TRAINING-FREE CROSS-ARCHITECTURE MERGING FOR GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Merging pretrained models without retraining is useful for deployment, but existing methods largely assume architecturally compatible networks. We study a more general graph setting: two class-specialized GNNs, trained on disjoint labeled node subsets of the same graph, may differ in architecture, depth, and width and must be merged without labels or gradient-based optimization. We introduce **MOSAIC** (Model Operator Space Alignment for Integration & Composition), which represents both GNNs in a shared Universal Message Passing Mixture (UMPM), a common basis of standard message-passing operators. MOSAIC matches layers with centered kernel alignment (CKA), aligns hidden representations with semi-orthogonal SVD maps, and estimates layer-wise interpolation weights from closed-form ridge reconstruction. A final Kernel Witness step—an MMD witness-function correction implemented with random Fourier features—reduces residual edge-message distribution shift. Across six node-classification benchmarks, MOSAIC preserves a large fraction of each parent's specialist accuracy for many matched-depth cross-architecture merges and yields a single merged GNN with $1.1\times$–$1.9\times$ inference speedup over two-parent ensembles.

1 INTRODUCTION

Training-free model merging aims to combine pretrained capabilities without optimizing a new model. For GNNs this is especially difficult: independently trained networks are misaligned not only in hidden coordinates but also in the *message-passing operations* that depend on graph topology. Graph-specific reuse work reports failures of topology-independent interpolation or alignment and motivates graph-aware message coordination or representation alignment (Jing et al., 2024; Garg et al., 2025); these methods assume homogeneous architectures or a different reuse setting.

We study the merging of class-specialized GNNs on a single transductive graph. Two parents f_A and f_B are trained on disjoint labeled node subsets of the same graph, using complementary 80/20 class-specialist splits following the split-data protocol of Ainsworth et al. (2023). They share the label space but may use different GNN families (GCN, GAT, GraphSAGE, or GIN), depths, and widths. At merge time, only the unlabeled graph, features, and parent forward passes are available. The goal is one merged GNN f_C with a single hidden-state stream that preserves each parent's held-out specialist performance; no labels, distillation, fine-tuning, or parallel parent inference are permitted.

Architectural heterogeneity creates two distinct incompatibilities. Even when two layers compute functionally similar representations, their hidden coordinates may be permuted, rotated, or dimension-mismatched. In addition, GNN families implement different graph operators, so corresponding weight tensors need not share parameter semantics. Rebasining and feature/width/depth alignment address coordinate or structural mismatch in conventional networks (Ainsworth et al., 2023; Stoica et al., 2024; Xu et al., 2024), but do not define how GCN normalization should compose with attention, mean aggregation, or GIN's coupled sum/self update. MOSAIC addresses these problems separately by first defining a common message-passing parameterization and then performing representation alignment and parameter fusion.

Contributions.

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.

- We formalize label-free, training-free merging of architecturally heterogeneous, class-specialized GNNs into a single model.
- We introduce UMPM, an exact reparameterization of the supported parent architectures in a shared message-passing basis, together with monotone CKA layer matching and semi-orthogonal hidden-space alignment for depth/width mismatch.
- We derive a closed-form, parent-specific reconstruction whose residuals determine layer-wise interpolation coefficients, including explicit handling of operators present in only one parent and label-free fusion of the output layer; the resulting model has one hidden-state stream rather than two parent streams.
- Across six node-classification benchmarks(and graph level benchmarks in the Appendix), we identify the regimes in which heterogeneous merging is reliable and the principal failure modes, and measure the deployment benefit of the resulting single merged model.

2 Related Work

Training-free merging and alignment. SWA and model soups average weights within compatible parameterizations (Izmailov et al., 2018; Wortsman et al., 2022); RegMean uses closed-form regression, task arithmetic combines task vectors, and TIES trims small updates and resolves sign conflicts (Jin et al., 2023; Ilharco et al., 2023; Yadav et al., 2023). Alignment-aware methods include Git Re-Basin, FedMA, OT-Fusion, and Sinkhorn rebasining (Ainsworth et al., 2023; Wang et al., 2020; Singh & Jaggi, 2020; Peña et al., 2023). ZipIt uses feature matching across tasks, AdaMerging learns merge coefficients, and TF-HMM addresses depth/width heterogeneity (Stoica et al., 2024; Yang et al., 2024; Xu et al., 2024); none is formulated around graph message-passing operators.

Graph model reuse. DistillGCN and AmalgamateGNN require additional training (Yang et al., 2020; Jing et al., 2021). Deep Graph Mating introduces message coordination and child-message calibration for learning-free GNN reuse (Jing et al., 2024); GNNMerge aligns node embeddings and provides an analytical variant (Garg et al., 2025). MOSAIC instead merges heterogeneous specialists with different message-passing rules, depths, and widths without labels or child optimization. Unlike output ensembling, it produces one merged GNN and does not retain both parent graphs at inference.

3 Preliminaries and Method

Let $G = (V, E)$ be a graph with $N = |V|$ nodes, node features $X \in \mathbb{R}^{N \times d_0}$, and K classes. We consider GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), mean-GraphSAGE (Hamilton et al., 2017), and GIN (Xu et al., 2019). Their relevant distinction for merging is the propagation rule: normalized neighborhood aggregation (GCN), learned edge weighting (GAT), separate self/neighbor-mean terms (GraphSAGE), or a coupled $(1 + \epsilon)$ self plus neighbor sum followed by an MLP (GIN), where ϵ is the GIN self-weight coefficient. MOSAIC represents these choices in one shared message-passing parameterization, then performs five deterministic phases: UMPM canonicalization (an exact reparameterization into the shared basis), layer matching, hidden-space alignment, parameter fusion, and Kernel Witness correction (an MMD witness-function step; Fig. 1).

3.1 Problem formulation

Let $f_P(\cdot; \theta_P)$, $P \in \{A, B\}$, denote the two frozen parent models, and let $\mathcal{T}_A, \mathcal{T}_B$ denote their disjoint labeled training subsets. The merging procedure must construct parameters for a single merged model,

$$\theta_C = \mathcal{M}(\theta_A, \theta_B; G, X, \mathcal{F}_A, \mathcal{F}_B), \tag{1}$$

where $\mathcal{F}_P$ denotes quantities obtainable from a parent forward pass (activations, messages, and predictions). Let $\mathcal{V}_{\text{cal}} \subseteq V$ denote the unlabeled nodes used to estimate merge statistics and $\mathcal{E}_{\text{cal}} \subseteq E$ the corresponding calibration edges. During merging, $\mathcal{M}$ may not access labels, $\mathcal{T}_A$, $\mathcal{T}_B$, or gradients with respect to merged-model parameters. It may solve deterministic closed-form subproblems, but performs no gradient-based optimization of f_C. At inference, only f_C is evaluated: parent activations or logits are not retained as parallel streams.

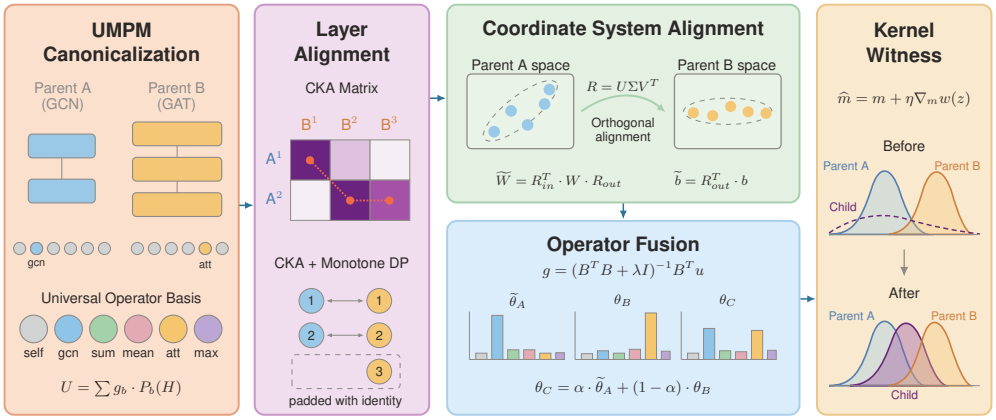

Figure 1: **Overview of MOSAIC.** Parents are reparameterized in UMPM, matched across layers with CKA, aligned in hidden space with semi-orthogonal SVD maps, fused using closed-form reconstruction and label-free interpolation, and corrected by Kernel Witness (an MMD witness-function step) to produce one merged GNN.

For evaluation only, let V_P be the held-out nodes associated with specialist P, and let $\mathrm{Acc}(f; V)$ denote classification accuracy of model f on node set V. We report $\mathrm{Ret}_P = \mathrm{Acc}(f_C; V_P)/\mathrm{Acc}(f_P; V_P)$ and focus on $\min(\mathrm{Ret}_A, \mathrm{Ret}_B)$, which penalizes a merge that preserves one specialist at the expense of the other. The class-specialist partition is therefore an evaluation device, not merge-time side information.

3.2 Universal Message Passing Mixture

UMPM uses the operator basis $\mathcal{B} = \{\mathrm{self}, \mathrm{gcn}, \mathrm{sum}, \mathrm{mean}, \mathrm{att}\}$. For layer input $H^{\ell-1}$, each basis operator $\mathcal{O}_b^\ell$ includes its feature transform and aggregation, $g_b^\ell \in \mathbb{R}$ is its scalar mixture coefficient, β^ℓ is the pre-update bias, and Ψ^ℓ denotes the architecture-specific post-aggregation update function, including any nodewise nonlinearity. For parent P, $\mathcal{B}_P^\ell \subseteq \mathcal{B}$ denotes the active basis elements at layer ℓ; inactive coefficients are zero:

$$U^\ell = \sum_{b \in \mathcal{B}} g_b^\ell \mathcal{O}_b^\ell(H^{\ell-1}) + \beta^\ell, \qquad H^\ell = \Psi^\ell(U^\ell). \tag{2}$$

For GCN, GAT, and GraphSAGE this directly selects the native message-passing computation. Standard GIN requires a coupled sum/self term because its MLP acts *after* self and neighbor terms are added. Let (W_1^ℓ, b_1^ℓ) denote GIN's first affine map, $\Psi_{\mathrm{tail}}^\ell$ its remaining nonlinear update, ϵ^ℓ its self-weight coefficient, and γ^ℓ the shared coefficient (equal to 1 in the unmerged parent). The exact parent representation is

$$U_{\mathrm{GIN}}^\ell = \gamma^\ell \left[\mathcal{O}_{\mathrm{sum}}^\ell(H^{\ell-1}; W_1^\ell) + (1+\epsilon^\ell)\mathcal{O}_{\mathrm{self}}^\ell(H^{\ell-1}; W_1^\ell)\right] + b_1^\ell, \quad H_{\mathrm{GIN}}^\ell = \Psi_{\mathrm{tail}}^\ell(U_{\mathrm{GIN}}^\ell). \tag{3}$$

The UMPM reparameterization is exact for each parent; convex fusion of arbitrary nonlinear GIN tails is not, so fusion uses the unlabeled local affine surrogate in Eq. (8).

The architecture mapping is explicit in this basis: GCN uses $\mathcal{O}_{\mathrm{gcn}}$, GAT uses $\mathcal{O}_{\mathrm{att}}$ (with attention internal to the operator), and mean-GraphSAGE uses separate $\mathcal{O}_{\mathrm{self}}$ and $\mathcal{O}_{\mathrm{mean}}$ blocks. GIN instead uses the coupled $\mathcal{O}_{\mathrm{sum}} + (1+\epsilon)\mathcal{O}_{\mathrm{self}}$ term before its nonlinear MLP tail.

The distinction between *exact reparameterization* and *exactness after interpolation* is important. For every supported parent, the UMPM form reproduces the native layer before fusion. In particular, Eq. (3) moves only GIN's first affine map through the sum; it does not split the nonlinear MLP into independent branches. The approximation enters later, when two different nonlinear update functions must be represented by one merged update function.

3.3 LAYER MATCHING AND HIDDEN-SPACE ALIGNMENT

Let L_P be the number of layers of parent P including the output layer and $\bar{L}_P = L_P - 1$ its internal depth. Let $\mathcal{U}_P = (U_P^0, \ldots, U_P^{\bar{L}_P})$ denote the centered layer representations used for matching, with $U_P^0 = X - \mathbf{1}\bar{x}^\top$, $\bar{x} = N^{-1}X^\top\mathbf{1}$, and U_P^i ($i \geq 1$) the centered internal pre-activation. We compute $S_{ij} = \mathrm{CKA}(U_A^i, U_B^j)$ using linear CKA, $\mathrm{CKA}(U, V) = \|U^\top V\|_F^2/(\|U^\top U\|_F \|V^\top V\|_F)$ (Kornblith et al., 2019). To exclude non-monotone layer correspondences, we use a dynamic-programming score $\mathrm{dp}[i, j]$ with nonnegative gap penalty κ and require an order-preserving path anchored at $0 \leftrightarrow 0$ and $\bar{L}_A \leftrightarrow \bar{L}_B$:

$$\begin{aligned} \mathrm{dp}[i, j] &= S_{ij} + \max\{\mathrm{dp}[i-1, j-1],\ \mathrm{dp}[i-1, j] - \kappa\delta_{ij},\ \mathrm{dp}[i, j-1] - \kappa\delta_{ij}\}, \\ \delta_{ij} &= \left| \frac{i}{\bar{L}_A} - \frac{j}{\bar{L}_B} \right|. \end{aligned} \tag{4}$$

The boundary states are initialized so that the path begins at $(0, 0)$ and terminates at $(\bar{L}_A, \bar{L}_B)$. Backtracking retains only diagonal moves as the injective correspondence set $\mathcal{C}$; horizontal or vertical moves represent unmatched depth rather than many-to-one layer matching.

For every matched pair (i, j), let d_i and d_j be the corresponding hidden widths and define the cross-covariance $C_{ij} = (U_A^i)^\top U_B^j$. We align parent A to parent B's hidden space with the semi-orthogonal map

$$R_{i\to j} = \arg\max_R \mathrm{tr}(R^\top C_{ij}), \quad R^\top R = I_{d_j}\ (d_i \geq d_j) \text{ or } RR^\top = I_{d_i}\ (d_i < d_j). \tag{5}$$

For a thin SVD $C_{ij} = \Phi\Sigma\Theta^\top$, the maximizer is $R_{i\to j} = \Phi\Theta^\top$; when $d_i = d_j$, this reduces to the classical orthogonal Procrustes solution (Schönemann, 1966). Tildes denote parent-A quantities after mapping them into parent B's coordinate system. For a linear operator, R_{in} and R_{out} are the maps associated with its input and output spaces, giving $\widetilde{W} = R_{\mathrm{in}}^\top W R_{\mathrm{out}}$; the pre-update bias transforms as $\widetilde{\beta} = R_{\mathrm{out}}^\top \beta$. GAT query/key/value projections follow the same rule and attention combination vectors transform as $\widetilde{a} = R_{\mathrm{out}}^\top a$. The coupled GIN sum/self term is transformed without distributing its nonlinear tail across branches.

The child has $\bar{L}_C = \max(\bar{L}_A, \bar{L}_B)$ internal slots. The deeper parent's order defines the reference layer slots; a matched shallower block is inserted at its aligned slot, while an unmatched slot receives a nodewise identity projection $\mathcal{I}^\ell(H) = HI_{\mathrm{rect}}^\ell$, where I_{rect}^ℓ is the rectangular identity map required by the adjacent widths. This padding is introduced only to define a corresponding layer for fusion. After fusion there is still one sequential hidden state, so depth mismatch does not introduce skip paths or a second inference stream.

3.4 CLOSED-FORM FUSION IN THE SHARED BASIS

At aligned layer ℓ, mixture-coefficient estimation uses the shared unlabeled input representation $H_*^{\ell-1} = \frac{1}{2}(\widetilde{H}_A^{\ell-1} + H_B^{\ell-1})$. For each parent separately, define $B_{P,b}^\ell = \mathcal{O}_{b,P}^\ell(H_*^{\ell-1})$ and $U_{P,\mathrm{tgt}}^\ell = \sum_{b\in\mathcal{B}_P^\ell} B_{P,b}^\ell$. This regression reconstructs the pre-update message-passing contribution only; the layer pre-update bias β_P^ℓ is transformed and fused separately. With $\mathbf{b}_{P,b}^\ell = \mathrm{vec}(B_{P,b}^\ell)$, $s_{P,b}^\ell = \max(\|\mathbf{b}_{P,b}^\ell\|_2, \varepsilon_s)$ for a fixed stabilizer $\varepsilon_s > 0$, normalized design matrix $\overline{\mathbf{B}}_P^\ell$ whose columns are $\mathbf{b}_{P,b}^\ell/s_{P,b}^\ell$, and $u_P^\ell = \mathrm{vec}(U_{P,\mathrm{tgt}}^\ell)$, MOSAIC solves the ridge problem with regularization $\lambda \geq 0$,

$$\widehat{c}_P^\ell = \arg\min_c \|\overline{\mathbf{B}}_P^\ell c - u_P^\ell\|_2^2 + \lambda\|c\|_2^2, \qquad \widehat{g}_{P,b}^\ell = \widehat{c}_{P,b}^\ell/s_{P,b}^\ell. \tag{6}$$

The regression is parent-specific but uses the same state; normalized columns are de-normalized before fusion. This shared-state construction makes the residuals comparable: r_P^ℓ measures how well the set of operators available to parent P reconstructs its own aligned target when both parents are probed at the same input state. Thus the subsequent interpolation coefficient is determined directly by the reconstruction residuals, not an architecture-pair lookup or a tuned validation parameter.

Tied GIN reconstruction. GIN cannot fit the sum and self terms independently without changing the parent semantics. For a GIN parent P, define $\mathbf{b}_{P,\mathrm{GIN}}^\ell = \mathrm{vec}(B_{P,\mathrm{sum}}^\ell + (1 + \epsilon_P^\ell)B_{P,\mathrm{self}}^\ell)$,

$s^{\ell}_{P,\mathrm{GIN}} = \max(\|\mathbf{b}^{\ell}_{P,\mathrm{GIN}}\|_2, \varepsilon_s)$, and $\overline{\mathbf{b}}^{\ell}_{P,\mathrm{GIN}} = \mathbf{b}^{\ell}_{P,\mathrm{GIN}}/s^{\ell}_{P,\mathrm{GIN}}$. Let $\overline{\mathbf{B}}^{\ell}_{P,-\mathrm{GIN}}$ contain the remaining normalized columns. We solve the ridge problem directly on this tied subspace,

$$(\widehat{c}^{\ell}_{P,\gamma}, \widehat{c}^{\ell}_{P,-\mathrm{GIN}}) = \arg\min_{c_\gamma, c} \left\| c_\gamma \overline{\mathbf{b}}^{\ell}_{P,\mathrm{GIN}} + \overline{\mathbf{B}}^{\ell}_{P,-\mathrm{GIN}} c - u^{\ell}_P \right\|_2^2 + \lambda(c_\gamma^2 + \|c\|_2^2). \tag{7}$$

After de-normalization, $\widehat{\gamma}^{\ell}_P = \widehat{c}^{\ell}_{P,\gamma}/s^{\ell}_{P,\mathrm{GIN}}$ is the single coefficient for the coupled GIN block. This is the exact ridge optimizer under the GIN tying constraint; the approximation in GIN-containing merges enters only when the nonlinear update function itself must be fused. For that step, because U already includes the first affine bias b_1, write the remaining two-layer GIN tail as $\Psi(U) = \mathrm{ReLU}(U)W_2 + b_2$. Let $\mu = |\mathcal{V}_{\mathrm{cal}}|^{-1} \sum_{v \in \mathcal{V}_{\mathrm{cal}}} U_v$ and let $p = |\mathcal{V}_{\mathrm{cal}}|^{-1} \sum_{v \in \mathcal{V}_{\mathrm{cal}}} \mathbf{1}[U_v > 0]$ be the vector of coordinatewise ReLU activation frequencies. Then

$$J = \mathrm{diag}(p) W_2, \qquad c_\Psi = \Psi(\mu) - \mu J, \qquad \widetilde{\Psi}(U) = UJ + c_\Psi. \tag{8}$$

This is exact only for affine update functions and is the principal approximation in GIN-containing merges. Let $\widehat{U}^{\ell}_{P,\mathrm{op}}$ be the reconstructed pre-update activation, using $\widehat{\gamma}^{\ell}_P$ for the tied GIN block when present and $\widehat{g}^{\ell}_{P,b}$ otherwise. With numerical stabilizer $\varepsilon_{\mathrm{rec}} > 0$, define

$$r^{\ell}_P = \frac{\|U^{\ell}_{P,\mathrm{tgt}} - \widehat{U}^{\ell}_{P,\mathrm{op}}\|_F}{\|U^{\ell}_{P,\mathrm{tgt}}\|_F + \varepsilon_{\mathrm{rec}}}, \tag{9}$$

$$\alpha_\ell = \frac{1/\max(r^{\ell}_A, \varepsilon_{\mathrm{rec}})}{1/\max(r^{\ell}_A, \varepsilon_{\mathrm{rec}}) + 1/\max(r^{\ell}_B, \varepsilon_{\mathrm{rec}})}. \tag{10}$$

If both residual-derived weights are degenerate, we set $\alpha_\ell = 0.5$. The same interpolation coefficient is used for all basis operators and update-function parameters at layer ℓ. Operators absent from one parent are zero-filled before fusion, and the raw merged parameters and mixture coefficients are

$$W^{\ell}_{C,b} = \alpha_\ell \widetilde{W}^{\ell}_{A,b} + (1-\alpha_\ell) W^{\ell}_{B,b}, \qquad g^{\ell}_{C,b} = \alpha_\ell \widehat{g}^{\ell}_{A,b} + (1-\alpha_\ell) \widehat{g}^{\ell}_{B,b}, \qquad \beta^{\ell}_C = \alpha_\ell \widetilde{\beta}^{\ell}_A + (1-\alpha_\ell)\beta^{\ell}_B. \tag{11}$$

Parameters of Ψ^{ℓ}_C follow the same interpolation rule, using the affine surrogate for GIN when needed. When a basis operator exists in only one parent, zero-filling the absent weight and also interpolating its coefficient would multiply that branch by α_ℓ^2 (or $(1-\alpha_\ell)^2$). MOSAIC therefore defines an effective coefficient that corrects this double attenuation:

$$g^{\ell,\mathrm{eff}}_{C,b} = \begin{cases} g^{\ell}_{C,b}/\alpha_\ell, & b \in \mathcal{B}^{\ell}_A \setminus \mathcal{B}^{\ell}_B,\ \alpha_\ell > 0, \\ g^{\ell}_{C,b}/(1-\alpha_\ell), & b \in \mathcal{B}^{\ell}_B \setminus \mathcal{B}^{\ell}_A,\ \alpha_\ell < 1, \\ g^{\ell}_{C,b}, & \text{otherwise.} \end{cases} \tag{12}$$

This preserves the intended single factor of the parent interpolation weight. The merged model then evolves a single hidden-state sequence,

$$H^{\ell}_C = \Psi^{\ell}_C \left(\sum_{b \in \mathcal{B}} g^{\ell,\mathrm{eff}}_{C,b} \mathcal{O}^{\ell}_{b,C}(H^{\ell-1}_C) + \beta^{\ell}_C \right). \tag{13}$$

All active basis operators consume the same $H^{\ell-1}_C$; neither parent is run as a separate stream at inference. Because class-logit axes correspond to labels, they must not be Procrustes-rotated. The output layer is transformed only on its input side and fused columnwise from unlabeled confidence. For parent logits $z_P(v)$, let $\widehat{y}_P(v) = \arg\max_k z_{P,k}(v)$, let $\rho_P(v)$ be the difference between the largest and second-largest components of $\mathrm{softmax}(z_P(v))$, and define $\mathcal{V}_{P,k} = \{v \in \mathcal{V}_{\mathrm{cal}} : \widehat{y}_P(v) = k\}$ with mean margin $\bar{\rho}_{P,k} = |\mathcal{V}_{P,k}|^{-1} \sum_{v \in \mathcal{V}_{P,k}} \rho_P(v)$ (zero if the set is empty). We use

$$\pi_k = \frac{\bar{\rho}_{A,k}}{\bar{\rho}_{A,k} + \bar{\rho}_{B,k}}, \qquad W^{\mathrm{out}}_{C,b}[:,k] = \pi_k \widetilde{W}^{\mathrm{out}}_{A,b}[:,k] + (1-\pi_k) W^{\mathrm{out}}_{B,b}[:,k], \tag{14}$$

with $\pi_k = 0.5$ when the denominator vanishes; biases follow the same rule. Thus the merged model has one K-dimensional output layer. Parent logits estimate merge coefficients once and are not ensembled at inference, and the class-specialist partition is never used to choose π_k.

3.5 KERNEL WITNESS AS AN MMD WITNESS CORRECTION

Fusion can still shift edge-message distributions after hidden-space alignment because interpolation changes both message values and how architecture-specific operators respond to local topology. For edge $e = (u \to v) \in \mathcal{E}_{\text{cal}}$ and model index $Q \in \{A, B, C\}$, let $m^{\ell}_{Q,e}$ denote the edge message and let $d_{\text{out}}(u)$ and $d_{\text{in}}(v)$ denote endpoint degrees. We use structural context $q_e = [\log \max\{d_{\text{out}}(u), 1\}, \log \max\{d_{\text{in}}(v), 1\}, \widehat{A}_{uv}]$, where $\widehat{A}$ denotes the symmetrically normalized adjacency with self-loops and $\widehat{A}_{uv}$ is used only as a fixed edge-structure feature, and define $\zeta^{\ell}_{Q,e} = [m^{\ell}_{Q,e} \| q_e]$. A seeded random Fourier map $\varphi^{\ell}(\zeta) = \sqrt{2/D_{\text{RFF}}} \cos(\Omega^{\ell}\zeta + \xi^{\ell})$, where Ω^{ℓ} is the frequency matrix and ξ^{ℓ} the phase vector for D_{RFF} features with Gaussian bandwidth σ_{KW}, approximates a Gaussian kernel (Rahimi & Recht, 2007); hence each model induces $\bar{\varphi}^{\ell}_{Q} = |\mathcal{E}_{\text{cal}}|^{-1} \sum_{e \in \mathcal{E}_{\text{cal}}} \varphi^{\ell}(\zeta^{\ell}_{Q,e})$. The target kernel mean uses the same layer-wise interpolation coefficient as parameter fusion:

$$\bar{\varphi}^{\ell}_{*} = \alpha_{\ell} \bar{\varphi}^{\ell}_{A} + (1 - \alpha_{\ell}) \bar{\varphi}^{\ell}_{B}, \qquad \Delta^{\ell} = \bar{\varphi}^{\ell}_{*} - \bar{\varphi}^{\ell}_{C}. \tag{15}$$

$\|\Delta^{\ell}\|_2^2$ approximates kernel maximum mean discrepancy (MMD) (Gretton et al., 2012). Defining the witness $w^{\ell}(\zeta) = \langle \Delta^{\ell}, \varphi^{\ell}(\zeta) \rangle$, one analytic gradient step on the witness function with respect to the message, with step size $\eta > 0$, $\widehat{m}^{\ell}_{C,e} = m^{\ell}_{C,e} + \eta \nabla_m w^{\ell}(\zeta^{\ell}_{C,e})$, moves the child message distribution toward the target kernel mean. The gradient is with respect to the message, not network parameters; weights remain frozen throughout.

4 EXPERIMENTAL SETUP

Benchmarks and models. We report six node-classification datasets: Cora and CiteSeer (Yang et al., 2016; Sen et al., 2008), Actor and Amazon-Ratings (Pei et al., 2020; Platonov et al., 2023), and ogbn-arxiv and ogbn-products (Hu et al., 2020). Parents use GCN, GAT, mean-GraphSAGE, or GIN; tables and figures abbreviate mean-GraphSAGE as SAGE. We evaluate homogeneous and heterogeneous pairs over ordered depth pairs $(L_A, L_B) \in \{(2, 2), (2, 3), (3, 2)\}$ and hidden-width pairs $(d_A, d_B) \in \{(64, 64), (64, 128), (128, 64)\}$; GAT width is the total across eight heads.

Parent specialization and merging. We use the complementary 80/20 class-specialist construction introduced in Section 3. Parent training and implementation details appear in Appendix A.4. Merge-time access follows the restrictions in Eq. (1); KW is applied only at the final message-passing layer with $D_{\text{RFF}} = 64$, $\sigma_{\text{KW}} = 1$, and $\eta = 0.1$.

Metric and baselines. We use the retention metric defined in Section 3, and reported MOSAIC cells average five seeds unless stated otherwise. We compare with GRAMA-PMC/CMC (Jing et al., 2024), GNNMerge++ (Garg et al., 2025), Git Re-Basin (Ainsworth et al., 2023), OT-Fusion (Singh & Jaggi, 2020), TF-HMM (Xu et al., 2024), and PLeaS (Nasery et al., 2025) where applicable.

5 RESULTS

5.1 CROSS-ARCHITECTURE RETENTION

Table 1 reports the architecture×depth×width grid for Actor and Amazon-Ratings, the two highest-mean-retention datasets over the evaluated sweep; grids for Cora, CiteSeer, ogbn-arxiv, and ogbn-products appear in Appendix Tables 6 and 7. Width effects are configuration-specific: averaging every reported cell across all six datasets gives mean retention 0.736, 0.726, and 0.729 for widths 64–64, 64–128, and 128–64, respectively.

On Actor and Amazon-Ratings, most non-GIN heterogeneous configurations exceed 0.80 retention across the evaluated depth and width combinations, including many GCN–GAT, GAT–GCN, GCN–SAGE, and SAGE–GAT merges. GIN-containing configurations show a different pattern: Actor remains comparatively robust in several settings, whereas Amazon-Ratings exhibits substantially lower retention for GIN pairs. The remaining datasets in the appendix exhibit the same overall distinction and include larger degradations under depth mismatch on the OGB graphs.

Depth and width sensitivity. The effect of depth mismatch is also visible in individual configurations. On ogbn-arxiv, for example, GCN–SAGE at width 64–64 drops from 0.82 retention at

Table 1: Retention on Actor and Amazon-Ratings, the two highest-mean-retention datasets over the evaluated six-dataset sweep. Cells are mean ± standard deviation of minimum retention over five seeds. Boldface/underlining denote ≥ 0.80/≥ 0.70 using unrounded means. Other dataset grids appear in Appendix Tables 6 and 7.

		Actor			**Amazon-Ratings**		
Pair	Depth	64–64	64–128	128–64	64–64	64–128	128–64
GCN–GCN	2-2	**.95±.08**	**.87±.12**	**.81±.09**	**.98±.02**	**.95±.05**	**.96±.02**
	2-3	**.89±.08**	**.80±.16**	.75±.13	**.97±.05**	**.93±.10**	**.93±.06**
	3-2	.79±.10	.70±.18	.76±.19	**.96±.08**	**.99±.02**	**.93±.06**
GAT–GAT	2-2	**.90±.13**	**.86±.11**	**.91±.07**	**.98±.01**	**.95±.04**	**.97±.01**
	2-3	**.82±.09**	**.91±.12**	**.90±.07**	**.96±.02**	**.97±.03**	**.95±.06**
	3-2	**.94±.05**	**.91±.05**	**.93±.06**	**.95±.02**	**.91±.05**	**.93±.08**
GCN–SAGE	2-2	**.97±.01**	.74±.07	**.86±.09**	**.90±.11**	**.83±.15**	**.82±.16**
	2-3	**.82±.10**	**.89±.19**	.79±.14	**.88±.10**	**.85±.19**	**.88±.06**
	3-2	**.91±.05**	.67±.18	.79±.17	**.97±.06**	**.98±.05**	**.89±.11**
GCN–GAT	2-2	**.97±.04**	**.96±.06**	**.95±.06**	**.98±.02**	**.97±.05**	**.96±.05**
	2-3	**.95±.01**	**.97±.04**	**.92±.10**	**.89±.12**	**.93±.05**	**.94±.03**
	3-2	**.87±.11**	**.87±.11**	**.80±.17**	**.95±.10**	**.97±.07**	**.93±.11**
GAT–GCN	2-2	**.86±.12**	**.82±.10**	**.97±.02**	**.95±.03**	**.90±.07**	**.95±.02**
	2-3	**.86±.13**	**.82±.14**	**.91±.11**	**.91±.08**	**.85±.13**	**.89±.09**
	3-2	**.96±.04**	**.95±.03**	**.95±.03**	**.93±.06**	**.85±.11**	**.94±.06**
SAGE–GAT	2-2	**.94±.03**	**.93±.03**	**.91±.05**	**.82±.08**	**.82±.11**	**.87±.07**
	2-3	**.87±.07**	**.88±.05**	**.85±.04**	**.83±.12**	**.85±.10**	**.92±.09**
	3-2	.74±.10	.74±.07	**.92±.12**	**.86±.15**	**.91±.11**	**.93±.15**
GCN–GIN	2-2	**.80±.20**	.77±.19	.70±.23	.46±.38	.79±.22	.47±.28
	2-3	.73±.15	.68±.24	.71±.18	.45±.25	.67±.27	.49±.25
	3-2	.70±.23	.62±.15	.77±.14	.30±.24	**.85±.32**	.38±.29
SAGE–GIN	2-2	**.83±.09**	**.82±.11**	**.83±.12**	.39±.30	.69±.16	.44±.36
	2-3	.72±.09	.73±.11	.72±.06	.49±.28	.56±.19	.42±.29
	3-2	.79±.11	.64±.14	**.84±.13**	.41±.32	.71±.27	.52±.34
GAT–GIN	2-2	.78±.23	**.81±.20**	.75±.23	.54±.36	.75±.28	.45±.36
	2-3	.65±.16	.63±.22	.68±.18	.39±.30	.62±.23	.38±.29
	3-2	**.81±.14**	.76±.21	.79±.22	.56±.37	.70±.26	.56±.37

depth 2–2 to 0.42 at 2–3, while the reverse 3–2 setting recovers to 0.63. GIN-containing pairs are more fragile: ogbn-products GCN–GIN at width 64–64 falls from 0.48 at 2–2 to 0.27 at 2–3. Semi-orthogonal alignment makes unequal widths feasible but does not remove all width sensitivity; ogbn-arxiv GCN–GIN at depth 3–2 varies from 0.74 (64–64) to 0.54 (64–128) and 0.17 (128–64).

5.2 Comparison with training-free merging

Table 2 shows that homogeneous results place MOSAIC among the stronger reported training-free methods, while the heterogeneous columns expose a much larger separation from our cross-architecture adaptation of PLeaS (Nasery et al., 2025), whose published formulation assumes a shared architecture.

5.3 Efficiency and component validation

To quantify the computational benefit of single-child inference, Table 3 reports measured inference speedup relative to executing both parents. The child is $1.13\times$–$1.91\times$ faster across the tested pairs.

Table 2: Training-free baselines on the restricted ogbn-arxiv non-GIN specialist protocol (mean ± std.). Dashes denote unavailable heterogeneous results; boldface/underlining denote the best/second-best reported value per column.

	Homogeneous (Same-Arch)		**Heterogeneous (Cross-Arch)**	
Method	Ret_A	Ret_B	Ret_A	Ret_B
GRAMA-PMC	**0.92±0.05**	**0.93±0.06**	—	—
GRAMA-CMC	0.86±0.08	0.85±0.12	—	—
GNNMerge++	0.81±0.29	0.74±0.13	—	—
Git Re-Basin	0.83±0.07	0.81±0.11	—	—
OT-Fusion	**0.92±0.07**	0.84±0.06	—	—
TF-HMM	0.70±0.20	0.67±0.22	—	—
PLeaS	0.27±0.23	0.37±0.30	<u>0.19±0.17</u>	<u>0.24±0.26</u>
MOSAIC (ours)	<u>0.87±0.11</u>	<u>0.90±0.09</u>	**0.86±0.13**	**0.88±0.12**

Table 3: Inference speedup of the merged MOSAIC model over evaluating both parents in the separate three-seed ogbn-arxiv latency benchmark.

Pair	Speedup	Pair	Speedup
GAT–GAT	$(1.44 \pm 0.30)\times$	GAT–GIN	$(1.91 \pm 1.20)\times$
GAT–SAGE	$(1.23 \pm 0.31)\times$	GCN–GIN	$(1.13 \pm 0.11)\times$
GCN–SAGE	$(1.32 \pm 0.64)\times$	GIN–GIN	$(1.31 \pm 0.11)\times$
SAGE–GIN	$(1.14 \pm 0.22)\times$	SAGE–SAGE	$(1.25 \pm 0.22)\times$

The first three phases transform parent representations before the merged classifier is defined; we therefore evaluate these transformations through equivalence and alignment diagnostics rather than predictive accuracy. UMPM reparameterization reproduces native parent forward passes with 95th-percentile maximum absolute discrepancy at most 7.81×10^{-3} on the four small graphs and 1.22×10^{-4} on the two OGB graphs. For rectangular width-changing matches, semi-orthogonal alignment increases mean matched-layer CKA by 1.9×10^{-4}–8.0×10^{-4}; square orthogonal maps are excluded because linear CKA is invariant to orthogonal transformations (Kornblith et al., 2019).

Kernel Witness is applied after parameter fusion as a final MMD witness-function correction. Table 4 compares the merged model before and after KW. The changes are positive on five datasets and neutral at displayed precision on ogbn-products.

Table 4: **Kernel Witness MMD correction.** Parent specialist accuracy and minimum merged-model retention after fusion and after KW.

	Parent specialist accuracy		**Minimum retention**	
Dataset	Parent A	Parent B	After fusion	After KW
Cora	.9238	.7922	.5276	.5367
CiteSeer	.7170	.8029	.7132	.7204
Actor	.4748	.5882	.7997	.8082
Amazon-Ratings	.6143	.7688	.7290	.7380
ogbn-arxiv	.6346	.5413	.4966	.5008
ogbn-products	.7789	.8033	.5622	.5663

5.4 Learned message-passing mixtures in the merged model

Because UMPM exposes named message-passing operators, the merged model can be inspected through the learned mixture coefficients. Appendix Figure 4 shows three CiteSeer examples. The non-GIN GCN–SAGE merge assigns substantial GCN, self, and mean contributions at every depth rather than reproducing only one parent. GCN–GIN and GIN–SAGE similarly preserve operators

originating from both families, while the GIN sum term remains structurally coupled to its scaled-self counterpart during fitting. These visualizations are descriptive: operators from both parent families receive nonzero coefficients within the merged GNN rather than being combined only at the logits.

6 Discussion and Conclusion

The experiments identify the main conditions under which merging is reliable. Non-GIN matched-depth pairs often retain both specialists well, whereas depth mismatch and GIN-containing merges are less reliable; KW contributes a small residual correction in the reported settings. This pattern is consistent with which steps of the method are exact and which are approximate: UMPM reconstructs the supported parent layers exactly before fusion, the semi-orthogonal cross-covariance alignment has the stated SVD optimum (reducing to classical orthogonal Procrustes at equal width (Schönemann, 1966)), and both mixture-coefficient estimation and tied GIN reconstruction solve their stated ridge objectives in closed form. By contrast, CKA layer matching is a correspondence heuristic (Kornblith et al., 2019), semi-orthogonal contraction can discard directions, GIN update function fusion uses a local affine surrogate, and KW uses a finite random-feature approximation to kernel mean matching (Rahimi & Recht, 2007; Gretton et al., 2012). The largest degradations therefore coincide with unmatched depth or the nonlinear update-function approximation rather than failure of the exact UMPM reparameterization.

The deployment objective is also distinct from distillation: knowledge distillation optimizes a student from teacher-derived supervision (Hinton et al., 2015; Yang et al., 2020), whereas MOSAIC performs no post-training child optimization and deploys only the merged graph measured in Table 3.

Natural extensions include richer coupled nonlinear update functions, more expressive training-free representation alignment, and message-passing bases with edge features, positional encodings, or additional message functions. The shared message-passing parameterization is most useful when architectural heterogeneity changes intermediate computations rather than only hidden coordinates.

Reproducibility Statement

Appendix A.4 specifies parent training, class-specialist construction, merge implementation details, and aggregation conventions. Table 1 and Appendix Tables 6 and 7 provide the reported depth/width grids across all six datasets, and Algorithm 1 specifies the complete merge procedure. All main-text aggregate claims are derived from these reported measurements.

AI Use Statement

A large language model was used to assist with manuscript editing, compression, consistency checking, and LaTeX restructuring. It was not used to generate experimental results or modify the underlying reported measurements. The authors reviewed the resulting text and remain responsible for the paper's technical claims, analyses, and presentation.

References

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Vipul Garg, Ishita Thakre, and Sayan Ranu. Merging of GNN models without accessing training data. arXiv preprint, 2025.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL `https://www.jmlr.org/papers/v13/gretton12a.html`.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 1024–1034, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint, 2015.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 876–885, 2018.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations*, 2023.

Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Amalgamating knowledge from heterogeneous graph neural networks. In *Conference on Computer Vision and Pattern Recognition*, 2021.

Yongcheng Jing, Seok-Hee Hong, and Dacheng Tao. Deep graph mating. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

Anshul Nasery, Jonathan Hayase, Pang Wei Koh, and Sewoong Oh. Pleas: Merging models with permutations and least squares. In *Conference on Computer Vision and Pattern Recognition*, 2025.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Nicolás Guerrero Peña, Youssef Mroueh, and Panagiotis Toulis. Re-basing neural networks via implicit sinkhorn differentiation. In *Conference on Computer Vision and Pattern Recognition*, 2023.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv*, 2023.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In John Platt, Daphne Koller, Yoram Singer, and Sam Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20, pp. 1177–1184. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Peter H. Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008. doi: 10.1609/aimag.v29i3.2157.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, 2020.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. ZipIt! merging models from different tasks without training. In *International Conference on Learning Representations (ICLR)*, 2024.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Zhengqi Xu, Han Zheng, Jie Song, Li Sun, and Mingli Song. Training-free heterogeneous model merging. *arXiv*, 2024.

Rahul Yadav, Aditya Kusupati, Suchin Gururangan, Noah A. Smith, and Luke Zettlemoyer. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024.

Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, 2020.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, 2016.