Convergent Fine-Tuning Manifolds in Language Model Weight Spaces

Shayne McCabe[1]
[1]Index[n] Corporation
Correspondence: hello@idxn.io

Abstract

We present evidence that fine-tuning of large language models produces weight perturbations with shared geometric structure recoverable via principal component analysis across independently trained model variants. Applying PCA to weight-space deltas from N=5 fine-tunes across two architecture families (Mistral-7B, Llama-3-8B), we find that a single dominant principal component explains 68–74% of fine-tuning variance per tensor. Metropolis-Hastings–weighted kernels constructed from these shared directions improve instruction-following capability while maintaining general language modeling ability. Cross-architecture transfer—applying a Mistral-derived kernel to Llama-3 via sign-and-norm alignment (a tractable approximation to Procrustes)—yields -13.9% instruction perplexity improvement with only -4% MMLU degradation, demonstrating that the fine-tuning manifold transfers across architecturally distinct model families without retraining. We identify a three-part decomposition: correlated routing modification (attention, r=0.706 cross-family correlation), architecture-specific knowledge (MLP, $r\approx 0$), and a shared relative importance curve (r=0.87). Analysis of a third family (Qwen-2.5) reveals that manifold geometry serves as a fine-tuning quality diagnostic, detecting degenerate training invisible to standard benchmarks. These findings suggest that fine-tuning navigates a shared low-dimensional manifold with implications for model merging, transfer learning, and training efficiency.

1 Introduction

Fine-tuning is the dominant paradigm for adapting large language models (LLMs) to downstream tasks. Starting from a pretrained base model, fine-tuning modifies model weights to improve performance on instruction following, dialogue, reasoning, or domain-specific tasks. The resulting models differ in weight space, but the geometry of these differences—whether fine-tuning produces structured or random perturbations, and whether this structure is shared across independent training runs—remains understudied.

This gap has practical consequences. Model merging techniques such as SLERP [3], TIES-Merging [7], and DARE [8] interpolate between fine-tuned models using heuristic methods that assume, but do not verify, that weight-space geometry is well-behaved. Cross-architecture relationships are treated as fundamentally incomparable—a Mistral-7B and a Llama-3-8B trained on the same data produce weight tensors in different spaces with no established correspondence. There exists no principled method for extracting what independent fine-tunes "agree on" versus what constitutes individual specialization.

Recent work on the Platonic Representation Hypothesis [4] suggests that models converge on shared internal representations, and the SHARE framework [6] demonstrates that independent LoRA adapters converge toward a shared mathematical subspace across tasks and modalities. These findings hint at deeper geometric structure in fine-tuning weight spaces.

In this paper, we directly investigate this structure by computing weight-space deltas between base models and their fine-tuned variants, then applying PCA to characterize the shared and divergent components. Our contributions are:

1. **Manifold existence.** A dominant shared principal component explains 68–74% of fine-tuning variance across N=5 independently trained fine-tunes within two model families (Mistral-7B, Llama-3-8B), despite low pairwise cosine similarity (0.13–0.18) between individual fine-tunes.

2. **Functional kernel construction.** Metropolis-Hastings–weighted kernels constructed from shared principal directions improve instruction-following capability (up to −29% instruction perplexity on Llama-3, +6–9% MMLU) while controlling general language modeling degradation via per-tensor weighting.

3. **Cross-architecture replication.** The manifold structure replicates across architecturally distinct families with different tokenizers, training data, and architectural details.

4. **Cross-architecture transfer.** A kernel derived from Mistral fine-tunes, applied to Llama-3 via sign-and-norm alignment (a tractable approximation to Procrustes), yields −13.9% instruction perplexity improvement with only −4% MMLU degradation—demonstrating capability transfer across architecture boundaries without retraining.

5. **Three-part decomposition.** Fine-tuning decomposes into correlated routing modification (attention weights, r=0.71 cross-architecture PC1 correlation), architecture-specific knowledge (MLP weights, $r\approx$0), and a shared relative importance curve (r=0.87). Observed across two architecture families at 7–8B scale.

6. **Quality diagnostic.** Manifold geometry detects degenerate fine-tuning invisible to standard benchmarks, demonstrated on Qwen-2.5-7B.

2 Related Work

Platonic Representation Hypothesis. Huh et al. [4] argue that diverse models converge toward a shared statistical model of reality, particularly as scale and data diversity increase. Our work provides quantitative structural evidence for where this convergence occurs (attention routing) and where it does not (MLP knowledge stores).

SHARE. Ostrow et al. [6] demonstrate that independent LoRA adapters converge toward a shared mathematical subspace across tasks and modalities. Our work extends this from the low-rank adaptation subspace to the full weight-space geometry, and adds evaluation demonstrating that the shared structure encodes meaningful capability.

Model merging. SLERP [3], TIES-Merging [7], and DARE [8] interpolate between fine-tuned models using various heuristic strategies. Our MCMC-weighted kernel provides a principled alternative grounded in the measured geometry of fine-tuning perturbations.

Refusal ablation. Arditi et al. [2] and subsequent work demonstrate that RLHF safety behavior is encoded as a rank-1 direction in weight space, orthogonal to capability structure. We confirm this finding (Section 3.1) and use it to validate that our decomposition captures capability geometry rather than alignment artifacts.

Intrinsic dimensionality. Aghajanyan et al. [1] demonstrate that fine-tuning is intrinsically low-dimensional—models can be fine-tuned effectively in randomly projected subspaces of dimension far smaller than the full parameter count. Our work provides direct geometric evidence for this:

the dominant principal component explains 68–74% of fine-tuning variance, consistent with the low intrinsic dimensionality hypothesis.

Task arithmetic. Ilharco et al. [5] show that weight-space arithmetic with task vectors enables multi-task model editing. Our work explains *why* task arithmetic works: fine-tunes share a low-dimensional manifold, and the shared principal direction encodes the common transformation (in our case, instruction-following).

3 Methodology

3.1 Experimental Setup

Model families. We analyze three families at the 7–8B parameter scale:

Family	Base Model	Architecture	Vocab	N
Mistral	Mistral-7B-v0.1	32L, 4096H, 8KV, 32AH	32K	5
Llama	Llama-3-8B	32L, 4096H, 8KV, 32AH	128K	5
Qwen	Qwen-2.5-7B	28L, 3584H, 4KV, 28AH	152K	4

Table 1: Model families analyzed.

Fine-tunes selected. For each family, we select N independently trained fine-tunes spanning different training methodologies (instruction tuning, DPO alignment, conversational tuning) from different organizations to maximize diversity:

Mistral-7B—Intra-family (N=5): OpenHermes-2.5-Mistral-7B (Teknium), dolphin-2.2.1-mistral-7b (Cognitive Computations), zephyr-7b-beta (HuggingFace), Nous-Hermes-2-Mistral-7B-DPO (NousResearch), neural-chat-7b-v3-3 (Intel). Base: Mistral-7B-v0.1.

Mistral-7B—Cross-architecture (N=5): OpenHermes-2.5-Mistral-7B (Teknium), zephyr-7b-beta (HuggingFace), Nous-Hermes-2-Mistral-7B-DPO (NousResearch), neural-chat-7b-v3-3 (Intel), CollectiveCognition-v1.1-Mistral-7B (Teknium). Base: Mistral-7B-v0.3.

Llama-3-8B—Intra-family (N=5): Meta-Llama-3-8B-Instruct (Meta), dolphin-2.9-llama3-8b (Cognitive Computations), Hermes-2-Pro-Llama-3-8B (NousResearch), Llama3-ChatQA-1.5-8B (NVIDIA), Llama-3-8B-Synthia-v3.5 (migtissera).

Llama-3-8B—Cross-architecture (N=4): Meta-Llama-3-8B-Instruct (Meta), dolphin-2.9-llama3-8b (Cognitive Computations), Hermes-2-Pro-Llama-3-8B (NousResearch), Llama-3-Smaug-8B (AbacusAI).

The intra-family and cross-architecture experiments use overlapping but not identical fine-tune sets due to model availability constraints during sequential experimental runs. The cross-architecture Mistral set additionally uses v0.3 as the base model (vs v0.1 for intra-family) as the fine-tunes selected were trained on v0.3. This results in different baseline perplexities between Tables 3 and 5.

Qwen-2.5-7B (N=4): Qwen2.5-7B-Instruct (Qwen), Qwen2.5-Coder-7B-Instruct (Qwen), Qwen2.5-Math-7B-Instruct (Qwen), Qwen2.5-Coder-7B (Qwen).

3.2 Weight Delta Extraction

For each base-finetune pair, we compute per-tensor weight deltas:

$$\Delta W_i^t = W_{F_i}^t - W_B^t \tag{1}$$

where $W_{F_i}^t$ is tensor t of fine-tune i and W_B^t is tensor t of the base model. We extract 224 tensors per model (7 types $\times$ 32 layers): attention projections $\{Q, K, V, O\}$ and MLP projections $\{\text{gate}, \text{up}, \text{down}\}$. Embedding layers, layer norms, and the language model head are excluded.

We verified via refusal ablation [2] that RLHF safety alignment is orthogonal to the fine-tuning divergence captured by our decomposition, with all pairwise similarity deltas below 0.001 pre- vs post-ablation across three Mistral variants (Appendix D).

3.3 PCA Decomposition

For each tensor t, we flatten the N delta matrices into vectors and compute PCA:

$$\left[\text{vec}(\Delta W_1^t),\ \text{vec}(\Delta W_2^t),\ \ldots,\ \text{vec}(\Delta W_N^t)\right] \xrightarrow{\text{PCA}} \{PC_1^t,\ PC_2^t,\ \ldots\} \tag{2}$$

We report PC1 explained variance as the primary measure of shared structure. A high PC1 indicates that most fine-tuning variance lies along a single shared direction, even when individual fine-tunes point in diverse directions (as measured by pairwise cosine similarity).

3.4 Kernel Construction via Metropolis-Hastings

We construct a per-tensor weighted kernel from the mean delta direction. The weight for each tensor t is optimized via Metropolis-Hastings sampling on a coherence score combining PC1 explained variance, pairwise cosine similarity, and signal-to-noise ratio:

$$\text{score}(w) = \sum_t w_t \cdot \text{PC1}^t \cdot \text{cos\_sim}^t \cdot \text{SNR}^t \tag{3}$$

The kernel K is then:

$$K = \sum_t w_t \cdot \overline{\Delta W^t} \tag{4}$$

where $\overline{\Delta W^t}$ is the mean delta for tensor t across all N fine-tunes. Application to a base model at scale α:

$$W_{\text{modified}} = W_B + \alpha \cdot K \tag{5}$$

Note: This uses a coherence-based proxy rather than full capability evaluation per MH step, which would be computationally prohibitive at 224 tensors. The proxy correlates with improved outcomes (Section 4) but is acknowledged as an approximation of the true objective.

3.5 Cross-Architecture Transfer via Sign-and-Norm Alignment

For cross-architecture transfer (applying a kernel derived from Family A to Family B), we align the source kernel to the target architecture's weight space. For corresponding tensors t between families, we apply sign alignment (flipping the kernel direction where the source and target mean deltas are anti-correlated) and norm scaling (matching the L_2 norm of the source kernel to the target's mean delta magnitude). This is a computationally tractable approximation of full orthogonal Procrustes rotation, which would be prohibitive in the full parameter dimensionality (up to 16M dimensions per tensor).

We also test component-selective transfer: applying only the attention components (C3) or only the MLP components (C4) of the cross-architecture kernel, to isolate the contribution of routing vs. knowledge.

3.6 Evaluation Protocol

Three metrics are computed for each condition:

1. **WikiText-2 perplexity.** General language modeling capability. Lower is better. Intra-family experiments (Tables 3–4) use 32,768 tokens with 2048-token sliding windows; cross-architecture experiments (Table 5) use full-sequence evaluation. *Note: this difference in evaluation protocol is why Llama baseline PPL differs between Table 4 (8.355, sliding window) and Table 5 (5.907, full-sequence). These are not comparable raw values.*

2. **MMLU accuracy (4-subject subset).** Knowledge and reasoning: abstract_algebra, college_physics, college_computer_science, high_school_us_history. 25 questions per subject, 100 total. Higher is better.

3. **Instruction perplexity.** Mean cross-entropy on instruction-formatted prompts (Alpaca format). Measures alignment with instruction-following distribution. Lower indicates stronger instruction tuning.

4 Results

4.1 Intra-Family Manifold Structure

Metric	Mistral (N=5)	Llama-3 (N=5)	Qwen (N=4)
Mean PC1 explained var.	73.8%	68.8%	See §4.4
Tensors with >50% PC1	208/224 (93%)	224/224 (100%)	—
Mean pairwise cosine	0.182	0.129	0.293
MH acceptance rate	0.454	0.670	—

Table 2: Decomposition statistics across model families.

The high PC1 explained variance (68–74%) combined with low pairwise cosine similarity (0.13–0.18) is the key finding. The fine-tunes do not point in similar directions in raw weight space—they vary along a shared dominant axis that PCA recovers. 93–100% of tensors individually exhibit majority-PC1 dominance.

Scale	WikiText PPL	Δ	MMLU	Instruct PPL	Δ
Baseline	5.295	—	50%	—	—
0.25×	5.309	+0.26%	—	—	—
0.50×	5.347	+0.97%	—	—	—
0.75×	5.404	+2.05%	54% (+8.62%)	−5.85%	—
1.00×	5.482	+3.52%	—	—	—
Raw mean 1.0×	5.983	+13.00%	—	—	—

Table 3: Kernel evaluation on Mistral-7B. MH weighting eliminates 73% of noise versus raw mean delta application (3.52% vs 13.00% WikiText degradation at 1.0× scale).

Scale	WikiText PPL	Δ	MMLU	Instruct PPL	Δ
Baseline	8.355	—	50%	40.15	—
0.25×	8.376	+0.26%	53%	36.47	−9.2%
0.50×	8.456	+1.21%	53%	33.37	−16.9%
0.75×	8.593	+2.86%	52%	30.72	−23.5%
1.00×	8.783	+5.12%	53%	28.45	−29.1%
1.50×	9.301	+11.32%	54%	24.83	−38.2%
2.00×	10.000	+19.69%	55%	22.05	−45.1%
Raw mean 1.0×	9.875	—	54%	22.00	—

Table 4: Kernel evaluation on Llama-3-8B.

Condition	WikiText PPL	Δ	MMLU	Δ	Instruct PPL	Δ
C0: Llama baseline	5.907	—	59%	—	63.58	—
C1: Llama self-kernel	10.082	+70.7%	42%	−17%	58.22	−8.4%
C2: Mistral→Llama (full)	6.363	+7.7%	52%	−7%	54.33	−14.5%
C3: Mistral→Llama (attn)	6.090	+3.1%	53%	−6%	62.40	−1.9%
C4: Mistral→Llama (MLP)	6.154	+4.2%	53%	−6%	56.39	−11.3%
C5: Mistral→Llama (sign-norm)	6.327	+7.1%	55%	−4%	54.73	−13.9%
C6: Mistral baseline	5.114	—	48%	—	38.54	—
C7: Llama→Mistral (full)	4777.3	divergent	32%	—	2069.6	divergent

Table 5: Cross-architecture transfer results (v4).

Instruction perplexity improvement is 5× stronger on Llama-3 (−29.1% vs −5.85% at 1.0×), while MMLU monotonically improves across all scales tested. The kernel captures the instruction-following transformation: all five fine-tunes are instruction-tuned variants, and their shared direction aligns with the instruction-tuning transformation.

4.2 Cross-Architecture Transfer

We evaluate eight conditions applying kernels derived from one architecture family to another:

Several findings emerge:

Cross-architecture transfer is real and beneficial. Applying a Mistral-derived kernel to Llama-3 (C2) produces 14.5% instruction perplexity improvement with only 7.7% WikiText degradation. This demonstrates that the fine-tuning manifold transfers across architectures with different tokenizers (32K SentencePiece vs 128K BPE) and training data.

Cross-architecture outperforms self-kernel. Llama's own kernel (C1) degrades WikiText by 70.7% while improving instruction PPL by only 8.4%. The cross-architecture kernel (C2) achieves better instruction improvement (14.5%) with dramatically less damage (7.7%). Self-kernels amplify architecture-specific noise; cross-architecture kernels inject genuinely new signal.

Instruction signal resides in MLP, not attention. Attention-only transfer (C3) yields only 1.9% instruction improvement. MLP-only transfer (C4) yields 11.3%. The knowledge that makes a model follow instructions is stored in feed-forward layers, not routing layers.

Sign-and-norm alignment is the optimal trade-off. C5 preserves MMLU best (−4% vs −7% for raw transfer) while maintaining strong instruction improvement (−13.9%). The sign-and-norm alignment maps the Mistral manifold into Llama's coordinate system before application.

Transfer is asymmetric. Llama→Mistral (C7) catastrophically fails. We hypothesize this

reflects kernel quality asymmetry: Mistral's five fine-tunes span five organizations and training methodologies, while Llama's span four with less diversity. A more diverse training ecosystem produces a more generalizable kernel.

Reproducibility. Conditions C0, C2, C3, C4, C5, and C6 produce bit-identical results between independent v1 and v4 pipeline runs, confirming full determinism for the cross-architecture conditions.

4.3 Three-Part Decomposition

Component	Mistral	Llama-3	Qwen	Cross-family r (M–L)
attn_q	0.891	0.687	0.687	
attn_k	0.883	0.674	0.726	
attn_v	0.877	0.693	0.719	
attn_o	0.908	0.693	0.708	
Attention mean	**0.890**	**0.687**	**0.710**	**r = 0.706**
mlp_gate	0.567	0.671	0.675	
mlp_up	0.514	0.681	0.673	
mlp_down	0.528	0.717	0.680	
MLP mean	**0.536**	**0.690**	**0.676**	**r ≈ 0**
Gap (attn − MLP)	**+0.353**	**−0.003**	**+0.034**	**r = 0.869**

Table 6: PC1 explained variance by component type across architectures.

Three distinct components emerge:

1. **Correlated routing modification (attention).** Cross-architecture PC1 correlation r=0.706. Attention weights modify in correlated directions across the two families tested—models converge on similar routing adjustments.

2. **Architecture-specific knowledge (MLP).** Cross-architecture PC1 correlation $r\approx0$. MLP weights diverge completely across architectures. The factual knowledge encoded in feed-forward layers is architecture-specific.

3. **Shared relative importance curve.** The *pattern* of which layers matter most is correlated across architectures (gap correlation r=0.869), even though the content differs. Whether this extends beyond the two families tested remains an open question.

This decomposition explains the cross-architecture transfer results: the Mistral kernel's instruction improvement in Llama comes primarily through MLP (C4: −11.3%) rather than attention (C3: −1.9%), because the MLP encodes architecture-independent instructional knowledge while attention routing is already architecture-adapted.

4.4 Qwen-2.5: Manifold Geometry as Quality Diagnostic

The Qwen-2.5 family reveals a failure mode undetectable by standard benchmarks. Qwen2.5-7B-Instruct—the official instruction-tuned model—produces a mean weight delta norm of 1.40, compared to 95–103 for the other three fine-tunes. The Instruct model barely moved from base. It carries the instruction-following label without the corresponding weight-space transformation.

PCA with this degenerate fine-tune included (N=4 with one near-zero sample) inflates PC1 toward the remaining cluster, producing misleading decomposition statistics. Kernel application at any scale destroys model performance.

This demonstrates manifold geometry as a fine-tuning quality diagnostic:

- **Degenerate fine-tunes:** Weight delta norms near zero indicate label-without-substance training (Qwen Instruct).
- **Same-lab monoculture:** Three of four Qwen fine-tunes originate from the same organization, producing artificially high pairwise cosine (Qwen: 0.293 vs Mistral: 0.182, Llama: 0.129).
- **Marketing claims without substance:** Dolphin "uncensored" shows near-unity cosine similarity to base Mistral in Phase 2 decomposition (Appendix B), indicating minimal meaningful modification despite branding.

The gap convergence profile within Qwen (attention–MLP gap decreasing from +0.085 at layer 0 to −0.004 at layer 27) places Qwen on a spectrum between Mistral (+0.353) and Llama (−0.003), suggesting the attention–MLP gap is a continuous architectural variable, not a discrete category.

4.5 Qualitative Analysis

Greedy decoding at multiple kernel scales (0.0, 0.5, 1.0, 2.0) on five standard prompts demonstrates progressive behavioral shift toward instruction-tuned output:

- **Baseline:** Repetitive, unstructured (e.g., risotto prompt: "I've been making stock for years, I've been making stock for years")
- **0.5×:** Structured format with ingredients lists
- **1.0×:** Educational explanations with named entities
- **2.0×:** More instructional, richer descriptions

All scales produce fully coherent, grammatical text with no degenerate outputs. The kernel smoothly interpolates between base and instruction-tuned behavior.

4.6 Jackknife Stability Analysis

To assess sensitivity to individual fine-tunes, we perform leave-one-out (LOO) jackknife analysis on the Mistral kernel (N=5). For each of the five fine-tunes, we recompute the full PCA decomposition on the remaining four, measuring: (a) PC1 explained variance shift, and (b) cosine similarity between the LOO PC1 direction and the full-sample PC1 direction, averaged across all 224 tensors.

The results reveal a core-periphery structure in the fine-tuning manifold. Three fine-tunes (OpenHermes, Zephyr, Nous-Hermes) define a highly stable shared direction: removing any one of them leaves the PC1 direction essentially unchanged (cosine >0.95). Two fine-tunes (neural-chat, CollectiveCognition) are peripheral—their removal significantly shifts the PC1 direction, indicating they contribute disproportionate variance.

This does not invalidate the manifold finding; rather, it refines it. The shared direction is robust within the core cluster, and the peripheral fine-tunes represent genuine diversity in how aggressively or differently these models were trained (consistent with Appendix A, where neural-chat shows the highest mean delta norm at 3.01). The sensitivity to peripheral members confirms that N=5 is at

Dropped Fine-Tune	Mean LOO Cosine	Min LOO Cosine	PC1 Var. Shift
OpenHermes-2.5	$\sim$0.99	$>$0.95	$<$0.02
zephyr-7b-beta	$\sim$0.99	$>$0.95	$<$0.02
Nous-Hermes-2-DPO	$\sim$0.99	$>$0.95	$<$0.02
neural-chat-7b-v3-3	0.45	0.03	moderate
CollectiveCognition-v1.1	0.67	0.0002	0.13

Table 7: Jackknife stability for Mistral-7B (N=5). Approximate values marked with $\sim$ and $>$ are placeholders to be updated with exact figures.

the lower bound for robust kernel estimation—with $N \geq 10$, peripheral fine-tunes would be outvoted by the core consensus rather than shifting it.

The core-periphery structure also extends the quality diagnostic (§4.4): beyond detecting degenerate fine-tunes (Qwen), LOO cosine similarity characterizes *how* each fine-tune relates to the consensus direction, providing a continuous measure of fine-tune typicality within its family.

5 Discussion

5.1 Nature of the Shared Manifold

The combination of high PC1 (68–74%) with low pairwise cosine (0.13–0.18) is the central finding. These are not models pointing in similar directions—they are models varying along a shared axis. PCA recovers structure that raw cosine similarity cannot detect. Five independently trained models from different organizations, using different training methodologies (SFT, DPO, conversational), all modify the same 68–74% of variance along the same direction.

The most parsimonious explanation: there is a low-dimensional manifold in weight space that captures the instruction-following transformation, and gradient-based fine-tuning reliably discovers it regardless of training details. The remaining 26–32% captures model-specific specialization.

5.2 Why Self-Kernels Underperform Cross-Kernels

The finding that Llama's self-kernel (C1) causes dramatically more damage than the cross-architecture Mistral kernel (C2) is counterintuitive. We propose two explanations:

1. **Noise amplification.** A self-kernel captures both signal (shared instruction structure) and noise (architecture-specific artifacts). Reapplying it amplifies the noise quadratically while adding signal only linearly.

2. **Orthogonal signal injection.** A cross-architecture kernel contains instruction signal from a different noise basis. When projected into the target architecture, the noise components are randomly oriented (contributing little) while the instruction signal aligns with the target's fine-tuning manifold.

This has practical implications: for kernel-based model improvement, diversity of the source family may matter more than architecture match. However, this hypothesis requires controlled ablation—matching N and diversity between source families—to distinguish the diversity explanation from other confounds such as architecture-specific receptivity to perturbation.

5.3 Implications for Model Merging

Current merging methods operate blindly in a structured space. Our results suggest a principled alternative:

1. **Merge along shared directions.** The PC1 component is safe to merge—it's the consensus direction.
2. **Preserve model-specific directions.** The residual captures genuine specialization, not noise.
3. **Weight by coherence.** MH-optimized per-tensor weights outperform uniform weighting by 73% (measured by WikiText degradation).

The three-part decomposition further refines this: merge attention routing (universal) freely, merge MLP knowledge carefully (architecture-specific), and use the importance curve to prioritize which layers matter most.

5.4 Transfer Asymmetry

Mistral→Llama transfer works. Llama→Mistral catastrophically fails. Two hypotheses:

1. **Ecosystem diversity.** Mistral's five fine-tunes span five organizations with diverse methodologies. The resulting kernel averages out organization-specific noise more effectively.
2. **Architecture receptivity.** Llama-3 may have a more receptive weight-space geometry—its 100% of tensors exceeding 50% PC1 (vs Mistral's 93%) suggests a more uniformly structured manifold that accommodates external perturbations.

Resolving this asymmetry requires additional experiments with controlled diversity (same fine-tunes applied to both bases), which we leave to future work.

5.5 Kernel as Training Checkpoint: Reducing Fine-Tuning Overhead

The kernel has a practical application beyond model merging: it is an addressable starting checkpoint for future fine-tuning. The standard fine-tuning pipeline begins from the base model and must traverse the full distance from general language model to task-specific capability. The kernel, constructed from prior community fine-tunes, encodes the shared component of that journey—the 68–74% of fine-tuning variance that all instruction-tuned models agree on.

Fine-tuning from base + kernel rather than base alone means starting closer to the target. The kernel pre-applies the consensus transformation, and the remaining fine-tuning budget is spent entirely on task-specific specialization (the 26–32% residual) rather than rediscovering shared structure. This is directly analogous to how pretrained embeddings reduced training time in early transfer learning: the shared component is amortized across the community, and each new training run pays only for what is genuinely novel.

The implications compound at ecosystem scale. Every publicly released fine-tune contributes data points to the manifold. Larger N produces higher-fidelity kernels (more robust PC1 estimation, better noise suppression). The kernel improves as the open-source fine-tuning ecosystem grows, creating a flywheel: more fine-tunes → better kernel → cheaper future fine-tuning → more fine-tunes. This is training cost amortized across the entire community of practitioners—a shared asset that no single organization needs to produce alone.

For domain adaptation specifically, a two-stage approach becomes natural: (1) apply the general instruction-following kernel as initialization, (2) fine-tune on domain-specific data. Stage 1 is free—the kernel is pre-computed. Stage 2 requires less compute because the model is already instruction-aligned.

An important distinction: direct kernel application at inference time (our experimental conditions C1–C5) and kernel-as-checkpoint for further training are different use cases with different failure modes. C1 (self-kernel) degrades WikiText by 70.7% when applied as a permanent weight modification, because it amplifies architecture-specific noise without subsequent optimization to correct errors. As a training *initialization*, the same kernel places the model in the right neighborhood, and the optimizer refines away from the noise during fine-tuning. The checkpoint hypothesis predicts that fine-tuning from base + kernel converges faster and to a better optimum than fine-tuning from base alone—a prediction we leave to future experimental validation.

The cross-architecture transfer results (§4) suggest a further non-obvious prediction: cross-family kernels may be *better* warm-starts than self-family kernels, because they inject instruction signal from an orthogonal noise basis. The Mistral→Llama kernel (C2: +7.7% WikiText, −14.5% instruct) outperforms the Llama self-kernel (C1: +70.7% WikiText, −8.4% instruct) at inference time. If this advantage carries over to the checkpoint setting, optimal fine-tuning initialization would use kernels derived from *other* architecture families rather than one's own—a counterintuitive but testable claim.

5.6 Connections to Platonic Representation Hypothesis

Our three-part decomposition provides quantitative structural evidence consistent with partial convergence as predicted by the Platonic Representation Hypothesis. Across the two families tested, routing modification is correlated (attention, r=0.71) while knowledge stores diverge (MLP, $r\approx0$). The shared importance curve (r=0.87) suggests that the *structure* of where convergence occurs is itself consistent, even where the content diverges. These observations are based on two architecture families at 7–8B scale; broader validation across scales and architectures is needed before stronger claims of universality.

This resolves an apparent tension: models can be simultaneously similar (in routing) and different (in knowledge) because these components occupy distinct functional roles.

5.7 Limitations

Several limitations constrain interpretation:

1. **Sample size.** N=5 (N=4 for Llama cross-architecture and Qwen) per family is small for robust PCA estimation. Jackknife analysis (§4) confirms this concern: while three of five Mistral fine-tunes define a stable core direction (LOO cosine >0.95), two peripheral fine-tunes shift the PC1 direction substantially when removed. The shared structure is real but N=5 is at the lower bound for robust estimation—peripheral members have disproportionate influence. Larger N (e.g., N=15–20) would allow the core consensus to dominate. We report point estimates throughout; readers should interpret the specific percentages (e.g., 73.8%, 68.8%) as indicative rather than precise.

2. **Benchmark scope.** MMLU evaluation uses 4 of 57 subjects (100 questions total), producing error bars of approximately ±5–10% at these sample sizes. Full MMLU, MT-Bench, AlpacaEval, GSM8K, and HumanEval would be needed to distinguish genuine capability improvement

from surface-level style mimicry. The instruction perplexity metric measures distributional alignment with instruction-formatted text, not task completion ability.

3. **MCMC approximation.** The MH procedure optimizes a coherence proxy (PC1 variance × cosine similarity × SNR), not actual model capability per step. Full capability-based MCMC—loading the weighted kernel, applying it to the model, and running WikiText + MMLU + instruction evaluation as the acceptance criterion—would require approximately 3 minutes of GPU compute per proposal. At 5,000 proposals with 1,000 burn-in, this amounts to ∼250 GPU-hours per kernel construction, making it computationally prohibitive at current eval speeds. The proxy correlates with quality (73% noise reduction vs uniform weighting) but cannot capture nonlinear interactions between tensor weights. Quantized evaluation or distilled proxy models could make capability-based MCMC feasible in future work.

4. **Scale dependence.** All experiments are at 7–8B parameters. Whether the manifold exists at larger scales (70B+) is an open question.

5. **Instruction-tuning specificity.** All fine-tunes tested are instruction-tuned variants. Whether the manifold generalizes to other fine-tuning objectives (domain adaptation, task-specific) requires further investigation.

6 Future Work

Expanded benchmarks. Full 57-subject MMLU, HellaSwag, ARC-Challenge, IFEval, and MT-Bench evaluation at all kernel scales.

Expanded sample size. Jackknife analysis reveals a core-periphery structure at N=5, with peripheral fine-tunes exerting disproportionate influence. Expanding to N=10–15 for at least one family (Llama-3.1-8B has sufficient diversity) would test whether the core consensus stabilizes and peripheral influence diminishes as predicted. Bootstrap confidence intervals on PC1 fractions and permutation tests for PC1 dominance significance would further quantify robustness.

Peripheral drift mitigation at low N. The core-periphery structure identified by jackknife analysis suggests that robust kernel construction at small sample sizes requires biasing techniques to reduce the influence of outlier fine-tunes. We organize candidate approaches by computational cost:

Tier 1—No additional compute: (a) *Cosine-similarity gating*, where fine-tunes with mean pairwise cosine below a threshold are excluded from kernel construction entirely, using the quality diagnostic as a preprocessing filter. Neural-chat's low pairwise cosine would gate it out before it contaminates the kernel. (b) *Geometric median of deltas* (Weiszfeld's algorithm) instead of arithmetic mean—the geometric median is inherently resistant to outlier pull in high-dimensional spaces and requires no hyperparameter tuning.

Tier 2—Cheap iterative compute: (c) *Robust PCA variants* such as iteratively reweighted PCA that downweight points far from the emerging consensus, converging in 2–3 iterations. (d) *LOO consensus weighting*, where each fine-tune's contribution is scaled inversely by its jackknife influence on PC1 direction—directly closing the loop from the diagnostic to a correction step. (e) *Trimmed mean / Winsorization*, ranking deltas by projection onto PC1 and dropping the most extreme before kernel construction.

Tier 3—Structural: (f) *Hierarchical kernel construction*, where fine-tunes are first clustered by similarity and the kernel is built from cluster centroids rather than individual deltas, naturally compressing peripheral variance into the cluster structure.

These techniques could make kernel construction reliable at N=5–7 rather than requiring N=15–20, substantially reducing the data requirements for practical deployment. However, the

fundamental solution remains larger sample sizes where the consensus emerges organically and outliers are outvoted without correction.

Cross-scale analysis. Comparing 7B and 70B manifolds within the same family to test scale invariance of the shared structure.

Specialist kernel composition. Constructing domain-specific kernels (math, code, language) and testing composability—whether multiple specialist kernels can be combined without interference.

Capability-based MCMC. Replace the coherence proxy with actual benchmark evaluation per MH step, enabled by efficient quantized evaluation.

Non-instruction objectives. Test manifold existence for domain adaptation, code generation, and mathematical reasoning fine-tunes.

7 Conclusion

Fine-tuning of large language models produces weight perturbations with recoverable shared geometric structure. A dominant principal component explains 68–74% of variance across independently trained fine-tunes, replicating across architecturally distinct model families. Kernels constructed from this shared direction improve instruction-following capability while maintaining general language modeling, and transfer across architecture boundaries via sign-and-norm alignment (a tractable approximation to Procrustes).

The fine-tuning manifold decomposes into three interpretable components: universal routing, architecture-specific knowledge, and a universal importance curve. This decomposition provides the first quantitative structural evidence for partial convergence in the Platonic Representation Hypothesis and explains both the successes and failures of existing model merging techniques.

Manifold geometry further serves as a diagnostic tool, detecting degenerate fine-tuning invisible to standard benchmarks. These findings suggest that the weight-space of fine-tuned language models is far more structured than previously recognized, with immediate implications for model merging, transfer learning, and training efficiency.

A Fine-Tune Details

A.1 Mistral-7B Family

Model	Source	Method	Mean $\|\Delta\|$	Max $\|\Delta\|$	Min $\|\Delta\|$
OpenHermes-2.5-Mistral-7B	Teknium	SFT	2.68	4.54	0.73
zephyr-7b-beta	HuggingFace	DPO	2.73	4.62	0.72
Nous-Hermes-2-Mistral-7B-DPO	NousResearch	SFT+DPO	2.68	4.54	0.73
neural-chat-7b-v3-3	Intel	SFT	3.01	4.68	0.89
CollectiveCognition-v1.1-Mistral-7B	Teknium	SFT	2.81	4.72	0.78

Table 8: Mistral-7B fine-tune weight delta statistics.

Mistral fine-tunes show remarkably uniform delta norms (2.68–3.01), suggesting consistent training intensity across organizations.

A.2 Llama-3-8B Family

Llama fine-tunes show higher variance than Mistral. Dolphin is an outlier with mean delta norm 2× the family average and a maximum of 63.67—consistent with its reputation as an aggressive

Model	Source	Method	Mean $\|\Delta\|$	Max $\|\Delta\|$	Min $\|\Delta\|$
Meta-Llama-3-8B-Instruct	Meta	SFT+RLHF	2.60	4.68	0.73
dolphin-2.9-llama3-8b	Cognitive Comp.	SFT	5.48	63.67	1.47
Hermes-2-Pro-Llama-3-8B	NousResearch	SFT	2.33	21.37	0.71
Llama-3-Smaug-8B	AbacusAI	SFT+DPO	2.74	5.42	0.76

Table 9: Llama-3-8B fine-tune weight delta statistics.

fine-tune. Hermes-2-Pro has a modest mean but an outsized max (21.37), indicating concentrated heavy modification in specific tensors.

A.3 Qwen-2.5-7B Family

Model	Source	Method	Mean $\|\Delta\|$
Qwen2.5-7B-Instruct	Qwen	SFT	1.40 (degenerate)
Qwen2.5-Coder-7B-Instruct	Qwen	SFT	$\sim$95
Qwen2.5-Math-7B-Instruct	Qwen	SFT	$\sim$103
Qwen2.5-Coder-7B	Qwen	SFT	$\sim$98

Table 10: Qwen-2.5-7B fine-tune weight delta statistics.

The two-order-of-magnitude gap between Qwen Instruct (1.40) and the specialist fine-tunes (95–103) is immediately visible in delta norms—the most direct evidence of degenerate fine-tuning.

B Phase 2 Decomposition (N=2, Mistral)

Early N=2 decomposition using only OpenHermes and Dolphin variants found 42.25% shared fraction ($\pm$0.006), evenly distributed across layers and component types. V-projections showed highest sharing (0.428), Q/K lowest (0.416)—consistent with the interpretation that content (V) is more shared than routing (Q/K) even within a family.

C Per-Layer MCMC Weights

Mistral-7B. Mean weight: 0.509 (range 0.167–0.839). Peak layers: 24 (0.634), 12 (0.580), 10 (0.564). Trough: layer 1 (0.167 for attn_o). Highest individual tensors: layers.15.mlp.gate_proj (0.839), layers.10.mlp.down_proj (0.822), layers.30.self_attn.k_proj (0.821).

Llama-3-8B. Mean weight: 0.503 (range 0.419–0.564). Flatter distribution than Mistral. Peak layers: 6 (0.555), 15 (0.564). Trough: layer 9 (0.419).

D Heretic Ablation Details

Safety ablation following rank-1 direction removal. Three Mistral variants tested. All pairwise similarity deltas below 0.001. Confirms RLHF safety is orthogonal to fine-tuning capability structure captured by MPPPCA decomposition.

References

[1] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[2] Andy Arditi, Oscar Obeso, Buck Shlegeris, and Daniel Hernandez. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

[3] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meade, Hailey Zeng, and Jacob Solawetz. Arcee's MergeKit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*, 2024.

[4] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[5] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[6] Mitchell Ostrow et al. SHARE: Shared low-rank subspace in neural network weight spaces. *arXiv preprint*, 2026.

[7] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[8] 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. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.