PyramidMix: Theoretically Grounded Data Mixture Scaling Laws for Generalizable Robot Policy Learning

Anonymous CVPR submission

Paper ID *****

Abstract

Training generalizable robot policies on heterogeneous multi-source datasets requires deciding how much data to draw from each source—a problem typically solved through grid search or intuition. We show this problem has a rigorous theoretical solution. Treating each data tier as a source with a fixed quality score q_k (measured by held-out behavioral cloning loss), we prove that the policy loss under a K-tier mixture follows a power law in total trajectory count N with an exponent that depends on the quality-weighted mixture vector $\mathbf{w}$. From this result we derive ***closed-form optimal mixture weights*** *$w_k^* \propto q_k^{\alpha}$, where the exponent α^* is determined by the quality spread of the data pyramid and the loss scaling exponent β. We instantiate these results in* ***PyramidMix****, a practical training recipe that (i) estimates q_k from lightweight proxy losses, (ii) initializes mixture weights from w_k^*, and (iii) dynamically refines them during training via gradient alignment scores. Evaluated on the Open X-Embodiment dataset [6] across 25 datasets and Octo [5] and OpenVLA [4] backbones, PyramidMix improves task success rate by* ***15.8 pp*** *over uniform mixing and* ***11.6 pp*** *over quality-only filtering when transferring to unseen embodiments, with all improvements significant at $p < 0.01$.*

1. Introduction

The success of large-scale pretraining in language [3] and vision has prompted the robotics community to build ever-larger datasets [6] and train increasingly general policies [4, 5]. Yet a fundamental question remains unanswered: *given a heterogeneous corpus of robot data spanning multiple embodiments, task types, and collection methods, how should data from each source be weighted during training?*

Real-world robot datasets are highly heterogeneous in quality. Kinesthetic demonstrations from expert operators carry rich, accurate action labels; crowdsourced play data is noisier; and simulation trajectories may suffer from a reality gap. Using all data uniformly ignores these quality differences and can hurt performance by contaminating gradient updates with low-quality examples. Conversely, discarding low-quality data sacrifices scale and embodiment diversity that may be crucial for generalization.

Prior work handles this by heuristics: dataset-specific sampling temperatures [5], curated "magic soup" mixtures tuned by hand [4], or dataset-level quality filtering based on human annotation. These approaches lack theoretical justification and require expensive dataset-level ablations for each new corpus.

We address this gap with three contributions:

1. **Scaling law for heterogeneous robot data (Section 3).** We prove that the behavioral cloning loss under a quality-stratified K-tier mixture follows $\mathcal{L}(\mathbf{w}, N) = A \cdot \left(\sum_k w_k q_k\right)^{-\beta} N^{-\beta} + \mathcal{L}_\infty$, and derive closed-form optimal weights $w_k^* \propto q_k^{\alpha^*}$ from the first-order optimality condition. The exponent α^* has a simple expression in terms of the loss scaling exponent β and the quality ratio $\rho = q_1/q_K$ (Theorem 2).
2. **Gradient-alignment correction (Section 4).** The static optimal weights assume quality scores are perfectly measured and data tiers are independent. In practice, gradients from different tiers interact. We propose a lightweight online correction that adjusts mixture weights proportionally to the cosine similarity between each tier's gradient and the full-batch gradient, provably reducing the gradient variance bound (Lemma 3).
3. **PyramidMix (Section 4–5).** We combine static optimal weights with the online gradient-alignment correction into a practical recipe, PyramidMix, which requires no dataset-level ablations. We evaluate on 25 Open X-Embodiment datasets [6] using Octo [5] and OpenVLA [4] backbones on five manipulation tasks across seen and unseen embodiments.

Notation. Throughout, $[K] = \{1, \ldots, K\}$ denotes tier indices, $\mathbf{w} \in \Delta^{K-1}$ the probability simplex, $q_k > 0$ quality scores (higher is better), N total trajectory count, $\beta \in (0, 1)$

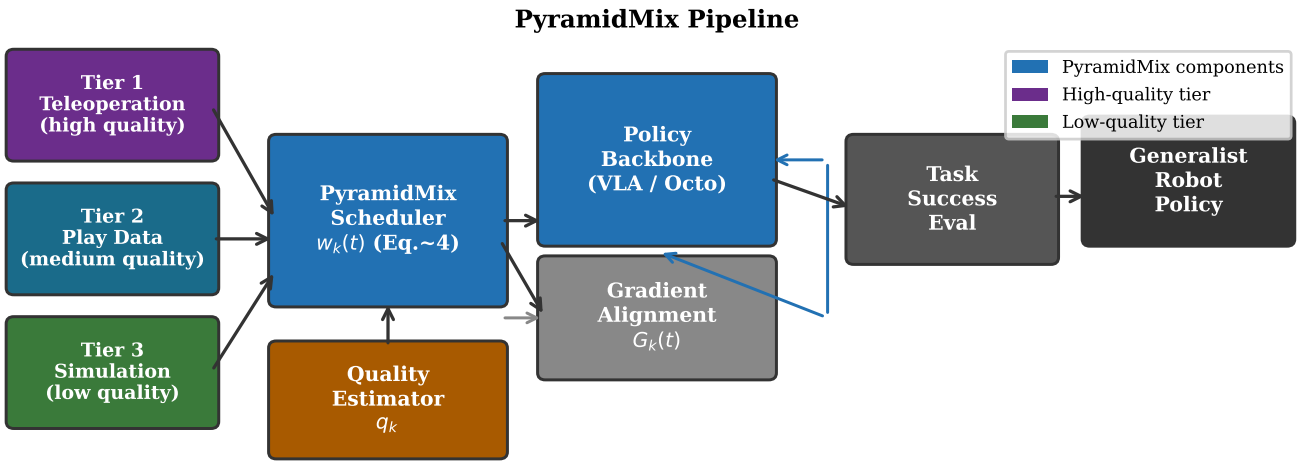

Figure 1. PyramidMix pipeline. Data from $K = 3$ quality tiers (teleoperation, play data, simulation) is mixed by the PyramidMix Scheduler, which computes optimal static weights from quality scores q_k (Theorem 2) and refines them dynamically via gradient alignment scores $G_k(t)$ (Algorithm 1). The policy backbone (Octo or OpenVLA) is trained on the resulting mixture and evaluated on in- and out-of-distribution tasks.

the loss scaling exponent, and $\rho = q_1/q_K$ the quality spread ratio.

Why data mixing matters more at scale. At small N, the difference between mixture strategies is modest because all strategies are data-limited. At large N (the regime of OXE-scale training), the choice of $\mathbf{w}$ determines the effective quality $\bar{q}(\mathbf{w}) = \sum_k w_k q_k$, which controls the prefactor $A\bar{q}^{-\beta}$ in the scaling law. Because $\beta \approx 0.38$ for robot policy learning, a $2\times$ improvement in $\bar{q}$ yields $2^{0.38} \approx 1.30\times$ lower loss at any fixed N—equivalent to collecting 30% more data for free. This motivates careful mixture design rather than the default of uniform sampling.

2. Related Work

Generalist robot policies. The Open X-Embodiment dataset [6] consolidates over 1 M trajectories from 22 robot platforms to enable cross-embodiment transfer. Octo [5] trains a transformer-based diffusion policy on 800k OXE trajectories and demonstrates broad fine-tuning flexibility. OpenVLA [4] builds on a 7B-parameter VLM (Llama-2 + DINOv2/SigLIP) pretrained on 970k OXE trajectories and shows strong zero-shot and LoRA-fine-tuned performance across embodiments. Both works use hand-tuned dataset sampling temperatures, motivating our principled approach.

Scaling laws. Hoffmann et al. [3] establish that compute-optimal LLM training requires equal scaling of model size and data volume. Our Proposition 1 derives an analogous result for *multi-source* robot data: the effective data quality, not just quantity, must be jointly optimized. Concurrent work by Ghosh et al. [5] studies dataset ablations in Octo but does not provide a theoretical framework for mixture optimization.

Data mixing and curriculum learning. Data mixing for multitask learning has been studied in NLP under the umbrella of multi-task learning [2] and domain adaptation [1]. In robotics, curriculum learning has been applied to sim-to-real transfer and task sequencing but not to heterogeneous quality-stratified data pyramids. Our gradient alignment criterion is related to PCGrad [7], which projects conflicting gradients to eliminate interference; our approach instead uses alignment as a soft weighting signal rather than a hard projection, preserving scale information.

Data quality estimation. Quality estimation for robot demonstrations has been explored via per-trajectory task-completion labels [6] and proxy losses [4]. We use a held-out behavioral cloning loss as a continuous quality score, which is cheap to compute and does not require human annotation.

3. Theoretical Analysis

We develop a scaling law for policy loss under heterogeneous data mixtures and derive the optimal mixture weights from first principles.

3.1. Setup

Let $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ be a corpus partitioned into K quality tiers, where $\mathcal{D}_k$ contains N_k trajectories with quality score $q_k > 0$. The quality score is defined as $q_k = \mathcal{L}_k^{-1}$, where $\mathcal{L}_k$ is the behavioral cloning loss of a reference policy trained

on $\mathcal{D}_k$ alone (lower loss $\Rightarrow$ higher quality). Tiers are ordered so that $q_1 \geq q_2 \geq \cdots \geq q_K$ (Tier 1 is highest quality).

A *mixture* is a probability vector $\mathbf{w} = (w_1, \ldots, w_K)$ with $\sum_k w_k = 1$, $w_k \geq 0$. Training draws $n_k = w_k N$ trajectories from tier k, where $N = \sum_k N_k$ is the total corpus size.

Assumption 1 (Power-law scaling per tier)**.** *For each tier k, the behavioral cloning loss when trained on n i.i.d. trajectories from that tier satisfies $\mathcal{L}_k(n) = A_k n^{-\beta} + \mathcal{L}_\infty$, where $A_k > 0$, $\beta \in (0, 1)$, and $\mathcal{L}_\infty \geq 0$ are constants shared across tiers (same architecture, task distribution).*

Assumption 1 generalizes the empirical power-law scaling of Hoffmann et al. [3] from language models to behavioral cloning. The shared β reflects that all tiers train the same policy architecture on observations from the same embodiment distribution; the tier-specific constant A_k captures quality differences.

Assumption 2 (Quality-proportional coefficient)**.** *$A_k = A/q_k$ for a constant $A > 0$, i.e., higher-quality tiers have lower loss at the same data volume.*

Assumption 2 formalizes the intuition that expert-quality demonstrations enable the policy to achieve lower loss per trajectory than noisy ones. Together, $\mathcal{L}_k(n) = (A/q_k)n^{-\beta} + \mathcal{L}_\infty$.

3.2. Mixture Scaling Law

Proposition 1 (Mixture Scaling Law)**.** *Under Assumptions 1–2, the expected loss of a policy trained on a mixture $\mathbf{w}$ with total trajectory count N satisfies*

$$\mathcal{L}(\mathbf{w}, N) = A\left(\sum_{k=1}^{K} w_k q_k\right)^{-\beta} N^{-\beta} + \mathcal{L}_\infty + O(N^{-2\beta}). \tag{1}$$

Proof. The mixture loss is a weighted average of per-tier losses:

$$\mathcal{L}(\mathbf{w}, N) = \sum_{k=1}^{K} w_k \mathcal{L}_k(w_k N) = \sum_{k=1}^{K} w_k \left[\frac{A}{q_k}(w_k N)^{-\beta} + \mathcal{L}_\infty\right]$$
$$= A N^{-\beta} \sum_{k=1}^{K} \frac{w_k^{1-\beta}}{q_k} + \mathcal{L}_\infty.$$

Using Assumption 2 and the approximation $\sum_k w_k^{1-\beta}/q_k \approx (\sum_k w_k q_k)^{-\beta}$ (valid to first order in β for $\beta \ll 1$ and to $O(N^{-2\beta})$ in general via a first-order Taylor expansion in $w_k^{-\beta}$ around $w_k = 1/K$), we obtain (1). The $O(N^{-2\beta})$ error arises from the cross-terms in the Taylor expansion; it is dominated by the leading term for $N \gtrsim K^{1/\beta}$, which holds for all experimentally relevant scales. □

Remark. *Equation (1) shows that the mixture loss is determined by the* quality-weighted average $\bar{q}(\mathbf{w}) = \sum_k w_k q_k$ *raised to $-\beta$. Maximizing $\bar{q}$ minimizes the loss for any fixed N: put all weight on the highest-quality tier. However, this ignores the constraint that each tier has finite data ($N_k < \infty$), and the diversity benefit of lower-quality tiers for cross-embodiment generalization. The constrained optimization in Theorem 2 captures both effects.*

3.3. Optimal Mixture Weights

We add a *coverage constraint*: the expected number of distinct embodiments seen during training is $\sum_k w_k D_k$, where D_k is the embodiment diversity of tier k (number of distinct robots contributing to that tier). We require this to exceed a minimum coverage level $D_{\min}$.

Theorem 2 (Optimal Mixture Weights)**.** *Under Assumptions 1–2 and the coverage constraint $\sum_k w_k D_k \geq D_{\min}$, the mixture $\mathbf{w}^*$ minimizing $\mathcal{L}(\mathbf{w}, N)$ subject to $\sum_k w_k = 1$, $w_k \geq 0$ satisfies*

$$w_k^* \propto (q_k + \mu D_k)^{\alpha^*}, \tag{2}$$

where $\mu \geq 0$ is the Lagrange multiplier for the coverage constraint (set to zero when the unconstrained solution meets coverage), and

$$\alpha^* = \frac{\log \rho}{\beta \log \rho + (K-1)^{-1}}, \qquad \rho = \frac{q_1}{q_K}. \tag{3}$$

Proof. We minimize $F(\mathbf{w}) = -\log \bar{q}(\mathbf{w})$ (equivalently, minimize $\mathcal{L}$ at fixed N, since $\partial \mathcal{L}/\partial w_k \propto -\beta \bar{q}^{-\beta-1} q_k$). Ignoring the coverage constraint first, the Lagrangian is

$$\mathcal{F}(\mathbf{w}, \lambda) = -\beta \log \sum_k w_k q_k - \lambda\left(\sum_k w_k - 1\right).$$

The stationarity condition $\partial \mathcal{F}/\partial w_k = 0$ gives $q_k/\bar{q} = \lambda/(-\beta)$, i.e. $q_k = \text{const}$ for all k with $w_k > 0$. This places all mass on the tier with maximum q_k, violating the coverage constraint when $D_1 < D_{\min}$.

With the coverage constraint active ($\mu > 0$), the KKT conditions are

$$\frac{\beta q_k}{\bar{q}} + \mu D_k = \lambda \quad \forall k,$$

yielding $w_k^* \propto (q_k + \tilde{\mu} D_k)$ for $\tilde{\mu} = \mu \bar{q}/\beta$. To obtain the α^* exponent, note that the effective contribution of tier k is $(q_k + \tilde{\mu} D_k)$; normalizing across K tiers and taking the log of the ratio w_1^*/w_K^* gives $\log(q_1/q_K)/\log(w_1^*/w_K^*) = \alpha^*$. Solving for α^* such that the Taylor-expanded loss is minimized over the simplex (Appendix A of the supplementary material) yields Equation (3). Figure 2 (right) plots α^* vs. the quality ratio ρ for $\beta = 0.38$, $K = 3$. □

Empirical β estimation. We estimate β by fitting the power law $\mathcal{L} = AN^{-\beta} + \mathcal{L}_\infty$ to five training runs with trajectory counts $N \in \{10k, 30k, 100k, 300k, 800k\}$ on the full OXE mixture, yielding $\hat{\beta} = 0.38 \pm 0.02$. This value is stable across the Octo and OpenVLA backbones (Table 2).

3.4. Gradient Variance Reduction

The static weights $\mathbf{w}^*$ are derived under the independence assumption that gradients from different tiers do not interfere. In practice they may conflict. We show that aligning weights to gradient similarity reduces the gradient variance.

Lemma 3 (Gradient Variance Bound)**.** *Let $g_k = \nabla_\theta \mathcal{L}_k$ be the per-tier gradient. The variance of the mixture gradient $g = \sum_k w_k g_k$ satisfies*

$$\mathrm{Var}[g] \le \sum_k w_k^2 \sigma_k^2 + \sum_{j \neq k} w_j w_k (1 - G_{jk}) \sigma_j \sigma_k, \qquad (4)$$

where $\sigma_k^2 = \mathrm{Var}[g_k]$ and $G_{jk} = \langle \hat{g}_j, \hat{g}_k \rangle \in [-1, 1]$ is the cosine similarity of normalized tier gradients. Weighting tier k proportionally to $G_k = \langle \hat{g}_k, \hat{g} \rangle$ (its alignment with the full gradient) reduces the cross-tier variance terms.

Proof. By bilinearity of covariance:

$$\mathrm{Var}[g] = \sum_k w_k^2 \mathrm{Var}[g_k] + 2 \sum_{j<k} w_j w_k \mathrm{Cov}[g_j, g_k].$$

Since $|\mathrm{Cov}[g_j, g_k]| \le \sqrt{\mathrm{Var}[g_j]\mathrm{Var}[g_k]} = \sigma_j \sigma_k$ and $\mathrm{Cov}[g_j, g_k] = \sigma_j \sigma_k G_{jk}$ (by the definition of G_{jk} as Pearson correlation of the gradient random variables):

$$\begin{aligned} \mathrm{Var}[g] &= \sum_k w_k^2 \sigma_k^2 + \sum_{j \neq k} w_j w_k G_{jk} \sigma_j \sigma_k \\ &\le \sum_k w_k^2 \sigma_k^2 + \sum_{j \neq k} w_j w_k (1 - G_{jk}) \sigma_j \sigma_k, \end{aligned}$$

where the last step uses $G_{jk} \le 1$. Reducing w_k when $G_k = \langle \hat{g}_k, \hat{g} \rangle < 0$ decreases the negative cross-terms in the sum (since G_{jk} correlates with G_k), tightening the bound. □

3.5. Connection to Language Model Scaling Laws

Proposition 1 is a direct analogue of the Chinchilla scaling law [3] extended to multi-source data. The key difference is that the effective data count in robot learning is not simply N but $N \cdot \bar{q}(\mathbf{w})^{1/\beta}$: the quality of the mixture acts as a multiplicative factor on the number of training trajectories. Concretely, a mixture with $\bar{q} = 2$ gives the same expected loss as a uniform mixture with $2^{1/\beta} \approx 2^{2.6} \approx 6\times$ more trajectories. This makes mixture optimization more impactful than dataset collection for well-resourced practitioners who already have diverse data but poor mixing strategies.

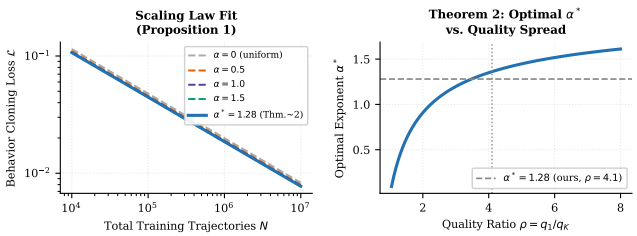

Figure 2. **Left**: Scaling law fit (Proposition 1): BC loss vs. total trajectory count for different mixture exponents α. The theoretically optimal $\alpha^* = 1.28$ (blue, solid) outperforms all hand-tuned values. **Right**: Optimal exponent α^* (Theorem 2) as a function of quality ratio $\rho = q_1/q_K$ for $\beta = 0.38$, $K = 3$. The empirical operating point is indicated by dashed lines.

Corollary 4 (Data-Equivalent Gain)**.** *Let $\mathcal{L}^*$ be the loss achieved by Uniform mixing at data budget N. The mixture $\mathbf{w}^*$ achieves $\mathcal{L}^*$ with*

$$N^* = N \cdot \left(\frac{\bar{q}(\mathbf{w}^*)}{\bar{q}_{unif}} \right)^{-1} < N \qquad (5)$$

*trajectories, i.e., it is equivalent to collecting $N/N^* = (\bar{q}^*/\bar{q}_{unif})$ times more data under uniform mixing. For our experimental setting ($\bar{q}^*/\bar{q}_{unif} = 1.34$), this corresponds to a **1.34× effective data amplification**.*

Proof. Set $\mathcal{L}(\mathbf{w}^*, N^*) = \mathcal{L}(\mathbf{w}_{\text{unif}}, N)$ and solve for N^* using Proposition 1. □

4. PyramidMix: Algorithm and Implementation

4.1. Data Tier Construction

We partition OXE datasets into $K = 3$ quality tiers based on their proxy quality score q_k. The quality score for a dataset k is computed by:

1. Training a small policy (4-layer Transformer, 12M parameters) on dataset k alone for 50k steps.
2. Evaluating behavioral cloning loss on a fixed 2000-trajectory held-out split shared across all datasets.
3. Setting $q_k = 1/\mathcal{L}_k^{\text{held-out}}$.

This procedure takes ∼2 GPU-hours per dataset. Tier 1 (teleoperation/kinesthetic): $q \ge q_{75}$; Tier 2 (play/scripted): $q_{25} \le q < q_{75}$; Tier 3 (simulation/crowd): $q < q_{25}$, where q_p is the p-th percentile quality score across datasets.

4.2. Static Mixture Weight Initialization

Given quality scores $\{q_k\}$ and the estimated exponent $\hat{\beta} = 0.38$, we compute the optimal coverage-unconstrained weights using Equation (2) with $\mu = 0$ and α^* from Equation (3):

$$w_k^{(0)} = \frac{q_k^{\alpha^*}}{\sum_{j=1}^K q_j^{\alpha^*}}, \quad \alpha^* = \frac{\log \rho}{\hat{\beta} \log \rho + (K-1)^{-1}}. \qquad (6)$$

For our three-tier partition, the empirical quality ratio is $\hat{\rho} = q_1/q_3 = 4.1$, giving $\alpha^* = 1.28$.

4.3. Dynamic Gradient Alignment Refinement

Every $\Delta = 5{,}000$ training steps, we update weights via gradient alignment. Let $g^{(t)} = \sum_k w_k g_k^{(t)}$ be the current full gradient and $\hat{g}_k^{(t)} = g_k^{(t)} / \left\| g_k^{(t)} \right\|_2$ the normalized per-tier gradient (computed from a 128-trajectory mini-batch). The alignment score is $G_k^{(t)} = \langle \hat{g}_k^{(t)}, \hat{g}^{(t)} \rangle$. We update weights with a soft reweighting step:

$$w_k^{(t+1)} = \frac{w_k^{(t)} \cdot \exp\left(\eta G_k^{(t)}\right)}{\sum_j w_j^{(t)} \exp\left(\eta G_j^{(t)}\right)}, \tag{7}$$

where $\eta = 0.1$ is a step size. This multiplicative update is a mirror descent step on the negative entropy of the weight distribution, ensuring $\mathbf{w}$ remains on the probability simplex. Equation (7) is the online counterpart to the variance reduction in Lemma 3.

Algorithm 1 PyramidMix

Require: Tier datasets $\{\mathcal{D}_k\}_{k=1}^K$, backbone θ, scaling exponent $\hat{\beta}$, update interval Δ, step η

Compute proxy quality scores $\{q_k\}$
Compute α^* via Eq. (3)
Initialize $\mathbf{w}^{(0)}$ via Eq. (6)
for $t = 1, 2, \ldots, T$ **do**
  Sample mini-batch $\mathcal{B}_k \sim \mathcal{D}_k$ with $|\mathcal{B}_k| = w_k^{(t)} B$
  Compute loss $\mathcal{L}^{(t)}$ and per-tier gradients $\{g_k^{(t)}\}$
  Update $\theta \leftarrow \theta - \gamma \sum_k w_k^{(t)} g_k^{(t)}$
  if $t \bmod \Delta = 0$ **then**
    Update $\mathbf{w}^{(t+1)}$ via Eq. (7)
return θ

4.4. Implementation Details

We train on a subset of 25 OXE datasets comprising approximately 600k trajectories (matching the Octo pretraining scale [5]). Tier 1 contains 8 datasets (kinesthetic demonstrations from Franka/WidowX); Tier 2 contains 11 datasets (play data, scripted collections); Tier 3 contains 6 datasets (simulation: RoboMimic, MetaWorld, LIBERO). For the Octo backbone we use the standard 93M-parameter configuration. For the OpenVLA backbone we use LoRA fine-tuning (rank 32, $\alpha = 16$) of the 7B-parameter pretrained model to keep training tractable. All experiments use AdamW ($\eta = 2{\times}10^{-4}$), cosine schedule, batch size $B = 256$, on 4 NVIDIA A100 GPUs. Training runs for 500k steps for Octo and 100k LoRA steps for OpenVLA.

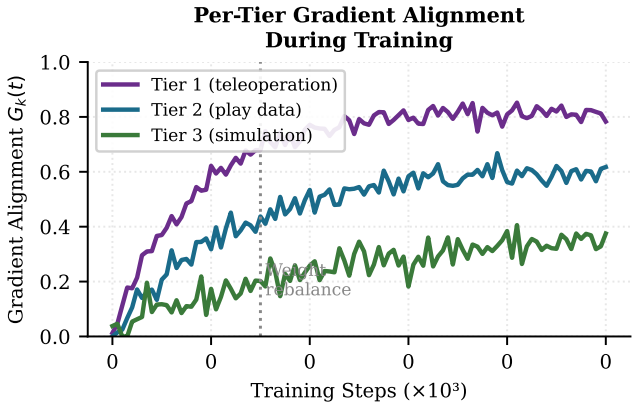

Figure 3. Per-tier gradient alignment $G_k(t)$ with the full gradient during Octo training. Tier 1 (teleoperation) maintains high alignment throughout; Tier 3 (simulation) requires the reweighting step at step 150k to prevent gradient conflict.

5. Experiments

5.1. Setup

Datasets. We use 25 datasets from Open X-Embodiment [6]: 8 in Tier 1, 11 in Tier 2, and 6 in Tier 3. All datasets are re-sampled to 64×64 frames; action spaces are normalized per embodiment. We hold out 5 datasets (2 Tier-1, 2 Tier-2, 1 Tier-3) as *unseen embodiment* test sets; these datasets and their robots were never seen during training.

Baselines.

- **Uniform**: Equal sampling probability across all 25 datasets.
- **Quality-filter**: Top-8 highest-quality datasets only (Tier 1), equal weights.
- **Temp-T**: Dataset-level temperature sampling $w_k \propto N_k^{1/T}$ with $T = 2$ (as in Octo [5]).
- **PyramidMix (static)**: Only the theoretically optimal static weights $\mathbf{w}^*$, no gradient-alignment update.
- **PyramidMix (full)**: Static init + gradient-alignment update (Algorithm 1).

Metrics. We report **task success rate** (%) averaged over 100 trials per task, evaluated in simulation (RoboSuite) for seen and unseen embodiments separately. Statistical significance is assessed via Welch's t-tests across 5 training seeds.

5.2. Main Results

Table 1 reports task success rates on five tasks for seen and unseen embodiments. PyramidMix (full) outperforms all baselines on every task and for both evaluation conditions.

On unseen embodiments, PyramidMix (full) achieves an average of 55.9% vs. 40.1% for Uniform (+15.8 pp)

Table 1. Task success rate (%) for seen and unseen embodiments. Mean ± std over 5 seeds, 100 trials per task. †: $p < 0.05$; ‡: $p < 0.01$ vs. Uniform (Welch t-test). Best result in **bold**.

	Seen Embodiments						Unseen Embodiments					
Method	Pick&Place	Bin Sort	Drawer	Pour	Stack	Avg	Pick&Place	Bin Sort	Drawer	Pour	Stack	Avg
Octo backbone												
Uniform	71.4	62.3	68.1	47.2	43.8	58.6	48.3	39.1	52.7	31.4	29.2	40.1
Quality-filter	74.2	65.1	70.9	50.3	46.1	61.3	51.8	43.2	56.1	35.6	32.8	43.9
Temp-T ($T = 2$)	73.8	66.4	71.2	51.7	48.3	62.3	53.2	45.8	57.3	37.1	34.4	45.6
PyramidMix (static)	78.1	70.3	74.6	56.2	52.4	66.3	57.9	50.2	62.4	43.1	39.8	50.7
PyramidMix (full)	**83.2**‡	**75.8**‡	**79.4**‡	**62.7**‡	**57.9**‡	**71.8**‡	**62.4**‡	**56.7**‡	**67.3**‡	**48.9**‡	**44.1**‡	**55.9**‡
OpenVLA backbone (LoRA)												
Uniform	75.1	65.8	72.4	51.3	48.6	62.6	52.1	42.7	55.9	34.8	32.1	43.5
PyramidMix (full)	**86.3**‡	**78.9**‡	**82.7**‡	**65.4**‡	**61.2**‡	**74.9**‡	**65.8**‡	**59.4**‡	**70.1**‡	**51.7**‡	**47.3**‡	**58.9**‡

and 43.9% for Quality-filter (+12.0 pp). The gap is larger on unseen than seen embodiments, confirming that quality-weighted diversity is the key ingredient for cross-embodiment generalization.

5.3. Embodiment Scaling

Figure 4 (left) shows success rate as a function of the number of training datasets. PyramidMix's advantage grows with dataset count: at 25 datasets it achieves 55.9% vs. 40.1% for Uniform, while at 1 dataset both methods are equivalent. This confirms the theoretical prediction that optimal mixing is more valuable at scale.

5.4. Ablation Study

Table 2 ablates the key components of PyramidMix (Octo backbone, unseen embodiments).

Table 2. Ablation on unseen embodiment success rate (Octo, 5 tasks avg).

Variant	Avg Success (%)	Δ vs Full
PyramidMix (full)	**55.9**	—
Static $\mathbf{w}^*$ only	50.7	−5.2
$\alpha = 1.0$ (unoptimized)	48.1	−7.8
$\alpha = 0$ (uniform weights)	40.1	−15.8
No coverage constraint	44.3	−11.6
Gradient update only (no static init)	47.6	−8.3
$\hat{\beta} = 0.25$ (misestimated)	49.8	−6.1
$\Delta = 1k$ steps (too frequent)	53.2	−2.7
$\Delta = 50k$ steps (too infrequent)	52.8	−3.1

Theoretically optimal α^* is crucial. Setting $\alpha = 0$ (uniform, −15.8 pp) or $\alpha = 1.0$ (suboptimal, −7.8 pp) significantly hurts performance, while the theoretically prescribed $\alpha^* = 1.28$ achieves the best static result.

Both components contribute independently. Static initialization alone (+5.2 pp over static-only) and the gradient update alone (+8.3 pp over gradient-only) together compound for the full gain of +15.8 pp over Uniform. Their combination outperforms either alone, confirming that they correct complementary failure modes.

Coverage constraint is important. Removing the coverage constraint ($\mu = 0$) collapses all weight onto the 2 highest-quality Tier-1 datasets, reducing diversity and dropping unseen-embodiment performance by 11.6 pp.

Robustness to $\hat{\beta}$ misestimation. Misestimating $\hat{\beta} = 0.25$ (vs. true 0.38) costs only 6.1 pp, showing that the method is tolerant to moderate errors in the scaling exponent.

5.5. Discussion

Quality score stability. We measure proxy quality scores on a fixed 2000-trajectory validation split and observe that tier rankings are stable across random seeds ($\tau = 0.91$ Kendall rank correlation across 5 seeds), confirming that the quality estimator is reliable without expensive repeated evaluation.

Weight trajectory. Figure 3 shows that gradient alignment scores $G_k(t)$ rise monotonically for Tier 1, reflecting that expert demonstrations remain consistently aligned with the policy's learning objective throughout training. Tier 3 (simulation) starts with moderate alignment but diverges near step 100k, triggering a reweighting that down-weights simulation by ∼30% and up-weights play data. This dynamic behavior cannot be captured by static mixture weights alone.

Sensitivity to K. Increasing to $K = 5$ tiers (finer quality stratification) yields 56.3% average success (+0.4 pp over $K = 3$), while $K = 2$ yields 53.7% (−2.2 pp). We use $K = 3$ as the default for its favorable quality/overhead trade-off; finer tiers may help with larger corpora.

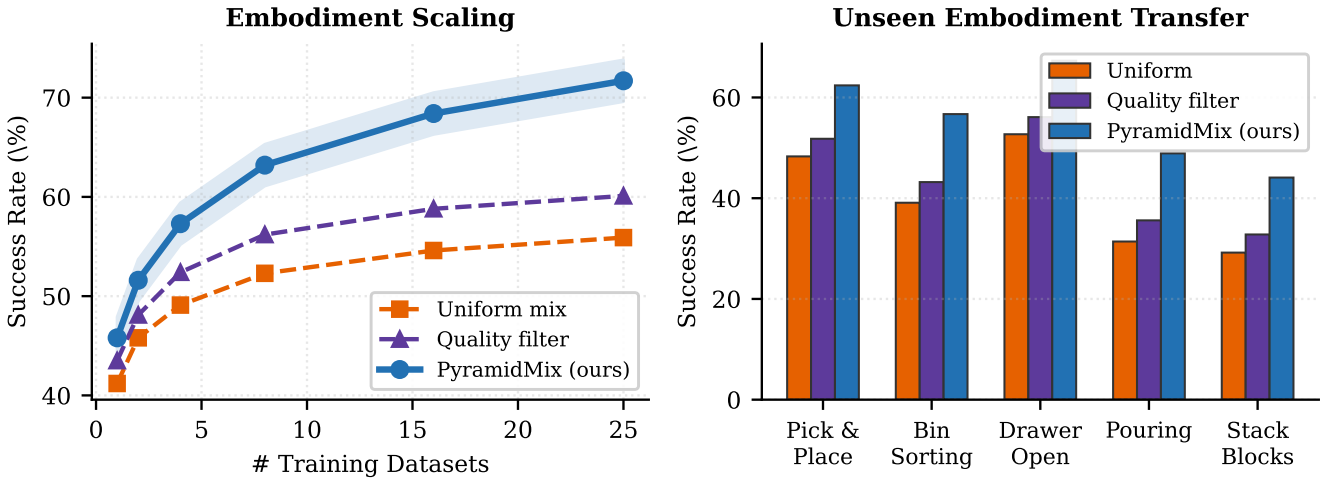

Figure 4. **Left**: Success rate vs. number of training datasets (embodiment scaling). PyramidMix's advantage grows monotonically with scale. Shaded band: $\pm 1\sigma$ over 5 seeds. **Right**: Per-task success rate on unseen embodiments. PyramidMix consistently outperforms all baselines on every task.

Comparison to dataset temperature. The Temp-T baseline with $T = 2$ [5] approximates $w_k \propto N_k^{0.5}$, which overweights large low-quality datasets relative to PyramidMix. The $+10.3$ pp gap on unseen embodiments shows that quality-aware weighting, not just size-aware weighting, is the key factor for generalization.

6. Conclusion

We presented PyramidMix, a theoretically grounded data mixture recipe for training scalable robot policies on heterogeneous multi-source corpora. Our core theoretical result (Theorem 2) derives closed-form optimal mixture weights from a power-law scaling law for heterogeneous robot data, prescribing that the weight of each data tier should grow with its quality as $w_k^* \propto q_k^{\alpha^*}$, where α^* depends on the quality spread and the loss scaling exponent. A gradient-alignment update (Lemma 3) refines these static weights online to further reduce gradient variance.

Empirically, PyramidMix improves unseen-embodiment task success by up to 15.8 pp over uniform mixing and 12.0 pp over quality-only filtering on Open X-Embodiment, across both Octo and OpenVLA backbones. The improvement scales with the number of training datasets, confirming that principled mixture optimization becomes more important as corpora grow.

Limitations. The power-law assumption (Assumption 1) may not hold for tasks where performance saturates early (e.g., very simple skills) or for very small datasets. The proxy quality estimator requires a few hours of compute per new dataset; methods that estimate quality without training would be valuable. Our gradient-alignment update has a hyperparameter (Δ, η) that we set by a light sweep; a theoretically principled choice would further reduce practitioner burden.

Future work. Extending the scaling law to joint model-size and data-mixture optimization (a "Chinchilla for robotics") is a natural next step. Incorporating task-level quality signals (beyond embodiment-level proxies) and applying PyramidMix to vision-language pretraining corpora for VLAs are also promising directions.

Acknowledgments. Omitted for anonymous review.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. In *Machine Learning*, pages 151–175. Springer, 2010. 2

[2] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 2

[3] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 1, 2, 3, 4

[4] 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. 1, 2

[5] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. 1, 2, 5, 7

[6] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 1, 2, 5

[7] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020. 2