SHARED SPECTRAL COORDINATES FOR COMPOSABLE EMBODIED SKILLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models are increasingly expected to accumulate skills across embodiments with different action spaces, without retraining for each addition. We represent each skill in the spectral coordinates of a frozen VLA backbone, so that independently learned skills can be composed without retraining the backbone, pooling their training data, or aligning them after the fact. For selected backbone projections, we freeze the pretrained singular basis and represent each skill as compact gain vectors that rescale the singular values, so all skills share one coordinate system from the start. Since alignment alone does not guarantee functional compatibility, we regularize the gains toward the pretrained configuration during learning and, after composition, recalibrate only the lightweight embodiment-specific readouts. Across autonomous driving, robot manipulation, and humanoid control, the method supports two- and three-skill composition within a single backbone using only 0.34M backbone adaptation parameters per skill, versus 1.09M for rank-1 and 8.7M for rank-8 LoRA experts, while retaining competitive offline performance and effective closed-loop control. Staying close to the pretrained configuration improves robustness to interference, and readout recalibration repairs most of the mismatch introduced by composition. Because setting all skill coefficients to zero recovers the pretrained backbone exactly, skills can be activated, combined, or removed at deployment time.

1 INTRODUCTION

Vision-language-action (VLA) models have emerged as a general foundation for embodied control. A critical next step is to accumulate multiple skills within a single pretrained model rather than train separate models for different embodiments. Recent VLA control systems have shown that large-scale pretraining can support diverse embodied decision-making tasks (Kim et al., 2024; Zitkovich et al., 2023; Shao et al., 2024; Sima et al., 2024; O'Neill et al., 2024). However, VLA skills are often learned independently from heterogeneous datasets and action spaces. Composing such skills within a shared model remains challenging without joint retraining or pooled dataset access.

Model merging offers a natural starting point. Many studies combine independently trained experts through arithmetic operations in parameter space (Ilharco et al., 2023; Yadav et al., 2023; Zhang et al., 2023b). However, later studies show that successful merging depends on how the experts are represented and trained. Low-rank updates can occupy misaligned subspaces and require additional alignment (Stoica et al., 2025), while the fine-tuning process itself can affect subsequent mergeability (Lee et al., 2025; Wei et al., 2026). Thus, independently trained experts are not composable by default. For VLA systems that accumulate skills across heterogeneous embodiments, this motivates a representation where independently acquired skills share one coordinate system from the start.

We therefore represent each skill in the spectral coordinates of a frozen VLA backbone. Using the singular value decomposition (SVD) of the pretrained vision-language backbone, we freeze its singular basis and represent each skill with compact spectral gain vectors that rescale the pretrained singular values. This parameterization builds on singular value fine-tuning (SVF) (Sun et al., 2022) and related compositional spectral experts (Sun et al., 2025). Because the singular basis is shared and fixed, the i-th gain of one skill acts on the same singular component as the i-th gain of any other skill. Skills can therefore be combined, scaled, or switched off through simple arithmetic on their

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.

gain vectors, and no alignment procedure is needed before merging because the experts share a basis by construction.

However, sharing coordinates does not guarantee functional compatibility. We address this at two stages. During independent training, each spectral gain vector is regularized toward $z = \mathbf{1}$, which exactly recovers the pretrained configuration (Li et al., 2018; Wortsman et al., 2022). After composition, the backbone and the composed gains are frozen, and only the lightweight embodiment-specific readouts are recalibrated on a small amount of task-local data, which absorbs the shift in backbone features that composition introduces.

We evaluate the framework on the CARLA driving simulator (Dosovitskiy et al., 2017; Yu et al., 2025), the LIBERO manipulation benchmark (Liu et al., 2023), and the Unitree G1 humanoid platform (Unitree Robotics, 2024). Each spectral skill uses 0.34M backbone adaptation parameters. The smallest nonzero LoRA rank requires 1.09M on the same projections, and the rank-8 experts in our main comparisons require 8.7M. With this budget, two- and three-skill composition within a single backbone achieves competitive offline performance and effective closed-loop control on all three embodiments. Composability depends on how far each skill drifts from the pretrained configuration, on the composition coefficients, and on readout recalibration. We analyze each factor and show that adding a third skill to an existing pair requires calibrating only its own readout and coefficient. Our main contributions are as follows:

- We represent independently learned VLA skills as gain vectors in the spectral coordinates of a frozen backbone, so that skills with different action spaces can be added, scaled, and removed exactly.
- We show that shared coordinates alone are not sufficient, and that regularizing gains toward the pretrained configuration and recalibrating the embodiment-specific readouts are both needed for functional composition.
- We evaluate two- and three-skill composition on driving, manipulation, and humanoid control, offline and in closed loop, and analyze the coefficient geometry, incremental skill addition, and failure cases that shape composability.

2 Related Work

Spectral and parameter-efficient adaptation. Parameter-efficient fine-tuning (PEFT) adapts pretrained models by updating a small number of task-specific parameters, exemplified by LoRA (Hu et al., 2022). Spectral variants instead adapt the singular structure of the weights. AdaLoRA and PiSSA keep the update in SVD form (Zhang et al., 2023b; Meng et al., 2024), and SVF tunes only the singular values (Sun et al., 2022). Transformer^2 learns one SVF expert per task and mixes them at inference with dispatch or CEM-searched coefficients (Sun et al., 2025), and BOLT builds a shared spectral basis from source task vectors (Park et al., 2025). Both adapt one model to language tasks with a single output space. We use the pretrained singular basis as a fixed coordinate system for control skills learned independently on embodiments with different action spaces.

Model merging and expert composition. Task Arithmetic (TA) combines independently trained experts through arithmetic on task vectors (Ilharco et al., 2023), and TIES-Merging reduces interference by pruning redundant updates and resolving sign conflicts (Yadav et al., 2023). The same idea extends to PEFT modules, with coefficients learned or searched (Zhang et al., 2023a; Huang et al., 2024; Yang et al., 2024b). Several studies exploit spectral structure. Task Singular Vectors analyzes interference along layer-wise singular directions (Gargiulo et al., 2025), AdaRank selects singular components useful for merging (Lee et al., 2026), and KnOTS projects LoRA updates into a shared SVD space before merging (Stoica et al., 2025). SAFT-Merge and OptMerge constrain training for mergeability (Lee et al., 2025; Wei et al., 2026), and Representation Surgery corrects the merged representation afterward (Yang et al., 2024a). All of these merge separately parameterized experts. We place every skill in the same spectral coordinates from the beginning, so composition is arithmetic on their gain vectors.

VLA models. VLA models extend large-scale multimodal pretraining to embodied control, in manipulation with RT-2 and OpenVLA (Zitkovich et al., 2023; Kim et al., 2024) and in driving with LMDrive and DriveLM (Shao et al., 2024; Sima et al., 2024). Cross-embodiment policies such as

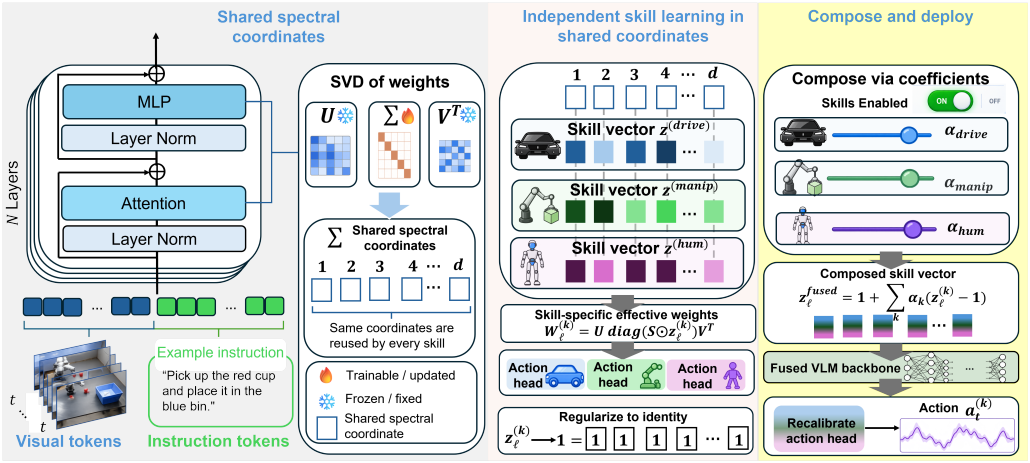


Figure 1: Overview. Left: the SVD of selected projections defines shared spectral coordinates. Middle: each skill learns gain vectors $z^{(k)}$ in these coordinates. Right: skills are composed via α_k and deployed after recalibrating the action heads.

CrossFormer, HPT and π_0 cover several robots at once by training jointly on mixed-embodiment data (Doshi et al., 2024; Wang et al., 2024b; Black et al., 2024; Vuong et al., 2023). HPT in particular attaches embodiment-specific stems and heads to a shared trunk, a structure our readouts share, but the trunk itself is learned jointly. We instead acquire skills independently on each embodiment and compose them afterward within one backbone.

3 Method

3.1 Problem Setting and Overview

We consider K VLA skills built on a common pretrained backbone. Skill k is learned from dataset $\mathcal{D}_k$ and predicts actions in its own continuous action space through an embodiment-specific action head h_k, which we also refer to as the readout. Our goal is to acquire these skills independently and later compose them without retraining the backbone or accessing their training datasets together.

Figure 1 summarizes the framework. We first build a shared spectral coordinate system from selected linear projections of the pretrained backbone. Each skill is then learned independently as a set of spectral gain vectors $z^{(k)}$ in these coordinates with its action head. The learned skills are composed by arithmetic on their gain vectors and deployed after a lightweight recalibration of each action head.

3.2 Shared Spectral Skill Representation

For each selected linear projection $W_\ell \in \mathbb{R}^{n_\ell \times m_\ell}$ in the pretrained backbone, we compute its thin singular value decomposition

$$W_\ell = U_\ell \Sigma_\ell V_\ell^\top, \qquad \Sigma_\ell = \mathrm{diag}(s_\ell), \tag{1}$$

where $s_\ell \in \mathbb{R}^{r_\ell}$ contains the pretrained singular values and $r_\ell = \min(n_\ell, m_\ell)$.

Following singular-value adaptation (Sun et al., 2022; 2025), we keep U_ℓ, V_ℓ, and s_ℓ fixed and represent skill k with a multiplicative spectral gain vector $z_\ell^{(k)} \in \mathbb{R}^{r_\ell}$:

$$W_\ell^{(k)} = U_\ell \,\mathrm{diag}\left(s_\ell \odot z_\ell^{(k)}\right) V_\ell^\top. \tag{2}$$

Only $z_\ell^{(k)}$ is learned. All gain vectors are initialized as $z_\ell^{(k)} = \mathbf{1}$, which exactly recovers the pretrained projection.

A complete skill consists of the gain vectors from all adapted projections,

$$z^{(k)} = \{z_\ell^{(k)}\}_{\ell \in \mathcal{P}}. \tag{3}$$

Because every skill uses the same pretrained decomposition, the i-th entry of $z_\ell^{(k)}$ always scales the same singular component of W_ℓ. Each coordinate has a fixed meaning across independently learned skills, which is what allows their gains to be combined entry by entry. We retain this diagonal parameterization because it preserves the pretrained singular directions (Lingam et al., 2024).

3.3 Independent Skill Learning

As shown in the middle panel of Fig. 1, each skill is trained independently in the shared spectral coordinate system. For skill k, we optimize its spectral gain vectors $z^{(k)}$ together with its embodiment-specific action head h_k using only the task-local dataset $\mathcal{D}_k$. We denote the corresponding control objective by $\mathcal{L}_{\text{task},k}$.

Since $z_\ell^{(k)} = \mathbf{1}$ exactly recovers the pretrained projection, the pretrained configuration provides a common reference for all skills. We define the deviation of skill k at projection ℓ as

$$\Delta z_\ell^{(k)} = z_\ell^{(k)} - \mathbf{1}. \tag{4}$$

These deviations are later combined during skill composition. To keep each independently learned skill close to the pretrained configuration, we regularize its spectral gains toward this common reference. Let $D = \sum_{\ell \in \mathcal{P}} r_\ell$ denote the total number of trainable spectral scalars in one skill. The training objective is

$$\mathcal{L}_k = \mathcal{L}_{\text{task},k} + \lambda \frac{1}{D} \sum_{\ell \in \mathcal{P}} \left\| \Delta z_\ell^{(k)} \right\|_2^2. \tag{5}$$

We use $\lambda = 0.01$ for all skills. The second term measures the mean squared deviation of the spectral gains from the pretrained configuration. Because the composition rule operates on $\Delta z_\ell^{(k)}$, this regularization discourages large skill-specific deviations before composition. This motivation is consistent with prior work showing that fine-tuning trajectories and parameter drift can affect subsequent mergeability (Lee et al., 2025; Wei et al., 2026). Section 4.3 evaluates the effect of this regularization on composition robustness.

After training, skill k is represented by its spectral gain vectors $z^{(k)}$ and embodiment-specific action head h_k. Each skill is acquired independently, without cross-task batches or access to data from other embodiments.

3.4 Skill Composition and Post-Composition Recalibration

The right panel of Fig. 1 shows the composition and deployment stage. Since all skills are learned in the same spectral coordinates, their deviations from the pretrained configuration can be combined entry by entry. Given K independently trained skills, we define

$$z_\ell^{\text{fused}} = \mathbf{1} + \sum_{k=1}^{K} \alpha_k \Delta z_\ell^{(k)}, \qquad \alpha_k \geq 0, \quad \ell \in \mathcal{P}, \tag{6}$$

where α_k controls the contribution of skill k and is shared across all adapted projections of that skill. The complete composed configuration is $z^{\text{fused}} = \{z_\ell^{\text{fused}}\}_{\ell \in \mathcal{P}}$.

Equation 6 is the linear composition rule used for task vectors and for PEFT modules (Ilharco et al., 2023; Zhang et al., 2023a; Huang et al., 2024), applied here in the spectral coordinates to skills with distinct action spaces. Setting $\alpha_k = 0$ removes skill k, while $\alpha_k = 1$ retains its full learned deviation. Setting all coefficients to zero exactly recovers the pretrained configuration. The coefficients therefore provide a simple mechanism for selecting and scaling skills at deployment.

For each adapted projection, the composed gains define the effective weight

$$W_\ell^{\text{fused}} = U_\ell \, \text{diag}\left(s_\ell \odot z_\ell^{\text{fused}}\right) V_\ell^\top. \tag{7}$$

The pretrained spectral factors remain fixed, and composition changes only the gain applied to each shared singular component.

Each action head h_k is initially trained with the isolated skill $z^{(k)}$. After composition, the head receives features generated by z^{fused} instead. We therefore recalibrate the head on a small task-local

calibration set $\mathcal{C}_k$ while keeping the composed backbone configuration fixed. Updating only the head follows the linear-probing principle that head-only adaptation does not distort pretrained features (Kumar et al., 2022), and parallels post-merging representation correction (Yang et al., 2024a). The heads serve as lightweight readouts from the shared representation to different action spaces, which lets one composed backbone emit driving waypoints, robot action chunks, and humanoid controls with different dimensionalities and semantics.

Let $f(x; z^{\text{fused}})$ denote the backbone representation under the composed configuration. Starting from the independently trained head h_k, we optimize only the readout

$$h_k^* = \arg\min_h \frac{1}{|\mathcal{C}_k|} \sum_{(x,a)\in\mathcal{C}_k} \ell_k \left(h(f(x; z^{\text{fused}})), a \right), \tag{8}$$

where ℓ_k denotes the embodiment-specific control loss. At deployment, embodiment k produces its action as

$$a_t^{(k)} = h_k^* \left(f(x_t; z^{\text{fused}}) \right). \tag{9}$$

No cross-task batches or joint backbone optimization are required at any point in this stage.

4 Experiments

4.1 Experimental Setup

Backbone and adaptation. We use Qwen3-VL-2B-Instruct as the shared pretrained backbone (Team, 2025). Spectral adaptation is applied to seven linear projections in each of the 28 decoder blocks, giving 196 adapted matrices and 344,064 spectral gains per skill, or approximately 0.34M parameters. Optimization settings and implementation details are in Appendix A.

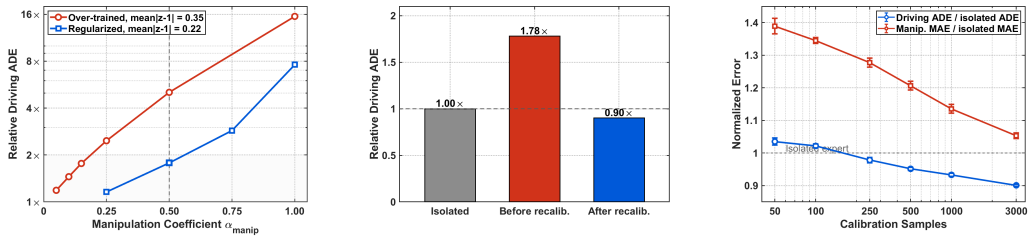
Embodied tasks. For CARLA, we use 52,430 frames and hold out 2% for evaluation. For LIBERO, we use a 5% held-out split. For Unitree G1, we evaluate humanoid block stacking with 16-dimensional control targets. Offline metrics are average displacement error (ADE) for CARLA, action mean absolute error (MAE) for LIBERO, and normalized action mean squared error (nMSE) for G1. All three embodiments are also evaluated in closed loop over 100 episodes per task, reporting success rate, plus closed-loop nMSE for G1.

Baselines and evaluation. We report isolated spectral and LoRA experts and compare composition against TA, TIES, and KnOTS+TIES (Hu et al., 2022; Ilharco et al., 2023; Yadav et al., 2023; Stoica et al., 2025), using rank-8 LoRA experts and a rank-1 control on the same projections. We also report joint training as a reference, and a Spectral-TIES variant of our method that applies TIES trimming and sign election directly to $\Delta z^{(k)}$ (Appendix C.2). All methods use the same held-out evaluation examples and post-composition calibration budget, during which the composed backbone is fixed and only the action heads are updated.

4.2 Skill Composition Across Embodiments

Table 1 summarizes the main results. Two-skill additive composition reaches a driving ADE of 0.291 and a manipulation MAE of 0.119, against 0.323 and 0.113 for the isolated spectral experts. Giving the isolated experts the same readout and recalibration budget leaves them essentially unchanged, at 0.3256 ± 0.0031 and 0.1121 ± 0.0015, so the driving improvement is positive transfer from composition rather than extra head training. Manipulation retains a modest residual cost. Applying Spectral-TIES to the same skills improves both tasks to $0.285/0.110$ at no extra parameter cost. Additive composition remains the default operator in all later analyses, and Spectral-TIES serves as an alternative conflict-aware rule. Joint training reaches $0.326/0.117$ but requires both datasets at once.

Against LoRA merging, the spectral representation trades a small amount of accuracy for a large reduction in parameters. Rank-1 LoRA with TA and the same recalibration reaches $0.291/0.126$, matching our additive result with $3.18\times$ more adaptation parameters per skill. Rank-8 merges are stronger, with TA at $0.278/0.108$ and KnOTS+TIES at $0.273/0.110$, but require 8.7M parameters



(a) (b) (c)

Figure 2: Functional composability. (a) Regularization toward the pretrained configuration improves robustness as α_{manip} increases. (b) Post-composition readout recalibration restores driving performance. (c) Recalibration error as a function of task-local calibration data, with three-seed standard deviations.

per skill, over $25\times$ ours. Spectral-TIES matches KnOTS+TIES on manipulation without the alignment step that KnOTS requires.

The composed skills remain effective in closed loop. CARLA success moves from 97% to 95% after two-skill composition and stays there when the G1 skill is added. LIBERO drops from 84% to 77% and then 75%. G1 reaches 75% success with 0.0054 nMSE, against 89% and 0.0049 for its isolated expert. Manipulation and humanoid control thus pay a larger composition cost than driving, but all three skills remain functional within one composed backbone without retraining the backbone or the spectral vectors.

Composition targets multi-skill coexistence rather than replacing isolated experts, which remain the simpler choice when a single embodiment suffices. Keeping all three gain sets resident costs 4.13 MB of GPU memory, recombining them takes 3.53 ms, and swapping a readout from CPU takes 0.31 ms. Appendix A.6 reports the full measurements.

Table 1: Multi-embodiment skill composition with 100 closed-loop episodes per task.

Offline evaluation			
Method	Driving ADE ↓	Manip. MAE ↓	Params / skill
Spectral expert	0.323	0.113	**0.34M**
Spectral expert + recalib.	0.326	0.112	**0.34M**
LoRA expert ($r = 1$)	0.321	0.115	1.09M
LoRA expert ($r = 8$)	0.310	0.093	8.7M
Joint training	0.326	0.117	–
TA ($r = 1$)	0.291	0.126	1.09M
TA ($r = 8$)	0.278	0.108	8.7M
TIES ($r = 8$)	0.315	0.138	8.7M
KnOTS+TIES ($r = 8$)	0.273	0.110	8.7M
Ours (additive)	0.291	0.119	**0.34M**
Ours + Spectral-TIES	0.285	0.110	**0.34M**

Closed-loop deployment			
Configuration	CARLA	LIBERO	G1 Succ./nMSE
Isolated expert	97%	84%	89% / .0049
Two-skill	95%	77%	–
Three-skill	95%	75%	75% / .0054

4.3 Functional Composability

Shared spectral coordinates align independently learned skills entry by entry, but functional compatibility also depends on how the skills are trained and how the composed representation is read out. We examine the effects of regularization toward the pretrained configuration, post-composition readout recalibration, and the amount of task-local calibration data. We also study whether previously calibrated readouts remain valid when a new skill is introduced.

Figure 2(a) tests whether keeping a skill closer to the pretrained configuration improves robustness during composition. The regularized driving skill has a mean absolute spectral deviation of 0.22, compared with 0.35 for the over-trained skill. As α_{manip} increases, the regularized skill degrades more slowly. At $\alpha_{\text{manip}} = 0.5$, its driving ADE is $1.78\times$ the isolated-expert error, compared with $5.05\times$ for the over-trained skill. At $\alpha_{\text{manip}} = 1.0$, the corresponding values are $7.59\times$ and $15.51\times$. Smaller spectral deviations from the pretrained configuration are therefore associated with greater robustness to composition.

Composition also changes the representation presented to each embodiment-specific action head. Figure 2(b) isolates this effect at $\alpha_{\text{manip}} = 0.5$. Composition increases normalized driving ADE

from 1.00 to 1.78 before recalibration. Updating only the action head reduces it to 0.90, while the backbone and composed spectral configuration remain fixed. As shown in Section 4.2, matched recalibration leaves the isolated experts unchanged, so this recovery and the resulting driving gain are not explained by head training alone. The transfer is asymmetric. Driving improves beyond both the isolated expert and its recalibrated control, whereas manipulation retains a modest residual cost.

Figure 2(c) further shows that recalibration does not require the full task dataset. Performance improves rapidly over the first few hundred task-local examples and changes more gradually as the calibration set grows.

We next test whether introducing a new skill requires recalibrating previously deployed readouts. Reusing the calibrated driving and manipulation heads while updating only the new G1 head gives 0.2927 ADE, 0.1210 MAE, and 0.0068 G1 nMSE. Recalibrating all three heads gives 0.2925, 0.1204, and 0.0068, respectively. The nearly identical results indicate that, in this three-skill setting, adding the G1 skill does not require updating the previously calibrated driving and manipulation readouts.

Table 2: Offline incremental readout recalibration after adding G1.

Configuration	Drive ADE ↓	Manip. MAE ↓	G1 nMSE ↓
No new recalib.	0.2927	0.1210	0.5175
G1 only	0.2927	0.1210	0.0068
All heads	0.2925	0.1204	0.0068
Existing only	0.2925	0.1204	0.5175

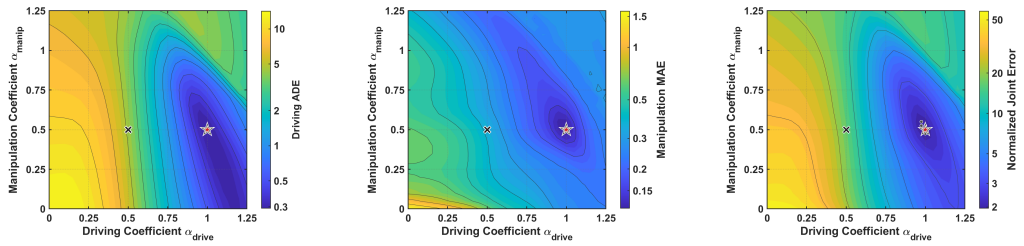
We further test whether readout recalibration can replace the G1 spectral contribution. Keeping the recalibrated G1 head fixed while setting $\alpha_{\mathrm{hum}} = 0$ increases G1 nMSE from 0.0068 to 0.0434, a $6.4\times$ increase. Restoring the selected $\alpha_{\mathrm{hum}} = 0.25$ reduces the error by approximately 84%. Thus, the G1 spectral contribution remains necessary even with the same readout, proprioceptive input, and recalibration budget.

Finally, Table 3 separates the roles of the spectral configuration and the action head. When the spectral skill is removed by setting $z^{(k)} = \mathbf{1}$, recalibrating the head alone gives a driving ADE of 1.238 and a manipulation MAE of 0.184. Conversely, with the action head fixed, changing only the spectral configuration produces large task-dependent changes. Driving ADE decreases from 14.18 under the pretrained configuration to 0.375 with the driving skill and 0.291 with the composed configuration. For manipulation, the corresponding MAEs are 1.612, 0.273, 0.243, and 0.119 under the pretrained, driving, manipulation, and composed configurations. The spectral configuration thus carries the task-specific behavior. The action heads adapt the shared representation to each space and absorb the mismatch that composition introduces between the representation and the actions.

4.4 Composition Coefficient Selection and Sensitivity

Composition coefficients are selected on the validation split with the independently trained action heads held fixed, before any post-composition recalibration. For the two-skill setting, a cross-entropy method (CEM) search (de Boer et al., 2005; Huang et al., 2024) with 77 evaluations per run returns $(1.02, 0.48)$, $(0.97, 0.52)$, and $(1.05, 0.47)$ across three random seeds, each within 0.05 of the selected operating point $(1.0, 0.5)$ in Table 4. The selected region is therefore recovered without readout recalibration and at low cost. Learned coefficient selection by entropy minimization (Yang et al., 2024b) assumes a classification output and does not apply to continuous control. The readouts are recalibrated once at the selected point, and only this point is evaluated on the held-out set for Table 1.

We then measure how sensitive the deployed configuration is to the coefficients. With the readouts recalibrated at $(1.0, 0.5)$ and held fixed, we sweep α_{drive} and α_{manip} over $[0, 1.25]^2$ at a resolution of 0.05, plus 16 local configurations at 0.025 spacing, 692 configurations in total. Figures 3(a) and 3(b) show that the two embodiments prefer different composition strengths, with their low-error regions overlapping near $(1.0, 0.5)$, and Figure 3(c) combines the two errors after normalizing each by its isolated expert. Only 0.87% of the configurations fall within 2% of the operating point, while perturbations of 0.025–0.05 remain within 2%. Uniform weighting $(0.5, 0.5)$ gives 4.680 ADE and 0.370 MAE. Part of the error away from the operating point comes from the readout, which was calibrated there. The landscape therefore measures how much a deployed configuration tolerates coefficient changes.



(a) (b) (c)

Figure 3: Post-recalibration coefficient sensitivity. Readouts are recalibrated once at $(1.0, 0.5)$ and held fixed. (a) Driving ADE, (b) manipulation MAE, and (c) normalized joint error over the composition coefficients. The star marks the selected operating point and the cross marks uniform weighting $(0.5, 0.5)$.

Adding a third skill does not require repeating the two-dimensional search. We retain $(\alpha_{\text{drive}}, \alpha_{\text{manip}}) = (1.0, 0.5)$ and vary only α_{hum}, again with the independently trained G1 readout fixed. A five-point sweep over $\{0, 0.25, 0.5, 0.75, 1.0\}$ followed by full evaluation of the two leading nonzero candidates selects $\alpha_{\text{hum}} = 0.25$. The previously selected coefficients can therefore be retained while only the new skill's coefficient is calibrated.

Table 3: Role of spectral skills and action heads. In the lower block, the heads are recalibrated once at z^{fused} and then held fixed.

Configuration	Drive ADE ↓	Manip. MAE ↓
$z = \mathbf{1}$ + head recalib.	1.238	0.184
Fixed head, $z = \mathbf{1}$	14.18	1.612
Fixed head, $z^{(\text{drive})}$	0.375	0.273
Fixed head, $z^{(\text{manip})}$	6.68	0.243
Fixed head, z^{fused}	0.291	0.119

Finally, we test whether a single scalar coefficient per skill is overly restrictive. We optimize 14 projection-specific coefficients, seven projection types for each of the two skills. A 241-evaluation search improves the normalized joint score by only 1.31% over the scalar solution, so the additional flexibility provides little gain. Further coefficient and composition-operator ablations are provided in Appendices C.3 and C.4.

4.5 Skill Switching and General Capability

Because setting all coefficients to zero restores $z = \mathbf{1}$ exactly, skills can be switched off without touching the backbone. Table 5 evaluates what this reversibility preserves, using GSM8K with 100 problems and MMLU with four subjects of 50 examples each. With all skills off, the backbone reproduces its pretrained accuracy of 78.0% on GSM8K and 45.0% on MMLU. Active skills reduce both, and the size of the reduction depends on the configuration. The manipulation skill keeps most of the MMLU accuracy, the regularized driving skill reaches 53.0% on GSM8K, and the composed configuration reaches 41.0% and 37.0%. The overtrained driving skill, whose mean spectral deviation is 0.35, drops to 5.0% on GSM8K.

Table 4: Coefficient selection on the validation split with independently trained readouts fixed.

Search	Coefficients	Dist.	Budget
Selected	$(1.00, 0.50)$	–	–
CEM seed 0	$(1.02, 0.48)$	0.02	77
CEM seed 1	$(0.97, 0.52)$	0.03	77
CEM seed 2	$(1.05, 0.47)$	0.05	77
Adding the G1 skill			
Sweep	$\alpha_{\text{hum}} \in \{0, .25, .5, .75, 1\}$		5 cand.
Finalists	$0.25,\ 0.50 \to 0.25$		2 evals.

Regularization toward the pretrained configuration therefore preserves general capability as well as composability. Deviation magnitude alone does not determine the loss, since the composed configuration deviates more than the regularized driving skill yet scores higher on MMLU. Together with the switching costs in Section 4.2, these results support a deployment model in which control skills are activated for a task and removed exactly afterward.

5 LIMITATIONS

Our experiments use a 2B vision-language backbone, and all closed-loop evaluations are conducted in simulation. Whether the same composition behavior and parameter efficiency persist at larger scales, and under real perception noise and dynamics mismatch, remains to be established.

Table 5: General language performance under different spectral configurations.

Configuration	Mean $\lvert z - \mathbf{1}\rvert$	GSM8K	MMLU
Pretrained / all skills off	0.000	78.0%	45.0%
Manipulation skill	0.088	55.0%	43.0%
Regularized driving skill	0.219	53.0%	34.0%
Composed $(1.0, 0.5)$	0.240	41.0%	37.0%
Over-trained driving skill	0.350	5.0%	33.5%

Composition requires task-local calibration. The action readouts are recalibrated after composition, and composition coefficients must be selected for the skills being combined. These costs remain modest in our two- and three-skill settings, but calibration-free composition is beyond the current scope.

Active spectral skills interfere with the backbone's general language capabilities. Setting all skill coefficients to zero exactly recovers the pretrained configuration, so skill activation is reversible, but preserving language behavior while control skills remain active is not addressed by the current formulation. Recent work addresses this by training on dialogue and control data jointly (Zhou et al., 2025). Our setting acquires each skill independently and therefore cannot use that approach.

Shared coordinates also do not guarantee functional compatibility for every pair of skills. Appendix D.4 reports a combination for which regularization and readout recalibration are insufficient, and predicting such conflicts before composition remains open.

Finally, the current representation restricts each skill to singular-value scaling within a fixed pretrained basis. Richer spectral updates and larger collections of independently acquired skills remain natural extensions.

6 CONCLUSION AND FUTURE WORK

We represented embodied skills as spectral gain vectors in the coordinates of a frozen VLA backbone, so that skills learned independently on embodiments with different action spaces can be composed by arithmetic on their gains. Shared coordinates alone are not sufficient. Regularization toward the pretrained configuration and lightweight post-composition readout recalibration are both needed for the composed skills to remain functional.

Across autonomous driving, robot manipulation, and humanoid control, the representation supports two- and three-skill composition with competitive offline performance and effective closed-loop control in simulation, using 0.34M backbone adaptation parameters per skill against 1.09M for rank-1 and 8.7M for rank-8 LoRA. Composition yields positive transfer on driving, adding a third skill requires calibrating only its own readout and coefficient, and controlled experiments show that the spectral configuration carries task-specific behavior that readout optimization alone cannot recover.

The compactness of the representation points to a use beyond storage. A driving skill occupies 0.69 MB in FP16, or roughly 3 MB with its action head, whereas a full checkpoint of the backbone is on the order of gigabytes. When a fleet of agents shares one pretrained backbone, a new skill can be distributed as its gain vector rather than as a model, and activated or removed by setting its coefficient. In cooperative driving, a roadside unit could transmit a skill adapted to a particular intersection, weather condition, or traffic regime to arriving vehicles over the vehicular link, and vehicles could switch skills as they move between road segments (Yu et al., 2026). Real-world experiments on physical vehicles, manipulators, and humanoid robots are the natural next step for testing this setting under perception noise, dynamics mismatch, and communication latency. Scaling to larger backbones and larger skill collections, and predicting skill compatibility from the spectral coordinates before composition, are the other directions we consider most important.

STATEMENT ON THE USE OF AI ASSISTANTS

AI assistants were used for language polishing and occasional assistance with document formatting. All research questions, methodology, experiments, analyses, and results were developed and verified by the authors.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024. URL https://doi.org/10.48550/arXiv.2410.24164.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=AuJnXGq3AL.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705, 2025. doi: 10.1109/CVPR52734.2025.01742.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Chao Du, Tianyu Pang, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition, 2024. URL https://openreview.net/forum?id=w8eCnnq57m.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

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

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Chanhyuk Lee, Jiho Choi, Chanryeol Lee, Donggyun Kim, and Seunghoon Hong. Adarank: Adaptive rank pruning for enhanced model merging. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fTygcJVOni.

Yeoreum Lee, Jinwook Jung, and Sungyong Baik. Mitigating parameter interference in model merging via sharpness-aware fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eaTqsptDPL.

Xuhong Li, Yves GRANDVALET, and Franck DAVOINE. Explicit induction bias for transfer learning with convolutional networks, 2018. URL https://openreview.net/forum?id=rye7IMbAZ.

Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham Krishna Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski, and Sujay Sanghavi. SVFT: Parameter-efficient fine-tuning with singular vectors. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024. URL https://openreview.net/forum?id=JDPK4sY6Ks.

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*, 2023.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. URL https://arxiv.org/abs/1608.03983.

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

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: dynamic integration of modular expertise in model merging. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17703–17716. Curran Associates, Inc., 2022. doi: 10.52202/068431-1287. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/70c26937fbf3d4600b69a129031b66ec-Paper-Conference.pdf.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6ZBHIEtdP4.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun,

Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Junghwan Park, Woojin Cho, Junhyuk Heo, Darongsae Kwon, and Kookjin Lee. Basis-oriented low-rank transfer for few-shot and test-time adaptation, 2025. URL https://arxiv.org/abs/2512.02441.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L. Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15120–15130, June 2024.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Zhang Hanxue, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with Graph Visual Question Answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. doi: 10.1007/978-3-031-72943-0_15. URL https://mlanthology.org/eccv/2024/sima2024eccv-drivelm/.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the knots. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=67X93aZHII.

Qi Sun, Edoardo Cetin, and Yujin Tang. Transformer-squared: Self-adaptive llms, 2025. URL https://arxiv.org/abs/2501.06252.

Yanpeng Sun, Qiang Chen, Xiangyu He, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Jian Cheng, Zechao Li, and Jingdong Wang. Singular value fine-tuning: Few-shot segmentation requires few-parameters fine-tuning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=LEqYZz7cZOI.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Unitree Robotics. Unitree g1 humanoid robot. https://www.unitree.com/g1/, 2024. Accessed: 2026-08-22.

Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, Chelsea Finn, Max Du, Moo Jin Kim, Alexander Khazatsky, Jonathan Heewon Yang, Tony Z. Zhao, Ken Goldberg, Ryan Hoque, Lawrence Yunliang Chen, Simeon Adebola, Gaurav S. Sukhatme, Gautam Salhotra, Shivin Dass, Lerrel Pinto, Zichen Jeff Cui, Siddhant Haldar, Anant Rai, Nur Muhammad Mahi Shafiullah, Yuke Zhu, Yifeng Zhu, Soroush Nasiriany, Shuran Song, Cheng Chi, Chuer Pan, Wolfram Burgard, Oier Mees, Chenguang Huang, Deepak Pathak, Shikhar Bahl, Russell Mendonca, Gaoyue Zhou, Mohan Kumar Srirama, Sudeep Dasari, Cewu Lu, Hao-Shu Fang, Hongjie Fang, Henrik I Christensen, Masayoshi Tomizuka, Wei Zhan, Mingyu Ding, Chenfeng Xu, Xinghao Zhu, Ran Tian, Youngwoon Lee, Dorsa Sadigh, Yuchen Cui, Suneel Belkhale, Priya Sundaresan, Trevor Darrell, Jitendra Malik, Ilija Radosavovic, Jeannette Bohg, Krishnan Srinivasan, Xiaolong Wang, Nicklas Hansen, Yueh-Hua Wu, Ge Yan, Hao Su, Jiayuan Gu, Xuanlin Li, Niko Suenderhauf, Krishan Rana, Ben Burgess-Limerick, Federico Ceola, Kento Kawaharazuka, Naoaki Kanazawa, Tatsuya Matsushima, Yutaka Matsuo, Yusuke Iwasawa, Hiroki Furuta, Jihoon Oh, Tatsuya Harada, Takayuki Osa, Yujin Tang, Oliver Kroemer, Mohit Sharma, Kevin Lee Zhang, Beomjoon Kim, Yoonyoung Cho, Junhyek Han, Jaehyung Kim, Joseph J Lim, Edward Johns, Norman Di Palo, Freek Stulp, Antonin Raffin, Samuel Bustamante, João Silvério, Abhishek Padalkar, Jan Peters, Bernhard Schölkopf, Dieter Büchler, Jan Schneider, Simon Guist, Jiajun Wu, Stephen Tian, Haochen Shi, Yunzhu Li, Yixuan Wang, Mingtong Zhang, Heni Ben Amor, Yifan Zhou, Keyvan Majd, Lionel Ott, Giulio Schiavi, Roberto Martín-Martín, Rutav Shah, Yonatan Bisk, Jeffrey T Bingham, Tianhe Yu, Vidhi Jain, Ted Xiao, Karol Hausman, Christine Chan, Alexander Herzog, Zhuo Xu, Sean Kirmani, Vincent Vanhoucke, Ryan Julian, Lisa Lee, Tianli Ding, Yevgen Chebotar, Jie Tan, Jacky Liang, Igor Mordatch, Kanishka Rao, Yao Lu, Keerthana Gopalakrishnan, Stefan Welker, Nikhil J Joshi, Coline Manon Devin, Alex Irpan, Sherry Moore, Ayzaan Wahid, Jialin Wu, Xi Chen, Paul Wohlhart, Alex Bewley, Wenxuan Zhou, Isabel Leal, Dmitry Kalashnikov, Pannag R Sanketi, Chuyuan Fu, Ying Xu, Sichun Xu, brian ichter, Jasmine Hsu, Peng Xu, Anthony Brohan, Pierre Sermanet, Nicolas Heess, Michael Ahn, Rafael Rafailov, Acorn Pooley, Kendra Byrne, Todor Davchev, Kenneth Oslund, Stefan Schaal, Ajinkya Jain, Keegan Go, Fei Xia, Jonathan Tompson, Travis Armstrong, and Danny Driess. Open x-embodiment: Robotic learning datasets and RT-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, 2023. URL https://openreview.net/forum?id=zraBtFgxT0.

Kui Wang, Kazuma Nonomura, Zongdian Li, Tao Yu, Kei Sakaguchi, Omar Hashash, Walid Saad, Changyang She, and Yonghui Li. Augmented intelligence in smart intersections: Local digital twins-assisted hybrid autonomous driving. *IEEE Transactions on Intelligent Vehicles*, pp. 1–15, 2024a. doi: 10.1109/TIV.2024.3485132.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=Pf7kdIjHRf.

Yongxian Wei, Runxi Cheng, Weike Jin, Enneng Yang, Li Shen, Lu Hou, Sinan Du, Chun Yuan, Xiaochun Cao, and Dacheng Tao. Optmerge: Unifying multimodal llm capabilities and modalities via model merging. In *ICLR*, 2026.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7949–7961, 2022. doi: 10.1109/CVPR52688.2022.00780.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning, 2024b. URL https://arxiv.org/abs/2310.02575.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2024. URL https://arxiv.org/abs/2311.03099.

Taoyuan Yu. Optimizing computational efficiency in autonomous vehicles: Integrative edge and cloud computing strategies in vehicular networks. In *Proceedings of the 2024 11th International Conference on Wireless Communication and Sensor Networks*, icWCSN '24, pp. 5–12, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400709005. doi: 10.1145/3657529.3657538. URL https://doi.org/10.1145/3657529.3657538.

Taoyuan Yu, Kui Wang, Zongdian Li, Tao Yu, Kei Sakaguchi, and Walid Saad. Digital twin-based cooperative autonomous driving in smart intersections: A multi-agent reinforcement learning approach. In *2025 IEEE Globecom Workshops (GC Wkshps)*, pp. 1857–1862, 2025. doi: 10.1109/GCWkshps68340.2025.11590933.

Taoyuan Yu, Kui Wang, Zongdian Li, Tao Yu, Walid Saad, and Kei Sakaguchi. How roadside units enhance intersection safety? cooperative autonomous driving system design and a proof of concept, 2026. URL https://arxiv.org/abs/2608.09144.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=5r3e27I9Gy.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=lq62uWRJjiY.

Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Yaxin Peng, Chaomin Shen, Feifei Feng, and Yi Xu. ChatVLA: Unified multimodal understanding and robot control with vision-language-action model. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5377–5395, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.273. URL https://aclanthology.org/2025.emnlp-main.273/.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R Sanketi, Grecia Salazar, Michael S Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J Joshi, Alex Irpan, brian ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=XMQgwiJ7KSX.

A Experimental Details

A.1 Backbone, Spectral Adaptation, and Training

We use Qwen3-VL-2B-Instruct (Team, 2025) as the shared pretrained backbone. Spectral adaptation is applied to the query, key, value, output, gate, up, and down projections of each of the 28 decoder blocks, giving 196 adapted projections. The query, output, gate, up, and down projections have rank 2048 and the key and value projections have rank 1024, so each block contributes 12,288 spectral gains and each skill contains $28 \times 12{,}288 = 344{,}064$, or approximately 0.34M, backbone adaptation parameters. The thin SVD of each projection is computed once before training, and the singular vectors and singular values stay fixed throughout. Spectral gains are initialized at $\mathbf{1}$, which exactly recovers the pretrained projection before adaptation.

Table 6: Skill training configurations.

Setting	Driving	LIBERO	G1
Gain LR	2×10^{-3}	2×10^{-3}	2×10^{-3}
Head LR	1×10^{-3}	1×10^{-3}	1×10^{-4}
Weight decay	0.01	0.01	0
Max. epochs	30	30	8
Early stop	3	3	–
Schedule	cosine	cosine	linear
Warmup	–	–	100 steps

Each skill is trained independently on data from its own embodiment, with the spectral gains and the action head optimized jointly. We use AdamW (Loshchilov & Hutter, 2019) with an effective batch size of 16 and regularize the gains toward the pretrained configuration with $\lambda = 0.01$, following L2-SP (Li et al., 2018). Table 6 lists the remaining settings. Driving and LIBERO use cosine annealing (Loshchilov & Hutter, 2017) to 10^{-6} with early stopping after three epochs without improvement on the validation split. G1 is trained for a fixed eight epochs with 100 warmup steps followed by linear decay. The over-trained driving variant used in Section 4.3 and Appendix D.1 uses the same data, regularization, and learning rate but is trained for 100 epochs without early stopping, well past the point at which the regularized skill stops. Its mean absolute spectral deviation is 0.35, against 0.22 for the regularized skill.

A.2 Embodied Tasks, Metrics, and Action Readouts

Table 7 summarizes the data and action spaces of the three embodiments. Each task uses three disjoint subsets. The training split is used for skill acquisition and provides the post-composition calibration examples, the validation split is used for early stopping and for selecting composition coefficients, and the evaluation split is held out from every training and selection step and is used only for the reported offline errors.

Table 7: Embodied tasks and offline evaluation settings.

Task	Data and split	Input	Action output
CARLA	52,430 frames, 2% held-out eval.	512×384	10×2 waypoints
LIBERO	5% held out, 805 eval.	256×256	8×7 action chunk
G1 simulation	76,700 frames, 90/5/5 split, 3,835 eval.	256×256	16-D control

For CARLA (Dosovitskiy et al., 2017), the 52,430 frames are collected in Town01 following the data collection pipeline of Yu et al. (2025); Wang et al. (2024a); Yu (2024). The data are partitioned at the episode level, so frames from the same driving episode do not appear across the training, validation, and evaluation splits. All three splits are collected under the same conditions, without pedestrians and under fixed weather. The evaluation episodes in Appendix D.1 add pedestrians, denser traffic flow, and weather variation, and therefore measure generalization beyond the collection conditions rather than in-distribution fit. The policy predicts 10 future waypoints covering a three-second horizon. Closed-loop evaluation is described in Appendix A.3.

LIBERO (Liu et al., 2023) uses the LIBERO goal suite of 10 tasks with eight-step action chunks. The Unitree G1 (Unitree Robotics, 2024) block-stacking task is simulated in Isaac Lab using a custom control module, with training data collected in the same environment. The 76,700 frames are split 90/5/5 into training, validation, and evaluation examples. The 16-dimensional control output

consists of 14 arm joint commands and two gripper commands, normalized per dimension using statistics computed on the training split.

Let N denote the number of evaluation examples. For CARLA, $p_{i,t}, \hat{p}_{i,t} \in \mathbb{R}^2$ denote the target and predicted waypoints, with $T = 10$. For LIBERO, $a_{i,t,d}$ and $\hat{a}_{i,t,d}$ denote action components, with chunk length $H = 8$ and action dimension $D = 7$. For G1, $\tilde{a}_{i,d}$ and $\hat{\tilde{a}}_{i,d}$ denote the normalized target and predicted actions, with $D = 16$. The offline metrics are

$$\mathrm{ADE} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left\| \hat{p}_{i,t} - p_{i,t} \right\|_2 ,$$

$$\mathrm{MAE} = \frac{1}{NHD} \sum_{i=1}^{N} \sum_{t=1}^{H} \sum_{d=1}^{D} \left| \hat{a}_{i,t,d} - a_{i,t,d} \right| ,$$

$$\mathrm{nMSE} = \frac{1}{ND} \sum_{i=1}^{N} \sum_{d=1}^{D} \left(\hat{\tilde{a}}_{i,d} - \tilde{a}_{i,d} \right)^2 .$$

Each embodiment uses a separate three-layer MLP readout for its native action space. The driving and LIBERO readouts take the 2048-dimensional backbone representation as input, use ReLU activations with dropout 0.1 after the first hidden layer, and output 10×2 waypoints and 8×7 action chunks. For G1, the backbone representation is concatenated with the 16-dimensional proprioceptive state before entering the readout. Backbone adaptation parameter comparisons exclude these embodiment-specific readouts, which are kept identical across adaptation methods.

A.3 Post-Composition Recalibration and Closed-Loop Evaluation

Each embodiment-specific action head is trained jointly with its spectral skill during independent skill acquisition. After composition, we initialize the readout from this trained head and recalibrate it while keeping the composed spectral gains and pretrained backbone fixed, following the head-only adaptation principle of Kumar et al. (2022). Calibration examples are drawn from the training split of each task and are disjoint from both the validation and evaluation splits. All three embodiments use 3,000 task-local calibration examples for post-composition readout recalibration. Driving and LIBERO are recalibrated for 40 epochs with a learning rate of 10^{-3}, while G1 is recalibrated for four epochs with a learning rate of 3×10^{-4}. For the matched isolated expert control, we apply the same recalibration procedure to the corresponding independently trained head while keeping its isolated spectral gains fixed, which separates the effect of additional readout optimization from the change introduced by composition.

All three embodiments are evaluated in closed loop, with 100 episodes per configuration for CARLA and G1 and 100 episodes per task for LIBERO. Table 8 summarizes the settings. CARLA evaluation runs in synchronous Town01 simulation. The policy predicts a waypoint trajectory every 0.3 seconds, and a PID controller tracks it between replanning steps. Each episode contains 40 surrounding NPC vehicles. Success requires reaching within 6 meters of the route endpoint without collision. The timeout is the nominal traversal time at 1.5 m/s plus 600 simulator ticks. The same routes are used for every configuration.

Table 8: Closed-loop evaluation settings.

Setting	CARLA	LIBERO	G1 simulation
Episodes	100	1,000 (100 per task)	100
Execution	replan every 0.3 s	8-step chunks	interactive control
Success	route completion	task predicate success	task success
Termination	success / collision / timeout	success / 300 steps	task termination
Metric	success	success	success / nMSE

LIBERO uses the official initialization states followed by five no-op actions. The policy executes each predicted eight-step chunk before replanning, and episodes terminate on the environment success predicate or after 300 steps. We evaluate the 10 LIBERO-goal tasks with 100 episodes each, 1,000 episodes in total. G1 is evaluated in the same Isaac Lab environment on the block-stacking task. We report task success together with nMSE computed over the actions executed during the closed-loop trajectories.

A.4 Baseline Implementations

We compare against independently trained rank-1 and rank-8 LoRA experts (Hu et al., 2022) and post-hoc merging baselines. LoRA is applied to the same seven projections as our spectral adaptation, giving approximately 1.09M and 8.7M backbone adaptation parameters per skill for rank 1 and rank 8.

Task Arithmetic (Ilharco et al., 2023) combines task updates relative to the pretrained backbone in weight space and is evaluated for both ranks. TIES (Yadav et al., 2023) removes low-magnitude components and resolves sign disagreement before merging, with a 20% retention ratio. KnOTS+TIES (Stoica et al., 2025) first aligns the LoRA updates of each projection through a shared low-rank SVD representation, applies TIES in the aligned space, and maps the merged update back to weight space. TIES and KnOTS+TIES use the rank-8 experts. Merging coefficients for all baselines are selected on the same validation split and with the same search budget as our composition coefficients. Spectral-TIES, which applies the same procedure directly to $\Delta z^{(k)}$, is described in Appendix C.2.

We do not evaluate merging methods designed for fully fine-tuned checkpoints, such as Fisher-weighted averaging (Matena & Raffel, 2022), RegMean (Jin et al., 2023), and DARE (Yu et al., 2024), nor routing-based composition (Wu et al., 2024; Lu et al., 2024), which requires a router trained on all tasks jointly.

Joint training is included as a reference and uses simultaneous access to the task datasets. All composed baselines use the same held-out evaluation examples and the same post-composition readout calibration budget, with the merged backbone fixed and only the action heads updated.

A.5 Statistical Protocol

The main full-pipeline offline experiments use three independent training seeds, 42, 43, and 44. Each run includes independent skill training, composition, and post-composition readout recalibration.

Adaptation-only experiments keep the learned spectral gains fixed at the seed-42 checkpoints and vary only the randomness of readout recalibration. The matched isolated expert control follows the same variance scope, with the isolated gains held fixed while the readout is recalibrated under the matched budget. These results are reported separately from full pipeline variation. For a fixed trained model, confidence intervals use 1,000 bootstrap resamples of the held-out evaluation set (Efron & Tibshirani, 1993).

Coefficient landscape experiments use action heads recalibrated once at the selected operating point and then held fixed while the composition coefficients are varied. Coefficient selection, including the CEM search and the α_{hum} sweep, uses a validation split disjoint from the held-out evaluation set, and all reported errors are computed on the held-out set.

A.6 Deployment Storage and Switching Cost

We measure the storage and switching overhead of the spectral representation with the pretrained backbone resident on GPU. Each skill contains 344,064 spectral gain scalars, corresponding to approximately 0.69 MB in FP16 or 1.38 MB in FP32. Table 9 reports the measured overhead. Keeping all three FP32 gain sets resident requires 4.129 MB in addition to the shared backbone. Resident coefficient changes take a few milliseconds, and the embodiment-specific readout can be switched independently with sub-millisecond overhead, so multi-skill deployment does not require a separate backbone instance for each embodiment.

Table 9: Measured storage and switching overhead.

Quantity	Measured value
Backbone GPU memory	8.377 GB
All three spectral gain sets resident	4.129 MB
Resident spectral recombination	3.525 ms
Switch all skills off	0.693 ms
Load spectral gains from CPU	1.386 ms
Swap action readout from CPU	0.312 ms

B Extended Composition Results

This section reports the complete rank-1 LoRA composition trajectory summarized in Table 1, and the coefficient selection and spectral-contribution control for the third skill.

Table 10: Left: rank-1 LoRA composition trajectory. Right: G1 spectral-contribution control with the recalibrated G1 readout fixed.

Configuration	Driving ADE ↓	Manip. MAE ↓
Isolated rank-1 expert	0.3213	0.1153
Direct Task Arithmetic	1.0456	0.1807
Task Arithmetic + recalibration	0.2909	0.1255

Configuration	G1 nMSE ↓
$\alpha_{\text{hum}} = 0$	0.0434
$\alpha_{\text{hum}} = 0.25$	0.0068

B.1 Rank-1 LoRA Composition

To control for the parameter count of the LoRA baselines, we evaluate rank-1 LoRA on the same adapted projections as the rank-8 experts. Rank 1 is the smallest possible LoRA rank and requires 1.09M backbone adaptation parameters per skill, compared with 0.34M for the spectral representation. Table 10 (left) reports the two-skill composition trajectory. The isolated rank-1 experts reach a driving ADE of 0.3213 and a manipulation MAE of 0.1153. Direct Task Arithmetic degrades both tasks to 1.0456 and 0.1807, relative increases of 225.4% and 56.7%. Post-composition readout recalibration recovers most of the degradation without changing the merged backbone, reaching 0.2909 and 0.1255. Rank-1 LoRA therefore reaches composed performance comparable to ours while using $3.18\times$ more backbone adaptation parameters per skill.

B.2 Incremental Three-Skill Composition

When adding the G1 humanoid skill to the driving and manipulation composition, we retain $(\alpha_{\text{drive}}, \alpha_{\text{manip}}) = (1.0, 0.5)$ and vary only α_{hum}. We first screen $\alpha_{\text{hum}} \in \{0, 0.25, 0.50, 0.75, 1.00\}$ on the validation split, then evaluate the two leading nonzero candidates, 0.25 and 0.50, under the full three-skill protocol. This selects $\alpha_{\text{hum}} = 0.25$ and the final configuration $(\alpha_{\text{drive}}, \alpha_{\text{manip}}, \alpha_{\text{hum}}) = (1.0, 0.5, 0.25)$, as summarized in Table 4.

At this configuration, reusing the calibrated driving and manipulation readouts while recalibrating only the new G1 readout gives a driving ADE of 0.2927, a manipulation MAE of 0.1210, and a G1 nMSE of 0.0068. Recalibrating all three readouts gives 0.2925, 0.1204, and 0.0068. The previously calibrated readouts can therefore be retained when G1 is introduced, and adding the third skill requires calibrating only its own readout.

We further isolate the contribution of the G1 spectral skill by keeping the recalibrated G1 readout fixed and varying only α_{hum}. Table 10 (right) reports the result. Removing the G1 spectral contribution increases nMSE from 0.0068 to 0.0434, a $6.4\times$ increase, while restoring $\alpha_{\text{hum}} = 0.25$ reduces the error by approximately 84%. The readout and the 16-dimensional proprioceptive input are identical in both configurations. The G1 spectral contribution therefore remains necessary for the recovered three-skill performance.

C Additional Ablations

C.1 Matched Readout Recalibration Control

We test whether the post-composition results can be explained by the additional readout optimization applied after composition. We therefore give the isolated spectral experts the same readout-recalibration budget while keeping their spectral gains fixed. Table 11 reports the result. Matched recalibration leaves the isolated experts essentially unchanged. For driving, the composed model reaches 0.291, compared with 0.323 for the isolated expert and 0.3256 ± 0.0031 after matched recalibration. Driving therefore exhibits positive transfer under composition, and this gain is not explained by additional readout optimization alone. For manipulation, the isolated expert remains

near 0.113, while the composed model reaches 0.119, a modest residual composition cost. The two tasks thus exhibit asymmetric transfer under the same composition procedure.

Table 11: Matched readout-recalibration control.

Configuration	Driving ADE ↓	Manip. MAE ↓
Isolated expert	0.323	0.113
Isolated + matched recalib.	0.3256 ± 0.0031	0.1121 ± 0.0015
Composed + recalib.	0.291	0.119

C.2 SPECTRAL-TIES

The shared spectral coordinates also support composition that resolves conflicts between skills. We apply TIES directly to the spectral deviations $\Delta z^{(k)} = z^{(k)} - \mathbf{1}$. Because all skills share the same fixed singular basis, corresponding entries of $\Delta z^{(k)}$ refer to the same singular component, so the three steps of TIES, trimming, sign election, and disjoint merging, apply entry by entry without the alignment space that KnOTS constructs for LoRA updates. We use the same 20% retention ratio as the LoRA TIES baseline and apply the selected coefficients $(\alpha_{\text{drive}}, \alpha_{\text{manip}}) = (1.0, 0.5)$ to the merged deviation, followed by the same readout recalibration as additive composition. Table 12 reports the result. Spectral-TIES reduces driving ADE from 0.291 to 0.285 and manipulation MAE from 0.119 to 0.110 at the same parameter count. This shows that the shared spectral coordinates also provide a native interface for resolving conflicts between skills at the level of individual coordinates, without a separate alignment stage after training.

Table 12: TIES applied in the shared spectral space.

Composition rule	Driving ADE ↓	Manip. MAE ↓	Params / skill
Additive	0.291	0.119	0.34M
Spectral-TIES	0.285	0.110	0.34M

C.3 COMPOSITION OPERATOR

Our main experiments use additive composition around the pretrained configuration, Eq. 6. We compare it with a multiplicative rule $z_\ell^{\text{mult}} = \prod_k \big(\mathbf{1} + \alpha_k \Delta z_\ell^{(k)}\big)$ using the same trained skills under the adaptation-only protocol of Appendix A.5. Table 13 reports the results. Multiplicative composition reduces driving ADE from 0.293 to 0.282, while manipulation MAE changes only slightly from 0.134 to 0.133. However, it produces 14 nonpositive entries across the adapted projections, which must be truncated before forming the effective weights. Additive composition instead keeps each coefficient tied directly to one skill and removes a skill exactly at $\alpha_k = 0$. We therefore retain the additive rule as the default.

Table 13: Additive and multiplicative spectral composition under the adaptation-only protocol.

Composition rule	Driving ADE ↓	Manip. MAE ↓
Additive	0.293	0.134
Multiplicative	0.282	0.133

C.4 SEPARATE COEFFICIENTS PER PROJECTION TYPE

The main method assigns one composition coefficient to each skill and applies it across all adapted projections. To test whether this scalar parameterization is overly restrictive, we optimize separate coefficients for each of the seven projection types and each of the two skills, a 14-dimensional search space. We use CEM with a population of 24 for 10 generations, 241 evaluations in total. The resulting driving coefficients lie between 0.915 and 1.050 and the manipulation coefficients between 0.423 and 0.616, concentrated around the scalar solution $(\alpha_{\text{drive}}, \alpha_{\text{manip}}) = (1.0, 0.5)$.

Table 14 summarizes the comparison. The additional flexibility improves the normalized joint score by only 1.31%. Most of the useful variation in composition strength is therefore captured by one coefficient per skill.

Table 14: One coefficient per skill versus one per projection type.

Parameterization	# Coeff.	Coefficient range	Joint-score gain
Per-skill scalar	2	$(1.00,\ 0.50)$	–
Projection-type-specific	14	Drive: 0.915–1.050 Manip.: 0.423–0.616	1.31%

C.5 Distribution of Spectral Skill Adaptation

We examine how spectral adaptation is distributed across the 28 decoder blocks. For each block, we compute the mean absolute spectral deviation $\text{mean}\,|\Delta z|$ over its adapted projections. Because the absolute magnitude differs across tasks, each depth profile is normalized by its own mean when comparing its shape across the backbone. The regularized and over-trained driving skills have mean absolute spectral deviations of 0.219 and 0.350, LIBERO 0.088, and G1 0.036. These magnitudes are reported before the per-skill normalization used in Fig. 4.

The normalized depth profiles show that driving and manipulation adaptations are distributed broadly across the decoder, with blockwise Gini coefficients of 0.016 for regularized driving, 0.010 for over-trained driving, and 0.038 for LIBERO. G1 is more concentrated toward later decoder blocks, with a Gini coefficient of 0.147.

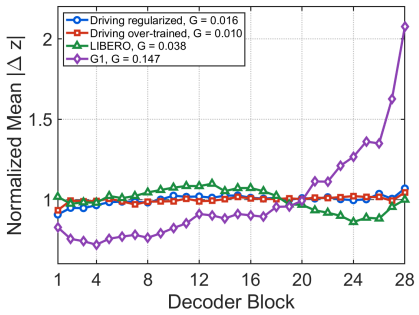


Figure 4: Distribution of spectral adaptation across decoder depth. Blockwise displacement is normalized by each skill's mean. The legend reports the blockwise Gini coefficient G.

D Robustness, Switching, and Failure Cases

D.1 Out-of-Distribution Robustness

We evaluate the driving skill under four distribution shifts. The training data and the in-distribution evaluation split are collected without pedestrians and under fixed weather. New Town01 episodes are newly generated routes in the training town with added pedestrians, denser traffic flow, and weather variation, none of which appear in the training data. Town03 is an unseen town under the same added conditions. The traffic-light subset contains signalized intersections, and the dynamics stress set perturbs the communication latency between perception and control. These shifts explain the larger absolute errors relative to Table 1. Table 15 compares the over-trained driving skill, the regularized driving skill used in the main method, and the composed model after post-composition readout recalibration.

Table 15: Driving ADE under distribution shift.

Evaluation setting	Over-trained	Regularized	Composed + recalib.
New Town01 episodes	1.122	**1.021**	1.098
Town03	1.476	**1.288**	1.345
Traffic-light intersections	0.256	0.333	**0.250**
Dynamics stress set	3.658	**3.093**	3.366

The regularized driving skill gives the lowest ADE on the new Town01 episodes, Town03, and the dynamics stress set, while the composed model performs best on the traffic-light subset. The relative advantage therefore varies across distribution shifts rather than favoring a single configuration uniformly. The over-trained skill is not the best performing configuration under any of the four shifts,

consistent with the regularization analysis in Section 4.3. The composed model remains competitive with the isolated driving variants across all four evaluations.

D.2 Trajectory-Level Safety Diagnostics

We additionally report three trajectory diagnostics for driving. Minimum separation measures the closest predicted distance to surrounding traffic. The violation rate counts cases with separation below 2 m within the prediction horizon, and time to collision (TTC) is computed from the predicted trajectories. Table 16 reports these quantities on the main evaluation set, Town03, and the traffic-light subset.

Table 16: Trajectory-level driving diagnostics. Higher minimum separation and TTC are preferred, while lower violation rate is preferred.

Set	Configuration	Min. sep. (m) ↑	Viol. (%) ↓	TTC (s) ↑
Main	Ground truth	15.50	0.1	1.80
	Over-trained	15.45	0.2	2.07
	Regularized	15.47	0.2	2.13
	Composed	15.44	0.2	1.78
Town03	Ground truth	19.10	0.6	2.32
	Over-trained	19.37	0.1	2.05
	Regularized	19.38	0.2	2.47
	Composed	19.36	0.1	2.15
Traffic light	Ground truth	12.98	0.1	2.50
	Over-trained	12.96	0.1	2.60
	Regularized	12.96	0.1	2.60
	Composed	12.98	0.1	2.55

The composed configuration remains close to the isolated driving variants across all three diagnostics and does not increase the measured violation rate on Town03 or the traffic-light subset. These quantities are diagnostic rather than a complete safety evaluation.

D.3 General Capability Evaluation Protocol

Table 5 in Section 4.5 reports general language performance under different spectral configurations. GSM8K uses 100 test problems with exact-match scoring of the final answer. MMLU uses four subjects, high-school mathematics, college computer science, formal logic, and high-school computer science, with 50 examples each, scored by multiple-choice accuracy over 200 questions. All configurations use identical prompts and decoding settings, with the action heads detached so that the backbone generates text. Setting all skill coefficients to zero restores the pretrained spectral configuration exactly and reproduces the pretrained accuracy on both benchmarks. Storage and switching overhead for this activation and removal are reported in Appendix A.6.

D.4 Compatibility Boundaries

Shared spectral coordinates do not guarantee functional compatibility for every pair of independently learned skills. As a negative case, we train a pedestrian behavior-prediction skill that classifies the intended behavior of surrounding pedestrians and compose it with the manipulation skill using the same regularization, coefficient search, and readout recalibration as in the main experiments. The behavior-prediction skill reaches 99.9% classification accuracy in isolation but only 52.8% after composition. The tested coefficient adjustments and readout recalibration do not recover the isolated performance. Shared coordinates therefore provide the structure for composition rather than a guarantee of functional compatibility, and predicting such conflicts before composition remains open.