbb{E}[\mathbf{Y} \mid \mathbf{X} = \mathbf{x}] = \mathbb{E}[\mathbf{Y}] + \mathrm{Cov}(\mathbf{Y}, \mathbf{X})\,\mathrm{Cov}(\mathbf{X})^{-1}\big(\mathbf{x} - \mathbb{E}[\mathbf{X}]\big).$$

Applying this formula at fixed c, with $\mathbf{Y} = \mathbf{M} - \mathbf{Z}$ and $\mathbf{X} = \mathbf{X}_t$, gives

$$\begin{aligned} \mathbf{v}(\mathbf{x}, t, c) &= \boldsymbol{\mu} + \mathbf{A}_t(\mathbf{x} - t\boldsymbol{\mu}), \\ \mathbf{A}_t &= \big[t\boldsymbol{\Sigma} - (1-t)\mathbf{I}\big]\mathbf{C}_t^{-1}. \end{aligned} \tag{15}$$

For $0 < t < 1$, solving for the covariance gives

$$\boldsymbol{\Sigma} = \frac{1-t}{t}\big[\mathbf{I} - t\mathbf{A}_t\big]^{-1}\big[\mathbf{I} + (1-t)\mathbf{A}_t\big]. \tag{16}$$

Thus the field is affine in $\mathbf{x}$, with coefficients determined by the target mean, covariance and interpolation time.

At the noise endpoint, $\mathbf{X}_0 = \mathbf{Z}$ is independent of $\mathbf{M}$ given c, so

$$\mathbf{v}(\mathbf{x}, 0, c) = \mathbb{E}[\mathbf{M} \mid c] - \mathbf{x}, \qquad \mathbf{x} + \mathbf{v}(\mathbf{x}, 0, c) = \mathbb{E}[\mathbf{M} \mid c]. \tag{17}$$

This identity also holds for non-Gaussian targets with finite second moments. A single Euler step of size one from $t = 0$ therefore returns the conditional mean.

Covariance affects the Gaussian velocity at intermediate times but is absent at $t = 0$. To motivate predicting variance from time-zero features, Appendix A.2 compares the teacher's internal features across these times at fixed context and noisy input.

A.1 DIAGONAL VARIANCE RECOVERY

For diagonal $\Sigma(c)$, each velocity coordinate depends only on the corresponding action coordinate. Given the mean $\mu(c)$, a velocity query at $0 < t < 1$ determines

$$\alpha_i(t, c) = \frac{v_i(m_t, t \mid c) - \mu_i(c)}{m_{t,i} - t\mu_i(c)} = \frac{t\sigma_i^2(c) - (1-t)}{(1-t)^2 + t^2\sigma_i^2(c)}, \tag{18}$$

provided $m_{t,i} \neq t\mu_i(c)$. Multiplying by the denominator and collecting the terms containing $\sigma_i^2(c)$ gives

$$t\big[1 - t\alpha_i(t, c)\big]\sigma_i^2(c) = (1-t)\big[1 + (1-t)\alpha_i(t, c)\big].$$

Hence,

$$\sigma_i^2(c) = \frac{(1-t)\big[1 + (1-t)\alpha_i(t, c)\big]}{t\big[1 - t\alpha_i(t, c)\big]}.$$

Thus, once the mean is known, one intermediate-time velocity query recovers all diagonal variances, provided every query coordinate satisfies the condition above. A nondegenerate Gaussian query satisfies these conditions with probability one.

For full covariance, the coordinates are coupled through $A_t(c)$. One velocity query determines only the matrix–vector product

$$v(m_t, t \mid c) - \mu(c) = A_t(c)\big(m_t - t\mu(c)\big). \tag{19}$$

Recovering the full matrix generally requires additional independent query directions or the velocity Jacobian, $\nabla_{m_t} v = A_t(c)$.

A.2 FEATURE CHANGES NEAR THE INITIAL TIME

At $t = 0$, the ideal velocity gives us the mean but says nothing about variance. The internal features may still carry this information, since at nearby times they support a velocity that depends on variance. If these features change little with time, we may be able to read out both moments from the same evaluation. This motivates comparing the features at $t = 0$, 0.01, and 0.1, keeping the context and noisy action input fixed.

For each teacher, we use 32 saved validation observations from distinct Human300 tasks and three Gaussian noise draws per observation, giving 96 pairs for each comparison with $t = 0$. GR00T uses the first validation observation from each of 32 task indices spaced evenly across the 300-task bank. Xiaomi uses 32 evenly spaced entries from an existing 50-task validation bank, each containing the first validation observation for that task and its native four-frame history. The task subsets differ between teachers; every within-teacher comparison holds images, instruction, robot state and the exact noisy action tensor fixed, changing only time. The noise seeds are 10817–10819 for GR00T and 47017–47019 for Xiaomi, including each deployed readout's fixed query.

We capture the action-transformer output h and the readout features u. The deployed moment readouts use u for both teachers. All sixteen action-token feature vectors are flattened before computing

$$\cos(f_0, f_t) = \frac{\langle f_0, f_t \rangle}{\|f_0\|_2 \|f_t\|_2}, \qquad \Delta_{\mathrm{rel}} = \frac{\|f_t - f_0\|_2}{\|f_0\|_2}, \qquad r = \frac{\|f_t\|_2}{\|f_0\|_2}. \tag{20}$$

To check whether a common feature offset dominates cosine similarity, we also subtract the across-observation mean separately for each time and noise seed before computing centered cosine similarity. Table 3 reports descriptive means across the 96 pairs.

Table 3: Feature changes with time at an identical noisy input. Each row averages 32 validation observations and three noise draws, relative to $t = 0$. Norm change is $100(r - 1)$; centered cosine removes the mean feature at each time and noise seed.

Teacher	Features	t	Cosine	Δ_{rel} (%)	Norm change (%)	Centered cosine
GR00T	h	0.01	0.99993	1.68	+1.16	0.99958
	u	0.01	0.99997	1.29	+1.06	0.99961
	h	0.10	0.99937	11.39	+10.76	0.99684
	u	0.10	0.99981	11.12	+10.94	0.99635
Xiaomi	h	0.01	0.99998	1.03	+0.76	0.99979
	u	0.01	0.99999	1.01	+0.89	0.99982
	h	0.10	0.99901	6.66	+4.84	0.99764
	u	0.10	0.99936	9.73	+8.98	0.99778

For each fixed observation and noisy input, feature directions remain highly similar across the tested times for both teachers. The readout features u mainly increase in magnitude.

The predicted velocities also remain nearly parallel across these times. Between $t = 0$ and $t = 0.1$, their mean cosine similarity is 0.99987 for GR00T and 0.99993 for Xiaomi, measured over the twelve action channels used for execution. Their norms increase by about 11% for both teachers.

High cosine similarity alone does not establish preservation of variance information, which could occupy a small component of the representation. The held-out moment-prediction experiments test that information's accessibility directly.

B EXPERIMENTAL SETUP

B.1 ROBOCASA365

RoboCasa365 is a simulation benchmark with 365 household tasks across 2,500 kitchen environments (Nasiriany et al., 2026). A mobile robot with an arm performs basic skills, such as opening a drawer, and multistep activities, such as preparing coffee. Both teachers use released RoboCasa365 checkpoints trained on Human300 demonstrations. Xiaomi uses the Xiaomi-Robotics-1-RoboCasa365 checkpoint with five flow updates (Xiaomi Robotics, 2026a; Xiaomi Robotics Team et al., 2026). GR00T N1.5 uses the `gr00t_n1-5/multitask_learning/checkpoint-120000` checkpoint from the RoboCasa365 checkpoint release, with four flow updates (NVIDIA et al., 2025).

Evaluation covers the 50-task target suite on the `pretrain` environment split: 18 Atomic-Seen, 16 Composite-Seen and 16 Composite-Unseen tasks. Xiaomi inputs include four observations spaced two simulator steps apart, the released 0.95 image crop and the robot state; GR00T uses its native current-frame inputs and preprocessing. Both policies execute 16 actions before replanning. The evaluator retains native task horizons, scene randomization and persistent workers. The main comparison uses 50 episodes per task and policy. Context selection and readout fitting are specified in Appendices C.1 and C.2.

The benchmark comparisons use independently drawn physical scenes, while the separate fixed-scene diagnostics restore saved starts. Episode indices alone do not reproduce the native scene draws because policies use different numbers of servers and persistent GPU random streams.

Raw one-step execution tests how well the teacher performs with a single flow update: it uses GAR's fixed noise input and the teacher's native action decoding, without fitted readouts or added action noise. Each GAR setting in Table 2 uses one fitted readout.

B.2 LIBERO-LONG

We evaluate the released Xiaomi-Robotics-0 LIBERO checkpoint (Xiaomi Robotics, 2026b) using five flow steps for the teacher and one action-transformer evaluation for GAR and raw one-step execution. All policies execute ten actions before replanning, with ten initial settling steps and a 520-step control limit. Evaluation uses fifty common official initial states per task, giving 500 episodes per policy.

C READOUT FITTING

C.1 ROBOCASA365 CONTEXT SAMPLING

We sample contexts to cover every Human300 task and different stages of its demonstrations. For each of the 300 tasks, we randomly select five of the first 100 human demonstrations without replacement and take four evenly spaced interior states from each, giving 6,000 fitting contexts. For a demonstration of L frames, the zero-based indices are $\lfloor 6 + j(L-23)/(K+1) \rfloor$, $j = 1, \ldots, K$, with $K = 4$ for fitting. These indices spread the states between frames 6 and $L - 17$, leaving room for the observation history and action chunk. Both teachers use the same contexts, each with its native observation and history inputs.

Tuning and evaluation use separate demonstrations from fitting and from each other. Two additional demonstrations per task supply two states each ($K = 2$); we assign one demonstration to tuning and the other to evaluation, giving 600 contexts in each set. Demonstration selection uses seed 66017 plus the task index; the tuning/evaluation assignment uses seed 11717. The evaluation bank has also appeared in earlier exploratory analyses.

Eligible training states are frames 6 through $L_d - 17$, inclusive, of demonstration d, where L_d is its number of frames. Among the first 100 demonstrations per task, excluding those reserved for tuning and evaluation leaves 29,400 demonstrations, giving

$$N = \sum_{d=1}^{29,400} (L_d - 22) = 26{,}612{,}851$$

eligible training states. The 6,000 fitting contexts therefore constitute $100 \times 6{,}000/N \approx 0.02\%$ of this pool.

To test the effect of more fitting data, the 24,000-context fits add fifteen training demonstrations per task using the same four-state rule. The original fitting, tuning and evaluation contexts are preserved, and command readouts use the same 6,000-context fit for both fitting-set sizes.

At each context, 384 teacher samples determine the empirical mean and regularized covariance target. GR00T has 144 varying continuous coordinates; its remaining 16 constant coordinates are kept at zero. Xiaomi uses all 160 normalized continuous coordinates.

C.2 RoboCasa365 readout fitting

Numerically inverting the variance-recovery identity can be sensitive to errors near $t = 0$ or $m_{t,i} = t\mu_i(c)$. GAR therefore fits empirical teacher moments estimated from samples.

Let d_m and d_b denote the numbers of continuous and binary coordinates in an action chunk. The mean and command readouts act on the frozen readout features at each action position. The mean readout adds an affine correction to the continuous coordinates of the teacher's one-step prediction. The covariance readout averages features across positions and applies a single 128-unit GELU hidden layer. For full covariance, it predicts a Cholesky factor with positive diagonal entries; the diagonal variant predicts positive variances. Full covariance includes dependence between continuous coordinates at different chunk positions. Command latents are independent of one another and of continuous actions.

Gaussian latents and Bernoulli commands. Thresholding a unit-variance Gaussian can represent any Bernoulli probability $p \in (0, 1)$. For each binary command, we sample $z_j = \eta_{\theta,j}(c) + \xi_j$ with $\xi_j \sim \mathcal{N}(0, 1)$ and set $b_j = \mathbf{1}\{z_j \geq 0\}$. Its activation probability is

$$\begin{aligned} p_{\theta,j}(c) &= \Pr(b_j = 1 \mid c) = \Pr(\xi_j \geq -\eta_{\theta,j}(c)) \\ &= 1 - \Phi(-\eta_{\theta,j}(c)) = \Phi(\eta_{\theta,j}(c)) \,, \end{aligned} \tag{21}$$

where Φ is the standard normal cumulative distribution function. Thus $b_j \mid c \sim \text{Bernoulli}(p_{\theta,j}(c))$. Choosing $\eta = \Phi^{-1}(p)$ gives the desired probability, with zero and one obtained as limits. Fixing the latent variance to one therefore does not restrict individual command probabilities. Binary cross-entropy against teacher command frequencies trains the probabilities of the executed commands.

The marginal result also holds for correlated Gaussian latents. For $z \mid c \sim \mathcal{N}(\eta_\theta(c), C(c))$, with $C_{jj}(c) > 0$, thresholding gives

$$\begin{aligned} b_j &= \mathbf{1}\{z_j \geq 0\}, \\ p_{\theta,j}(c) &= 1 - \Phi\left(-\frac{\eta_{\theta,j}(c)}{\sqrt{C_{jj}(c)}}\right) = \Phi\left(\frac{\eta_{\theta,j}(c)}{\sqrt{C_{jj}(c)}}\right). \end{aligned} \tag{22}$$

Each binary marginal is therefore fully specified by its mean, regardless of correlations between latents. The joint binary probabilities depend on those correlations and generally require Gaussian orthant integrals. GAR uses independent unit-variance command latents, also independent of continuous actions, yielding the factorized likelihood in Equation (11).

Offline teacher statistics. We select a small set of contexts $\mathcal{C}$ and draw N teacher action chunks $a^{(i)} = (m^{(i)}, b^{(i)})$ at each context. Binary commands are obtained using the teacher's execution thresholds. We estimate continuous-action moments and command frequencies as

$$\widehat{\mu}_c = \frac{1}{N}\sum_{i=1}^{N} m^{(i)}, \tag{23}$$

$$\widehat{S}_c = \frac{1}{N}\sum_{i=1}^{N}(m^{(i)} - \widehat{\mu}_c)(m^{(i)} - \widehat{\mu}_c)^\top,$$

$$\widehat{p}_{c,j} = \frac{1}{N}\sum_{i=1}^{N} b_j^{(i)}.$$

We shrink $\widehat{S}_c$ by 5% toward its diagonal and floor eigenvalues at the larger of 10^{-5} times its mean diagonal and 10^{-10} to obtain $\widehat{\Sigma}_c$. This regularization weakens correlations and prevents nearly singular covariance targets; it is separate from the readout loss. Shrinkage toward the diagonal preserves marginal variances, and eigenvalue flooring can only increase them. These statistics are cached together with the readout features and one-step teacher predictions.

Moment matching. We fit the affine mean readout by least squares:

$$\mathcal{L}_\mu(\theta) = \mathbb{E}_{c\sim\mathcal{C}}\left[\frac{1}{d_m}\,\|\mu_\theta(c) - \widehat{\mu}_c\|_2^2\right]. \tag{24}$$

We then fit the covariance readout while keeping the mean and command readouts fixed. Covariance targets are centered on the empirical teacher mean, so covariance fitting is independent of mean-prediction error.

To account for differences in scale between action coordinates, let $D = \mathrm{diag}(s_1, \ldots, s_{d_m})$, where s_k^2 is the average target variance of coordinate k over the original 6,000 fitting contexts, for both fitting-set sizes. For full covariance, we minimize

$$\mathcal{L}_\Sigma(\theta) = \mathbb{E}_{c\sim\mathcal{C}}\Bigg[\frac{1}{d_m^2}\left\|D^{-1}\big(\Sigma_\theta(c) - \widehat{\Sigma}_c\big)D^{-1}\right\|_F^2 + \lambda_{\mathrm{var}}\,\mathcal{R}_{\mathrm{var}}(c)\Bigg], \tag{25}$$

where

$$\mathcal{R}_{\mathrm{var}}(c) = \frac{1}{d_m}\sum_{k=1}^{d_m}\left(\log\frac{\Sigma_{\theta,kk}(c)}{\widehat{\Sigma}_{c,kk}}\right)^2, \tag{26}$$

and $\lambda_{\mathrm{var}} = 0.1$. The first term matches covariance entries after coordinate-wise normalization, balancing action coordinates with different scales. The logarithmic term also gives weight to errors at contexts with small target variances: doubling and halving a variance incur the same penalty, regardless of its scale. For the diagonal variant, the first term averages normalized squared errors over the d_m diagonal entries only. This training penalty is separate from the shrinkage and eigenvalue flooring used to estimate covariance targets.

Covariance optimization. Both covariance variants use AdamW (Loshchilov & Hutter, 2019) with learning rate 10^{-3}, weight decay 10^{-2}, batch size 64, gradient clipping at norm one and seed 17. We evaluate the corresponding moment loss on the 600 tuning contexts every 1,000 updates and select its lowest-loss checkpoint. From update 12,000, four tuning evaluations without an improvement exceeding the larger of 10^{-4} in relative loss and 10^{-7} in absolute loss reduce the learning rate by a factor of 0.3, down to 10^{-6}. At this minimum rate, eight evaluations without such an improvement stop fitting. Appendix E.1 reports the selected updates and convergence curves.

Command fitting. The command readout learns the teacher's activation probabilities by minimizing binary cross-entropy against the sampled command frequencies:

$$\mathcal{L}_b(\theta) = \mathbb{E}_{c\sim\mathcal{C}}\left[\frac{1}{d_b}\sum_{j=1}^{d_b}\mathrm{BCE}(\widehat{p}_{c,j}, \Phi(\eta_{\theta,j}(c)))\right], \tag{27}$$

where BCE denotes binary cross-entropy. This objective fits command activation probabilities through the Gaussian latent means. For each teacher, the 6,000- and 24,000-context GAR variants share the same command readout, trained on 6,000 contexts. During command training, the Bernoulli term above is combined with a Gaussian KL term for continuous motion. We use 3,000 AdamW (Loshchilov & Hutter, 2019) updates, batch size 64, learning rate 10^{-4}, weight decay 10^{-5} and gradient clipping at norm one, with checkpoint selection by validation motion-and-command likelihood. The mean and covariance readouts described above replace the continuous-action predictors from this joint fit; the command parameters stay fixed. All readout fitting uses cached features and teacher statistics, without further teacher evaluations or action sampling.

C.3 LIBERO-Long readout fitting

For each task, thirty demonstrations supply twenty contexts each, giving 6,000 fitting contexts overall. Separate sets of five demonstrations per task supply twelve contexts each for tuning and moment evaluation, giving 600 contexts in each set. At each context, 384 teacher samples estimate continuous-action moments and gripper-command frequencies.

GAR adds an affine correction to the teacher's one-step continuous-action prediction and predicts diagonal covariance through a pooled 128-unit GELU readout. The teacher networks remain frozen.

We use a probit readout with shared feature weights and a separate bias for each of the ten action positions. Its coefficients were fitted by binary cross-entropy on the training contexts and selected by tuning cross-entropy.

C.4 Fitting and preparation cost

For Xiaomi, the main cost is preparing the teacher data. Once the readout features and targets are cached, the recorded fitting times total about two minutes for either the diagonal or full-covariance readouts on 6,000 contexts.

Fitting cost. The Xiaomi 6,000-context affine mean is solved by CPU least squares in 3.7 seconds. Its diagonal and full covariance fits take 94.4 and 100.3 seconds on an RTX 4090. The initial joint motion-and-command fit that supplies the fixed command readout adds 16.7 seconds on an RTX 4090, including optimization, tuning evaluations and checkpoint saving but excluding untimed data loading and setup. The recorded fitting times therefore total 114.8 seconds for diagonal GAR and 120.7 seconds for full covariance; teacher sampling and feature extraction are counted separately below.

Preparation cost. The Xiaomi preparation measurements cover a cache of 7,200 fitting, tuning and evaluation contexts, each with 512 teacher chunks (3,686,400 chunks in total). At each context, 384 chunks estimate the targets and the remaining 128 are reserved for density and variance comparisons.

Table 4: Preparation cost for Xiaomi's cache of 7,200 contexts and 3,686,400 teacher chunks. The measurements exclude teacher pretraining and policy evaluation; checkpoint download was not timed.

Preparation step	Resources	Time
Download, extraction and data checks	CPU/network	2.71 h wall time
Feature extraction and teacher sampling	4 RTX 4090 GPUs	3.84 GPU-hours total 1.14 h longest parallel job
Motion and command target estimation	12 CPU workers	6.9 s wall time

D GAUSSIAN APPROXIMATION IN PHYSICAL ACTION UNITS

D.1 SHARED PROCEDURE

Samples and units. Both teachers use the same 300 saved observation identities, selecting one validation observation uniformly within each Human300 task bank. Each observation supplies 1,024 teacher chunks, generated with FP32 action-network computation and Euler accumulation while retaining frozen vision-language conditioning. Xiaomi uses its native four-frame conditioning and five flow steps; GR00T uses current-frame conditioning and four flow steps. Binary commands remain unrestricted during sampling but are excluded from the continuous-action comparison. The first 512 samples fit the maximum-likelihood Gaussian mean and covariance; the remaining 512 evaluate the approximation. Both teachers decode translation and rotation outputs directly into controller inputs. The controller clips each coordinate to $[-1, 1]$, then scales translation by 50 mm and rotation-vector coordinates by 0.5 radians, which we convert to degrees. We fit before clipping and apply the controller mapping to both teacher and Gaussian commands before projecting them.

Sliced comparison. Sliced 1-Wasserstein distance compares the teacher and Gaussian distributions along many one-dimensional projections (Villani, 2009; Bonneel et al., 2015). Each projection combines the action coordinates into one number, allowing the comparison to detect dependence that separate coordinate comparisons can miss. Along each direction, we measure how far command values must move to match the two distributions, then average these distances across directions. Small shifts in command values therefore give small errors, whereas KL divergence can be large for small shifts between narrowly concentrated distributions.

Let $\theta_1, \ldots, \theta_L$ be the unit projection directions, and let $\widehat{F}^{-1}_{c,\ell}$ and $\widehat{G}^{-1}_{c,\ell}$ denote the empirical quantile functions of the projected teacher and Gaussian reference points at context c. For the n held-out teacher commands $x_i(c)$ in physical units, we define

$$
\begin{aligned}
D(c) &= \frac{1}{L} \sum_{\ell=1}^{L} \int_0^1 \left| \widehat{F}^{-1}_{c,\ell}(u) - \widehat{G}^{-1}_{c,\ell}(u) \right| du, \\
A(c) &= \frac{1}{Ln} \sum_{\ell=1}^{L} \sum_{i=1}^{n} \left| \theta_\ell^\top x_i(c) \right|, \qquad R(c) = \frac{D(c)}{A(c)}.
\end{aligned}
\tag{28}
$$

The projected command magnitude $A(c)$ averages the absolute projections of the teacher commands, without centering them. Thus $R(c)$ measures approximation error relative to command size, including any nonzero mean. We compute this ratio within each context, then report its median across contexts; all observed denominators are nonzero.

For each action-group dimension, we use $n = 512$ held-out teacher commands and $L = 256$ unit directions sampled uniformly on the sphere, shared across teachers and covariance choices. To reduce numerical error in each Gaussian reference, we use 2,048 points from a scrambled Sobol sequence (Sobol', 1967; Matoušek, 1998). Sobol sequences are designed to cover the unit cube evenly; scrambling randomizes the points while preserving this balanced coverage. We transform these points through the standard-normal quantile function, multiply by the fitted covariance's symmetric square root, and add the fitted mean. The reference points undergo the same clipping, scaling and projection as the teacher commands. Sorting the projected points lets us evaluate each quantile integral exactly for these empirical distributions.

Gaussian sampling controls. Finite samples produce approximation error even when the underlying distribution is Gaussian, so we use Gaussian sampling controls to measure this baseline. At each context and for each group, we draw eight independent sets of 1,024 samples from the fitted Gaussian and repeat the 512/512 split, refitting and projected comparison. Sampling and refitting use the same covariance family as the corresponding teacher comparison. We average the eight control errors before dividing by the original teacher's projected command magnitude. These controls measure finite-sample discrepancy and reference integration error; they are not subtracted as an unbiased correction.

Rotation uses Euclidean rotation-vector coordinates, not geodesic orientation distance. These measurements describe command distributions, not positioning error or a count of density modes; their consequences for task performance are evaluated separately in Section 5.2.

D.2 Action-chunk comparison

The full-chunk comparison asks how well a Gaussian captures the joint distribution of commands across all sixteen action positions (Table 5). We group the translation commands into one 48-dimensional vector and the rotation commands into another. Full covariance retains dependence within and across action positions; diagonal covariance retains the same fitted means and marginal variances but sets all off-diagonal entries to zero. Both approximations fit teacher samples directly, independently of the learned GAR readouts.

We apply the shared procedure to the 48-dimensional translation and rotation groups. Both covariance choices use the same projected teacher-command magnitude as denominator.

Table 5: Gaussian approximation of translation and rotation across sixteen-action chunks at 300 fixed observations per teacher. Each group has 48 coordinates. Entries are median relative sliced 1-Wasserstein errors (%), normalized by the teacher's projected command magnitude; lower is better. Controls repeat fitting and evaluation on samples from the corresponding full or diagonal Gaussian (Appendix D).

		Full covariance		Diagonal covariance	
Teacher	Action group	Error (%)	Control (%)	Error (%)	Control (%)
Xiaomi	Translation	0.8	0.7	1.8	0.7
	Rotation	1.8	1.6	3.0	1.5
GR00T	Translation	4.6	3.3	9.1	3.4
	Rotation	9.5	6.3	19.9	6.4

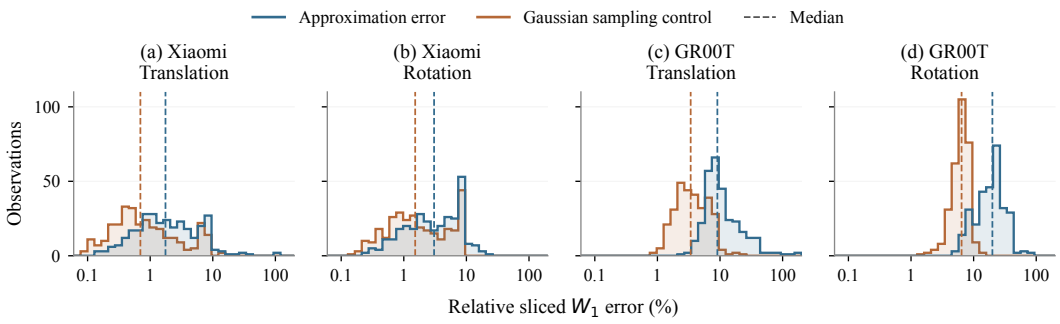


Figure 8: Diagonal-covariance Gaussian approximation under the same procedure as Figure 2, with 300 observations per panel. Approximation errors and Gaussian sampling controls use the same logarithmic bins and axis limits as the full-covariance comparison. Dashed lines mark medians, and all observations are retained.

Numerical checks and scope. We vary the number of projection directions and Gaussian reference points to check numerical accuracy in the 48-dimensional groups. Using 64, 128 or 256 directions changes aggregate median relative errors by at most 0.15 percentage points with full covariance and 0.75 with diagonal covariance. On twelve matched contexts per teacher, increasing the reference to 8,192 points with an independent scramble changes individual relative errors by at most 0.59 and 0.24 percentage points, respectively.

The chunk comparison includes temporal dependence within translation and within rotation, while translation–rotation coupling and other commands remain outside its scope. The physical distances concern projected command distributions, not accumulated displacement or orientation errors.

Table 6: Physical sliced errors and upper-tail relative errors for the sixteen-action comparison on 300 contexts per teacher. Physical errors are medians across contexts, in millimetres for translation and degrees for rotation-vector commands. The last two columns give 90th percentiles of relative error (%); lower values are better. These are descriptive summaries without confidence intervals; Table 5 gives median relative errors and Gaussian controls.

		Physical error		Relative error, 90th percentile	
Teacher	Covariance	Translation	Rotation	Translation	Rotation
Xiaomi	Full	0.0590	0.0071	7.6	8.8
	Diagonal	0.1207	0.0119	7.7	8.8
GR00T	Full	0.4054	0.0726	14.9	19.3
	Diagonal	0.9867	0.1316	29.2	35.4

Effect of covariance. Removing correlations increases median approximation errors for both teachers (Table 5). The 90th-percentile relative errors remain nearly unchanged for Xiaomi and roughly double for GR00T (Table 6), showing a larger effect of removing correlations in GR00T's upper-tail errors.

E MOMENT RECOVERY

Small moment errors relative to action magnitude coexist with substantial relative errors in individual variances (Section 5.1). We examine these errors more closely, compare them with uncertainty in the teacher estimates, and test how they change with the fitting loss.

E.1 MOMENT PREDICTION AND CONVERGENCE

Errors relative to action magnitude. The SD metric in Table 1 measures marginal-scale errors relative to the magnitude of the teacher's mean action chunk. For diagonal Gaussian sampling, holding the mean fixed and sharing a standard Gaussian noise vector ϵ gives

$$\sqrt{\mathbb{E}_\epsilon\left[\|\left(\sigma_\theta(c)-\widehat{\sigma}_c\right)\odot\epsilon\|_2^2\right]}=\|\sigma_\theta(c)-\widehat{\sigma}_c\|_2. \tag{29}$$

The numerator is therefore the root mean square change in the sampled action due to SD prediction error, measured in the teacher's fitted action coordinates. For full-covariance readouts, this marginal metric omits correlation errors.

Table 7 complements the medians and interquartile ranges in Table 1 with 90th-percentile errors, showing how the errors grow toward the upper end of the distribution. For the 6,000-context diagonal readouts, SD errors at this percentile reach 16.19% for Xiaomi and 29.07% for GR00T. All 600 evaluation means per teacher have nonzero norm; no contexts are excluded and no denominators are floored. Percentiles use linear interpolation between ordered values.

Table 7: 90th-percentile moment errors on the same 600 evaluation contexts per teacher as Table 1. All errors are percentages of the teacher mean's norm; lower is better.

Teacher	Fitting contexts	Mean error (%)	SD error (%)	
			Diagonal	Full
Xiaomi	6,000	13.63	16.19	10.65
Xiaomi	24,000	13.40	13.90	10.34
GR00T	6,000	20.08	29.07	25.26
GR00T	24,000	19.44	25.86	24.65

Diagonal variance accuracy. The readouts underestimate the largest variances beyond the discrepancy between teacher sample estimates. Among the largest 1% of target variances, median prediction-to-target ratios are 0.577 for Xiaomi and 0.633 for GR00T with the 6,000-context diagonal readouts, compared with 0.976 and 1.001 for independent teacher estimates (Table 8).

At each evaluation context, the targets use regularized covariance estimates from 384 teacher action chunks, while the independent estimates use unbiased sample variances from 128 disjoint teacher chunks. For each teacher, we pool all context–coordinate pairs and select the largest 1% by target variance: 960 pairs for Xiaomi and 864 for GR00T. We divide each independent variance estimate by its corresponding target and take the median of these individual ratios over the selected pairs. GAR's ratios use the same targets and selected pairs, with the predicted variance in the numerator.

Figure 4 shows how predictions vary across the range of teacher variances. It groups all 96,000 Xiaomi and 86,400 GR00T pairs into 24 equal-width bins in log variance, placing each point at the median target variance in its bin. The independent teacher curve uses sample variances from 128 additional actions at each context. To show the spread behind these summaries, Figure 9 includes every prediction from all eight deployed checkpoints, including errors outside the central bands.

Table 8: Marginal variance accuracy on 600 evaluation contexts per teacher, relative to regularized 384-sample targets $\widehat{v}$. Independent teacher rows use unbiased variances from 128 disjoint teacher chunks per context; other rows use GAR predictions. Tail columns use the largest 1% of target variances pooled across contexts and coordinates, reporting the median individual ratio $v/\widehat{v}$ (one is exact) and median relative error $|v/\widehat{v} - 1|$ (lower is better). These targets exceed 0.00784 for Xiaomi and 0.0786 for GR00T, in their respective fitted action coordinates. Here v denotes the independent estimate or GAR's predicted marginal variance.

Teacher	Fitting contexts	Estimate	Tail ratio	Tail error (%)
Xiaomi	—	Independent teacher	0.976	10.9
Xiaomi	6,000	Diagonal	0.577	53.0
Xiaomi	6,000	Full	0.503	56.5
Xiaomi	24,000	Diagonal	0.669	45.5
Xiaomi	24,000	Full	0.545	52.4
GR00T	—	Independent teacher	1.001	7.4
GR00T	6,000	Diagonal	0.633	37.5
GR00T	6,000	Full	0.647	37.1
GR00T	24,000	Diagonal	0.710	31.1
GR00T	24,000	Full	0.667	36.5

Convergence. We track tuning loss through learning-rate reductions to check whether the fits have reached a plateau. All eight covariance fits stop at a plateau, with selected checkpoints between 18,000 and 53,000 updates.[1] Figure 10 shows the loss curves for the two 6,000-context diagonal readouts, including continued fitting after the selected checkpoints.

E.2 Teacher-sample uncertainty

The teacher moments used as references are themselves estimated from finite samples. We examine how the estimated mean changes with the number of teacher samples.

Mean estimates at different sample counts. For each of the original 6,000 Xiaomi fitting contexts, we estimate the continuous-action mean from nested subsets of 32, 128 and 384 cached teacher chunks. The mean of $\|\widehat{\mu}_{c,N} - \widehat{\mu}_{c,384}\|_2^2/160$ across contexts is 1.23×10^{-5} for $N = 32$ and 2.24×10^{-6} for $N = 128$, in squared normalized-action units. The 128 reserved teacher samples at each context are excluded from this comparison.

[1]The 24,000-context Xiaomi covariance fits use CPUs, while the 6,000-context fits use a GPU, so their minibatch streams are not paired despite the common seed.

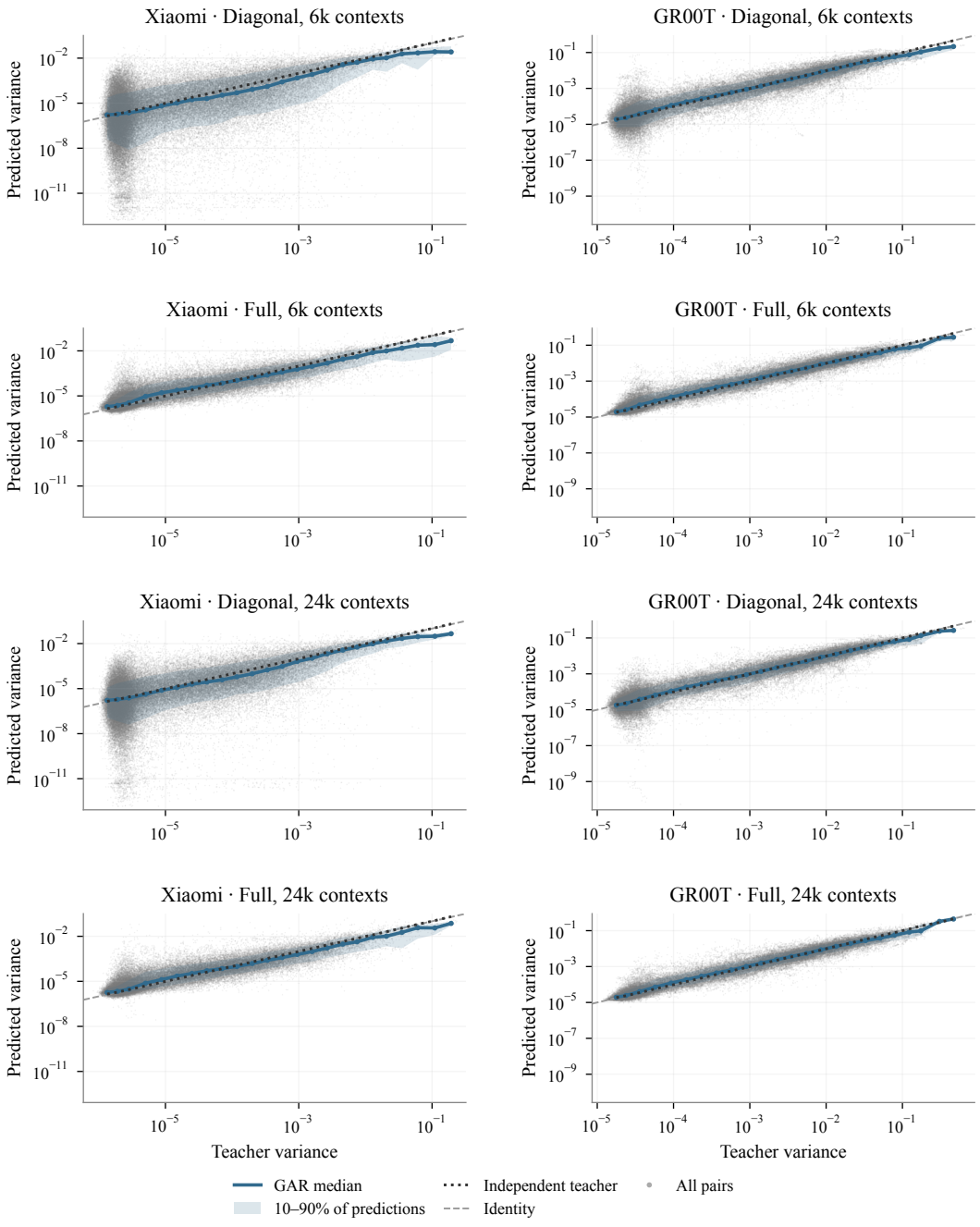


Figure 9: All held-out marginal-variance predictions from the deployed diagonal and full readouts. Gray points include every context–coordinate pair; curves and bands follow Figure 4. Each teacher uses common axis limits across its four panels, spanning the complete target and prediction ranges without clipping. GR00T's 16 structurally constant coordinates are excluded. Xiaomi uses normalized action coordinates and GR00T uses decoded coordinates.

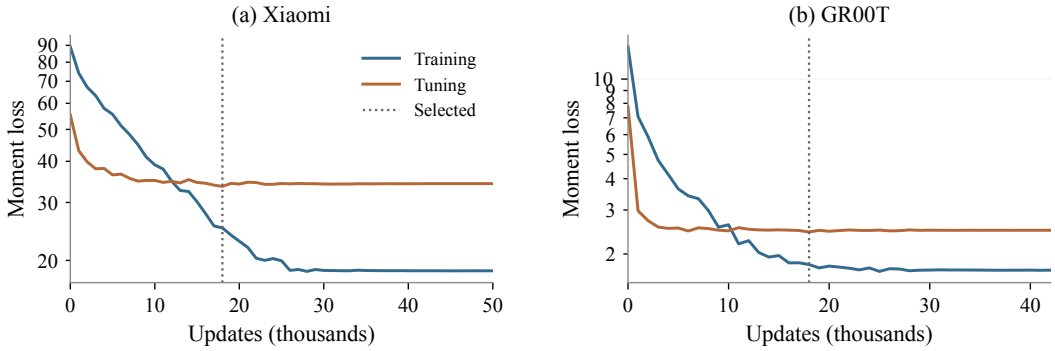


Figure 10: Convergence of the 6,000-context diagonal covariance readouts. Lower moment loss is better; vertical dotted lines mark the deployed checkpoints selected by tuning loss. Training curves use a fixed 1,200-context subset and tuning uses 600 contexts from separate demonstrations. Both fits select update 18,000 and stop at 50,000 updates for Xiaomi and 42,000 for GR00T. Each panel shows one fit, with the full recorded update range and a logarithmic loss axis.

The mean squared difference from the 384-draw reference falls by a factor of 5.5 when increasing the sample count from 32 to 128. The shared reference is itself estimated, so these differences measure stability under changes in sample count rather than error against the true teacher mean.

E.3 Loss ablation for variance prediction

We next test how much variance prediction depends on the fitting loss. Changing the fitting loss reduces typical Xiaomi SD errors without changing the readout architecture. The two log-space losses give similar improvements, while applying a Huber penalty to the original variance residual gives a smaller gain (Table 9). These improvements come with stronger underestimation of the largest target variances (Figure 11).

Comparing fitting losses. The original loss combines squared variance residuals with a log-ratio penalty. We compare two ways of changing this emphasis: reducing the influence of large variance residuals with a Huber penalty, and measuring multiplicative discrepancies in log space. Let $r_{ck} = (v_{\theta,k}(c) - \widehat{v}_{ck})/s_k^2$ be the variance residual normalized by the training mean variance, and let $\ell_{ck} = \log(v_{\theta,k}(c)/\widehat{v}_{ck})$ be its log ratio. The four losses average the following penalties over contexts and coordinates:

$$\begin{aligned} \text{Original:} \quad & r_{ck}^2 + 0.1\ell_{ck}^2, & \text{Variance Huber:} \quad & h(r_{ck}) + 0.1\ell_{ck}^2, \\ \text{Log MSE:} \quad & \ell_{ck}^2, & \text{Log Huber:} \quad & h(\ell_{ck}), \end{aligned} \tag{30}$$

where $h(u) = u^2$ for $|u| \le 1$ and $h(u) = 2|u| - 1$ otherwise.

Matched fits. To isolate the effect of the loss, all four objectives use the same frozen Xiaomi readout features, 128-unit diagonal covariance readout, and 6,000/600/600 fitting, tuning and evaluation contexts specified in Appendix C.1. Targets use the same 384 teacher samples per context, and all 600 evaluation contexts are retained. Each objective is fitted with seeds 17 and 29, pairing initialization and minibatch sampling within each seed. Table 9 selects each checkpoint by its own tuning objective.

Accuracy and upper-tail trade-off. With log Huber, the mean tail ratio across the two seeds falls from 0.58 to 0.24, and raw variance MSE increases by 29%. The loss therefore changes the balance between typical SD error and errors on large variances.

These fits have not been evaluated in control rollouts, so the ablation establishes improved typical moment prediction rather than improved task performance. Figure 4 and the policy comparisons use the original deployed readouts.

Table 9: Xiaomi diagonal variance-loss ablation on 600 evaluation contexts, with fitting seeds reported separately. SD error is $100\|\sigma_\theta - \widehat{\sigma}\|_2/\|\widehat{\mu}\|_2$, shown as median [25th, 75th percentile]; lower is better. Tail ratio is median predicted/target variance among the largest 1% of targets; one is exact.

Objective	Seed	SD error (%)	Tail ratio
Original	17	5.58 [3.61, 9.03]	0.577
	29	5.29 [3.30, 8.67]	0.581
Variance Huber	17	3.16 [1.78, 5.64]	0.410
	29	3.24 [1.76, 5.71]	0.411
Log MSE	17	2.81 [1.49, 5.55]	0.244
	29	2.85 [1.48, 5.42]	0.248
Log Huber	17	2.71 [1.49, 5.39]	0.250
	29	2.77 [1.47, 5.51]	0.238

Fitting details. All fits use AdamW with learning rate 10^{-3}, weight decay 10^{-2}, batch size 64 and gradient clipping at norm one. Tuning is evaluated every 1,000 updates; after 12,000 updates, four evaluations without improvement reduce the learning rate by a factor of 0.3, down to 10^{-6}, and eight further evaluations without improvement at that rate stop fitting. All eight fits reach this stopping condition after 40,000–50,000 updates.

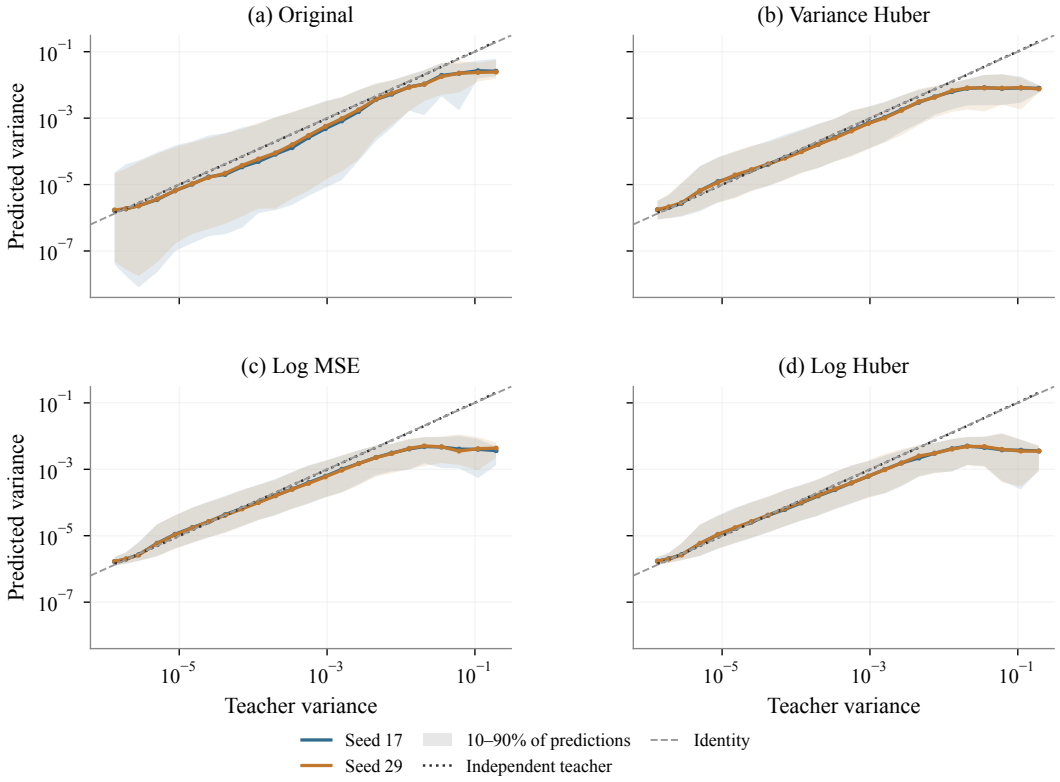


Figure 11: Effect of the fitting loss on Xiaomi variance prediction, with checkpoints selected by their respective tuning objectives. Both seeds use all 600 evaluation contexts and 160 coordinates per context. Curves show median predictions in the same 24 logarithmic bins; shading shows the 10th–90th percentiles for each seed, not confidence intervals. Axes are shared across panels and span the full target-variance range. The dotted curve uses 128 independent teacher samples, while targets use 384 samples. Log-space fitting improves typical predictions but increases underestimation of the largest variances.

F TASK PERFORMANCE AND RECOVERY

F.1 INFERENCE COST OF THE MOMENT READOUTS

To measure how much a single action-transformer evaluation saves in practice, we time the full policy call, including vision-language processing. GAR is $1.91\times$ faster for Xiaomi and $1.57\times$ for GR00T on the main system (Table 10).

Table 10: Batch-one latency of diagonal policies on two RTX 4090 systems, including vision-language processing. Mean and command readouts use 6,000 contexts; covariance fitting counts are shown separately. Speedup is teacher/GAR latency; lower latency and higher speedup are better. Three-session measurements use 64 calls per policy per session on 32 matched contexts and report medians of session means and speedups, with observed speedup ranges in brackets. The separate EPYC protocol uses 400 calls per policy on fifty contexts after ten warmups.

Model	Covariance contexts	Teacher (ms)	GAR (ms)	Speedup
Threadripper PRO 5955WX: three sessions				
Xiaomi	6,000	220.25	115.36	$\mathbf{1.91}\times$ [1.909, 1.915]
GR00T	6,000	68.01	43.30	$\mathbf{1.57}\times$ [1.566, 1.571]
EPYC 9554: earlier three sessions				
Xiaomi	24,000	202.08	111.84	$1.80\times$ [1.797, 1.807]
GR00T	6,000	60.87	42.98	$1.42\times$ [1.407, 1.421]
EPYC 9554: separate 400-call protocol				
Xiaomi	6,000	221.31	132.92	$1.66\times$
GR00T	6,000	59.77	41.74	$1.43\times$

Full and diagonal readouts have similar measured latency; variation across hosts and sessions is larger than their difference.

For Xiaomi, GAR reduces action-transformer time by replacing five evaluations with one (Table 11). Vision-language processing takes similar time for both policies and accounts for approximately 29% of teacher latency and 55% of GAR latency in this profile.

Table 11: Xiaomi latency breakdown on the Threadripper PRO 5955WX / RTX 4090 system, using the 6,000-context mean and diagonal covariance readouts. Values are means over ten matched observations in a separate profiling session. Module times use CUDA events; total call latency uses synchronized wall-clock timing and also includes preprocessing, transfers and decoding.

Component	Teacher (ms)	GAR (ms)
Vision-language processing	65.4	66.8
Action transformer	135.4	28.3
GAR readout	—	0.5
Total call latency	**225.9**	**120.7**

Timing protocol. The main timing comparison uses the exact 6,000-context mean, diagonal covariance and command readouts from Table 2 for both teachers. Three independent sessions each use 32 matched held-out contexts, five warmup calls per policy and two timed repetitions per context, with policy order rotated. Teacher and student share an RTX 4090 and an AMD Threadripper PRO 5955WX host (NVIDIA driver 550.120); CUDA is synchronized before and after each call. Timing starts from in-memory observations and includes preprocessing, transfers, vision-language processing, action generation and decoded CPU actions. Model loading, communication, queueing and simulation are excluded.

F.2 Uncertainty in task-success differences

Differences from the teacher. For both teachers, the intervals for the 6,000-context diagonal readout include zero but also allow losses of several percentage points; they do not establish practical equivalence.

Table 12 reports success differences for every readout in Table 2. We independently resample the fifty binary episode outcomes within each task and policy, retaining equal weights across the fifty tasks, and repeat this 20,000 times. Brackets give the 2.5th and 97.5th percentiles of the policy-minus-teacher differences. Native scenes are unpaired, so episodes are not paired across policies by their indices. These pointwise intervals describe repeated-episode uncertainty for the fixed task suite and fitted checkpoints; they omit fitting-seed and new-task variation and are not adjusted for multiple comparisons.

Table 12: Success differences on fifty tasks with fifty episodes per policy and task (2,500 total), using the settings of Table 2. Teachers succeed on 1,456 Xiaomi and 577 GR00T episodes. Difference is readout minus teacher success in percentage points; higher favors the readout. Pointwise 95% intervals use 20,000 independent within-task episode-bootstrap resamples (seed 85017), conditional on these tasks and fitted checkpoints.

Teacher	Fitting contexts	Readout	Successes	Difference	95% interval
Xiaomi	6,000	Mean only	1,416	-1.60	$[-3.84, 0.60]$
		Diagonal	1,441	-0.60	$[-2.84, 1.60]$
		Full	1,432	-0.96	$[-3.20, 1.20]$
	24,000	Mean only	1,411	-1.80	$[-4.08, 0.40]$
		Diagonal	1,443	-0.52	$[-2.72, 1.68]$
		Full	1,411	-1.80	$[-4.04, 0.40]$
GR00T	6,000	Mean only	493	-3.36	$[-5.24, -1.52]$
		Diagonal	548	-1.16	$[-3.04, 0.68]$
		Full	508	-2.76	$[-4.64, -0.88]$
	24,000	Mean only	505	-2.88	$[-4.76, -1.00]$
		Diagonal	536	-1.64	$[-3.52, 0.24]$
		Full	529	-1.92	$[-3.80, -0.04]$

Diagonal sampling versus mean-only execution. To isolate the contribution of continuous-action sampling, each comparison keeps the learned mean and stochastic command predictor fixed. All four observed success differences favor diagonal sampling, but only GR00T with 6,000 fitting contexts has a pointwise 95% interval entirely above zero (Table 13). We use the same independent within-task bootstrap as above.

Table 13: Diagonal-minus-mean success differences in percentage points, with pointwise 95% intervals from 20,000 independent within-task bootstrap resamples (seed 85017). Positive differences favor diagonal sampling; each policy has fifty episodes per task on the fixed fifty-task suite.

Teacher	Fitting contexts	Difference	95% interval
Xiaomi	6,000	$+1.00$	$[-1.20, 3.24]$
	24,000	$+1.28$	$[-0.96, 3.52]$
GR00T	6,000	$+2.20$	$[0.40, 3.96]$
	24,000	$+1.24$	$[-0.56, 3.04]$

F.3 CONDITIONAL VERSUS FIXED COVARIANCE

GAR's predicted covariance gives closer translation and rotation action-chunk distributions to the teacher than a fixed covariance, while its effect on task success remains uncertain. We compare the 6,000-context diagonal GAR with a *Fixed* policy that replaces its context-dependent covariance by the average predicted covariance over the 6,000 fitting contexts. Both policies retain identical learned means and stochastic command predictors, and use variance multiplier one. The fixed covariance is computed from the fitted GAR readout, so this baseline requires the same context collection, teacher sampling and readout fitting as GAR.

Task success. GAR has higher observed success for both teachers, but the intervals for GAR-minus-Fixed include zero (Table 14). Each policy uses fifty episodes on each of the fifty RoboCasa365 tasks, with independently drawn scenes and the execution settings of Table 2.

The Fixed evaluation combines five earlier episodes per task with forty-five additional episodes, without selecting a variance scale. To check the effect of reusing the earlier episodes, we also restrict both policies to the latter episode indices. This gives differences of $+1.91$ percentage points for Xiaomi and $+1.11$ for GR00T, with 95% intervals $[-0.36, 4.18]$ and $[-0.80, 3.02]$, respectively.

Table 14: GAR versus Fixed success on RoboCasa365, with 2,500 episodes per policy and teacher. Both policies share their mean and command readouts; only the covariance changes. Success is a percentage, and differences are GAR minus Fixed in percentage points; higher favors GAR. Brackets give pointwise 95% independent within-task episode-bootstrap intervals, conditional on the fitted checkpoints and fixed task suite.

Teacher	GAR	Fixed	Difference	95% interval
Xiaomi	57.64	55.92	$+1.72$	$[-0.44, 3.92]$
GR00T	21.92	20.76	$+1.16$	$[-0.64, 3.00]$

Distribution fit. Task success summarizes the outcome of an execution; we also compare the action distributions that produce it. GAR has lower mean distribution error in all four comparisons, with individual 95% paired context-bootstrap intervals excluding zero (Table 15). This supports more faithful translation and rotation chunk distributions from conditional covariance, although the task-success comparison does not establish a corresponding control benefit. The distances do not assess translation–rotation coupling, other continuous channels or binary commands.

We compare the deployed readouts directly with teacher samples on 143 held-out contexts per teacher, excluding contexts used for fitting or checkpoint selection. At each context, the final 512 samples from the FP32 teacher banks in Appendix D provide the reference distribution. We evaluate the saved readouts in FP32 on cached readout features and measure sliced 1-Wasserstein distance separately for the 48 translation and 48 rotation coordinates of each sixteen-action chunk. Both policies use the same clipping and physical scaling, 256 projection directions and 2,048 standard-normal Sobol reference points, following Appendix D.2. No Gaussian is fitted to the evaluation samples.

Table 15: Mean sliced 1-Wasserstein distance to teacher action chunks on 143 held-out contexts per teacher; lower is better. GAR and Fixed share their means and command predictors. Distances and reductions use the action group's stated unit; reduction is Fixed minus GAR, so positive values favor GAR. Brackets give pointwise 95% paired context-bootstrap intervals from 10,000 resamples, conditional on the fitted readouts and cached samples.

Teacher	Action group (unit)	GAR	Fixed	Reduction [95% interval]
Xiaomi	Translation (mm)	0.566	0.987	0.420 $[0.359, 0.483]$
	Rotation (degrees)	0.145	0.196	0.051 $[0.042, 0.061]$
GR00T	Translation (mm)	1.560	2.647	1.088 $[0.887, 1.299]$
	Rotation (degrees)	0.336	0.636	0.301 $[0.259, 0.343]$

F.4 GR00T RECOVERY FROM STUDENT STATES

Readout adaptation. Remaining failures need not all arise from the Gaussian policy family: the readouts may predict inaccurate moments at contexts encountered during execution. To test whether this error is recoverable, we first replace the learned continuous-action moments with moments estimated online from 384 fresh teacher samples at each query, keeping the command predictor fixed. On eight ScrubCuttingBoard scenes, this diagonal Gaussian policy succeeds in 5/8 cases, compared with 1/8 for the learned readouts. Although this diagnostic is too expensive for deployment, it shows that some failures can be corrected without changing the Gaussian family.

We then test whether a small offline readout update can recover some of this benefit. To target errors where the student acts, we add 256 contexts visited by GR00T's 6,000-context diagonal GAR on four seen tasks selected after inspecting failures. We query the teacher at these contexts and refit the readouts while keeping both pretrained networks frozen. An equal-size control instead adds demonstration contexts.

Evaluation restores identical initial scenes across the original student and both refits, using eight fresh scenes per targeted task and four per control task. The control tasks are OpenCabinet and PickPlaceCounterToStove. Evaluation scenes are separate from collection and tuning, and all policies retain native task horizons and sixteen-action execution.

Student-state refitting increases success from 3/32 to 6/32 on the selected tasks, compared with 1/32 for the demonstration control (Table 16). The main gain is on ScrubCuttingBoard, where it preserves the original success and recovers three additional scenes. It also retains all six original successes across eight scenes on two control tasks.

The gains are task-dependent, and this small, selected-task study does not establish a benchmark-wide improvement. It nevertheless identifies a practical way to improve some GAR failures: add teacher supervision where the student acts and update only the readouts, preserving the same inference procedure.

Table 16: GR00T readout adaptation on matched evaluation scenes, separate from collection and tuning. Entries are successes/episodes. Both refits add 256 training contexts.

Task	Original	Student states	Demonstrations
ScrubCuttingBoard	1/8	4/8	0/8
TurnOnMicrowave	0/8	1/8	0/8
RinseSinkBasin	2/8	1/8	1/8
StirVegetables	0/8	0/8	0/8
Targeted tasks	3/32	6/32	1/32
Two control tasks	6/8	6/8	7/8

Data and diagnostic details. The demonstration control draws its 256 added contexts from the original training demonstrations on the same four tasks. Both refits use the original 600 tuning contexts together with 64 additional contexts from separate student-execution scenes. At each added training or tuning context, 384 teacher samples estimate motion moments and command probabilities, and 128 additional samples support diagnostics. The sixteen unseen tasks and the original evaluation demonstrations are excluded from fitting and checkpoint selection.

The online-moment diagnostic uses eight further ScrubCuttingBoard scenes, shared across its five policies and excluded from fitting and selection. On these scenes, online full covariance succeeds in 3/8 cases, teacher continuous actions with student commands in 4/8, and the native teacher in 3/8. These small diagnostic panels do not establish superiority over the teacher.

Readout fitting. Both refits update the affine mean, diagonal covariance and command readouts, retaining the original feature normalization and frozen pretrained networks. Original and added training contexts receive equal total weight in the affine least-squares fit and the covariance and command objectives. The covariance and command heads use AdamW with learning rate 10^{-3}, weight decay 10^{-2}, gradient clipping at norm 1, and seed 17. Each update samples 32 original and 32 added contexts.

The covariance and command losses are divided by fixed initial training-loss values for their respective data sources, floored at 10^{-6}. Checkpoint selection uses the equally weighted sum of these normalized losses on the original and added tuning sets, evaluated every 1,000 updates.

After 12,000 updates, four checks without a relative improvement of 10^{-4} or an absolute improvement of 10^{-7}, whichever is larger, reduce the learning rate by a factor of 0.3, down to 10^{-6}. Eight checks without such improvement at the minimum rate stop fitting. Both refits select the 5,000-update checkpoint and stop at 40,000 updates.

F.5 LIBERO-LONG

LIBERO-Long tests whether GAR retains near-teacher performance beyond RoboCasa365. Across its ten tasks, GAR succeeds in 477/500 episodes, compared with 483/500 for the teacher and 480/500 for raw one-step execution (Table 17). Appendices B.2 and C.3 specify evaluation and fitting. Table 17 reports successes, with fifty episodes in each task row.

Table 17: LIBERO-Long successes on fifty common initial states per task. The teacher is Xiaomi-Robotics-0; diagonal GAR uses 6,000 fitting contexts. Overall entries are successes out of 500 episodes.

Task	Teacher	Raw one-step	GAR
Soup and tomato sauce into basket	49	47	48
Cream cheese and butter into basket	48	47	47
Turn on stove and place moka pot	50	50	50
Bowl into drawer and close it	48	46	48
Two mugs onto plates	50	50	50
Book into caddy	50	50	50
Mug onto plate and pudding beside it	44	45	41
Soup and cream cheese into basket	50	50	50
Two moka pots onto stove	45	47	45
Mug into microwave and close it	49	48	48
Overall	**483/500**	**480/500**	**477/500**

GAR's success difference is -1.2 percentage points relative to the teacher, with a paired 95% interval of $[-3.0, +0.6]$, and -0.6 points relative to raw one-step, with an interval of $[-2.2, +1.0]$. We compute these intervals using 10,000 bootstrap resamples of paired episodes within each task, retaining equal task weights. The intervals condition on these ten tasks and one fitted checkpoint.

G BEHAVIORAL DIVERSITY

G.1 CONSTRUCTED GAUSSIAN POLICY

The hand-designed policy in Figure 6 illustrates how Gaussian actions can produce two routes through state feedback. Near the start, the lateral mean amplifies small positive or negative deviations from $y = 0$, separating trajectories around the obstacle. Past the obstacle, the mean draws paths back toward $y = 0$ and the goal $(3, 0)$. The same policy thus follows either route while using a single Gaussian at every state.

For state $s_k = (x_k, y_k)$, actions update position directly:

$$a_k \sim \mathcal{N}(\mu(s_k), \Sigma), \qquad s_{k+1} = s_k + a_k, \qquad \Sigma = \mathrm{diag}(0.025^2, 0.035^2).$$

The mean depends only on the current state:

$$\begin{aligned} \mu_x(x, y) &= 0.08 \tanh(1.8(3 - x)), \\ \mu_y(x, y) &= 0.20 \left[0.92 g(x) \tanh(5y) - y\right], \\ g(x) &= \left[1 + \exp(5(x - 0.95))\right]^{-1}. \end{aligned}$$

As $g(x)$ decreases, the lateral mean changes from separating the paths to bringing them together.

This position update connects Gaussian action sampling to diffusion in physical time: it is an Euler–Maruyama step, with unit step size, for

$$ds_\tau = \mu(s_\tau)\, d\tau + \Sigma^{1/2}\, dW_\tau, \tag{31}$$

where τ denotes physical time and W_τ is standard two-dimensional Brownian motion. This interpretation applies to the constructed position model.

All 128 rollouts start at $(-3, 0)$ and execute 110 Gaussian actions, using NumPy's default random generator with seed 2026. The circle of radius 0.5 at the origin marks the obstacle; the Gaussian update is applied unchanged throughout. Every generated trajectory is displayed, with colors assigned afterward by the sign of y_{38}.

G.2 Diagonal behavioral diversity

We next test whether GAR produces different object-movement orders from the same initial scene, and compare its behavioral diversity with the teacher's. We use the Xiaomi 6,000-context Diagonal policy from the main benchmark, with its predicted variance and independent Bernoulli commands. The comparison covers two scenes each for loading a dishwasher, packing identical lunches and storing leftovers in a bowl, with 32 executions per policy and scene (384 episodes in total). These scenes were selected during exploratory search and do not estimate the prevalence of alternatives across the benchmark.

To hold the starting context fixed, both policies receive the same saved initial images, robot-state history and instruction. Both use the same evaluation runtime, native preprocessing and decoding, task horizons and sixteen-action replanning. For each policy and scene, a fixed-seed check confirms that actions and their sixteen-action physical replay are repeatable; these twelve checks are separate from the 384 episodes.

Different orders in successful executions. In the second lunch-packing scene, Diagonal succeeds in 3/32 attempts with two movement orders (Figure 7). Two successes move garlic A, garlic B, chicken B and chicken A; the third moves garlic A, chicken A, garlic B and chicken B. All three order labels remain unchanged at 3, 5 and 8 cm displacement thresholds. The teacher also succeeds in 3/32 attempts, with three distinct orders; only one matches an observed successful Diagonal order.

These executions demonstrate distinct successful behaviors under a Gaussian continuous-action policy. The policy also samples Bernoulli commands, so the experiment does not isolate continuous-action noise as the cause of the different orders.

Across the six starts, Diagonal succeeds in 13/192 attempts and the teacher in 12/192. Both policies succeed nine times in the second leftovers scene, always with the same observed order; Diagonal also succeeds once in the first lunch-packing scene, where its order label varies with the displacement threshold. Neither policy succeeds in the other three scenes.

Choice and order entropy. The successful examples establish that different orders are possible; we now compare how much behavior varies across all attempts, including failures. First-choice entropy measures variation in which object moves first, while full-order entropy measures variation in the sequence of objects moved. For category frequencies $\hat{p}_k$, Shannon entropy $-\sum_k \hat{p}_k \log_2 \hat{p}_k$ measures diversity in bits, and is zero when only one category is observed.

An object counts as moving after exceeding 5 cm displacement for three consecutive physical steps; thresholds of 3 and 8 cm test sensitivity. First-choice categories include ties and no movement, and full-order categories include incomplete attempts. We compute entropy over all 32 attempts per policy and scene, and give the six scene entropies equal weight. Figure 12 shows the scene entropies, and Table 18 gives the first-object counts.

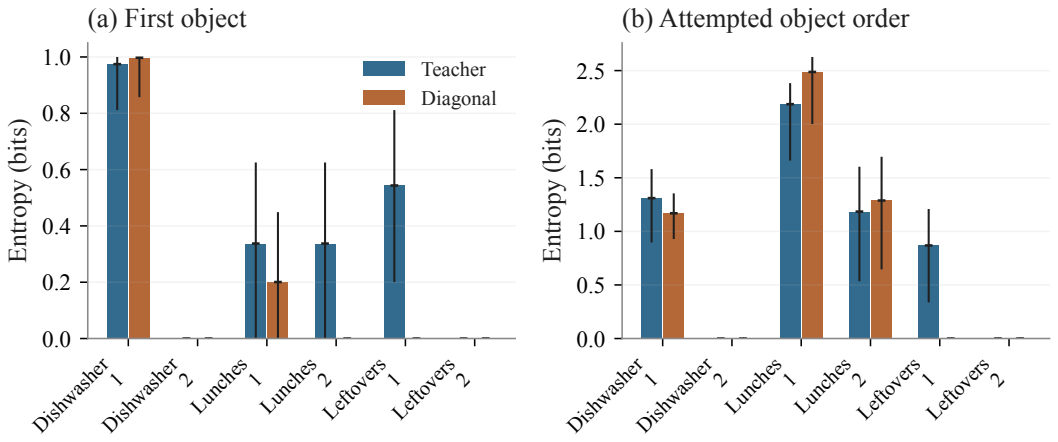


Figure 12: Behavioral diversity of diagonal GAR and the Xiaomi teacher from identical starts. Each scene has 32 executions per policy, including all successes and failures. Bars show first-object and full movement-order entropy at 5 cm; lines show 95% within-policy, within-scene bootstrap intervals. Zero observed entropy does not exclude rare alternatives, and equal entropy does not establish matching choices.

Table 18: First-object counts for diagonal GAR and the Xiaomi teacher across all 32 attempts per policy and scene, using the 5 cm threshold. These are movement choices rather than successful completions; no ties or no-movement outcomes occur in these scenes. A and B distinguish objects of the same type within a scene.

Scene	Teacher counts	Diagonal counts
Dishwasher 1	cup: 19; bowl: 13	cup: 15; bowl: 17
Dishwasher 2	dish A: 32	dish A: 32
Lunches 1	vegetable A: 2; vegetable B: 30	vegetable A: 1; vegetable B: 31
Lunches 2	chicken A: 2; garlic A: 30	garlic A: 32
Leftovers 1	chicken drumstick: 4; vegetable: 28	vegetable: 32
Leftovers 2	vegetable: 32	vegetable: 32

Diagonal produces less varied first-object choices on these scenes. Mean first-choice entropy is 0.365 bits for the teacher and 0.200 bits for Diagonal, a Diagonal-minus-teacher difference of -0.166 bits with a 95% within-scene bootstrap interval of $[-0.266, -0.043]$. Diagonal has lower first-choice entropy in three scenes, slightly higher entropy in the first dishwasher scene, and zero observed entropy for both policies in the other two scenes.

For full movement orders, mean entropy is 0.925 bits for the teacher and 0.824 bits for Diagonal, with a difference of -0.101 bits and interval $[-0.282, 0.088]$. This interval includes zero and does not establish equal entropy.

The first-choice reduction persists when the movement threshold changes: at 3 and 8 cm, differences remain negative at -0.225 and -0.110 bits, with intervals $[-0.313, -0.101]$ and $[-0.195, -0.006]$. All three full-order difference intervals include zero. Resampling is independent within each policy and scene; intervals condition on the observed categories and exclude unseen alternatives, scene-selection uncertainty and variation across new scenes.

H LIMITATIONS

- **Evaluation scope.** Our experiments evaluate selected pretrained policies on RoboCasa365 and LIBERO-Long, both simulation benchmarks. Generalization to physical robots, other embodiments and other pretrained policies remains untested. The task-success intervals describe episode variability for the evaluated tasks and fitted checkpoints; they exclude variation across readout fits and new tasks, and do not establish practical equivalence to the teachers (Appendices F.2 and F.5).
- **Distributional approximation.** Near-teacher task success does not imply that GAR reproduces the teacher's full action distribution. Diagonal GAR omits correlations between continuous coordinates and action positions, while both covariance variants assume binary commands are mutually independent and independent of continuous actions given the context. Our Gaussian-approximation measurements assess translation and rotation separately, leaving their joint dependence and other command groups outside the analysis (Appendix D).
- **Moment-prediction errors.** Moment prediction remains imperfect, with substantial errors at some contexts and underestimation of the largest variances, especially for Xiaomi (Appendix E.1). Readouts fitted on demonstration contexts may also encounter different states during execution. The GR00T recovery diagnostic shows that better moment estimates can correct some failures within the same diagonal Gaussian family, but the small, selected-task adaptation study does not establish a general remedy (Appendix F.4).
- **Behavioral diversity.** Gaussian action distributions can produce multimodal trajectories through state feedback, as our constructed example demonstrates, and GAR produces distinct successful object-movement orders. Preservation of the teacher's behavioral diversity is not established: across six selected scenes, first-object choice entropy is lower for GAR, while the difference in full movement-order entropy is inconclusive (Appendix G.2). These measurements include unsuccessful attempts and do not establish how frequently diverse successful behaviors occur across the benchmark.
- **Preparation and inference cost.** Fitting small readouts is inexpensive once teacher statistics are cached, but preparing those statistics requires repeated teacher sampling at each context (Appendix C.4). The small fraction of fitting contexts therefore does not describe the full distillation cost. At inference, GAR retains the teacher's vision-language processing, which limits the total speedup from replacing iterative action generation (Appendix F.1).