GeoMerge-OPD: Balanced Skill Integration in Vision-Language-Action Models

Anonymous authors
Paper under double-blind review

Abstract

Vision-language-action (VLA) models excel at individual robotic skills, yet integrating heterogeneous skills into a single policy remains a key challenge for general-purpose robotic control, especially in whole-body tasks with differing manipulation and locomotion requirements. Common multi-skill integration strategies, such as joint fine-tuning and parameter merging, can induce unintended actions during closed-loop execution, severely degrading individual skills. Task-conditioned analysis of expert updates reveals response energy concentrated in a few dominant directions and opposing expert responses along high-energy directions within a shared, task-relevant response subspace. This structure co-occurs with unintended action shifts after merging. We propose GeoMerge-OPD, combining response-space regularization, parameter merging, and on-policy distillation. During independent expert fine-tuning, G-isometry constrains update-response gains within task-activation subspaces; the expert parameters are then statically merged into a student with preliminary multitask capabilities. Fine-tuned task experts serve as teachers, providing velocity-field supervision at the environment observations and intermediate action-flow states visited by the student to calibrate residual behavioral deviations. Experiments with two VLA backbones show strong, balanced humanoid skill integration on SIMPLE and consistent performance on LIBERO and LIBERO-PRO, outperforming joint fine-tuning and multi-task reinforcement-learning baselines while maintaining robustness under perturbations.

1 Introduction

Vision-language-action (VLA) models adapt pretrained vision-language models to robotic control using demonstration data, combining semantic understanding and referential grounding with strong performance on individual manipulation tasks (Driess et al., 2025; Black et al., 2025; Hancock et al., 2026). Generalist robots, however, must integrate multiple skills within one policy; humanoids must additionally coordinate upper-body manipulation and lower-body locomotion (Jiang et al., 2026; Wei et al., 2026a). This makes reliable integration of heterogeneous skills a central challenge for VLA models.

A common approach is joint fine-tuning on demonstrations pooled across skills. Although effective for homogeneous tasks, it can improve some skills at the expense of others when task demands differ (Kim et al., 2025; Wang et al., 2026). These trade-offs also arise with a shared control interface: with a $\pi_{0.5}$ backbone, joint training on a stream of real-robot tasks improved one normalized task score from 52 to 92 while reducing another from 100 to 33.3 (Zhu et al., 2026). On whole-body tasks, jointly fine-tuned Ψ_0 underperformed its single-task counterparts across all evaluated tasks, with some skills nearly lost (Wei et al., 2026a).

An alternative is to fine-tune experts separately and merge their parameters, an approach effective in language and vision models (Yadav et al., 2023; Nguyen et al., 2026; Marczak et al., 2025). MergeVLA extends this approach to VLAs through task masks, dedicated action heads, and test-time routing (Fu et al., 2026). However, we find that directly merging whole-body VLA experts can severely degrade individual skills, even when the task-specific experts retain those skills. In our Ψ_0 experiments, the merged policy issued unintended locomotion commands during BendPick, a task requiring bending and grasping in place, resulting in task failure. These failures motivate our central

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.

question: how can heterogeneous task behaviors be retained in shared parameters to form a balanced, robust policy without maintaining separate models for individual tasks?

Prior work documents concentrated update spectra and interference arising from disagreements in parameter signs or singular directions (Yadav et al., 2023; Gargiulo et al., 2025; Marczak et al., 2025; Zhang et al., 2026). It also shows that the functional effects of updates depend on inputs and may concentrate in low-dimensional activation or output-shift subspaces (Cheng et al., 2025; Li et al., 2026). Building on these findings, we analyze task-conditioned responses of Ψ_0 experts, using task activation statistics to characterize the functional effects of their parameter updates. We observe opposing components within a shared task-relevant response subspace, concentrated along high-energy response directions. This structure co-occurs with unintended action shifts after merging, suggesting that energy concentration can amplify the impact of opposing responses. These observations motivate regularizing update responses before merging and calibrating the resulting policy afterward.

We propose GeoMerge-OPD, which combines regularized expert fine-tuning, static parameter merging, and on-policy distillation (OPD). Starting from a common pretrained backbone, each expert is fine-tuned with G-isometry to constrain update-response gains in task-activation subspaces. This regularizer acts on responses induced by task inputs, reducing excessive dominance by a few directions. Static merging then establishes a balanced initialization with preliminary multitask competence. Inspired by multi-teacher OPD for integrating heterogeneous expert capabilities in language and text-to-image models (Ma et al., 2026; Fang et al., 2026), we use OPD to consolidate skills from task-specific VLA experts into a unified student policy. The student interacts with the environment in closed loop and generates its own action-flow trajectories. The corresponding task expert provides dense velocity-field supervision at the environment observations and intermediate flow states actually visited by the student. This supervision calibrates residual behavioral deviations under the student's own state and action-generation distributions, further coordinating multitask skills within the shared model.

We evaluate GeoMerge-OPD on LIBERO (Liu et al., 2023), LIBERO-PRO (Zhou et al., 2025), and SIMPLE (Wei et al., 2026b), using $\pi_{0.5}$ and Ψ_0 as the two VLA backbones, covering tabletop manipulation, whole-body loco-manipulation, and generalization to perturbed evaluation conditions. Our main contributions are:

1. **Empirical diagnosis of multitask VLA integration.** We characterize failures in merging task-specific VLA experts and identify concentrated update responses and opposing components in shared task-relevant subspaces, connecting these observations to unintended behavioral shifts in closed-loop execution.
2. **A framework for balanced skill consolidation.** GeoMerge-OPD combines G-isometry regularization and static merging to establish a balanced initialization, followed by task-routed OPD to calibrate residual behavior on student-visited states and action-flow trajectories, consolidating heterogeneous skills in shared parameters.
3. **Skill consolidation and robustness across two backbones.** GeoMerge-OPD achieves 26/30 whole-body successes on SIMPLE versus 22/30 for Joint FT from the same G-regularized merge, reaches 97.1% on LIBERO versus 96.2% for Joint FT from the pretrained base, and attains the highest average LIBERO-PRO success (67.3%) across five perturbation axes among the evaluated policies.

2 Related Work

2.1 Model Merging and VLA Skill Integration

Model merging research has expanded from parameter combination to conflict resolution, task-subspace modeling, and spectral regulation to improve the integration of expert capabilities (Yadav et al., 2023; Gargiulo et al., 2025; Marczak et al., 2025). Mergeability has also been incorporated into expert training and architectural design to prepare models for subsequent consolidation (Yang et al., 2026; Fu et al., 2026). In parallel, studies of input-dependent update effects use input or feature subspaces to characterize how parameter updates influence task behavior (Cheng et al., 2025; Li et al., 2026). Together, these studies provide a basis for understanding post-merging capability

retention through both expert-update structure and task-dependent functional responses, informing the analysis of behavioral degradation in heterogeneous robotic skill integration.

2.2 Policy Distillation and Multi-Teacher Capability Integration

Behavioral data generated by specialized experts can supervise generalist robotic policies, consolidating experience across tasks or manipulated objects within a single policy (Xu et al., 2025; Wang et al., 2025). Applying teacher supervision to student-generated sequences or trajectories helps mitigate distribution mismatch between training and execution and correct errors induced by the student's behavior (Agarwal et al., 2024; Zhong et al., 2026). In large language models and text-to-image generation, multi-teacher OPD further integrates heterogeneous capabilities into a unified student through supervision from domain-specialized teachers on student-generated trajectories (Ma et al., 2026; Fang et al., 2026). To our knowledge, multi-teacher OPD for integrating heterogeneous skills across VLA tasks remains underexplored. Inspired by this capability integration paradigm, we use task-specific VLA teachers to provide dense velocity-field supervision along the student's own action-flow trajectories, conditioned on observations encountered during its environment rollouts, to coordinate multi-task skills after parameter merging.

3 Why Multi-Task Integration Fails

3.1 Capability collapse after integration

We refer to severe degradation of individual skills after multitask integration as capability collapse. On the BendPick, CloseDoor, and OpenOven tasks with Ψ_0, both joint fine-tuning and static parameter merging exhibited this phenomenon, with BendPick nearly failing completely. An examination of failed episodes showed that the merged policy continuously issued lower-body movement commands in BendPick scenes requiring bending and grasping in place, whereas the corresponding task-specific expert did not issue such commands. This unintended movement caused the merged policy to deviate from the control behavior required by the task.

3.2 Response energy concentration and opposing responses

To analyze this deviation, we examine the responses induced by expert updates on task inputs. For a fixed layer, let $G_t = \mathbb{E}_t[xx^\top]$ denote the input second moment of task t. The response energy induced by any weight change A is

$$\mathcal{E}_t(A) = \mathbb{E}_t\|Ax\|_2^2 = \|AG_t^{1/2}\|_F^2. \quad (1)$$

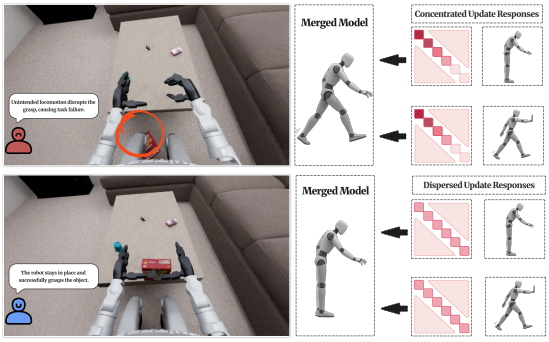

Figure 1: BendPick after merging: unintended locomotion (top) and stationary grasping with regularized experts (bottom). Response matrices are schematic.

For an expert update $\Delta W_t = W_t - W_0$, the squared singular values of $\Delta W_t G_t^{1/2}$ describe the energy along the principal response directions. If most energy concentrates in a few directions, deviations along those directions affect a large fraction of the task response. When other experts introduce opposing components along these directions, parameter merging may cancel the adjustments required by the task or allow the other experts' responses to dominate, causing the merged result to deviate from task requirements. Appendix E.1 analyzes this behavior under fixed activations. Degradation during joint fine-tuning also depends on task gradients at the current parameters, with the relevant conditions provided in Appendix E.3.

We examine the update responses of task-specific fine-tuned experts on Ψ_0, analyzing their energy distributions and directional relationships across experts (Appendix D). For the standard fine-tuned CloseDoor and BendPick experts, the top five principal response directions account for 93% and

78% of response energy, respectively. Within the principal response subspace defined with BendPick as the reference, the median directional cosine between the two experts' responses is -0.73, indicating a pronounced opposing relationship. The tasks also differ in their lower-body control requirements: CloseDoor requires movement, whereas BendPick requires remaining stationary. This response structure co-occurs with capability collapse on BendPick after merging.

To examine the connection between response structure and behavioral degradation, we constrain the update responses within task-activation subspaces during fine-tuning. With this constraint, the median energy shares of the top five directions for CloseDoor and BendPick are 14.3% and 17.6%, respectively, while the response cosine increases from -0.73 to -0.31. The corresponding statically merged policy produces substantially weaker redundant translation commands on BendPick, and the number of successful trials increases from 0/10 to 3/10.

These results indicate that constraining update responses helps mitigate post-merging capability degradation while preserving preliminary execution capabilities, establishing a more balanced starting point for subsequent training. Residual behavioral deviations still require further calibration. GeoMerge-OPD therefore builds on the merging of regularized experts and introduces OPD with task-specific teachers, providing supervision at the environment states and action-flow trajectories actually visited by the student to further integrate multitask skills.

4 Method

GeoMerge-OPD first trains task-specific experts independently with G-isometry regularization, then combines their parameters through static parameter merging to initialize a student with preliminary multitask execution capabilities. The corresponding task teachers subsequently provide velocity-field supervision at the environmental observations and action-generation states actually visited by the student, further correcting behavioral deviations left by the merge.

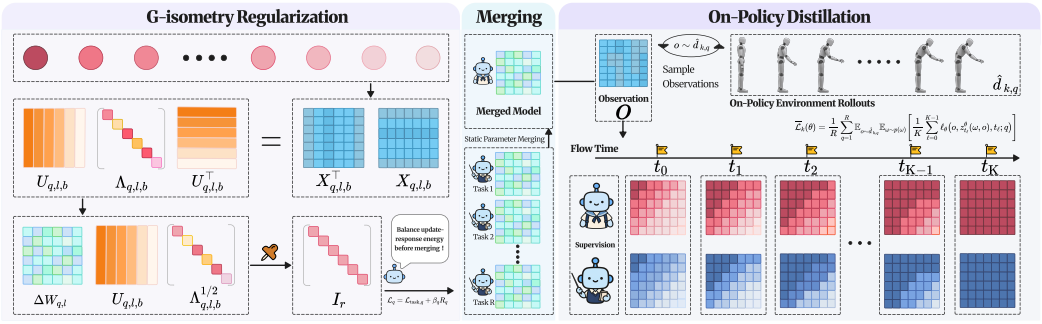

Figure 2: GeoMerge-OPD: G-isometry shapes task-specific update responses; static merging initializes one student; task teachers supervise its velocity field along student-visited observations and action-flow states.

4.1 G-Isometry Regularization

The analysis in Section 3 shows that the functional effect of an expert update depends on the response it produces on task inputs. Accordingly, following the response-energy expression in Eq. (1), we constrain the response of each expert update in the principal activation subspace of its task.

Suppose there are R task-specific experts, with q indexing an expert and its corresponding task group. For input activations at layer l and noise bin b, we combine the batch and token dimensions into a sample dimension, obtaining $X_{q,l,b} \in \mathbb{R}^{n_{q,l,b} \times d_{\text{in}}}$. Here, $n_{q,l,b}$ is the number of activation samples used for the statistics, and d_{in} is the input dimension of the layer. We compute the Gram matrix normalized by the number of samples and retain its leading r eigenvectors:

$$G_{q,l,b} = \frac{1}{n_{q,l,b}} X_{q,l,b}^{\top} X_{q,l,b} \approx U_{q,l,b} \Lambda_{q,l,b} U_{q,l,b}^{\top}. \tag{2}$$

Here, r is the number of retained directions, the columns of $U_{q,l,b}$ represent the principal activation directions, and $\Lambda_{q,l,b}$ is the diagonal matrix of their corresponding eigenvalues. Let $\Delta W_{q,l}$ denote the update at this layer relative to the pretrained weights, and define the weighted response

$$M_{q,l,b} = \Delta W_{q,l} U_{q,l,b} \Lambda_{q,l,b}^{1/2} \in \mathbb{R}^{d_{\text{out}} \times r}, \tag{3}$$

where d_{out} is the output dimension of the layer. Each column of $M_{q,l,b}$ represents the response of the weight update to a principal activation direction, weighted by the square root of that direction's activation energy so that the response scale reflects the role of different input directions in the task.

We penalize deviations of the weighted-response Gram matrix from the identity matrix:

$$R_{q,l,b} = \frac{1}{r} \left\| M_{q,l,b}^{\top} M_{q,l,b} - I_r \right\|_F^2. \tag{4}$$

Here, I_r is the r-dimensional identity matrix, and $\|\cdot\|_F$ is the Frobenius norm. The diagonal entries of the response Gram matrix represent the energies of the column responses, while its off-diagonal entries represent their inner products. Bringing this matrix close to the identity constrains both response scale and directional correlation, mitigating excessive dominance of a few directions in the response energy.

Aggregating the penalties over all noise bins and constrained layers gives the task regularizer and training objective:

$$R_q = \sum_{b=1}^{m} w_b \left(\frac{1}{|\mathcal{A}|} \sum_{l \in \mathcal{A}} R_{q,l,b} \right), \qquad \mathcal{L}_q = \mathcal{L}_{\text{task},q} + \beta_q R_q. \tag{5}$$

Here, m is the number of noise bins, and $\mathcal{A}$ is the set of constrained layers. $\mathcal{L}_{\text{task},q}$ is the original task loss, and β_q controls the regularization strength. The weights w_b decrease monotonically as the noise level increases, imposing stronger constraints in lower-noise bins. All task-specific experts are fine-tuned independently from the same pretrained VLA backbone, yielding their respective parameters θ_q.

4.2 Parameter Merging and Student Initialization

For task-specific experts derived from the same pretrained base, we merge both the vision-language and action-module parameters, including their LoRA updates, using a static merging operator $\mathcal{M}$, obtaining $\theta_{\text{merge}} = \mathcal{M}(\{\theta_q\}_{q=1}^{R})$. The student is initialized with θ_{merge}, providing a starting point that retains preliminary multitask execution capabilities for subsequent distillation. Each task-specific expert retains its own parameters and serves as a fixed teacher during distillation, supervising the student as it corrects the remaining behavioral deviations after merging.

4.3 On-Policy Distillation

We first let the student interact with the environment and then generate actions under the observations it actually encounters, with the corresponding task teacher supervising the action-generation process. This involves two levels of sampling: the robot's execution trajectories in the environment and the internal denoising trajectories from noise to action during each action generation. The teacher provides a velocity field at the intermediate action states actually traversed by the student, helping it correct its current generation errors.

4.3.1 Environment-Level Observation Distribution

Let the student parameters be θ, with initial parameters $\theta^{(0)} = \theta_{\text{merge}}$. The student parameters before update round k are $\theta^{(k)}$, and the corresponding action-generation distribution is $\pi_{\theta^{(k)}}$. Given an observation condition o, the student generates an action $u \sim \pi_{\theta^{(k)}}(\cdot \mid o)$ and interacts with the environment in closed loop.

For task group q, we collect G episodes and save the observations at each action replanning step. Let episode e contain $N_{k,q,e}$ such replanning observations. We construct the empirical distribution

$$\hat{d}_{k,q} = \frac{1}{G} \sum_{e=0}^{G-1} \frac{1}{N_{k,q,e}} \sum_{i=0}^{N_{k,q,e}-1} \delta_{o_{k,q,e,i}}. \tag{6}$$

Here, i indexes observations within an episode, and δ_o is a point mass at observation o. This distribution is normalized within each episode before averaging episodes with equal weight. It is then used to sample observation conditions for subsequent distillation.

4.3.2 Action-Level Velocity-Field Distillation

Given $o \sim \hat{d}_{k,q}$, we hold this observation fixed and rerun the student's action-generation process. In normalized action space, the student starts from standard Gaussian noise and progressively denoises it into an action. Using the flow time convention in which the noise endpoint is $t = 1$ and the action endpoint is $t = 0$, we divide denoising uniformly into K steps, with $t_0 = 1 > \cdots > t_K = 0$. For $\ell = 0, \ldots, K-1$, let $h_\ell = t_\ell - t_{\ell+1} > 0$. We denote by $z^\theta_{t_\ell}$ the intermediate latent action state actually traversed by the student at step ℓ, while z denotes a generic intermediate state.

At each step, we first compute the update mean from the student velocity field and then add a Gaussian perturbation:

$$z^\theta_{t_{\ell+1}} = \mu_\theta\big(z^\theta_{t_\ell}, o, t_\ell\big) + g_\ell \sqrt{h_\ell}\, \xi_\ell, \qquad \xi_\ell \sim \mathcal{N}(0, I). \tag{7}$$

Here, $g_\ell > 0$ is the diffusion coefficient, ξ_ℓ is an independent standard Gaussian perturbation at each step, and I is the identity matrix with the same dimension as the latent action state. The student velocity field $v_\theta(z, t; o)$ predicts the flow velocity from the current action state, flow time, and observation, and is used to construct the conditional mean of the denoising update. The Euler-Maruyama discretization of the reverse SDE gives

$$\mu_\theta(z, o, t_\ell) = \left(1 - \frac{h_\ell g_\ell^2}{2t_\ell}\right) z - h_\ell \left(1 + \frac{g_\ell^2(1 - t_\ell)}{2t_\ell}\right) v_\theta(z, t_\ell; o). \tag{8}$$

This update makes the next denoising state Gaussian conditional on the current state, observation, and time. To compare student and teacher denoising behavior, at the same z, o, t_ℓ, we query the task teacher v_{θ_q} and use the same noise schedule and covariance. The teacher mean μ_{θ_q} is obtained by substituting its velocity field into the equation above.

Let C be the set of valid scalar action coordinates included in the loss, P_C the corresponding coordinate-selection operator, and $D = |C|$. The student and teacher transition kernels on these coordinates can both be written as

$$K^C_\psi(\cdot \mid z, o, t_\ell) = \mathcal{N}\big(P_C \mu_\psi(z, o, t_\ell), g_\ell^2 h_\ell I_D\big), \qquad \psi \in \{\theta, \theta_q\}, \tag{9}$$

where ψ denotes the parameters of either the student or the current task teacher, and I_D is the D-dimensional identity matrix.

Because the covariances are identical, the local KL divergence can be expressed as a difference between means and then reduced to a velocity-field error:

$$D_{\mathrm{KL}}\Big(K^C_\theta \Big\| K^C_{\theta_q}\Big) = \frac{\big\|P_C(\mu_\theta - \mu_{\theta_q})\big\|_2^2}{2 g_\ell^2 h_\ell} = w_\ell \big\|P_C(v_\theta - v_{\theta_q})\big\|_2^2. \tag{10}$$

Here, $\|\cdot\|_2$ is the Euclidean norm. The abbreviated means and velocity fields are evaluated at the same observation, intermediate state, and flow time. The time-step weight $w_\ell = \frac{h_\ell}{2g_\ell^2}\left(1 + \frac{g_\ell^2(1-t_\ell)}{2t_\ell}\right)^2$ converts velocity-field differences into distributional differences for the corresponding denoising step. Derivations of the sampling mean and KL divergence are provided in Appendix E.4.

During distillation, the teacher provides supervision at the student-visited $z^{\theta}_{t_\ell}$. Normalizing by the number of valid action coordinates, we define the local loss

$$\ell_\theta\big(o, z^{\theta}_{t_\ell}, t_\ell; q\big) = \frac{w_\ell}{D} \left\| P_C\Big[v_\theta\big(z^{\theta}_{t_\ell}, t_\ell; o\big) - \mathrm{SG}\Big[v_{\theta_q}\big(z^{\theta}_{t_\ell}, t_\ell; o\big)\Big]\Big] \right\|_2^2 . \tag{11}$$

Here, SG treats the teacher output as a stop-gradient supervision target and also blocks the gradient of the teacher output with respect to its input state. Both the student velocity field and its action-generation path participate in backpropagation.

During the current update, the environment observation distribution $\hat{d}_{k,q}$ remains fixed. We collectively denote the initial noise for action generation and all stepwise perturbations by the base noise ω, whose joint distribution $p(\omega)$ is independent of the parameters being optimized. The student path can therefore be reparameterized as a differentiable function $\mathbf{z}^{\theta}(\omega, o)$, giving the overall training objective

$$\bar{\mathcal{L}}_k(\theta) = \frac{1}{R}\sum_{q=1}^{R} \mathbb{E}_{o\sim\hat{d}_{k,q}} \mathbb{E}_{\omega\sim p(\omega)} \left[\frac{1}{K}\sum_{\ell=0}^{K-1} \ell_\theta\big(o, z^{\theta}_{t_\ell}(\omega, o), t_\ell; q\big) \right] . \tag{12}$$

We compute the reparameterized gradient with the above stop-gradient operation, $\nabla_\theta \bar{\mathcal{L}}_k(\theta)\big|_{\theta=\theta^{(k)}}$, update the student parameters, and alternate between environment sampling and distillation updates.

5 Experiments

5.1 Experimental setup

Benchmarks and evaluation. We evaluate Ψ_0 on SIMPLE (Wei et al., 2026b) for whole-body control and $\pi_{0.5}$ on LIBERO (Liu et al., 2023) and LIBERO-PRO (Zhou et al., 2025) for manipulation. Task-specific SIMPLE and suite-specific LIBERO models are independently fine-tuned from their respective pretrained bases. Evaluation uses ten episodes per SIMPLE task (BendPick, CloseDoor, OpenOven) and 100 per LIBERO suite per seed over three seeds. LIBERO-PRO tests these policies under perturbations without retraining. We report equally weighted task, suite, or perturbation averages.

Comparisons and training. Task- and suite-specific fine-tuned models serve as references. TIES, WUDI, and RegMean++ use official implementations adapted only to Ψ_0 and $\pi_{0.5}$ and are evaluated after merging. Joint fine-tuning (Joint FT) is trained to convergence. GRPO-mix (Shao et al., 2024)

Table 1: **Whole-body skill consolidation on SIMPLE (Ψ_0).** Success counts.

Method	Init.	BendPick	CloseDoor	OpenOven	Total
Single-task Finetuned Model					
Standard FT (unreg.)	B	9/10	10/10	10/10	29/30
G-isometry FT	B	10/10	10/10	10/10	30/30
Integration by static merging					
TIES	G	2/10	4/10	1/10	7/30
WUDI	G	3/10	6/10	2/10	11/30
RegMean++	G	3/10	5/10	4/10	12/30
Integration with post-training					
Joint FT	B	0/10	6/10	**8/10**	14/30
Joint FT	G	5/10	9/10	**8/10**	22/30
GRPO-mix	G	5/10	7/10	7/10	19/30
GeoMerge-OPD (Ours)	G	**8/10**	**10/10**	**8/10**	**26/30**

B: pretrained base. S: unregularized experts. G: G-isometry experts (static merging) or their RegMean++ merge (post-training). Joint FT (G) and GRPO-mix share our G initialization. Joint FT and OPD both fully fine-tune the action backbone and interfaces. GRPO-mix and GeoMerge-OPD match preset episodes, updates, and observation exposures. OPD additionally uses teacher velocity-field supervision. Bold marks the best integration results, including ties.

Table 2: **Consolidation and robustness with $\pi_{0.5}$.** Mean success rates (%) over three evaluations.

		LIBERO					LIBERO-PRO					
Method	**Init.**	**Spat.**	**Obj.**	**Goal**	**Long**	**Avg.**	**Obj.**	**Pos.**	**Sem.**	**Task**	**Env.**	**Avg.**
Single-task Finetuned Model												
Standard FT (unreg.)	B	99.3	100.0	99.0	99.7	99.5	87.3	23.0	96.0	5.7	53.7	53.1
G-isometry FT	B	100.0	98.7	99.0	99.3	99.3	86.7	22.0	97.0	7.7	55.0	53.7
Integration by static merging												
TIES	G	77.3	74.7	66.3	50.7	67.3	42.3	17.3	57.0	14.3	18.3	29.9
WUDI	G	65.3	61.3	65.0	43.7	58.8	38.3	10.3	45.7	20.0	13.0	25.5
RegMean++	G	86.3	84.7	66.0	55.7	73.2	58.0	25.0	55.7	16.0	29.7	36.9
Integration with post-training												
Joint FT	B	**97.3**	95.7	**97.0**	94.7	96.2	85.3	33.3	97.0	42.0	64.7	64.5
Joint FT	G	97.0	95.7	96.3	91.7	95.2	**86.0**	35.3	**97.7**	40.7	65.0	64.9
GRPO-mix	G	93.3	91.7	90.7	84.7	90.1	79.3	37.7	89.0	33.7	59.3	59.8
GeoMerge-OPD	G	**97.3**	**98.7**	95.3	**97.0**	**97.1**	85.3	**42.7**	95.0	**46.7**	**67.0**	**67.3**

Codes and boldface follow Table 1. Avg. equally weights four LIBERO suites or five PRO axes. Sem.: language paraphrasing.

and GeoMerge-OPD match preset episodes, optimizer updates, and observation exposures. OPD additionally uses teacher velocity-field supervision. Expert and joint fine-tuning use VLM LoRA with $r = \alpha = 64$ and fully fine-tune learnable interfaces. Both use a global batch size of 128, with learning rates of 3×10^{-4} for $\pi_{0.5}$ and 1×10^{-4} for Ψ_0, and full configurations in Appendix A.

5.2 Whole-body skill integration

All three static merging operators retain preliminary execution capabilities on every SIMPLE task after merging G-isometry-regularized models (Table 1). RegMean++ yields the highest static average and most balanced task profile, so its merged model is our default post-training initialization. We also initialize Joint FT from this G-regularized merge to construct a strengthened baseline sharing our starting point. This raises Joint FT from 14/30 with the pretrained base to 22/30, extending the benefit of our initialization to joint fine-tuning. From the same initialization, GeoMerge-OPD reaches 26/30 and is best or tied-best across all three tasks among the evaluated integration methods, yielding strong, balanced whole-body performance.

5.3 Cross-backbone evaluation and robustness

We extend evaluation from Ψ_0 whole-body control to $\pi_{0.5}$ manipulation (Table 2). GeoMerge-OPD achieves the highest average among evaluated multi-skill integration methods on LIBERO and LIBERO-PRO, at 97.1% and 67.3%. On LIBERO, its average surpasses static merging, GRPO-mix, and joint fine-tuning from the pretrained base. On LIBERO-PRO, its average exceeds the 64.9% of joint fine-tuning from the G-regularized merge, with the best scores on position perturbations, task variations, and environment changes. That baseline remains slightly higher on object perturbations and language paraphrasing, by 0.7 and 2.7 percentage points, respectively. These results support benefits across VLA backbones and strong shared-policy performance under changing task conditions.

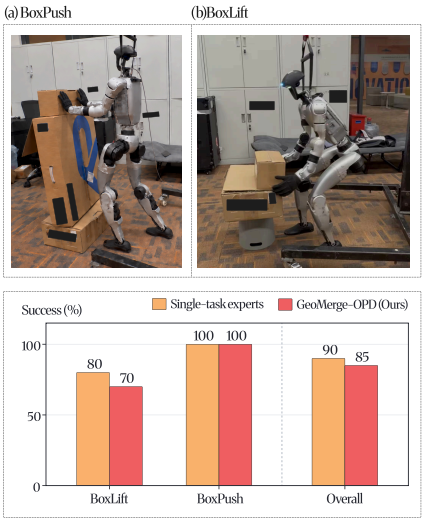

Figure 3: Unitree G1 evaluation, with 10 trials per policy per task.

5.4 Real-world experiments

Tasks. We evaluate GeoMerge-OPD on a Unitree G1 humanoid robot in two stationary manipulation tasks (Fig. 3). BoxPush requires pushing an elevated cardboard box forward a short distance, and BoxLift requires

bending down to lift a box from a low platform and hold
it stably. We collect 50 teleoperated demonstrations per task for fine-tuning. Each policy is evaluated over 10 trials per task using the same observation interface and low-level controller. Success requires autonomous task completion.

Results. GeoMerge-OPD achieves 85% overall success (17/20), compared with 90% (18/20) for the single-task experts. Both achieve 100% on BoxPush. On BoxLift, GeoMerge-OPD achieves 70%, compared with 80% for the corresponding specialist. These results provide preliminary evidence that one shared action expert can retain both real-world skills.

5.5 Ablations

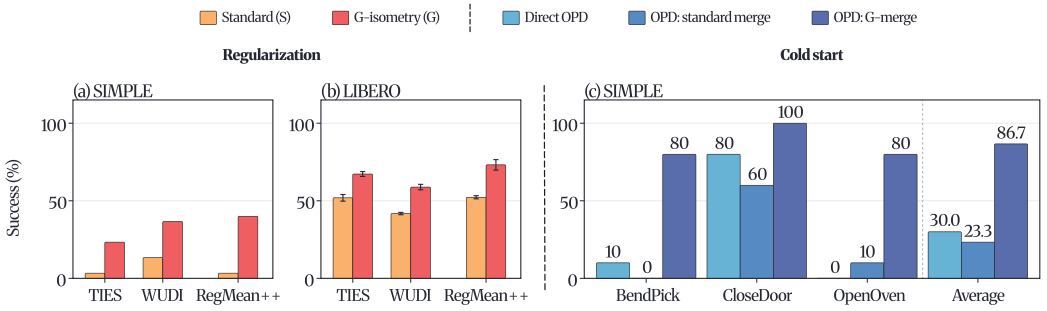

Figure 4: (a,b) Static merging from standard fine-tuned models (S, unregularized) and G-isometry fine-tuned models (G). LIBERO error bars show sample SD of seed-level suite averages. (c) SIMPLE success after direct OPD or OPD from standard (unregularized) and G-isometry merged initializations. The teachers are unchanged across these initializations. Average equally weights the three tasks.

Regularization before merging. G-isometry improves the average static-merging success of TIES, WUDI, and RegMean++ on both SIMPLE and LIBERO (Figure 4a,b). This consistency across operators supports adjusting task-specific parameter updates before merging. The resulting task competence provides the foundation for subsequent behavior-level refinement.

Initialization for on-policy distillation. OPD from the G-regularized merge achieves 86.7% mean success, compared with 30.0% for direct OPD and 23.3% for OPD from a standard (unregularized) merge (Figure 4c). The G-initialized policy retains all three skills, whereas each alternative exhibits complete collapse on one of the evaluated tasks. The two ablations support complementary roles: regularization prepares a merged policy with preliminary task competence, and OPD uses task-specific supervision to calibrate its behavior into a more capable shared policy.

6 Conclusion

GeoMerge-OPD combines response-space regularization, parameter merging, and on-policy distillation into a complete framework for integrating heterogeneous robotic skills. G-isometry improves the mergeability of expert updates during independent fine-tuning, while OPD uses task teachers to calibrate post-merging behavioral deviations on observations and action-generation trajectories actually visited by the student. Experiments with two VLA backbones demonstrate balanced multi-skill performance, and LIBERO-PRO further shows robustness under perturbations. Real-world experiments further demonstrate the feasibility of integrating multiple manipulation skills into a shared policy on a humanoid robot.

Limitations and future work. Current real-world validation mainly covers two stationary manipulation tasks, and the expert combinations studied are limited in scale. Future work will extend to real-world tasks that alternate locomotion and manipulation and involve richer contact modes, and examine skill retention, balance, and robustness as the number of experts increases.

AI USE STATEMENT

Generative AI tools (Codex) assisted with translation, manuscript editing, consistency checks, and summarizing and visualizing author-provided experimental results. The authors take full responsibility for the final manuscript.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, pp. 21246–21263, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40, 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever Started the Interference Should End It: Guiding Data-Free Model Merging via Task Vectors. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10121–10143, 2025. URL https://proceedings.mlr.press/v267/cheng25h.html.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge Insulating Vision-Language-Action Models: Train Fast, Run Fast, Generalize Better. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3439. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/94e936034d12bcd04834ec2773f02aff-Abstract-Conference.html.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, and Feng Zhao. Flow-OPD: On-Policy Distillation for Flow Matching Models, 2026. URL https://arxiv.org/abs/2605.08063.

Yuxia Fu, Zhizhen Zhang, Yuqi Zhang, Zijian Wang, Zi Huang, and Yadan Luo. MergeVLA: Cross-Skill Model Merging Toward a Generalist Vision-Language-Action Agent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22335–22347, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fu_MergeVLA_Cross-Skill_Model_Merging_Toward_a_Generalist_Vision-Language-Action_Agent_CVPR_2026_paper.html.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task Singular Vectors: Reducing Task Interference in Model Merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18695–18705, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gargiulo_Task_Singular_Vectors_Reducing_Task_Interference_in_Model_Merging_CVPR_2025_paper.html.

Yanggan Gu, Shuo Cai, Zihao Wang, Wenjun Wang, Yuanyi Wang, Pengkai Wang, Sirui Huang, Su Lu, Jianmin Wu, and Hongxia Yang. FeatCal: Feature calibration for post-merging models, 2026. URL https://arxiv.org/abs/2605.13030.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as Language: Fine-Tuning VLMs into VLAs Without Catastrophic Forgetting. In *International Conference on Learning Representations*, pp. 75333–75360, 2026. URL https://proceeding

s.iclr.cc/paper_files/paper/2026/hash/7a0f8055c838df8e62329a76c7c6403d-Abstract-Conference.html.

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Chuang Wang, Zhihui Peng, and Hongyang Li. WholeBodyVLA: Towards Unified Latent VLA for Whole-Body Loco-Manipulation Control. In *International Conference on Learning Representations*, pp. 157438–157461, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ff7373914a96956f2a7cacbdf3b0b8d8-Abstract-Conference.html.

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*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Model Merging in the Essential Subspace. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31145–31154, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Li_Model_Merging_in_the_Essential_Subspace_CVPR_2026_paper.html.

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://openreview.net/forum?id=xzEtNSuDJk.

Shangge Liu, Yuehan Yin, Lei Wang, Qi Fan, Yinghuan Shi, Wenbin Li, Yang Gao, and Dacheng Tao. Understanding and Enforcing Weight Disentanglement in Task Arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28744–28753, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liu_Understanding_and_Enforcing_Weight_Disentanglement_in_Task_Arithmetic_CVPR_2026_paper.html.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. MOPD: Multi-Teacher On-Policy Distillation for Capability Integration in LLM Post-Training, 2026. URL https://arxiv.org/abs/2606.30406.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No Task Left Behind: Isotropic Model Merging with Common and Task-Specific Subspaces. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43177–43199, 2025. URL https://proceedings.mlr.press/v267/marczak25a.html.

The-Hai Nguyen, Huu-Tien Dang, Takeshi Suzuki, and Le-Minh Nguyen. RegMean++: Enhancing Effectiveness and Generalization of Regression Mean for Model Merging. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=H5lDsSCS9i.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 66727–66754, 2023. doi: 10.52202/075280-2913. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d28077e5ff52034cd35b4aa15320caea-Abstract.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL https://arxiv.org/abs/2402.03300.

Wenbo Wang, Fangyun Wei, Lei Zhou, Xi Chen, Lin Luo, Xiaohan Yi, Yizhong Zhang, Yaobo Liang, Chang Xu, Yan Lu, Jiaolong Yang, and Baining Guo. UniGraspTransformer: Simplified Policy Distillation for Scalable Dexterous Robotic Grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12199–12208, June 2025. URL

https://openaccess.thecvf.com/content/CVPR2025/html/Wang_UniGraspTransformer_Simplified_Policy_Distillation_for_Scalable_Dexterous_Robotic_Grasping_CVPR_2025_paper.html.

Ye Wang, Sipeng Zheng, Hao Luo, Wanpeng Zhang, Haoqi Yuan, Chaoyi Xu, Haiweng Xu, Yicheng Feng, Mingyang Yu, Zhiyu Kang, Zongqing Lu, and Qin Jin. Rethinking Visual-Language-Action Model Scaling: Alignment, Mixture, and Regularization, 2026. URL https://arxiv.org/abs/2602.09722.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, Sheng Zang, Weiduo Yuan, Marco Pavone, Di Huang, and Yue Wang. Ψ_0: An Open Foundation Model Towards Universal Humanoid Loco-Manipulation. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026a. doi: 10.15607/RSS.2026.XXII.021. URL https://www.roboticsproceedings.org/rss22/p021.html. Extended version: https://arxiv.org/abs/2603.12263v1.

Songlin Wei, Zhenhao Ni, Jie Liu, Zhenyu Zhao, Junjie Ye, Hongyi Jing, Junkai Xia, Xiawei Liu, Michael Leong, Liang Heng, Di Huang, and Yue Wang. SIMPLE: Simulation-Based Policy Learning and Evaluation for Humanoid Loco-Manipulation, 2026b. URL https://arxiv.org/abs/2606.08278.

Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. RLDG: Robotic Generalist Policy Distillation via Reinforcement Learning. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.028. URL https://www.roboticsproceedings.org/rss21/p028.html.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. TIES-Merging: Resolving Interference When Merging Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115, 2023. doi: 10.52202/075280-0310. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

Enneng Yang, Qun Yang, Peng Wang, Anke Tang, Guibing Guo, Xiaochun Cao, and Li Shen. MergOPT: A Merge-Aware Optimizer for Robust Model Merging. In *The Fourteenth International Conference on Learning Representations*, pp. 48400–48425, 2026. URL https://openreview.net/forum?id=C21rz8mo65.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Han-Chen Zhang, Zi-Hao Zhou, Mao-Lin Luo, Shimin Di, Min-Ling Zhang, and Tong Wei. DC-Merge: Improving Model Merging with Directional Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22248–22258, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_DC-Merge_Improving_Model_Merging_with_Directional_Consistency_CVPR_2026_paper.html.

Zhide Zhong, Haodong Yan, Junfeng Li, Junjie He, Tianran Zhang, and Haoang Li. VLA-OPD: Bridging Offline SFT and Online RL for Vision-Language-Action Models via On-Policy Distillation, 2026. URL https://arxiv.org/abs/2603.26666.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards Robust and Fair Evaluation of Vision-Language-Action Models Beyond Memorization, 2025. URL https://arxiv.org/abs/2510.03827.

Jiarun Zhu, Yijun Hong, Xiaoquan Sun, Zetian Xu, Qijun He, Haijier Chen, Zhiyong Wang, Mingqi Yuan, Wenjun Zeng, and Jiayu Chen. Can Vision-Language-Action Models Learn from Real-World Data Continually without Forgetting?, 2026. URL https://arxiv.org/abs/2605.26820.

A Training and Method Settings

We use PaliGemma with a Gemma action expert for $\pi_{0.5}$ and Qwen3-VL-2B with an action expert for Ψ_0. All training runs use proprioception, four A100 80GB GPUs, and one training seed.

Table 3: Expert and joint fine-tuning settings.

Setting	$\pi_{0.5}$ **Expert**	$\pi_{0.5}$ **Joint FT**	Ψ_0 **Expert**	Ψ_0 **Joint FT**
Training scope	One suite	Four suites	One task	Three tasks
Effective global batch	128	128	128	128
Learning rate	3×10^{-4}	3×10^{-4}	1×10^{-4}	1×10^{-4}
VLM tuning	LoRA	LoRA	LoRA	LoRA
LoRA rank r	64	64	64	64
LoRA scale α	64	64	64	64
Action backbone	LoRA	Full	LoRA	Full
Interfaces	Full	Full	Full	Full
Training steps	15,000	30,000	30,000	60,000
Effective images	2	2	1	1

Image counts exclude empty cameras. Joint FT samples the three Ψ_0 tasks equally.

Table 4: Regularization, merging, and post-training settings.

Setting	$\pi_{0.5}$	Ψ_0 **(three skills)**
G-isometry		
Update / r_G / equal-width bins	$\Delta W = BA$ / 64 / 8	$\Delta W = BA$ / 64 / 8
Training obs. per expert / Gram	1,024 / fixed	1,024 / fixed
Regularization strength β_q	8	2
Merging		
Operator / α_{RM} / ridge	RegMean++ / 0.3 / 0	RegMean++ / 0.3 / 0
Experts / weight per expert	4 / $\frac{1}{4}$	3 / $\frac{1}{3}$
Calibration obs. per expert / total	1,024 / 4,096	1,024 / 3,072
Gram updates	Blockwise	Blockwise
OPD / GRPO-mix		
VLM LoRA / interfaces / action backbone	$r = \alpha = 64$ / Full / Full	$r = \alpha = 64$ / Full / Full
Module/interface LR: VLM / action	10^{-6} / 10^{-5}	10^{-6} / 10^{-5}
Optimizer / observation batch	AdamW / 128	AdamW / 128
Episodes per group / updates per round	2 / 1 (current round only)	2 / 1 (current round only)
Optimizer updates / collected episodes	800 / 6,400	1,000 / 6,000
Observation exposures	102,400	128,000
Action chunk length / executed steps	50 / 10	30 / 24
		1,080 / 1,080 (initial)
OPD loss / GRPO stochastic dims.	350 / 1,600	864 / 864 (6-step RTC prefix)
Uniform flow steps K / noise scale η	10 / 0.1	10 / 0.1
GRPO		
Group size / standardized-advantage ϵ	2 / 10^{-8}	2 / 10^{-8}
Clipping / reference-KL coefficient	0.2 / 0.01	0.2 / 0.01
Reward / reference policy	Binary success / initial merge	Binary success / initial merge

G-isometry uses Gram matrices precomputed from the common pretrained base. GRPO groups independent trajectories sharing a task and complete initial state. Observation exposures count draws with replacement. The unnormalized bin weights are $w_b = \exp[-(b-0.5)/(8\times0.3)]/8$, $b = 1,\dots,8$. Sampling and reductions follow Appendix B. With $t_\ell = 1-\ell/K$ and $h_\ell = 0.1$, the diffusion coefficient is

$$g_\ell = \eta\sqrt{\frac{t_\ell}{1-\bar{t}_\ell}}, \qquad \bar{t}_\ell = \begin{cases} t_{\ell+1}, & t_\ell = 1, \\ t_\ell, & t_\ell < 1, \end{cases} \quad \ell = 0,\dots,K-1. \tag{A.1}$$

Only the denominator substitutes t_1 at $t_0 = 1$. Drift and local-KL weights retain t_ℓ.

TIES, WUDI, and RegMean++ follow official implementations (Yadav et al., 2023; Cheng et al., 2025; Nguyen et al., 2026), adapted only to the two backbones. OPD and GRPO-mix share the G-regularized merge and matched episode, update, and observation-exposure budgets. OPD adds teacher supervision.

Numerical stability in static merging. During our reproduction of RegMean++ on $\pi_{0.5}$ and Ψ_0, we encountered severely ill-conditioned regression systems associated with calibration features of very low energy. The resulting solutions could exhibit disproportionately large merged weights despite remaining finite. To mitigate this behavior, we explored a stabilization in the spirit of FeatCal's anchored feature calibration (Gu et al., 2026): computing corrections around reference merged weights while suppressing amplification in weakly supported directions. This treatment alleviates the extreme numerical behavior, but does not by itself guarantee balanced closed-loop task performance.

B Algorithm and Method Details

This section details the flow of parameters, observations, and gradients through expert training, merging, and two-level on-policy distillation. We use R task groups, indexed by q, update round k, and flow step ℓ. The training settings are specified separately; the algorithm below uses the objectives and sampling convention of the main text.

B.1 Expert Training and Student Initialization

Each task expert is independently fine-tuned from the same pretrained base. Task activation statistics define its principal input directions and their energies. G-isometry acts on the corresponding weighted response of the expert update relative to that base. The expert for task group q minimizes $\mathcal{L}_{\text{task},q} + \beta_q R_q$, where R_q aggregates the response penalties over constrained layers and noise bins.

The static operator combines the experts' vision-language and action-module parameters, including their LoRA updates, to form $\theta_{\text{merge}} = \mathcal{M}(\{\theta_q\}_{q=1}^{R})$. The shared student is initialized with $\theta^{(0)} = \theta_{\text{merge}}$. The experts retain their own parameters and serve as fixed task teachers. The student provides a single policy for execution and receives subsequent updates; for observations from task group q, teacher θ_q provides the velocity-field targets. The task index organizes teacher supervision, while the resulting actions are generated by the shared student.

Train experts	$\longrightarrow$	**Merge parameters**	$\longrightarrow$	**Distill the student**
G-isometry		VLA initialization		two-level on-policy sampling

B.2 Complete Algorithm

Algorithm B.1 connects the three stages. Each OPD round collects environment observations, resamples student flow paths at those observations, aggregates the loss, and performs one student update. The per-task teachers remain fixed throughout distillation.

B.3 Two-Level Sampling and Episode Weights

Environment-level observations. For $\pi_{0.5}$, a task group is one LIBERO suite, with its ten tasks selected in cyclic order across rounds. For Ψ_0, each group is one SIMPLE task. Before update round k, student $\theta^{(k)}$ interacts with the environment in closed loop using $\pi_{\theta^{(k)}}$. For each task group, it collects G episodes and saves the observation at each action replanning step. These observations reflect the environment states reached through the student's preceding actions. If episode e contains $N_{k,q,e}$ saved observations, the empirical distribution is

$$\hat{d}_{k,q} = \frac{1}{G} \sum_{e=0}^{G-1} \frac{1}{N_{k,q,e}} \sum_{i=0}^{N_{k,q,e}-1} \delta_{o_{k,q,e,i}}. \tag{B.1}$$

Here δ_o is a point mass at o. Sampling from this distribution means selecting an episode uniformly, then selecting one of its saved replanning observations uniformly. Each episode therefore receives

Algorithm B.1 GeoMerge-OPD training and integration

Input: Common pretrained base θ_0, task data $\{\mathcal{D}_q\}_{q=1}^{R}$, merge operator $\mathcal{M}$, and training settings
Output: A shared student θ

Stage 1: independent regularized experts
for $q = 1, \ldots, R$ **do**
  Initialize expert θ_q from the common base
  Construct G-isometry from the task activation statistics
  Independently train with $\mathcal{L}_{\text{task},q} + \beta_q R_q$
end for
Stage 2: merging and student initialization
Merge vision-language and action-module parameters, including LoRA updates: $\theta_{\text{merge}} \leftarrow \mathcal{M}(\{\theta_q\}_{q=1}^{R})$
Initialize student $\theta \leftarrow \theta_{\text{merge}}$
Preserve and fix the task teachers $\{\theta_q\}_{q=1}^{R}$
Stage 3: two-level on-policy distillation
for each update round k **do**
  Set $\theta^{(k)} \leftarrow \theta$ and collect G environment episodes per task group
  Save replanning observations and construct $\hat{d}_{k,q}$ by Eq. (B.1)
  for each task group q and each of its observation samples **do**
    Uniformly sample an episode, then a saved observation o within it
    Hold o fixed; draw fresh $\omega = (z_{t_0}, \xi_0, \ldots, \xi_{K-1})$
    Initialize the path with the sampled z_{t_0}; retain its computation graph
    for $\ell = 0, \ldots, K-1$ **do**
      $v_\ell \leftarrow v_\theta(z_{t_\ell}^{\theta}, t_\ell; o)$
      $y_\ell \leftarrow \text{SG}[v_{\theta_q}(z_{t_\ell}^{\theta}, t_\ell; o)]$
      Accumulate $\frac{w_\ell}{D}\|P_C(v_\ell - y_\ell)\|_2^2$
      Advance the differentiable state using Eq. (B.2)
    end for
  end for
  Average observation/noise samples and flow steps within each task
  Average task-group losses as in Eq. (B.5)
  Backpropagate through the student field and path at $\theta = \theta^{(k)}$
  Perform one update of the student parameters
end for
return θ

equal total weight. Uniform sampling from a concatenated list of all observations would instead give longer episodes more weight and define a different training distribution.

Action-level resampling. For a sampled observation $o \sim \hat{d}_{k,q}$, we hold o fixed and rerun the student's action-generation process with fresh Gaussian noise. At every intermediate state $z_{t_\ell}^{\theta}$, the student and the corresponding teacher receive the same observation, intermediate state, and flow time. The teacher supplies a velocity target at a state generated by the student.

Table 5: Environment execution and flow resampling.

Environment execution	**Flow resampling for supervision**
Execute the student's actions and obtain subsequent observations.	Fix one saved observation and draw a new action-generation path.
Save replanning observations with their episode identities.	Query the teacher at each student-visited intermediate action state.
Define the empirical observation distribution $\hat{d}_{k,q}$.	Compute the loss along a differentiable student flow path.

The action-generation paths used during environment execution determine which observations are reached. The paths resampled at those observations provide the distillation loss. Their connection

is the saved observation condition; resampling does not require reusing the execution-time noise or reproducing the execution-time intermediate states.

In normalized action space, $z_{t_0} \sim \mathcal{N}(0, I)$ and the flow times are uniform with $t_0 = 1 > \cdots > t_K = 0$. Let $h_\ell = t_\ell - t_{\ell+1} > 0$. The resampled student path follows

$$z^\theta_{t_{\ell+1}} = \mu_\theta(z^\theta_{t_\ell}, o, t_\ell) + g_\ell \sqrt{h_\ell}\, \xi_\ell, \qquad \xi_\ell \sim \mathcal{N}(0, I), \tag{B.2}$$

where $g_\ell > 0$ and the perturbations are independent. The reverse-SDE mean used in the main text is

$$\begin{aligned} \mu_\theta(z, o, t_\ell) = & \left(1 - \frac{h_\ell g_\ell^2}{2t_\ell}\right) z \\ & - h_\ell \left(1 + \frac{g_\ell^2(1 - t_\ell)}{2t_\ell}\right) v_\theta(z, t_\ell; o). \end{aligned} \tag{B.3}$$

B.4 Loss Reduction and Gradient Flow

The student and teacher use the same noise schedule and covariance when comparing their transitions at a common o, z, t_ℓ. The equal-covariance Gaussian KL is therefore a time-weighted squared velocity difference. Let C contain the valid scalar action coordinates, let P_C select those coordinates, and let $D = |C|$. Subsequent Ψ_0 chunks hold the six-step RTC prefix fixed and exclude it from the loss and stochastic kernel. The local loss is

$$\begin{aligned} \ell_\theta(o, z^\theta_{t_\ell}, t_\ell; q) = \frac{w_\ell}{D} \Big\| P_C \Big[& v_\theta(z^\theta_{t_\ell}, t_\ell; o) \\ & - \mathrm{SG}\left[v_{\theta_q}(z^\theta_{t_\ell}, t_\ell; o)\right] \Big] \Big\|_2^2, \end{aligned} \tag{B.4}$$

with the main-text time-step weight

$$w_\ell = \frac{h_\ell}{2g_\ell^2} \left(1 + \frac{g_\ell^2(1 - t_\ell)}{2t_\ell}\right)^2.$$

The factor $1/D$ averages the sum of squared errors over valid coordinates. The full objective additionally averages over the K flow steps and the R task groups. These are separate reductions: once a coordinate mean has been computed, the same coordinate count is not divided out again.

During the current update, $\hat{d}_{k,q}$ is fixed. Let ω collect the initial noise and all stepwise perturbations, with a distribution $p(\omega)$ independent of the optimized parameters. The resampled student path is a differentiable function $\mathbf{z}^\theta(\omega, o)$. The reparameterized objective is

$$\begin{aligned} \bar{\mathcal{L}}_k(\theta) = \frac{1}{R} \sum_{q=1}^{R} & \mathbb{E}_{o \sim \hat{d}_{k,q}} \mathbb{E}_{\omega \sim p(\omega)} \\ & \left[\frac{1}{K} \sum_{\ell=0}^{K-1} \ell_\theta\big(o, z^\theta_{t_\ell}(\omega, o), t_\ell; q\big) \right]. \end{aligned} \tag{B.5}$$

Observation and noise samples estimate the two expectations within each task group. The Ψ_0 batch rotates task quotas of 43/43/42. Losses are averaged within each episode, then equally across episodes and task groups.

Gradients reach the student parameters through both the velocity prediction at the current intermediate state and the earlier student flow updates that produced this state. Stop-gradient is applied to the complete teacher output. It blocks derivatives with respect to teacher parameters and the teacher output's derivative with respect to its input state. The student field and student path retain their computation graph. Environment interaction and the saved observations are fixed conditions for the current update, and no derivative is propagated through the past environment execution history.

Evaluating the resampled path at the pre-update parameter values specifies where the gradient is computed; it does not detach the student path. With the teacher-output stop-gradient in Eq. (B.4), the algorithm evaluates $\nabla_\theta \bar{\mathcal{L}}_k(\theta)\big|_{\theta=\theta^{(k)}}$, updates the student once, and then returns to environment sampling for the next round. Appendix E.4 derives the local Gaussian KL and its relationship to path-level errors and task returns; the reduction and gradient rules above specify the training objective used here.

C Detailed Ablation Results

This appendix gives the task- and suite-level results underlying Figure 4. S denotes standard fine-tuning without G-isometry regularization, and G denotes G-isometry-regularized fine-tuning.

Regularization before merging. Tables 6 and 7 evaluate static merging before OPD. Source identifies the fine-tuned input models. LIBERO uses 100 episodes per suite per seed. Its Avg. equally weights the four suites, and the associated SD is computed across their three seed-level means, assuming corresponding seed order across suites.

Table 6: Static merging on SIMPLE (Ψ_0), underlying Figure 4a. Task entries are successful episodes out of ten. Avg. is the task-macro success rate (%).

Merger	Source	BendPick	CloseDoor	OpenOven	Avg.
TIES	S	0/10	1/10	0/10	3.3
WUDI	S	0/10	2/10	2/10	13.3
RegMean++	S	0/10	0/10	1/10	3.3
TIES	G	2/10	4/10	1/10	23.3
WUDI	G	3/10	6/10	2/10	36.7
RegMean++	G	3/10	5/10	4/10	40.0

Table 7: Static merging on LIBERO ($\pi_{0.5}$), underlying Figure 4b. Success rates (%) are mean $\pm$ sample SD across three seeds.

Merger	Source	Spatial	Object	Goal	Long	Avg.
TIES	S	66.7 ± 3.1	65.3 ± 2.1	46.3 ± 0.6	29.3 ± 4.7	51.9 ± 2.1
WUDI	S	56.3 ± 4.6	62.3 ± 2.1	35.7 ± 1.2	12.7 ± 0.6	41.8 ± 0.8
RegMean++	S	73.7 ± 2.3	75.3 ± 4.6	45.0 ± 3.6	15.0 ± 3.6	52.3 ± 1.0
TIES	G	77.3 ± 1.2	74.7 ± 2.3	66.3 ± 2.5	50.7 ± 6.8	67.3 ± 1.6
WUDI	G	65.3 ± 1.5	61.3 ± 2.3	65.0 ± 2.6	43.7 ± 3.8	58.8 ± 1.8
RegMean++	G	86.3 ± 5.5	84.7 ± 3.5	66.0 ± 5.2	55.7 ± 3.5	73.2 ± 3.4

OPD initialization. Table 8 reports final SIMPLE outcomes. B starts from the pretrained base, while S and G start from the corresponding merged models. SIMPLE has no repeated-seed uncertainty estimates.

Table 8: OPD initialization on SIMPLE (Ψ_0), underlying Figure 4c. The teachers are unchanged across initializations. Task entries are successful episodes out of ten. Avg. is the task-macro success rate (%).

Initialization	Code	BendPick	CloseDoor	OpenOven	Avg.
Pretrained base (direct OPD)	B	1/10	8/10	0/10	30.0
Standard merged model	S	0/10	6/10	1/10	23.3
G-isometry merged model	G	8/10	10/10	8/10	86.7

D Response Geometry and Behavioral Diagnostics

Comparison. We diagnose two-expert merging of Ψ_0 BendPick (B) and CloseDoor (C) experts under standard and G-isometry-regularized fine-tuning. Under both fine-tuning conditions, the two experts are merged using the same RegMean++ configuration into one two-task policy. We measure expert-update response geometry and merged-policy behavior separately. This diagnostic uses two experts, while the main SIMPLE comparison integrates three tasks.

Fixed inputs. For each task, we select 20 held-out reference trajectories and uniformly sample 16 valid replanning observations per trajectory. At the midpoint of each of eight equal-width noise bins, we use three fixed noise samples. Conditions share images, instructions, proprioception, normalization, and real-time chunking (RTC) prefixes and delays. We collect layer inputs x from the common pretrained base at the constrained linear layers and estimate the full input second moment $G = \mathbb{E}[xx^\top]$ for each task, layer, and bin. Invalid tokens, padded frames, and committed action prefixes are masked. The layer set is fixed in advance.

Response geometry. At each layer and bin, define the full response matrix $M_q^{\text{full}} = \Delta W_q G_q^{1/2}$. The training regularizer uses retained activation directions. Each model uses its own spectrum for C_5. For cosine evaluation in both conditions, fix the five leading right singular vectors v_j of the standard BendPick response matrix. Under the common BendPick reference, let $r_{q,j} = \Delta W_q G_B^{1/2} v_j$ and compute

$$C_5(q) = \frac{\sum_{j=1}^{5} \sigma_j^2(M_q^{\text{full}})}{\|M_q^{\text{full}}\|_F^2}, \qquad c_j = \frac{\langle r_{B,j}, r_{C,j} \rangle}{\|r_{B,j}\|_2 \|r_{C,j}\|_2}. \tag{D.1}$$

Cosines are first reduced by their median over the five directions. Both this statistic and C_5 are averaged equally across bins, then reduced by their median across layers. The aggregated cosine is denoted c_B. Near-zero responses are marked undefined using a prespecified threshold. We retain common valid directions and record their count. The denominator of C_5 is the full response energy.

Behavioral evaluation. The additional diagnostic uses 20 paired scenarios per task, restoring the complete initial environment state and fixing policy noise, controller, sampler, RTC, and success criteria. BendPick translation-command magnitude V_{xy} is the episode time average of $\sqrt{v_x^2 + v_y^2}$ after denormalization, averaged equally across episodes. It includes executed commands, excludes warm-up and unexecuted trailing chunks, and omits turning flags and target heading. Success counts on both tasks help identify overall action shrinkage or early failure. Table 9 reports the previously recorded results, whose success counts use the original ten-trial evaluations.

Table 9: Previously recorded response geometry and two-expert RegMean++ merging results. Success counts in (b) are from the original ten-trial evaluations.

(a) Expert-update responses

Expert fine-tuning	BendPick C_5 (%)	CloseDoor C_5 (%)	Cosine c_B
Standard FT	78.1	93.3	−0.73
G-isometry FT	17.6	14.3	−0.31

(b) Single-task reference and two-expert merging

Policy	BendPick V_{xy}	BendPick	CloseDoor
Standard FT BendPick expert	0.0013	9/10	0/10
Standard FT + RegMean++	0.45	0/10	5/10
G-isometry FT + RegMean++	0.034	3/10	7/10

Layer- and bin-level statistics and configurations are retained in the original results. This diagnostic examines correspondence between response structure and behavioral changes. It does not by itself establish that spectral concentration causes task failure.

E THEORY

This appendix develops four results for GeoMerge-OPD. They concern interference in high-energy response directions during parameter merging, the responses controlled by G-isometry, transient task degradation during joint training, and the link between two-level OPD error and task performance. The response analysis uses a linear layer with fixed activations. The behavioral analysis uses the finite-step action sampler in the main text. Related background includes functional interpretations of task arithmetic (Ortiz-Jimenez et al., 2023) and multitask gradient interference (Yu et al., 2020).

Let R be the number of task groups, indexed by $p, q \in \{1, \dots, R\}$. A group is one SIMPLE task or one LIBERO suite. The matrix norms $\|\cdot\|_F$ and $\|\cdot\|_{\text{op}}$ denote the Frobenius norm and the largest singular value, respectively. Vector norms without subscripts are Euclidean. $\langle A, B\rangle_F = \text{tr}(A^\top B)$ denotes the inner product of matrices with the same shape, tr denotes the trace, and I is the identity matrix of the dimension determined by context.

E.1 TASK-CONDITIONED RESPONSES AND MERGING DEVIATIONS

Fix a layer and a flow time bin. Let $h \in \mathbb{R}^{d_{\text{in}}}$ be the layer input, with input and output dimensions d_{in} and d_{out}, respectively. The update of expert weights W_q relative to the shared base W_0 is $D_q = W_q - W_0$. The reference input second-moment matrix for task q is $G_q = \mathbb{E}_q[hh^\top]$, where $\mathbb{E}_q$ denotes expectation over that task's reference input distribution. Retaining r_q directions gives a factor $S_q \in \mathbb{R}^{d_{\text{in}} \times r_q}$ corresponding to $\widehat{G}_q = S_q S_q^\top$. All comparisons across experts use the same task-specific reference factor:

$$B_{pq} = D_p S_q, \qquad M_q = B_{qq}, \qquad \mathcal{E}_q = \|M_q\|_F^2. \tag{E.1}$$

Here, B_{pq} is the response of expert p's update on the reference inputs of task q. The squared singular values of M_q give the response energies in the retained space. This relation is exact for fixed layer inputs. It does not account for nonlinearities between layers or parameter-dependent inputs in the full network.

First consider additive merging $D_{\text{m}} = \sum_p a_p D_p$, where $a_p \geq 0$ is the merging coefficient of expert p. On task q, the retained response error relative to its expert is

$$F_q = \|(D_{\text{m}} - D_q) S_q\|_F^2. \tag{E.2}$$

Let P_q be an orthogonal projection in the output space, for example onto M_q's leading k left singular directions, where k is the number of selected directions. Define

$$U_q = P_q M_q, \quad V_q = \sum_{p \neq q} a_p P_q B_{pq}, \quad \rho_q = \frac{\|U_q\|_F^2}{\mathcal{E}_q}, \quad b_q = \frac{\|V_q\|_F}{\|U_q\|_F}, \quad c_q = \frac{\langle U_q, V_q\rangle_F}{\|U_q\|_F \|V_q\|_F}. \tag{E.3}$$

The matrix U_q is the task's own response in the selected directions, and V_q is the merged response of other experts in those same directions. The quantity ρ_q is the energy fraction carried by these directions, b_q is the ratio of the other experts' response magnitude to the task's own response magnitude, and c_q is their directional cosine. Assume $\mathcal{E}_q > 0$ and $U_q \neq 0$. If $V_q = 0$, set $b_q = 0, c_q = 0$. We use a matrix Frobenius cosine. Medians of layerwise or directionwise cosines differ from this aggregate quantity.

Proposition E.1 Merging deviation under opposing responses

Under opposing interference, concentrating more response energy in the affected dominant directions increases the merging deviation in those directions. Specifically, fix $\mathcal{E}_q, a_q, b_q, c_q$. If $0 \leq a_q \leq 1$, $b_q > 0$, and $c_q \leq 0$, then

$$\begin{aligned} \|P_q(D_{\text{m}} - D_q)S_q\|_F^2 &= \rho_q \mathcal{E}_q \left[(1 - a_q)^2 + b_q^2 - 2(1 - a_q) b_q c_q\right] \\ &\leq F_q, \end{aligned} \tag{E.4}$$

where the right-hand side of the equality increases strictly with the energy fraction ρ_q.

Equation (E.4) separates the deviation in the dominant directions into three parts: the change in the task's own response caused by its merging weight, the response introduced by other experts, and

their interaction. When $c_q < 0$ and $a_q < 1$, the last term is positive, so opposing responses further increase the deviation. The factor $\rho_q \mathcal{E}_q$ determines how much task response these affected directions carry.

Proof. The projected error is $(a_q - 1)U_q + V_q$. Expanding its squared norm gives

$$(1 - a_q)^2 \|U_q\|_F^2 + \|V_q\|_F^2 - 2(1 - a_q)\langle U_q, V_q \rangle_F.$$

Substituting Eq. (E.3) yields the equality. Since P_q is an orthogonal projection, the full error additionally contains the nonnegative squared norm in the orthogonal complement, yielding the inequality. The final claim follows from the signs of the bracketed terms. □

This result describes the combined effect of concentration and opposing responses. Energy may concentrate because a skill requires only a few control directions. Concentration alone is not harmful. For BendPick, if responses from other experts cancel necessary adjustments in its dominant response space or shift the output toward the control requirements of other tasks, the resulting deviation lies in directions on which this task relies heavily. Internal response directions do not directly correspond to translation command directions. Subsequent network layers and environment dynamics also mediate the relationship between layer response deviations and closed-loop task failure.

Equation (E.4) applies to additive merging. For merging rules with parameter masks, matrix weighting, or other nonadditive operations, Eq. (E.2) should be evaluated using the actual D_{m}. If $D_{\mathrm{m}} = \sum_p a_p D_p + E_{\mathrm{m}}$ and Q_q denotes the equality value in Eq. (E.4), the triangle inequality still gives

$$F_q \geq \left[\max\{0, \sqrt{Q_q} - \|P_q E_{\mathrm{m}} S_q\|_F \} \right]^2.$$

Additional merging operations can therefore also correct the original additive deviation.

E.2 Gain and Coverage Controlled by G-isometry

The regularizer in the main text acts on each expert's own $M_p = D_p S_p$:

$$R_p = \frac{1}{r_p} \|M_p^\top M_p - I_{r_p}\|_F^2, \qquad \delta_p = \sqrt{r_p R_p}. \tag{E.5}$$

It bounds response gain weighted by task activations. We next relate this property to responses on other tasks without assuming identical activation subspaces across tasks.

To express activation coverage across tasks, assume each S_p has full column rank and define

$$A_{pq} = S_p^\dagger S_q, \qquad T_{pq} = (I - S_p S_p^\dagger) S_q, \qquad u_{pq} = \sqrt{1 + \delta_p} \|A_{pq}\|_F + \|D_p T_{pq}\|_F. \tag{E.6}$$

Here, $\dagger$ denotes the Moore–Penrose pseudoinverse. The matrix A_{pq} represents the part of task q covered by expert p's retained directions, T_{pq} represents the uncovered part, and u_{pq} bounds the response magnitude contributed by both parts.

Proposition E.2 Response gain under cross-task coverage

G-isometry limits response amplification in covered directions and thereby bounds the merging deviation. Under the preceding setup,

$$\|D_p S_q\|_F \leq u_{pq}, \qquad \sqrt{F_q} \leq |a_q - 1| \|M_q\|_F + \sum_{p \neq q} a_p u_{pq}. \tag{E.7}$$

The first inequality in Eq. (E.7) bounds expert p's response magnitude on task q. The second sums these magnitudes to bound the merged model's deviation from the task's own expert. The regularizer controls the gain of the covered part through $\sqrt{1 + \delta_p}$, while the uncovered response $\|D_p T_{pq}\|_F$ remains an explicit term.

Proof. The operator norm of a symmetric matrix is bounded by its Frobenius norm, so $\|M_p^\top M_p - I\|_{\mathrm{op}} \leq \delta_p$. Let the eigenvalues of $M_p^\top M_p$ be λ_j. Then $|\lambda_j - 1| \leq \delta_p$, giving

$$\|M_p\|_{\mathrm{op}} \leq \sqrt{1 + \delta_p}. \tag{E.8}$$

If $\delta_p < 1$, each retained singular value $\sigma_j(M_p)$ also satisfies $\sqrt{1-\delta_p} \le \sigma_j(M_p) \le \sqrt{1+\delta_p}$. Also, $S_q = S_p A_{pq} + T_{pq}$, hence

$$D_p S_q = M_p A_{pq} + D_p T_{pq}.$$

The inequality $\|MA\|_F \le \|M\|_{\mathrm{op}} \|A\|_F$ and the triangle inequality give the first response bound. Applying the triangle inequality again to $(D_{\mathrm{m}} - D_q)S_q = (a_q - 1)M_q + \sum_{p \ne q} a_p D_p S_q$ gives the second bound. □

Robot tasks share vision, proprioception, and action generation modules, making shared activation directions plausible. Different instructions, objects, and manipulation stages can also activate directions outside the retained space. Accordingly, Eq. (E.7) retains $\|D_p T_{pq}\|_F$ rather than assuming unconditional coverage across tasks. Reducing R_p tightens the gain factor for the covered component. Across training runs, S_p, relative directions, magnitudes, and uncovered components may also change.

The spectral meaning of the regularizer can also be expressed by an exact decomposition. Let λ_j be the eigenvalues of $M_p^\top M_p$ and let $\bar{\lambda} = r_p^{-1} \sum_j \lambda_j$. Then

$$R_p = (\bar{\lambda} - 1)^2 + \frac{1}{r_p} \sum_j (\lambda_j - \bar{\lambda})^2.$$

When $\delta_p < 1$, the energy fraction in the leading k response directions is at most $\min\{1, k(1+\delta_p)/(r_p(1-\delta_p))\}$, because the numerator is at most $k(1+\delta_p)$ and the denominator is at least $r_p(1-\delta_p)$. This worst-case fraction bound requires $\delta_p < 1$. The upper bound in Eq. (E.8) does not.

Controlled gain and compatibility across tasks are distinct conditions. For example, with $S_1 = S_2 = I$, $D_1 = I$, and $D_2 = -I$, both regularizers are zero, yet equal-weight merging produces a zero update that still deviates from each expert. G-isometry directly controls directional imbalance in update responses. See Liu et al. (2026) for related orthogonal regularization in parameter space. It does not constrain the entire spectrum of the full mapping $W_0 + D_p$, whose Gram matrix also contains base and cross terms. If the full input second-moment matrix is $G_q = \widehat{G}_q + E_q$, with $E_q \succeq 0$, the full layer error additionally contains $\mathrm{tr}((D_{\mathrm{m}} - D_q)E_q(D_{\mathrm{m}} - D_q)^\top)$.

E.3 When Directional Imbalance Affects Joint Training

Joint training determines its next update at a common parameter point. Let θ denote the shared parameters, $L_q(\theta)$ the task loss, and $g_q = \nabla_\theta L_q$ its gradient. Task weights satisfy $\alpha_q > 0$ and $\sum_q \alpha_q = 1$. The average loss and gradient are $\bar{L} = \sum_q \alpha_q L_q$ and $\bar{g} = \sum_q \alpha_q g_q$, respectively. For fixed sampling distributions, if L_q is twice continuously differentiable, using learning rate $\eta > 0$ for the update $\theta^+ = \theta - \eta\bar{g}$ gives

$$L_q(\theta - \eta\bar{g}) - L_q(\theta) = -\eta \left[\alpha_q \|g_q\|^2 + \sum_{p \ne q} \alpha_p g_q^\top g_p \right] + O(\eta^2). \tag{E.9}$$

Inside the brackets, $\alpha_q \|g_q\|^2$ is the task's own improvement term, and the remaining inner products describe the effects of other tasks. First-order degradation occurs only when the weighted negative terms outweigh this improvement. The term $O(\eta^2)$ contains higher-order effects that scale quadratically with the learning rate. This distinction is consistent with analyses of multitask gradient interference (Yu et al., 2020).

Assume that all inputs and targets below have finite second moments. Consider a fixed-feature linear predictor with current weights W. Let h_q, y_q be the input and supervision target for task q, respectively, and let $e_q = Wh_q - y_q$ be the prediction residual. The squared-loss gradients satisfy

$$g_q = \mathbb{E}_q[e_q h_q^\top], \qquad \langle g_q, g_p \rangle_F = \mathbb{E}_{q,p}[(e_q^\top e_p)(h_q^\top h_p)]. \tag{E.10}$$

The two tasks are sampled independently on the right-hand side. This identity follows by expanding the Frobenius inner product of the two expectations. For a shared linear layer in a deep network, e_q can be replaced by the backpropagated derivative at that layer's output. Hidden response directions therefore affect task losses through the subsequent network.

In robot tasks, the following conditions can jointly produce the negative effect in Eq. (E.9). First, two tasks use shared visual, proprioceptive, or locomotion features, and task conditioning has not yet fully separated the relevant gradients. Second, the current predictions require different corrections. If the model commands excessive motion during stationary grasping but insufficient motion when approaching a cabinet door, the two tasks require opposite local corrections to the shared output. The comparison concerns corrections relative to current predictions. Zero and nonzero motion targets are not inherently conflicting. Third, a larger residual magnitude, sampling weight, or local sensitivity can make one task's weighted contribution dominant. Finally, directions that distinguish tasks may remain available but be relatively weak, requiring more updates to compensate for a shift along the shared direction. The following realizable model makes these conditions concrete.

Consider two tasks sharing parameters $w \in \mathbb{R}^2$, with inputs $h_1 = (A, 1)^\top$ and $h_2 = (A, -1)^\top$ and targets $y_1 = 1, y_2 = -2$. Here, $A > \sqrt{3}$ represents the strength of the shared direction relative to the direction that distinguishes the tasks. Define $L_q(w) = \frac{1}{2}(w^\top h_q - y_q)^2$ and apply gradient descent to the equally weighted average loss from $w_0 = 0$. Let w_t denote the parameters after t updates. Both tasks admit the common zero-loss solution $w^\star = (-1/(2A), 3/2)^\top$.

Proposition E.3 Transient degradation of jointly realizable tasks

Even when two tasks can be learned simultaneously, joint training can initially worsen one task while the average loss keeps decreasing. In the preceding model, take $\eta = A^{-2}$. For every integer $t \geq 1$,

$$L_1(w_t) = L_2(w_t) = \frac{9}{8}(1 - A^{-2})^{2t}. \tag{E.11}$$

After the first step, task one's loss exceeds its initial value $1/2$. It then recovers gradually and converges to zero together with task two.

The factor $(1 - A^{-2})^{2t}$ in Eq. (E.11) describes the gradual decrease in error during training. A stronger shared direction means a larger A, bringing this decay factor closer to 1. The direction that distinguishes the tasks then takes longer to compensate for the initial shift. Task one remains worse than at initialization over the interval

$$1 \leq t < \frac{\log(3/2)}{-\log(1 - A^{-2})}. \tag{E.12}$$

Let the mixed input second-moment matrix be $\bar{G} = \frac{1}{2}(h_1 h_1^\top + h_2 h_2^\top) = \mathrm{diag}(A^2, 1)$. Its condition number $\kappa(\bar{G})$, the ratio of its largest to smallest eigenvalue, equals A^2. As $A \to \infty$, the threshold in Eq. (E.12) is approximately $\log(3/2)\kappa(\bar{G})$. Thus, greater directional imbalance can prolong this transient degradation. All occurrences of log denote natural logarithms.

Proof. Substituting $w^\star$ directly gives the targets 1 and -2. At initialization,

$$g_1 = (-A, -1)^\top, \quad g_2 = (2A, -2)^\top, \quad \bar{g} = (A/2, -3/2)^\top, \quad g_1^\top \bar{g} = (3 - A^2)/2.$$

For any w, the squared loss gives the exact change

$$L_q(w - \eta\bar{g}) - L_q(w) = -\eta g_q^\top \bar{g} + \frac{\eta^2}{2}(h_q^\top \bar{g})^2. \tag{E.13}$$

At initialization, when $A^2 > 3$, the first-order term for task one is strictly positive and the second-order term is nonnegative. The change in the average loss is $-\eta\|\bar{g}\|^2 + \frac{\eta^2}{2}\bar{g}^\top \bar{G}\bar{g}$, which is at most $-\eta(1 - \eta A^2/2)\|\bar{g}\|^2$. With $\eta = A^{-2}$, this upper bound becomes $-\eta\|\bar{g}\|^2/2$, which is strictly negative whenever the gradient is nonzero.

Let $e_t = w_t - w^\star$. Joint realizability gives $e_{t+1} = (I - \eta\bar{G})e_t$, and hence

$$e_t = \left(\frac{(1 - \eta A^2)^t}{2A}, -\frac{3}{2}(1 - \eta)^t\right)^\top.$$

With $\eta = A^{-2}$, the first component vanishes for $t \geq 1$, and the two prediction residuals are $-\frac{3}{2}(1 - A^{-2})^t$ and $+\frac{3}{2}(1 - A^{-2})^t$, respectively. Squaring yields Eq. (E.11). Rearranging $L_1(w_t) > L_1(w_0) = 1/2$ as $(1 - A^{-2})^t > 2/3$ gives Eq. (E.12). Finally, use $-\log(1 - A^{-2}) \sim A^{-2}$ and $\kappa(\bar{G}) = A^2$. □

The first coordinate represents a direction shared by both tasks, and the second retains information that distinguishes them. The shared direction initially dominates the correction, and the distinguishing direction subsequently restores task one. Degradation can therefore persist over a finite training budget, although this model eventually learns both tasks simultaneously.

A behavior that keeps the robot in place and has already been learned can also be perturbed without opposing nonzero residuals. Keeping h_1, h_2, take $y_1 = 0, y_2 = b > 0$, $w_0 = 0$, and task-one weight $\alpha \in (0, 1)$. Then $g_1 = 0$, and the joint step is $w^+ = \eta(1-\alpha) b\, h_2$, giving

$$L_1(w^+) = \frac{\eta^2}{2}(1-\alpha)^2 b^2 (A^2 - 1)^2. \tag{E.14}$$

When $A^2 \neq 1$, task one deteriorates from its original zero loss. The other task's update changes a shared output that was already correct, corresponding to the second-order effect in Eq. (E.13). A common solution $w^\star = (b/(2A), -b/2)^\top$ still exists.

These results use a local fixed-feature model and standard gradient descent. The fixed features are an analytical approximation at a given parameter point; upstream vision-language modules and the action expert can change their internal activations during fine-tuning. The example therefore establishes a possible optimization mechanism. For optimizers with momentum, preconditioning, or weight decay, let $\Delta\theta_{\text{actual}}$ denote the actual parameter change. The first-order loss change is $g_q^\top \Delta\theta_{\text{actual}}$. Unpreconditioned gradient angles cannot be substituted directly into the standard gradient descent result.

Expert responses cannot substitute for current gradients. If an unregularized expert W_q is a stationary point of the fixed-feature squared loss, then $\mathbb{E}_q[(W_q h - y_q)h^\top] = 0$, so at the shared base,

$$g_q(W_0) = -D_q G_q. \tag{E.15}$$

For a common input second-moment matrix G, further take the full factor $S_q = S = G^{1/2}$, so that $M_q = D_q S$. Then

$$\langle g_p(W_0), g_q(W_0)\rangle_F = \operatorname{tr}(M_p G M_q^\top), \qquad \langle M_p, M_q\rangle_F = \operatorname{tr}(M_p M_q^\top). \tag{E.16}$$

Equation (E.15) follows by substituting $W_0 = W_q - D_q$ into the gradient. Substituting $D_p G = M_p G^{1/2}$ into the inner product then gives Eq. (E.16).

Take $M_1 = \operatorname{diag}(10, 1)$ and $M_2 = \operatorname{diag}(-20, 1)$, and let each task's target be $W_q h$, with $W_0 = 0$ and $W_q = M_q G^{-1/2}$. The leading response direction accounts for $100/101$ and $400/401$ of the two experts' response energies, respectively, and the response inner product is always -199. Yet at initialization of equally weighted joint training,

G	$\langle g_1, g_2\rangle_F$	$\langle g_1, \bar{g}\rangle_F$
I	-199	-49
$\operatorname{diag}(10^{-4}, 1)$	0.98	0.995

Thus, identical concentrated and opposing expert responses can yield opposite first-order training conclusions. This counterexample establishes that response statistics alone are insufficient. Proposition E.3 separately establishes transient degradation even for jointly realizable tasks. For regularized experts, different activation distributions, or nonlinear fine-tuning trajectories, the stationarity assumption in Eq. (E.15) also cannot be directly transferred.

E.4 Two-Level On-Policy Distillation and Per-Task Returns

Merging error characterizes the deviation of the initialization from the experts. OPD subsequently calibrates the student's action generation behavior directly. This section analyzes finite-step sampling using the discrete Gaussian transitions from the main text, retaining the time-step weights and effective action dimension of the actual loss.

Fix task q. Let T_q denote its teacher and θ the student parameters, with velocity fields v_{T_q} and v_θ, respectively. Let j index environment decisions, K be the number of sampling steps in one action generation process, and $\ell = 0, \ldots, K-1$ index flow steps. Let h_j be the full environment history or a sufficient Markov state, and let o_j be the model conditioning determined by it. Full histories allow the environment to include contact states, controller memory, and fixed action prefixes. The following conditions specify the stochastic sampling policies considered in this section:

1. The student and teacher execute the same task under the same initial state distribution and environment transitions, with episode length $1 \le N \le H_q$ and the same termination rule. Finite timeouts and a common task evaluation protocol provide this setting. The task return $\mathcal{R}_q$ is the same function of the complete trajectory for both policies and takes values in $[0, 1]$, including success indicators.
2. Both policies use the same K-step noise schedule and initial noise distribution. The flow time t_ℓ decreases along $1 = t_0 > \cdots > t_K = 0$, $h_\ell = t_\ell - t_{\ell+1} > 0$ is the step size, and $g_\ell > 0$ is the noise coefficient at that step. Only the velocity field is replaced at each step, as in the equal-covariance Gaussian kernels of the main text.
3. The d_q effective action coordinates used in the loss constitute the complete stochastic sampling state. Other coordinates are fixed by functions shared by both policies and can be included in the conditioning. If stochastic coordinates omitted from the loss affect subsequent effective coordinates, they must also be included in the state and KL divergence for this analysis. Decoding, denormalization, and action execution mappings are identical for both policies.

These conditions specify the policies and interfaces being compared. They do not assume that neural network training necessarily attains low error.

Let d_q denote the total number of scalar coordinates in the effective stochastic state, counting all effective coordinates of a flattened action chunk. Let $z \in \mathbb{R}^{d_q}$ denote an intermediate action state and o fixed observation conditioning. We use the score parameterization constructed from the velocity field in the main text, $s_\theta(z, t, o) = -[z + (1 - t)v_\theta(z, t, o)]/t$, which gives the reverse-time SDE drift $v_\theta - \frac{1}{2}g_\ell^2 s_\theta$. Taking an Euler–Maruyama step of size h_ℓ in decreasing time gives the conditional mean $z - h_\ell[v_\theta - \frac{1}{2}g_\ell^2 s_\theta]$. Define

$$c_\ell = 1 - \frac{h_\ell g_\ell^2}{2t_\ell}, \qquad d_\ell = h_\ell \left(1 + \frac{g_\ell^2(1 - t_\ell)}{2t_\ell}\right).$$

Fix the sampling step ℓ below and omit its index from the one-step mean μ_θ and transition kernel K_θ. Within the effective coordinates, the one-step conditional mean is $\mu_\theta(z, o) = c_\ell z - d_\ell v_\theta(z, t_\ell, o)$ and the covariance is $g_\ell^2 h_\ell I_{d_q}$. The student kernel $K_\theta(\cdot \mid z, o)$ is the Gaussian distribution with this mean and covariance. The teacher kernel K_{T_q} replaces only v_θ by v_{T_q}. KL denotes Kullback–Leibler divergence with natural logarithms, which measures the difference between two probability distributions. Taking the expectation of the log ratio of the two Gaussian densities under the student kernel eliminates the centered linear term, giving

$$\mathrm{KL}(K_\theta(\cdot \mid z, o) \| K_{T_q}(\cdot \mid z, o)) = \frac{\|\mu_\theta - \mu_{T_q}\|^2}{2g_\ell^2 h_\ell} = w_\ell \|v_\theta - v_{T_q}\|^2, \qquad w_\ell = \frac{d_\ell^2}{2g_\ell^2 h_\ell}. \tag{E.17}$$

This w_ℓ matches the weight in the main text (Fang et al., 2026).

Let z_ℓ^θ be generated by the action sampler of the current fixed student, with complete path $\mathbf{z}^\theta = (z_0^\theta, \ldots, z_K^\theta)$ and corresponding teacher path $\mathbf{z}^{T_q}$. Define the average distillation error when this path is resampled given a history as

$$\ell_q(\mathsf{h}) = \mathbb{E}_{\mathbf{z}^\theta \mid o(\mathsf{h})}\left[\frac{1}{K}\sum_{\ell=0}^{K-1} \frac{w_\ell}{d_q}\|v_\theta(z_\ell^\theta, t_\ell, o) - v_{T_q}(z_\ell^\theta, t_\ell, o)\|^2\right]. \tag{E.18}$$

Projection onto the effective coordinates is absorbed into the velocity fields here. Further define

$$\varepsilon_q^{\text{self}} = \mathbb{E}_{\text{student environment episodes}}\left[\frac{1}{N}\sum_{j=0}^{N-1} \ell_q(\mathsf{h}_j)\right]. \tag{E.19}$$

The quantity $\ell_q(\mathsf{h})$ measures action generation error given a history, and $\varepsilon_q^{\text{self}}$ averages this error over the environment histories that the student actually visits. Selecting observations from student environment episodes and then resampling student flow paths at those observations matches the two-level sampling objective in the main text. Define $J_q(\theta) = \mathbb{E}_\theta[\mathcal{R}_q]$ and $J_q(T_q) = \mathbb{E}_{T_q}[\mathcal{R}_q]$ as the expected task returns of the student and teacher, respectively.

Theorem E.4 Per-task return bound under two-level OPD

When the student's distillation error on its own environment states and action paths is sufficiently small, its task performance is close to the teacher's. Under the preceding stochastic sampling conditions, if $\varepsilon_q^{\text{self}} < \infty$, then

$$|J_q(\theta) - J_q(T_q)| \le \min\left\{1, \sqrt{\frac{H_q K d_q}{2}\varepsilon_q^{\text{self}}}\right\}. \tag{E.20}$$

The left-hand side of Eq. (E.20) is the performance gap between student and teacher. The right-hand side bounds it at the scale of the square root of distillation error. A smaller $\varepsilon_q^{\text{self}}$ gives a smaller guaranteed gap. The quantity H_q is the episode horizon, while K and d_q correspond to the averages over steps and coordinates in the loss. When $\mathcal{R}_q$ is the success indicator, the left-hand side is the success-rate gap.

For the average multitask error $\bar{\varepsilon} = \sum_q \alpha_q \varepsilon_q^{\text{self}}$, we also obtain

$$J_q(\theta) \ge J_q(T_q) - \min\left\{1, \sqrt{\frac{H_q K d_q}{2\alpha_q}\bar{\varepsilon}}\right\}. \tag{E.21}$$

Here, $\alpha_q > 0$ and $\sum_q \alpha_q = 1$ are the task weights. For the same average distillation error, the guarantee is therefore looser for tasks with smaller weights.

Proof. First fix a history h. Let $\mathbb{P}_\theta(\mathbf{z} \mid \mathsf{h})$ and $\mathbb{P}_{T_q}(\mathbf{z} \mid \mathsf{h})$ denote the conditional distributions of the student's and teacher's complete action paths, respectively. The student and teacher action paths share the same initial noise distribution, so the log ratio of their path densities is the sum of the log ratios of the conditional transition kernels. Taking expectations under the student path distribution, the KL chain rule and Eq. (E.17) give

$$\mathrm{KL}(\mathbb{P}_\theta(\mathbf{z} \mid \mathsf{h}) \| \mathbb{P}_{T_q}(\mathbf{z} \mid \mathsf{h})) = \sum_{\ell=0}^{K-1} \mathbb{E}_{z_\ell^\theta \mid \mathsf{h}}[w_\ell \| v_\theta - v_{T_q}\|^2] = K d_q \ell_q(\mathsf{h}). \tag{E.22}$$

In each term, the teacher is evaluated at the student's intermediate state.

Let $\mathbb{P}_\theta^{\text{ext}}$ and $\mathbb{P}_{T_q}^{\text{ext}}$ denote the extended episode distributions containing the environment history and every complete action path. After termination, both processes can be padded with a common absorbing process to H_q steps. The shared initial state distribution, environment transitions, and execution mappings contribute zero conditional KL divergence. Applying the chain rule again gives

$$\begin{aligned} \mathrm{KL}(\mathbb{P}_\theta^{\text{ext}} \| \mathbb{P}_{T_q}^{\text{ext}}) &= K d_q \, \mathbb{E}_\theta\left[\sum_{j=0}^{N-1} \ell_q(\mathsf{h}_j)\right] \\ &\le H_q K d_q \, \mathbb{E}_\theta\left[\frac{1}{N}\sum_{j=0}^{N-1} \ell_q(\mathsf{h}_j)\right] \\ &= H_q K d_q \varepsilon_q^{\text{self}}. \end{aligned} \tag{E.23}$$

The inequality uses $N \le H_q$ and $\ell_q \ge 0$ in each episode. It does not require the random episode length to be independent of the error.

Use the total variation convention $\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)|$, where P, Q are two probability distributions and the supremum ranges over all measurable events A. For any return function with values in $[0, 1]$, its expectation difference is bounded by total variation. This follows by taking expectations under both distributions in $\mathcal{R}_q = \int_0^1 \mathbf{1}_{\{\mathcal{R}_q > u\}} \, du$. Pinsker's inequality gives $\mathrm{TV}(P, Q) \le \sqrt{\mathrm{KL}(P\|Q)/2}$. Substituting Eq. (E.23) and using the fact that the return gap is at most 1 yields Eq. (E.20). Finally, $\alpha_q \varepsilon_q^{\text{self}} \le \bar{\varepsilon}$ gives Eq. (E.21). □

Equation (E.22) explains why action-level supervision is evaluated along student paths, and Eq. (E.23) explains why environment-level sampling uses states visited by the student. The result

directly covers discontinuous success indicators because it compares complete stochastic trajectory distributions and does not require the success function to be Lipschitz in the actions. The factors K and d_q arise because the main-text loss averages over steps and coordinates. They do not imply that increasing the number of sampling steps or the action dimension necessarily reduces performance. Whether the bound is numerically informative still depends on the residual magnitude.

Policy and optimization scope. The theorem compares teachers and students satisfying the same stochastic sampling protocol above. Equation (E.20) does not directly carry over if deployment uses a deterministic ODE, a different noise schedule, or different action processing. In particular, when $g_\ell = 0$, the KL formula for nondegenerate Gaussians with equal covariance does not apply. The theorem also does not identify the empirical training loss with the current $\varepsilon_q^{\text{self}}$. States collected by previous parameters, finite data, and the updated policy may correspond to different distributions. It compares policies given the current error and provides no convergence guarantee for nonconvex training.

Applying stop-gradient to the teacher does not change the numerical loss in Eq. (E.17). However, when the student path z^θ depends on the parameters, applying stop-gradient to the entire teacher output removes the teacher's derivative with respect to its input. The probabilistic bound therefore neither requires nor implies that the update used is the full gradient of the numerical path KL divergence. Task conditioning helps the student express different skills for different observations and instructions. Optimization of shared parameters can still exhibit the gradient interference analyzed in the preceding section.