From Parameters to Behaviors: A Survey of Model Fusion for Large Language Models

Shuo Cai *, Yanggan Gu, Zihao Wang, Yuanyi Wang, Yibo Yan, Wenjun Wang, Yuhang Liu, Guanghao Zhu, Sirui Huang, Ming Li, Hongxia Yang *

Posted Date: 29 May 2026

doi: 10.20944/preprints202605.2007.v1

Keywords: model fusion; large language models; knowledge transfer

 doi:10.20944/preprints202605.2007.v1

Article

From Parameters to Behaviors: A Survey of Model Fusion for Large Language Models

Shuo Cai [1,*], Yanggan Gu [1], Zihao Wang [2], Yuanyi Wang [1], Yibo Yan [3], Wenjun Wang [1], Yuhang Liu [5], Guanghao Zhu [1], Sirui Huang [1], Ming Li [1,4,†] and Hongxia Yang [1,4,5,†,*]

[1] The Hong Kong Polytechnic University (PolyU)
[2] The Chinese University of Hong Kong
[3] The Hong Kong University of Science and Technology (Guangzhou)
[4] PolyU-Daya Bay Technology and Innovation Research Institute
[5] InfiX.ai
* Correspondence: shuo1031.cai@connect.polyu.hk (S.C.); hongxia.yang@polyu.edu.hk (H.Y.)
† These authors contributed equally to this work.

Abstract

Model fusion integrates the capabilities from source models into a single target model. As the open-source AI ecosystem matures, Hugging Face has hosted more than 2M models. This growing pool provides a rich base for model reuse and capability integration. Yet existing surveys often cover only separate parts of this space, and they do not provide a unified definition or a systematic taxonomy. This survey defines model fusion and organizes prior work into three levels: parameter-level, representation-level, and behavior-level fusion. We also review related metrics, benchmarks, and applications, summarize current challenges, and identify future directions. Our goal is to provide a clear map of this area and support future work on model fusion.

Keywords: model fusion; large language models; knowledge transfer

1. Introduction

As large language models and the open-source ecosystem continue to grow, the number and variety of available models have increased quickly. Hugging Face now has hosted over 2M open-source models[1], which provides a rich and diverse base for model reuse and capability integration. Therefore, reusing and integrating existing model capabilities within a single model is becoming an important direction [1,2].

Given multiple source models with diverse capabilities, model fusion integrates their parameters, representations, or behaviors into a single target model, as shown in Figure 1. Besides, the target model does not depend on any source model at inference time. Under this definition, traditional model merging and knowledge distillation methods can be view as parameter-level and behavior-level model fusion [1,3–5].

As a broad framework for capability integration, model fusion has multiple attractive advantages. First, it enables efficient reuse of existing models and integrates their capabilities into one target model. This can be done by fusion parameters, using representations to diagnose and repair drift, or distilling output behaviors [6–9]. As shown in Figure 2, model fusion has attracted steadily increasing researcher attention since 2023. This trend is also reflected in industrial practice, where DeepSeek-V4 [10], NVIDIA's Nemotron-Cascade 2 [11], and GLM-5 [12] adopt on-policy distillation to integrate or recover model capabilities. Second, model fusion supports continual learning by absorbing new task signals while preserving earlier capabilities. AIMMerging [13], RECALL [14], and NUFILT [15] show how fusion can add new knowledge while reducing forgetting.

[1] https://huggingface.co/models

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2026 doi:10.20944/preprints202605.2007.v1

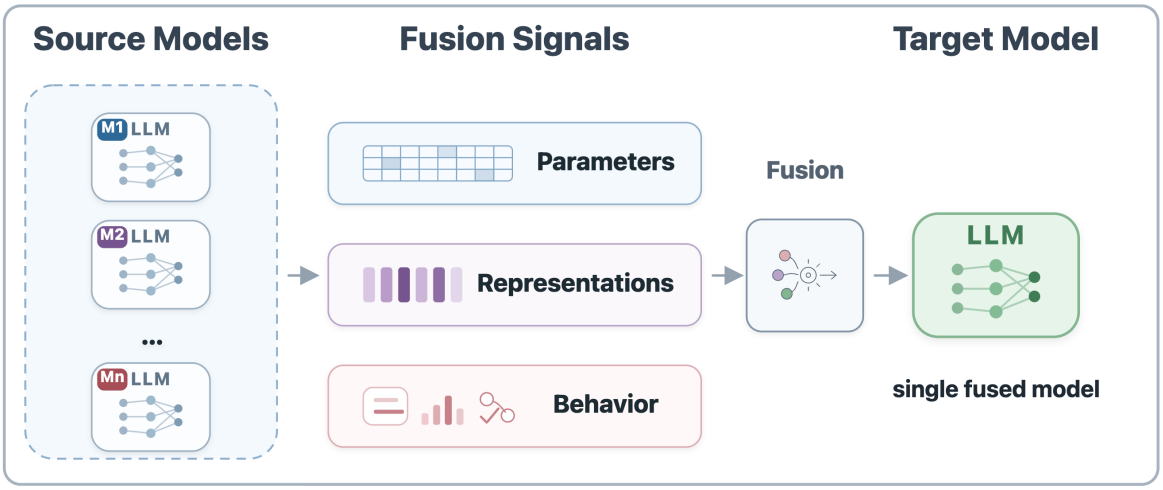

Figure 1. Model fusion overview. Source models provide different signals to form a single target model.

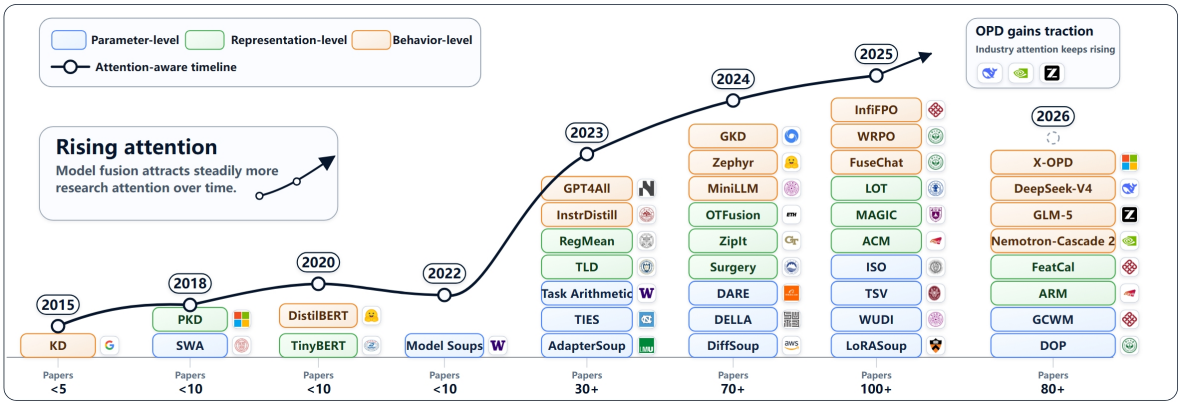

Figure 2. Progress timeline of model fusion.

Despite these advantages, recent studies also show that model fusion remains far from settled. Weight averaging and alignment can improve accuracy and robustness, but some task-level combinations may collapse, and current theory cannot yet predict when fusion will succeed [16–18]. Moreover, fusion becomes harder when source models differ in architecture, tokenizer, or modality, because parameter and representation alignment can be unstable [19–21]. In evaluation, recent benchmarks improve standardization, but average scores can still hide local degradation and cross-capability interference [18,22–24].

Table 1. Coverage of related surveys.

Survey	Venue & Year	Param. level	Repre. level	Behav. level
Gou et al. [25]	IJCV'21		✓	✓
Xu et al. [26]	arXiv'24		✓	✓
Yadav et al. [5]	TMLR'25	✓		
Yang et al. [27]	TIST'25		✓	✓
Qin et al. [28]	IJIS'25	✓		✓
Yang et al. [3]	CSUR'26	✓	✓	
Song and Zheng [4]	arXiv'26	✓	✓	
Li et al. [1]	TNNLS'26	✓	✓	
Song and Zheng [29]	arXiv'26			✓
Fang et al. [30]	AIR'26		✓	✓
Ours		✓	✓	✓

As shown in Table 1, existing surveys mainly systematize model merging or knowledge transfer as separate topics [25–30]. They still lack a unified view of the scope, boundary, and evaluation of model fusion. To fill this gap, this paper gives a formal definition of model fusion and organizes its methods into three levels: parameter-level fusion, representation-level fusion, and behavior-level fusion. We also discuss evaluation, applications, open challenges and future directions for model fusion.

 doi:10.20944/preprints202605.2007.v1

We survey more than 150 papers on model fusion and organize the paper as follows. Section 2 introduces the definition and formulation of model fusion. Section 3 presents the taxonomy of parameter-, representation-, and behavior-level fusion, together with evaluation settings. Sections 4–6 summarize practical takeaways, discuss challenges and future directions, and conclude the paper.

2. Definition and Formulation

This section gives a general definition and formulation of model fusion.

Definition.

We define model fusion as follows:

Given a set of source models, model fusion aims to integrate their capabilities, knowledge, representations, or behaviors into a single target model, so that the resulting model can operate at inference time without relying on complete source models.

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input space and the output space. Given n source models

$$\mathcal{S} = \{M_i^{\text{src}}\}_{i=1}^{n}, \tag{1}$$

where M_i^{src} is the i-th source model. For input $x \in \mathcal{X}$, each source model gives a conditional output distribution $p_i^{\text{src}}(y \mid x)$, where $y \in \mathcal{Y}$. The goal is to build a target model M_θ^{tgt} with parameters θ. Its conditional output distribution is $p_\theta^{\text{tgt}}(y \mid x)$. Model fusion can be written as a mapping from source models to the target model:

$$\theta = \Phi(\mathcal{S}, \mathcal{D}). \tag{2}$$

Here, Φ is the fusion mapping. $\mathcal{D}$ is the dataset used during fusion, and it can be an empty set. The above fusion goal can be written as:

$$\theta^\star = \arg\min_\theta \sum_{i=1}^{n} \mathop{\mathbb{E}}_{x \sim \mathcal{T}_i} [\mathrm{D}_{\text{out}}(p_\theta^{\text{tgt}}(\cdot \mid x), p_i^{\text{src}}(\cdot \mid x))]. \tag{3}$$

Here, $\mathcal{T}_i$ is the input distribution on the task of the i-th source model. D_{out} is the distance in the output space.

Inference Independence.

Once θ is fixed, the target model no longer needs the source models during inference:

$$p_\theta^{\text{tgt}}(y \mid x, \mathcal{S}) = p_\theta^{\text{tgt}}(y \mid x). \tag{4}$$

3. Taxonomy of Model Fusion

This section builds a taxonomy of model fusion based on the fusion signal. The taxonomy asks which kind of information from source models is mainly used in fusion, not the surface form of the final result. We therefore distinguish parameter-, representation-, and behavior-level fusion.

 doi:10.20944/preprints202605.2007.v1

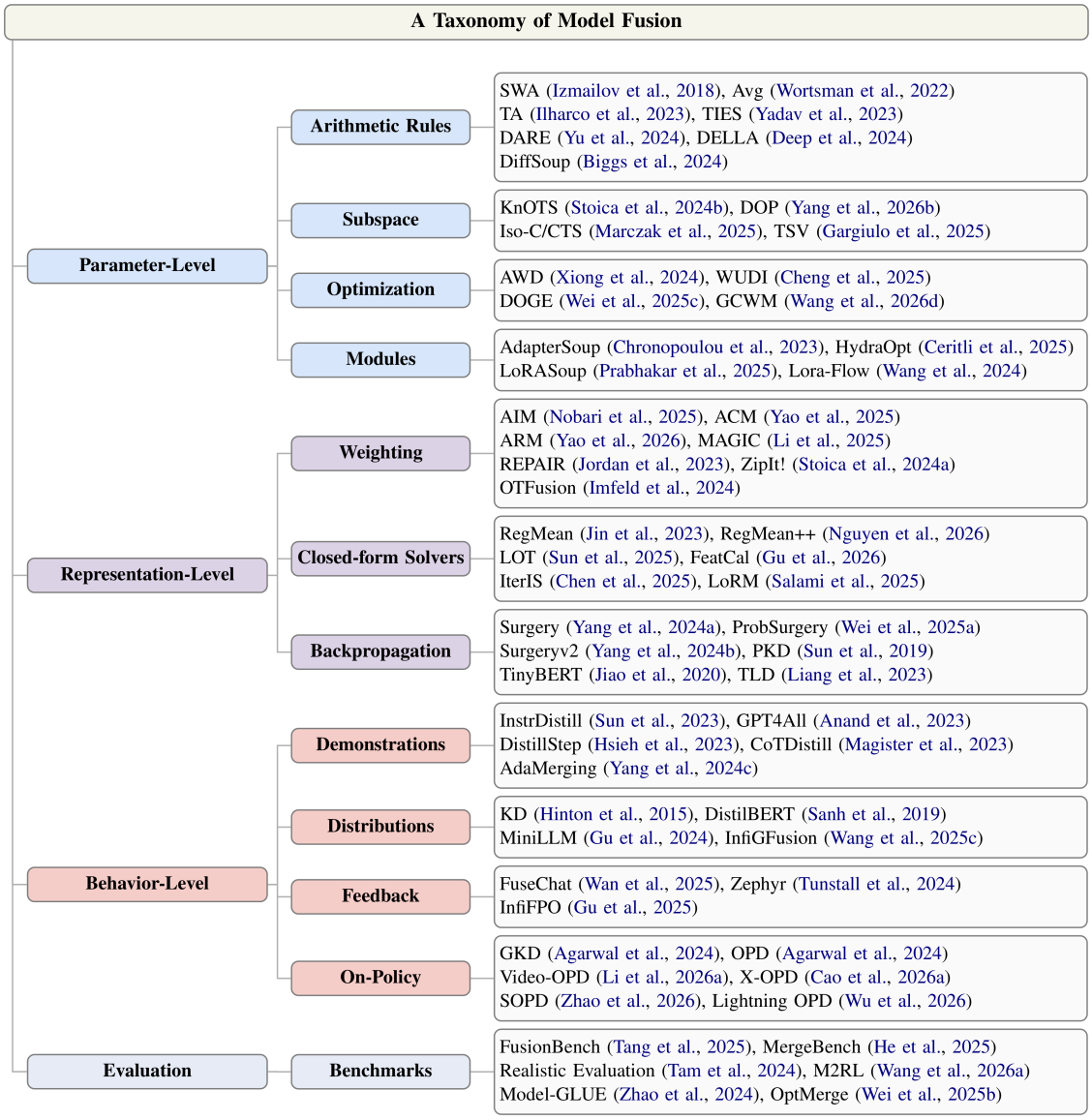

Figure 3. A taxonomy of model fusion for LLMs and MLLMs. The method branches are organized by the main object being fused or aligned: parameters, representations, or behaviors. The evaluation branch summarizes representative benchmark resources. Representative methods and resources are illustrative rather than exhaustive.

3.1. Parameter-Level Fusion

Definition. Parameter-level fusion directly operates on source parameters or modules to form a single target model, i.e., $\theta = \Phi^{\text{param}}(\mathcal{S})$. Here, Φ^{param} maps source parameters or modules into target parameters. These methods use source weights or modules as the main signal and typically do not require a dataset.

Related work and methods. Parameter-level fusion methods can be organized by how they manipulate parameters. *Arithmetic rules* combine source weights or parameter deltas with fixed or lightly tuned coefficients. SWA [31] averages parameter snapshots sampled along an SGD trajectory with a cyclical or constant learning rate, approximating an ensemble with a single model and improving generalization with little additional cost. Model soups [16] show that multiple fine-tuned models can be averaged when they lie in a nearby parameter basin, while task arithmetic [32] represents the difference between a fine-tuned model and its base model as a task vector, enabling capability composition or behavior editing through vector addition and subtraction. Subsequent methods further address conflicts and redundancy among parameter deltas. For example, TIES-Merging, DARE, and DELLA-Merging reduce interference through sign consistency, random dropping or rescaling [6,33,34].

 doi:10.20944/preprints202605.2007.v1

Subspace methods identify, reshape, or constrain structured directions in weight or update space to improve alignment and reduce interference. SVD-based methods use singular directions to reshape update spaces, separate shared and task-specific components, and reduce interference [35–37]. For continual fusion, DOP [38] approximates unavailable data subspaces with SVD subspaces of task vectors and applies dual orthogonal projections to balance stability and plasticity without accessing task data.

Optimization methods formulate merge coefficients, task vectors, or transformation variables as explicit optimization problems. AWD [39] optimizes a decomposition of task vectors into redundant and disentangled components, improving orthogonality while preserving task-specific performance. WUDI [40] uses task vectors to identify and guide the sources of interference in a data-free setting, so that the components responsible for conflicts can be corrected during fusion. GCWM [41] and DOGE [42] use geometric or projected-gradient objectives to reduce interference during multi-task fusion.

Module fusion is distinguished by the fusion object rather than by a specific merge operator: it fuses LoRA, adapters, projectors, or other pluggable modules instead of full model parameters, making it particularly suitable for parameter-efficient fine-tuning. Representative methods include AdapterSoup [43], which averages selected domain adapters, and LoRA soups [44,45], which average, concatenate, or learn coefficients over skill-specific modules for composition tasks.

Parameter-level fusion offers a direct, inference-efficient route to a deployable model, with methods ranging from simple arithmetic to subspace optimization, and module-based merging. Its main challenges are source-model compatibility and interference control.

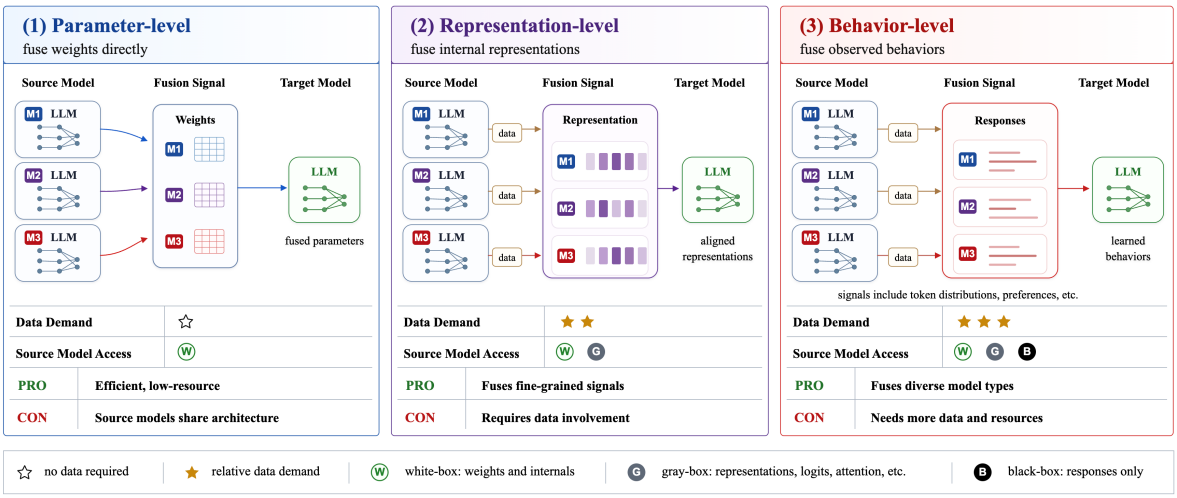

Figure 4. Three levels of model fusion and their practical trade-offs.

3.2. Representation-Level Fusion

Definition. Representation-level fusion uses intermediate representations as the main signal for capability integration.

$$\Phi^{\text{repr}}(\mathcal{S}, \mathcal{D}) = \arg\min_{\theta} \sum_{i=1}^{n} \sum_{\ell} \mathop{\mathbb{E}}_{x \sim \mathcal{T}_i} \left[\mathrm{D}_{\text{repr}} \left(r_{\theta}^{\ell}(x), r_i^{\ell}(x) \right) \right]. \tag{5}$$

Here, $r_{\theta}^{\ell}(x)$ and $r_i^{\ell}(x)$ denote target and source representation at layer ℓ. D_{repr} measures their representation discrepancy. We classify a method as representation-level fusion when hidden representations are the main signal, rather than source parameters or source output behaviors.

Related work and methods. Representation-level fusion asks how intermediate representations can guide the construction or repair of a target model. Existing methods mainly use representations in

 doi:10.20944/preprints202605.2007.v1

three ways: to derive merge signals, solve local matching problems, or train repair and distillation objectives.

Weighting methods compute fusion weights from representations and then combine models in parameter space. These weights can be defined over parameters, layers, modules, or matched components. AIM [46] estimates weight saliency from activation magnitudes on a task-agnostic calibration set. MAGIC [47] calibrates representation and weight magnitudes, while Merging Beyond [48] uses activation subspaces to form rotation-aware updates. Related alignment methods compute correspondence from representations before fusion: REPAIR [49] rescales preactivations, ZipIt! [50] matches units by activation similarity, and Transformer Fusion [51] aligns Transformer components with optimal transport. These methods are efficient, but they depend on calibration data, layer correspondence, and reliable representation similarity.

Closed-form solvers formulate representation matching as local regression problems and solve them analytically, which is most practical for linear modules. RegMean [52] uses input covariance to merge each linear module so that its output matches source-module outputs. RegMean++ [53] improves this local view by adding intra-layer and cross-layer dependencies. LOT-Merging [54] and FeatCal [55] further treat representation drift as the main target: the former derives layer-wise analytic updates, while the latter calibrates merged weights in forward order by separating upstream propagation from local mismatch. For LoRA fusion, LoRM [56] applies output matching to low-rank modules, and IterIS [57] refines the matching objective through iterative inference-solving. Compared with weighting methods, these solvers use representations more directly by fitting local matching objectives, not only by estimating fusion weights.

Backpropagation methods train the target model or added repair modules with representation losses, allowing nonlinear repair and the use of multiple internal signals such as hidden states and attention maps. Patient Knowledge Distillation [58] matches hidden states from selected source layers, while TinyBERT [59] extends the signal to embeddings, attention maps, hidden states, and predictions. Task-aware layer-wise distillation [60] further filters hidden representations before alignment, so the target model focuses on task-relevant parts. Fusion repair methods use the same idea after an initial parameter-level fusion step. Representation Surgery [7] learns a lightweight module to correct final-layer representation bias. Surgeryv2 [61] extends this repair across multiple layers. ProbSurgery [62] models the correction as a distribution to capture uncertainty from parameter interference. Compared with closed-form solvers, these methods can handle more complex mismatch, but they require slower training, larger repair data, and careful regularization to avoid overfitting.

In short, representation-level fusion is most useful when hidden states are available and fusion errors appear as representation drift or layer mismatch. Weighting methods are low cost but sensitive to calibration and similarity signals. Closed-form solvers fit local matching objectives and are more direct, but they need more representation samples and aligned linear modules. Backpropagation methods handle more complex mismatch, but they are slower and more data hungry. Future work may focus on data-efficient repair, robust drift diagnosis, and alignment across heterogeneous source models.

3.3. Behavior-Level Fusion

Definition. Behavior-level fusion uses observable source behaviors to train a target model.

$$\Phi^{\text{behav}}(\mathcal{S}, \mathcal{D}) = \arg\min_{\theta} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{x \sim \mathcal{T}_i} \left[\mathrm{D}_{\text{behav}}\left(q_\theta(x), q_i(x)\right) \right]. \tag{6}$$

Here, q_θ and q_i denote behavior function of the target and source models on input x, e.g., token distribution. $\mathrm{D}_{\text{behav}}$ measures their discrepancy in behavior. Source models therefore act as *behavior providers*, rather than parameter or representation providers. Methods that only use the target model's own entropy, confidence, or uncertainty to guide parameter fusion, without external source behavior as supervision, fall outside this category.

 doi:10.20944/preprints202605.2007.v1

Related work and methods. Behavior-level fusion can be grouped by the transferred behavior type into distribution fusion, demonstration fusion, and feedback fusion, with an orthogonal distinction between off-policy supervision on fixed data and on-policy supervision on target-induced states.

Distribution fusion transfers source-provided soft labels, output distributions, token probabilities, or logits. Classical knowledge distillation matches a teacher's softened output distribution [63], while DistilBERT shows its effectiveness for language model compression [64]. For model fusion, such signals can integrate complementary capabilities across models: InfiGFusion further models logits as relational graphs and aligns their geometry via an efficient Gromov–Wasserstein approximation, moving beyond independent token-level matching [65]. This category should be distinguished from behavior-guided parameter weighting, where behavioral signals guide merge coefficients but are not themselves distilled as source supervision. For example, AdaMerging learns task-wise fusion weights from output entropy [66]; MWA weights checkpoints by training metrics such as loss or training step [67]; and Fisher Merging uses sample-estimated Fisher information to approximate posterior precision for parameter-wise averaging [68].

Demonstration fusion learns from source-generated responses, rationales, reasoning traces, tool-use traces, or trajectories. Instruction distillation and GPT4All-style training use stronger-model outputs to train independently deployable targets [69,70], while rationale or step-level distillation transfers intermediate reasoning processes [71,72]. These methods require only sampled outputs, but may inherit source errors, spurious reasoning, or stylistic bias.

Feedback fusion transfers preferences, scores, critiques, corrections, reward signals, or verifier labels, making it useful for alignment and safety transfer when parameters, hidden states, or full distributions are unavailable. Chat-oriented fusion can construct data from multi-source responses, rankings, and preferences, as in FuseChat and Zephyr [73,74]. InfiFPO further formulates fusion as implicit preference optimization, absorbing source-model advantages without direct pivot model access [75].

From the state-distribution perspective, *off-policy fusion* uses fixed behavior data and is simple to scale, but suffers from mismatch when the target visits poorly covered states. *On-policy fusion* instead lets the target generate prefixes, responses, or trajectories, and then obtains supervision on these target-induced states. This connects to dataset aggregation in imitation learning [76]; in LLMs, GKD instantiates it by distilling from teacher feedback on student-generated sequences [9]. Recent OPD variants study self-distillation, black-box or semi-on-policy supervision, offline logit reuse, token-efficient supervision, and stabilization [77–81], and extend OPD to multimodal trajectories such as video grounding and speech LLM alignment [82,83]. This formulation is especially suitable for heterogeneous fusion, where source and target models may differ in architecture, tokenizer, modality interface, decoding policy, or capability profile.

Overall, behavior-level fusion is well suited to closed-source and heterogeneous source models because it avoids parameter and hidden-state access. Demonstration fusion is broadly applicable but prone to imitation bias; distribution fusion provides dense token-level supervision but often requires probability or logit access; and feedback fusion supports alignment and safety transfer but depends on verifier or reward quality. Emerging directions include robust multi-source behavior aggregation, reliable verifier supervision, budget-aware on-policy querying, and unified process- and outcome-level feedback.

3.4. Evaluation

Metrics.

Evaluation for model fusion can start from two simple metrics. (1) *Avg performance* reports the average performance of the target model over the task pool. It gives a direct view of overall quality and is easy to compare across methods. (2) *Normalized performance* compares the target model with the corresponding source model on each task. MergeBench uses this metric to measure how much source task performance is retained by the target model [23]. This is important because a target model can improve the average score while losing one source capability. Other metrics can examine interference,

 doi:10.20944/preprints202605.2007.v1

generalization, internal alignment, cost, and safety when the setting supports them. Appendix Table A6 gives a compact summary.

Benchmarks.

Model fusion benchmarks involve more than a task leaderboard. They usually define a model pool and a task pool, so methods can be compared under shared source and evaluation settings. FusionBench [22] gives unified settings for comparing many parameter-level fusion methods across model and task pools. MergeBench [23] focuses on domain source models and reports retention, generalization, and cost. Appendix Table A5 compares representative resources by modality coverage, model pool, task pool, heterogeneity, fusion type, and evaluation focus. The comparison shows that current resources still mainly support parameter-level fusion. Representation-level fusion often relies on drift analysis in method papers. Behavior-level fusion often borrows task, response, or safety benchmarks from distillation studies [26,29]. Future directions include shared source settings and clearer reports of representation drift, behavior transfer, judge settings, and total fusion cost [7–9,55,84].

Table 2. Quantitative analysis for Takeaway 2, using results from M2RL [85] and MergeBench [23]. Avg. and Norm. denote the average and normalized performance mentioned at Sec. 3.4. (P), (R) and (B) denote parameter-, representation-, and behavior-level fusion.

Method	Level	Avg.	Norm.
RLVR Expert Fusion: Qwen3-4B, 5 domains			
TIES	P	61.00	103.8
DARE	P	60.99	101.2
MT-OPD	B	60.46	102.1
Domain Expert Fusion: Llama-3.1-8B, 5 domains			
TIES	P	46.8	81.4
DARE	P	45.2	78.7
Task Arithmetic	P	48.7	84.8
RegMean	R	46.3	80.6

Table 3. Quantitative analysis for Takeaway 3, using results from FeatCal [55] and TinyBERT [59].

Method	Level	Avg.	Norm.
Post-Merge Calibration: Llama-3.1-8B, 6 tasks			
TA	P	63.5	90.2
TA+Surgery	P R	64.0	90.7
TA+ProbSurgery	P R	64.4	91.4
TA+FeatCal	P R	65.8	93.1
Teacher-Student Fusion: BERT-base, 3 GLUE tasks			
TinyBERT w/o Logit Fusion	R	73.5	95.2
TinyBERT	R B	75.6	98.1

4. Practical Takeaways

❶ Fusion methods should be selected under practical constraints. Figure 4 summarizes the fusion signals, data and access needs of the three levels. The key choice is which signal is available and which failure mode is most likely. When source models share architecture, initialization, and tokenizer, and their weights are available, parameter-level fusion is often a simple first option. When drift or internal loss appears, representation-level fusion can use hidden states to find and calibrate layer mismatch. When source models only expose responses, or when models differ greatly, behavior-level fusion becomes more practical.

❷ Representation- and behavior-level fusion do not necessarily outperform parameter-level fusion. A simple reason is that, when source models are derived from the same base, their parameter coordinates are often well aligned, and task vectors or parameter deltas can directly encode the acquired capabilities. In this setting, direct parameter fusion may already be sufficient, while representation-

 doi:10.20944/preprints202605.2007.v1

or behavior-level methods introduce additional data collection, estimation, or training costs. Table 2 reports the M2RL and MergeBench results for this point. In M2RL, parameter-level TIES and DARE reach 61.00 and 60.99 on Avg.; behavior-level MT (Multi-Teacher)-OPD reaches 60.46 on Avg. with extra 967 GPU-hours. In MergeBench, Task Arithmetic reaches 48.7 on Avg., and TIES, DARE, and RegMean are slightly lower.

❸ **Combining fusion levels can yield a stronger practical pipeline.** Different fusion levels can address complementary failure modes. Parameter-level fusion can provide a low-cost initial target when source parameters are aligned; representation-level fusion can then calibrate residual representation drift or layer mismatch; behavior-level supervision can further recover missing output behavior when cheaper signals are insufficient. Table 3 shows this pattern. On Llama-3.1-8B, FeatCal improves Task Arithmetic from 63.5 to 65.8 on Avg., outperforming Surgery and ProbSurgery. In TinyBERT, adding logit fusion to intermediate representation fusion improves the performance from 73.5 to 75.6 on Avg.. Common application settings are summarized in Appendix B.

5. Challenges and Future Directions

In model fusion, several challenges still limit reliable capability integration and practical use. Addressing these challenges can help build target models that are more robust, scalable, and safe.

❶ **Unclear Theoretical Foundations and Applicability Conditions.** Model fusion still lacks a clear account of when each type of method works. Existing theory mainly explains parameter-level fusion under shared initialization, nearby loss basins, or hidden unit alignment [16,17,86]. These findings are useful, but they do not cover many LLM settings with different architectures, tokenizers, tasks, or data distributions. For representation-level fusion, it is still unclear when representation spaces can be aligned and when calibration is enough to reduce drift [7,55]. For behavior-level fusion, including on-policy behavior fusion, the field still lacks clear rules for when source feedback helps and when state mismatch or query cost makes it less useful [9,29]. Future work can study the conditions for all three levels, such as source compatibility, task conflict, data access, and target model capacity.

❷ **Difficulty in Aligning Heterogeneous Source Models.** In real settings, source models often have different architectures, tokenizers, or modality interfaces. This breaks parameter correspondence in parameter-level fusion, makes spatial alignment harder for representation-level fusion, and complicates behavior-level fusion because output distributions and reasoning styles are hard to unify. Transport and Merge [20] uses optimal transport for cross-architecture LLM fusion, while AdaMMS [21] learns coefficients for heterogeneous MLLMs. Future work can study representation translation and architecture-agnostic transfer when direct alignment fails.

❸ **Evaluation Remains Incomplete.** Existing evaluation systems often center on average scores. This can hide local capability drops and may mistake differences in source access or tuning budget for method advantages. FusionBench [22] and MergeBench [23] begin to fix source pools and task settings, while Realistic Evaluation [24] points out that compositional generalization can expose interference that single-task evaluation cannot see. Behavior-level fusion is hard to evaluate, because many works report what the target model retains but not what it loses. Papers use different base models, budgets, and rollout settings, which makes fair comparison difficult [29,87,88]. Model fusion evaluation can report source ability retention, worst-task drop, cost, and whether the target model satisfies the single-model inference condition. Future benchmarks can use shared source settings, budget reports, and evaluation suites for different fusion levels.

❹ **Toward Trustworthy Model Fusion.** Model fusion can carry unsafe behavior, backdoors, private data, or unclear ownership from source models into the target model. LoRA-as-an-Attack [89] and Merge Hijacking [90] show that harmful updates can survive fusion. Merger-as-a-Stealer [91] studies private information leakage, while MergeGuard [92] studies fingerprints and IP protection. Among Us [93] further studies malicious contributions in model collaboration. Future work can combine source screening, provenance, contribution attribution, and privacy-aware fusion [94].

We further discuss continual forgetting and large-scale deployment costs in Appendix H.

 doi:10.20944/preprints202605.2007.v1

6. Conclusions

This paper defines model fusion as integrating the capabilities of source models into a single target model, and sets inference without relying on complete source models as the boundary. We organize existing methods into parameter-level fusion, representation-level fusion, and behavior-level fusion. We identify several core challenges in model fusion and propose future research directions. We hope this paper provides a clear framework for model fusion research and helps make model fusion research and practice more systematic, safer, and more efficient.

7. Limitations

This survey may not cover every recent work on model fusion, especially fast-moving preprints and industrial systems with limited public details. Some relevant papers may also be missed because model fusion is studied under different names, such as model merging and knowledge transfer. To reduce this risk, we collected papers from related surveys, benchmark papers, and method papers, and checked the taxonomy and references in several rounds. Human errors may still remain in the categorization or citation of some papers. In addition, our benchmark summary is based on reported results and public resources, which may not fully reflect differences in model scale, data access, and tuning budget. Even with these limits, this survey provides a broad and clear map of model fusion, and summarizes its main methods, evaluation issues, applications, and open challenges.

Appendix A. Symbol Definitions

Table A1. Notation used in the model fusion formulation.

Symbol	Meaning
$\mathcal{X}$	Input space.
$\mathcal{Y}$	Output space.
x	An input instance.
y	An output instance.
n	Number of source models.
$\mathcal{S}$	Set of source models, $\mathcal{S} = \{M_i^{\text{src}}\}_{i=1}^{n}$.
M_i^{src}	The i-th source model.
$p_i^{\text{src}}(y \mid x)$	Conditional output distribution of the i-th source model.
$\mathcal{T}_i$	Input distribution associated with the i-th source model.
M_θ^{tgt}	Target model parameterized by θ.
θ	Parameters of the target model.
$\theta^\star$	Target parameters produced by the fusion process.
$p_\theta^{\text{tgt}}(y \mid x)$	Conditional output distribution of the target model.
ℓ	Layer index.
$r_i^\ell(\cdot \mid x)$	Representation distribution of the i-th source model at layer ℓ.
$r_\theta^\ell(\cdot \mid x)$	Representation distribution of the target model at layer ℓ.
q_i^x	Behavior distribution of the i-th source model on input x.
q_θ^x	Behavior distribution of the target model on input x.
$\mathcal{D}$	Auxiliary information used during fusion; it can be empty.
Φ	Mapping from source models and auxiliary information to target parameters.
D_{out}	Discrepancy measure for output, representation, or behavior gaps.

Appendix B. Applications

Model fusion is useful when capabilities from existing models are integrated into one target model. We group its common uses into four settings: continual learning, capability integration, safety control, and model compression.

 doi:10.20944/preprints202605.2007.v1

Model Fusion in Continual Learning.

In continual learning, model fusion can add new task or domain knowledge while limiting forgetting. AIMMerging [13] and NUFILT [15] apply parameter-level fusion to add new task updates while reducing forgetting and interference. K-Merge [95] extends this setting to online LoRA fusion for on-device LLMs. RECALL [14] uses hidden representations for hierarchical fusion without historical data. SDFT [96] uses behavior-level fusion to learn new skills while reducing forgetting.

Multi-Task Learning and Domain Capability Integration.

Model fusion integrates source models trained for different tasks, domains, or languages. Compared with training one model on mixed task data, it can reuse existing source models and reduce reliance on original data or full retraining [52,66]. Language Specific Model Merging [97] fuses language-specific models to lower multilingual training and update costs. SurgeryV2 [61] and FeatCal [55] repair representation drift after fusion. FuseLLM [8] and DeepSeek-V4 [10] use behavior-level fusion to integrate source capabilities.

Safety and Control.

For safety control, model fusion can transfer, keep, or weaken behavior attributes after training. SafeMERGE [98] and Fuse to Forget [99] use parameter-level fusion to preserve safety or reduce unwanted behavior. Safety Realignment [100] uses subspace-oriented model fusion to realign unsafe models. Multilingual Safety Alignment via Self-Distillation [101] transfers safety behavior across languages through behavior-level fusion. However, unsafe source models can also propagate misalignment during fusion [102].

Model Compression.

Model compression uses source models to build smaller target models with similar capabilities. LoRA soups [45] and LoRM [56] can fold several lightweight modules into one target module. DeepSeek-R1 [103] transfers reasoning patterns into six dense models with 1.5B to 70B parameters. Nemotron-Cascade 2 [11] builds a compact 30B MoE model, with 3B active parameters, for math, code, and agentic tasks. Smaller target models can lower serving cost and speed up inference in resource-limited settings.

Appendix C. Parameter-Level Fusion Analysis

This appendix summarizes representative parameter-level fusion methods and organizes them by source-model relation, fusion object, and evaluated model backbones or settings.

 doi:10.20944/preprints202605.2007.v1

Table A2. Parameter-level fusion methods compared by source-model relation, fusion object, and evaluated backbones or settings. Method groups follow the taxonomy in Section 3 and Figure 3.

Method	Venue	Source relation	Fusion object	Evaluated backbones/settings
Arithmetic rules				
SWA [31]	UAI'18	C checkpoint trajectory	W checkpoints	CV CNN/CV classifiers
Model Soups / Avg [16]	ICML'22	S same-base fine-tuned sources	W full weights	CV CLIP/ViT
Task Arithmetic [32]	ICLR'23	S same-base task vectors	T per-task vectors	CV LLM ED CLIP, GPT-2, T5
TIES-Merging [6]	NeurIPS'23	S same-base task vectors	T per-task deltas	CV ED ViT and T5
DARE [33]	ICML'24	S homologous same-base models	T per-task deltas	LM LLM BERT/RoBERTa, Llama
DELLA-Merging [34]	arXiv'24	S same-base task vectors	T per-task deltas	LLM Llama-2 experts
DiffSoup [104]	ECCV'24	S shared diffusion checkpoint	W diffusion weights	DIF text-to-image diffusion
Subspace-based methods				
KnOTS [35]	ICLR'25	S same-base LoRA sources	P T LoRA task updates	CV LLM CLIP-ViT, Llama3
DOP [38]	NeurIPS'25	S sequential same-base experts	T A task and merged updates	CV ED ViT, Flan-T5
Iso-C / CTS [36]	ICML'25	S same-base task matrices	A aggregated task matrix	CV LLM CLIP-ViT, LLMs
TSV [37]	CVPR'25	S same-base task matrices	T per-task matrices	CV CLIP-ViT
Optimization-based methods				
AWD [39]	arXiv'24	S same-base task vectors	T disentangled task vectors	CV LM ViT, RoBERTa
WUDI [40]	ICML'25	S same-base task vectors	A merged task vector	CV LM LLM ViT, RoBERTa, Llama
DOGE [42]	ICML'25	S same-base task vectors	T A modified merged update	CV LM LLM vision and NLP models
GCWM [41]	arXiv'26	C continual same-backbone updates	A cumulative update state	LLM Qwen3
Module merging				
AdapterSoup [43]	EACL Findings'23	S shared-backbone adapters	P adapters	LLM GPT-2
HydraOpt [105]	EMNLP'25	S shared-backbone adapters	P low-rank adapters	LLM Llama/Qwen-style LLMs
LoRASoup [45]	COLING Industry'25	S shared-backbone LoRAs	P LoRA modules	LLM Llama-7B
Lora-Flow [106]	ACL'24	S shared-backbone LoRAs	P LoRA modules	LLM Llama-2

Source relation S same base, tokenizer, and parameter coordinates; C checkpoint trajectory or continual same-backbone updates.

Fusion object W full model weights or checkpoints; T per-task updates before aggregation; A aggregated or cumulative updates after aggregation; P PEFT modules, LoRA, or adapters.

Backbones/settings CV CV encoder, CLIP-ViT, or CNN; LM encoder-only LM, such as BERT or RoBERTa; ED encoder–decoder LM, such as T5 or Flan-T5; LLM decoder-only LLM; DIF diffusion or text-to-image model.

Classification notes Labels are descriptive and non-exclusive. Methods are grouped according to the parameter-level branch in Figure 3; when a method manipulates multiple parameter objects, all applicable object badges are shown. KnOTS is marked as both PEFT-module and per-task-update based because it aligns LoRA task updates before merging.

 doi:10.20944/preprints202605.2007.v1

Appendix D. Representation-Level Fusion Analysis

This appendix summarizes representative representation-level fusion methods and organizes them by source-model relation, fused parameter/module scope, and evaluated model backbones or settings.

 doi:10.20944/preprints202605.2007.v1

Table A3. Representation-level fusion methods compared by source-model relation, fused parameter/module scope, and evaluated model backbones or settings. Method groups follow the taxonomy in Section 3.

Method	Venue	Source relation	Fused params/modules	Evaluated backbones/settings
Weighting and representation matching				
†REPAIR [49]	ICLR'23	A same architecture after permutation alignment	N	E CNN classifiers
ZipIt! [50]	ICLR'24	A architecture-compatible sources	L A	E vision Transformers/classifiers
Transformer Fusion [51]	ICLR'24	A aligned Transformer variants	L A	E T Transformer encoders/enc–decoders
AIM [46]	NeurIPS'25	S same-base LLM checkpoints	F	D decoder-only LLMs
ACM [107]	arXiv'25	S same-base LLM checkpoints	F	D decoder-only LLMs
MAGIC [47]	arXiv'25	S same-base or aligned sources	F N	E D CV and Llama merging
Merging Beyond [48]	arXiv'26	S sequential same-backbone updates	F L	D streaming LLM updates
Closed-form representation solvers				
RegMean [52]	ICLR'23	S same architecture and tokenizer	L	T language models
RegMean++ [53]	TMLR'26	A same-family compatible models	L	E T D encoder, enc–dec, decoder-only
LoRM [56]	ICLR'25	S PEFT modules over compatible bases	L P	E D LoRA-equipped models
IterIS [57]	CVPR'25	S compatible LoRA adapters	P	M text-to-image, VLM, and LLM adapters
LOT-Merging [54]	NeurIPS'25	S same-base task-vector checkpoints	L N	D Transformer/LLM checkpoints
FeatCal [55]	arXiv'26	H mismatch handled by projection/alignment	L N P	D post-merging decoder-only LLMs
Backpropagation-based representation transfer				
Patient KD [58]	EMNLP'19	T teacher–student fusion	F	E BERT-style encoders
TinyBERT [59]	EMNLP Findings'20	T teacher–student fusion	F	E BERT-style encoders
Task-aware LWD [60]	ICML'23	T teacher–student fusion	F P	E language-model compression
Representation Surgery [7]	ICML'24	S same-base multi-task merged models	P	E encoder-based multi-task models
Surgeryv2 [61]	arXiv'24	S same-base multi-task merged models	P	E aligned-tokenizer settings
ProbSurgery [62]	ICML'25	S same-base multi-task merged models	P	E multi-task model merging
RECALL [14]	EMNLP'25	S continual same-family checkpoints	L N	C in-domain checkpoint sequences
NUFILT [15]	ICLR'26	S continual same-backbone checkpoints	F P	C data-free continual merging

Source relation S same base, tokenizer, or checkpoint trajectory; A architecture-compatible sources requiring alignment/matching; H heterogeneous or projection-needed sources; T teacher–student representation fusion.

Fused scope F full parameters or task vectors; L linear/projection weights; A attention components; N normalization, bias, or activation statistics; P projection, LoRA, adapter, or repair module.

Backbones/settings E encoder or vision backbone; D decoder-only LLM; T encoder–decoder; M multimodal or vision–language setting; C checkpoint sequence or continual-merging setting.

Boundary cases †Non-LLM representation repair included because it motivates representation-level post-merge correction; teacher–student methods are treated as representation-level fusion when intermediate hidden states or attention maps provide the main fusion signal.

 doi:10.20944/preprints202605.2007.v1

Appendix E. Behavior-Level Fusion Analysis

This appendix summarizes representative behavior-level fusion methods and organizes them by source-model relation, behavior signal, and evaluated model backbones or settings.

Table A4. Behavior-level fusion methods compared by source-model relation, behavior signal, state distribution, and evaluated backbones or settings. Method groups follow the taxonomy in Section 3.3.

Method	Venue	Source relation	Behavior signal	Evaluated backbones/settings
Distribution fusion				
Knowledge Distillation [63]	NeurIPS'15	T teacher–student	D O soft outputs	general neural networks
DistilBERT [64]	NeurIPS'19	T BERT teacher–student	D O token distributions	E BERT encoders
InfiGFusion [65]	NeurIPS'25	M multi-source LLMs	D O logit geometry	D decoder-only LLMs
Demonstration fusion				
Instruction Distillation [69]	arXiv'23	T stronger teacher	X O instruction responses	D LLM rankers
GPT4All [70]	GitHub'23	B API teacher	X O assistant demos	C chatbot tuning
Distilling Step-by-Step [71]	ACL'23	T larger LLM teacher	X O rationales and labels	D small reasoning LMs
Teaching Small LMs to Reason [72]	ACL'23	T reasoning teacher	X O CoT rationales	D small LMs
Feedback fusion				
FuseChat [73]	EMNLP'25	M chat-model sources	X F O responses and preferences	C chat fusion
Zephyr [74]	COLM'24	T aligned teacher	F O preference data	C chat alignment
InfiFPO [75]	NeurIPS'25	M source preferences	F O preference optimization	D LLM fusion
On-policy and trajectory-level fusion				
†DAgger [76]	AISTATS'11	T expert policy	R P learner-state actions	A imitation learning
GKD / OPD [9]	ICLR'24	T teacher on student states	D R P self-generated sequences	D autoregressive LMs
Self-Distilled Reasoner [77]	arXiv'26	S self-distillation	X R P reasoning traces	D reasoning LLMs
SODA [78]	arXiv'26	B black-box teacher	X R S semi-on-policy data	D black-box distillation
Lightning OPD [79]	arXiv'26	T teacher-logit source	D S offline OPD signals	D reasoning LLMs
TIP [80]	arXiv'26	T sampled-token teacher	D R P token-importance signals	D token-efficient OPD
Demystifying OPD [81]	arXiv'26	T rollout teacher	D R P stabilized token signals	D OPD stabilization
Video-OPD [82]	arXiv'26	H multimodal teacher	R P video trajectories	M video grounding MLLMs
X-OPD [83]	arXiv'26	H cross-modal teacher	R P speech trajectories	M speech LLM alignment

Source relation T teacher–student or expert–learner relation; M multiple source models or model zoo; B black-box or API-access source; S self-distillation source; H heterogeneous or cross-modal source–target setting.

Behavior signal D output distributions, token probabilities, or logits; X demonstrations, responses, rationales, or traces; F preferences, rankings, scores, critiques, rewards, or verifier labels; R target-induced states, rollouts, or trajectories.

State distribution O off-policy fixed behavior data; P on-policy supervision on states induced by the target model; S semi-on-policy, cached, or offline-reused on-policy-style supervision.

Backbones/settings E encoder-only LM; D decoder-only LLM; C chat or instruction-following LLM; M multimodal, speech, or video-language setting; A imitation-learning or agent-policy setting.

Boundary cases †DAgger is included as the classical on-policy imitation-learning analogue of behavior-level fusion; methods are grouped according to the behavior-level branch in Section 3.3. Labels are descriptive and non-exclusive, since a method may combine distributions, demonstrations, and feedback.

Appendix F. Model Fusion Benchmarks

This appendix summarizes representative model fusion benchmarks and related evaluation resources. We organize them by modality coverage, model and task settings, heterogeneity support, fusion type, and evaluation focus.

 doi:10.20944/preprints202605.2007.v1

Resource	Venue	Vision	Text	LLM	MLLM	Model Pool	Task Pool	Hetero.	Fusion Type	Evaluation Focus	Open
Realistic Evaluation [24]	arXiv'24	Y	Y	N	N	Y	Y	P	Vision / text merging	Acc., compositionality	Y
Model-GLUE [108]	NeurIPS D&B'24	N	N	Y	N	Y	Y	Y	Heterogeneous LLM merging	Model selection, aggregation	Y
MergeKit [109]	EMNLP-I'24	N	N	Y	N	P	P	P	Recipe-based merging	Leaderboard perf.	Y
EMR-Merging [110]	NeurIPS'24	Y	Y	P	P	Y	Y	P	Tuning-free merging	Acc., scalability	Y
Merging at Scale [111]	arXiv'24	N	N	Y	N	Y	Y	N	LLM merging	Scaling, expert count	N
H3Fusion [112]	EACL'26	N	N	Y	N	Y	Y	N	Alignment merging	Helpful., honest., harmless.	P
SMM-Bench [113]	AutoML-N'25	N	N	Y	N	Y	Y	P	Surrogate merge search	Search cost, ranking	Y
Systematic Study [114]	TMLR'26	N	N	Y	N	Y	Y	N	LLM merging study	Method reliability	N
Mergenetic [115]	ACL Demo'25	N	N	Y	N	P	P	P	Evolutionary merging	Fitness, search efficiency	Y
FusionBench [22]	JMLR'25	Y	Y	Y	P	Y	Y	P	Merging / ensemble / mixing	Acc., robust., OOD	Y
MergeBench [23]	NeurIPS D&B'25	N	N	Y	N	Y	Y	N	Domain LLM merging	Acc., forgetting, runtime	Y
OptMerge [116]	ICLR'26	N	N	N	Y	Y	Y	Y	MLLM merging	VQA, OCR, grounding	Y
Merging Scaling Law [117]	ICML'26	N	N	Y	N	Y	Y	N	Large-scale LLM merging	Scaling law, expert count	Y
M2RL [118]	arXiv'26	N	N	Y	N	Y	Y	N	RLVR merging / OPD	Synergy, interference, efficiency	Y

Pool definition	*Model Pool* indicates whether the resource explicitly defines source models, expert models, or fine-tuned checkpoints to be fused; *Task Pool* indicates whether it defines downstream tasks or domain pools for post-merge evaluation.
Heterogeneity	*Hetero.* indicates whether the resource explicitly evaluates heterogeneous fusion, including cross-family, cross-architecture, cross-modal, or heterogeneous-output-space settings.
Open	*Open* indicates whether the resource provides public code, scripts, model pools, evaluation resources, or reproducible configurations.
Symbols	Y explicit support; P partial, implicit, or recipe-dependent support; N not covered or not the focus.
Boundary cases	Toolkits and search ecosystems, such as MergeKit [109] and Mergenetic [115], are included when they provide reusable fusion pipelines or practical evaluation settings. They are not treated as fixed benchmark suites.

Table A5. Representative model fusion benchmarks and related evaluation resources. Unlike traditional LLM benchmarks that mainly define task instances and metrics, model fusion benchmarks often define source model pools, target tasks, fusion settings, and cost or retention axes.

Appendix G. Model Fusion Metrics

This appendix summarizes common evaluation dimensions for model fusion and clarifies when each metric is most useful.

 doi:10.20944/preprints202605.2007.v1

Table A6. Common metrics for model fusion. Avg score and normalized performance are the most direct metrics. Other metrics are useful but depend on the setting, access level, or safety goal.

Dimension	Question	Representative metrics	Use condition
Overall quality	Is the target model strong overall?	Avg score, mean task score, mean capability score.	Core metric for most benchmarks.
Capability retention	How much source capability is kept?	Normalized performance, retention ratio, worst source-task drop.	Core metric when source tasks or source capabilities are known.
Interference and transfer	Does fusion hurt or help related tasks?	Local task drop, negative transfer rate, held-out task score.	Useful when task combinations or held-out settings are defined.
Internal alignment	Are the fusion signals well matched?	Representation drift, output-distribution gap, calibration error.	Requires hidden states, logits, or output distributions.
Efficiency	How costly is fusion and use?	Fusion compute, source-query count, inference latency.	Needed when comparing practical fusion methods.
Safety and risk	Does fusion keep task constraints?	Harmful response rate, backdoor attack success rate, privacy leakage.	Use under a clear safety goal or threat model.

Appendix H. Additional Deployment Challenges

This appendix discusses two deployment-oriented challenges that complement the main challenges in Section 5.

Continual Fusion Can Easily Cause Forgetting.

In real deployment, the target model may continually absorb new source models, domain updates, or safety patches. Each fusion step can overwrite earlier knowledge or weaken previously aligned behavior, especially when old training data, source models, or evaluation signals are unavailable. AIMMerging [13], NUFILT [15], and K-Merge [95] study continual fusion for language models, but stable long-term fusion remains open. Behavior-level methods also face forgetting when new skills are learned from source feedback [96]. A useful direction is to preserve old capabilities, new capabilities, and safety behavior together without full retraining.

Large-Scale Fusion Remains Underexplored.

Model fusion can be cheaper than retraining or using all source models at inference time, but large-scale fusion brings new costs. For parameter-level fusion, MergeKit [109] makes LLM fusion easier to run. MergePipe [119] further shows that expert-parameter I/O and repeated scans become key bottlenecks as the source pool grows. For method search, FusionBench [22] and MergeBench [23] improve standard comparison, but large models still make candidate evaluation costly. For behavior-level fusion, source feedback can also be expensive. Lightning OPD [79] and TIP [80] reduce live teacher serving or token-level supervision cost. Future work can scale model fusion by reducing parameter I/O, candidate search, and source-query cost while preserving source capabilities.

References

1. Li, W.; Peng, Y.; Zhang, M.; Ding, L.; Hu, H.; Shen, L. Deep Model Fusion: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* **2026**, *37*, 2008–2024. https://doi.org/10.1109/TNNLS.2025.3628666.
2. Zheng, H.; Shen, L.; Tang, A.; Luo, Y.; Hu, H.; Du, B.; Wen, Y.; Tao, D. Learning from Models Beyond Fine-Tuning. *Nature Machine Intelligence* **2025**, *7*, 6–17. https://doi.org/10.1038/s42256-024-00961-0.

 doi:10.20944/preprints202605.2007.v1

3. Yang, E.; Shen, L.; Guo, G.; Wang, X.; Cao, X.; Zhang, J.; Tao, D. Model Merging in LLMs, MLLMs, and Beyond: Methods, Theories, Applications, and Opportunities. *ACM Comput. Surv.* **2026**, *58*. https://doi.org/10.1145/3787849.
4. Song, M.; Zheng, M. Model Merging in the Era of Large Language Models: Methods, Applications, and Future Directions, 2026, [arXiv:cs.CL/2603.09938].
5. Yadav, P.; Raffel, C.; Muqeeth, M.; Caccia, L.; Liu, H.; Chen, T.; Bansal, M.; Choshen, L.; Sordoni, A. A Survey on Model MoErging: Recycling and Routing Among Specialized Experts for Collaborative Learning. *Transactions on Machine Learning Research* **2025**.
6. Yadav, P.; Tam, D.; Choshen, L.; Raffel, C.A.; Bansal, M. TIES-Merging: Resolving Interference When Merging Models. In Proceedings of the Advances in Neural Information Processing Systems, 2023.
7. Yang, E.; Shen, L.; Wang, Z.; Guo, G.; Chen, X.; Wang, X.; Tao, D. Representation Surgery for Multi-Task Model Merging. In Proceedings of the Proceedings of the International Conference on Machine Learning, ICML 2024, 2024.
8. Wan, F.; Huang, X.; Cai, D.; Quan, X.; Bi, W.; Shi, S. Knowledge Fusion of Large Language Models. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
9. Agarwal, R.; Vieillard, N.; Zhou, Y.; Stanczyk, P.; Ramos, S.; Geist, M.; Bachem, O. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In Proceedings of the Proceedings of ICLR, 2024.
10. DeepSeek-AI. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. Technical report, DeepSeek-AI, 2026.
11. Yang, Z.; Liu, Z.; Chen, Y.; Dai, W.; Wang, B.; Lin, S.; Lee, C.; Chen, Y.; Jiang, D.; He, J.; et al. Nemotron-Cascade 2: Post-Training LLMs with Cascade RL and Multi-Domain On-Policy Distillation. *CoRR* **2026**, *abs/2603.19220*, [2603.19220]. https://doi.org/10.48550/ARXIV.2603.19220.
12. GLM-5 Team. GLM-5: from Vibe Coding to Agentic Engineering. *CoRR* **2026**, *abs/2602.15763*. https://doi.org/10.48550/arXiv.2602.15763.
13. Feng, Y.; Li, J.; Dong, X.; Xu, P.; Zhou, X.; Zhang, Y.; LU, Z.; Wang, Y.; Zhao, A.; Chu, X.; et al. AIMMerging: Adaptive Iterative Model Merging Using Training Trajectories for Language Model Continual Learning. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 2025; pp. 13420–13437. https://doi.org/10.18653/v1/2025.emnlp-main.678.
14. Wang, B.; Wan, H.; Shi, L.; Yang, C.; He, P.; Ma, Y.; Han, H.; Li, W.; Tan, T.; Li, Y.; et al. RECALL: REpresentation-aligned Catastrophic-forgetting ALLeviation via Hierarchical Model Merging. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 2025; pp. 16381–16395. https://doi.org/10.18653/v1/2025.emnlp-main.829.
15. Qiu, Z.; Wang, L.; Cao, Y.; Zhang, R.; Su, B.; Xu, Y.; Meng, F.; Xu, L.; Wu, Q.; Li, H. Null-Space Filtering for Data-Free Continual Model Merging: Preserving Stability, Promoting Plasticity. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
16. Wortsman, M.; Ilharco, G.; Gadre, S.Y.; Roelofs, R.; Lopes, R.G.; Morcos, A.S.; Namkoong, H.; Farhadi, A.; Carmon, Y.; Kornblith, S.; et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Proceedings of the Proceedings of the International Conference on Machine Learning, ICML 2022, 2022.
17. Ainsworth, S.K.; Hayase, J.; Srinivasa, S.S. Git Re-Basin: Merging Models modulo Permutation Symmetries. In Proceedings of the Proceedings of the International Conference on Learning Representations, ICLR 2023, 2023.
18. Cao, Y.; Ran, D.; Guo, Y.; Wu, M.; Chen, S.; Li, L.; Yang, W.; Xie, T. An Empirical Study and Theoretical Explanation on Task-Level Model-Merging Collapse. *CoRR* **2026**, *abs/2603.09463*.
19. Sung, Y.L.; Li, L.; Lin, K.; Gan, Z.; Bansal, M.; Wang, L. An Empirical Study of Multimodal Model Merging. In Proceedings of the Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, 2023.
20. Cui, C.; Yang, B.; Shen, F.; Chen, Y.; Zheng, J.; Wang, X.; Zhang, A.; Chua, T.S. Transport and Merge: Cross-Architecture Merging for Large Language Models. *CoRR* **2026**, *abs/2602.05495*.
21. Du, Y.; Wang, X.; Chen, C.; Ye, J.; Wang, Y.; Li, P.; Yan, M.; Zhang, J.; Huang, F.; Sui, Z.; et al. AdaMMS: Model Merging for Heterogeneous Multimodal Large Language Models with Unsupervised Coefficient Optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, 2025.
22. Tang, A.; Shen, L.; Luo, Y.; Yang, E.; Hu, H.; Zhang, L.; Du, B.; Tao, D. FusionBench: A Unified Library and Comprehensive Benchmark for Deep Model Fusion. *Journal of Machine Learning Research* **2025**, *26*, 1–38.

 doi:10.20944/preprints202605.2007.v1

23. He, Y.; Zeng, S.; Hu, Y.; Yang, R.; Zhang, T.; Zhao, H. MergeBench: A Benchmark for Merging Domain-Specialized LLMs. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2025.
24. Tam, D.; Kant, Y.; Lester, B.; Gilitschenski, I.; Raffel, C. Realistic Evaluation of Model Merging for Compositional Generalization. *CoRR* **2024**, *abs/2409.18314*.
25. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge Distillation: A Survey. *Int. J. Comput. Vision* **2021**, *129*, 1789–1819. https://doi.org/10.1007/s11263-021-01453-z.
26. Xu, X.; Li, M.; Tao, C.; Shen, T.; Cheng, R.; Li, J.; Xu, C.; Tao, D.; Zhou, T. A Survey on Knowledge Distillation of Large Language Models. *CoRR* **2024**, *abs/2402.13116*.
27. Yang, C.; Zhu, Y.; Lu, W.; Wang, Y.; Chen, Q.; Gao, C.; Yan, B.; Chen, Y. Survey on Knowledge Distillation for Large Language Models: Methods, Evaluation, and Application. *ACM Trans. Intell. Syst. Technol.* **2025**, *16*. https://doi.org/10.1145/3699518.
28. Qin, L.; Zhu, T.; Zhou, W.; Yu, P.S. Knowledge Distillation in Federated Learning: A Survey on Long Lasting Challenges and New Solutions. *International Journal of Intelligent Systems* **2025**, *2025*. https://doi.org/10.1155/int/7406934.
29. Song, M.; Zheng, M. A Survey of On-Policy Distillation for Large Language Models, 2026, [arXiv:cs.LG/2604.00626].
30. Fang, L.; Yu, X.; Cai, J.; Chen, Y.; Wu, S.; Liu, Z.; Yang, Z.; Lu, H.; Gong, X.; Liu, Y.; et al. Knowledge Distillation and Dataset Distillation of Large Language Models: Emerging Trends, Challenges, and Future Directions. *Artificial Intelligence Review* **2026**, *59*. https://doi.org/10.1007/s10462-025-11423-3.
31. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.P.; Wilson, A.G. Averaging Weights Leads to Wider Optima and Better Generalization. In Proceedings of the Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI 2018, 2018.
32. Ilharco, G.; Ribeiro, M.T.; Wortsman, M.; Schmidt, L.; Hajishirzi, H.; Farhadi, A. Editing models with task arithmetic. In Proceedings of the Proceedings of the International Conference on Learning Representations, 2023.
33. Yu, L.; Yu, B.; Yu, H.; Huang, F.; Li, Y. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning. PMLR, 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 57755–57775.
34. Deep, P.T.; Bhardwaj, R.; Poria, S. DELLA-Merging: Reducing Interference in Model Merging through Magnitude-Based Sampling. *CoRR* **2024**, *abs/2406.11617*. https://doi.org/10.48550/arxiv.2406.11617.
35. Stoica, G.; Ramesh, P.; Ecsedi, B.; Choshen, L.; Hoffman, J. Model merging with SVD to tie the Knots. In Proceedings of the Proceedings of the International Conference on Learning Representations, 2024.
36. Marczak, D.; Magistri, S.; Cygert, S.; Twardowski, B.; Bagdanov, A.D.; van de Weijer, J. No Task Left Behind: Isotropic Model Merging with Common and Task-Specific Subspaces. In Proceedings of the Proceedings of the International Conference on Machine Learning, ICML 2025, 2025.
37. Gargiulo, A.A.; Crisostomi, D.; Bucarelli, M.S.; Scardapane, S.; Silvestri, F.; Rodola, E. Task Singular Vectors: Reducing Task Interference in Model Merging. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025.
38. Yang, E.; Tang, A.; Shen, L.; Guo, G.; Wang, X.; Cao, X.; Zhang, J. Continual model merging without data: Dual projections for balancing stability and plasticity. *Advances in Neural Information Processing Systems* **2026**, *38*, 39275–39305.
39. Xiong, F.; Cheng, R.; Chen, W.; Zhang, Z.; Guo, Y.; Yuan, C.; Xu, R. Multi-Task Model Merging via Adaptive Weight Disentanglement. *CoRR* **2024**, *abs/2411.18729*.
40. Cheng, R.; Xiong, F.; Wei, Y.; Zhu, W.; Yuan, C. Whoever Started the Interference Should End It: Guiding Data-Free Model Merging via Task Vectors. In Proceedings of the Proceedings of the 42nd International Conference on Machine Learning. PMLR, 2025, Vol. 267, *Proceedings of Machine Learning Research*, pp. 10121–10143.
41. Wang, Y.; Yang, Y.; Lu, S.; Gu, Y.; Wang, P.; Wang, W.; Yan, Z.; Xie, C.; Wu, J.; Cao, J.; et al. Geometry Conflict: Explaining and Controlling Forgetting in LLM Continual Post-Training. *arXiv preprint arXiv:2605.09608* **2026**.
42. Wei, Y.; Tang, A.; Shen, L.; Hu, Z.; Yuan, C.; Cao, X. Modeling Multi-Task Model Merging as Adaptive Projective Gradient Descent. In Proceedings of the Proceedings of the International Conference on Machine Learning, ICML 2025, 2025.

 doi:10.20944/preprints202605.2007.v1

43. Chronopoulou, A.; Peters, M.E.; Fraser, A.; Dodge, J. AdapterSoup: Weight Averaging to Improve Generalization of Pretrained Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EACL 2023, 2023.
44. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the Proceedings of the International Conference on Learning Representations, ICLR 2022, 2022.
45. Prabhakar, A.; Li, Y.; Narasimhan, K.; Kakade, S.; Malach, E.; Jelassi, S. LoRA Soups: Merging LoRAs for Practical Skill Composition Tasks. In Proceedings of the Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, 2025.
46. Nobari, A.H.; Alimohammadi, K.; ArjomandBigdeli, A.; Srivastava, A.; Ahmed, F.; Azizan, N. Activation-Informed Merging of Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems, 2025.
47. Li, Y.; Zhang, J.; Guo, J.; Cheng, Z.; Qi, L.; Shi, Y.; Gao, Y. MAGIC: Achieving Superior Model Merging via Magnitude Calibration. *CoRR* **2025**, *abs/2512.19320*, [2512.19320]. https://doi.org/10.48550/ARXIV.2512.19320.
48. Yao, Y.; Sheng, H.; Lv, Q.; Wu, H.; Liu, S.; Liu, Z.; Liu, Z.; Gao, J.; Tan, H.; Fu, X.; et al. Merging Beyond: Streaming LLM Updates via Activation-Guided Rotations. *CoRR* **2026**, *abs/2602.03237*.
49. Jordan, K.; Sedghi, H.; Saukh, O.; Entezari, R.; Neyshabur, B. REPAIR: Renormalizing Permuted Activations for Interpolation Repair. In Proceedings of the Proceedings of the International Conference on Learning Representations, ICLR 2023, 2023.
50. Stoica, G.; Bolya, D.; Bjorner, J.; Ramesh, P.; Hearn, T.; Hoffman, J. ZipIt! Merging Models from Different Tasks without Training. In Proceedings of the Proceedings of the International Conference on Learning Representations, ICLR 2024, 2024.
51. Imfeld, M.; Graldi, J.; Giordano, M.; Hofmann, T.; Anagnostidis, S.; Singh, S.P. Transformer Fusion with Optimal Transport. In Proceedings of the Proceedings of the International Conference on Learning Representations, ICLR 2024, 2024.
52. Jin, X.; Ren, X.; Preotiuc-Pietro, D.; Cheng, P. Dataless Knowledge Fusion by Merging Weights of Language Models. In Proceedings of the Proceedings of the International Conference on Learning Representations, ICLR 2023, 2023.
53. Nguyen, T.H.; Huu-Tien, D.; Suzuki, T.; Nguyen, L.M. RegMean++: Enhancing Effectiveness and Generalization of Regression Mean for Model Merging. *Transactions on Machine Learning Research* **2026**. Expert Certification.
54. Sun, W.; Li, Q.; Wang, W.; Liu, Y.; Geng, Y.a.; Li, B. Towards minimizing feature drift in model merging: Layer-wise task vector fusion for adaptive knowledge integration. In Proceedings of the Advances in Neural Information Processing Systems, 2025.
55. Gu, Y.; Cai, S.; Wang, Z.; Wang, W.; Wang, Y.; Wang, P.; Huang, S.; Lu, S.; Wu, J.; Yang, H. FeatCal: Feature Calibration for Post-Merging Models, 2026, [arXiv:cs.LG/2605.13030].
56. Salami, R.; Buzzega, P.; Mosconi, M.; Bonato, J.; Sabetta, L.; Calderara, S. Closed-Form Merging of Parameter-Efficient Modules for Federated Continual Learning. In Proceedings of the Proceedings of the International Conference on Learning Representations, ICLR 2025, 2025.
57. Chen, H.; Li, R.; Zhu, B.; Wang, Z.; Chen, L. IterIS: Iterative Inference-Solving Alignment for LoRA Merging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, 2025.
58. Sun, S.; Cheng, Y.; Gan, Z.; Liu, J. Patient Knowledge Distillation for BERT Model Compression. In Proceedings of the Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019.
59. Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; Liu, Q. TinyBERT: Distilling BERT for Natural Language Understanding. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.372.
60. Liang, C.; Zuo, S.; Zhang, Q.; He, P.; Chen, W.; Zhao, T. Less is more: Task-aware layer-wise distillation for language model compression. In Proceedings of the International Conference on Machine Learning, 2023.
61. Yang, E.; Shen, L.; Wang, Z.; Guo, G.; Wang, X.; Cao, X.; Zhang, J.; Tao, D. Surgeryv2: Bridging the gap between model merging and multi-task learning with deep representation surgery. *arXiv preprint arXiv:2410.14389* **2024**.
62. Wei, Q.; He, S.; Yang, E.; Liu, T.; Wang, H.; Feng, L.; An, B. Representation Surgery in Model Merging with Probabilistic Modeling. In Proceedings of the Proceedings of the International Conference on Machine Learning, ICML 2025, 2025.

 doi:10.20944/preprints202605.2007.v1

63. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *CoRR* **2015**, *abs/1503.02531*.
64. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* **2019**.
65. Wang, Y.; Yan, Z.; Zhang, Y.; Zhou, Q.; Gu, Y.; Wu, F.; Yang, H. InfiGFusion: Graph-on-Logits Distillation via Efficient Gromov-Wasserstein for Model Fusion. In Proceedings of the Advances in Neural Information Processing Systems, 2025.
66. Yang, E.; Wang, Z.; Shen, L.; Liu, S.; Guo, G.; Wang, X.; Tao, D. AdaMerging: Adaptive Model Merging for Multi-Task Learning. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
67. Yu, S.J.; Choi, S. Parameter-Efficient Checkpoint Merging via Metrics-Weighted Averaging. *CoRR* **2025**, *abs/2504.18580*, [2504.18580]. https://doi.org/10.48550/ARXIV.2504.18580.
68. Matena, M.; Raffel, C. Merging Models with Fisher-Weighted Averaging. In Proceedings of the Advances in Neural Information Processing Systems 35, NeurIPS 2022, 2022.
69. Sun, W.; Chen, Z.; Ma, X.; Yan, L.; Wang, S.; Ren, P.; Chen, Z.; Yin, D.; Ren, Z. Instruction Distillation Makes Large Language Models Efficient Zero-shot Rankers. In Proceedings of the 2023, 2023.
70. Anand, Y.; Nussbaum, Z.; Duderstadt, B.; Schmidt, B.; Mulyar, A. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. In Proceedings of the GitHub, 2023.
71. Hsieh, C.Y.; Li, C.L.; Yeh, C.K.; Nakhost, H.; Fujii, Y.; Ratner, A.; Krishna, R.; Lee, C.Y.; Pfister, T. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In Proceedings of the Findings of ACL, 2023.
72. Magister, L.C.; Mallinson, J.; Adamek, J.; Malmi, E.; Severyn, A. Teaching Small Language Models to Reason. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2023. https://doi.org/10.18653/v1/2023.acl-short.151.
73. Wan, F.; Yang, Z.; Zhong, L.; Quan, X.; Huang, X.; Bi, W. FuseChat: Knowledge Fusion of Chat Models. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, 2025.
74. Tunstall, L.; Beeching, E.; Lambert, N.; Rajani, N.; Rasul, K.; Belkada, Y.; Huang, S.; von Werra, L.; Fourrier, C.; Habib, N.; et al. Zephyr: Direct Distillation of LM Alignment. In Proceedings of the First Conference on Language Modeling, 2024.
75. Gu, Y.; Yan, Z.; Wang, Y.; Zhang, Y.; Zhou, Q.; Wu, F.; Yang, H. InfiFPO: Implicit Model Fusion via Preference Optimization in Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems, 2025.
76. Ross, S.; Gordon, G.J.; Bagnell, J.A. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Proceedings of the Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, 2011.
77. Zhao, S.; Xie, Z.; Liu, M.; Huang, J.; Pang, G.; Chen, F.; Grover, A. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *CoRR* **2026**, *abs/2601.18734*.
78. Chen, X.; Wang, J.; Zhu, W.; Qiu, P.; Dong, X.; Sang, H.; Wang, Z.; Geramifard, A.; Luo, F. SODA: Semi On-Policy Black-Box Distillation for Large Language Models. *arXiv preprint arXiv:2604.03873* **2026**.
79. Wu, Y.; Han, S.; Cai, H. Lightning OPD: Efficient Post-Training for Large Reasoning Models with Offline On-Policy Distillation. *arXiv preprint arXiv:2604.13010* **2026**.
80. Xu, Y.; Sang, H.; Zhou, Z.; He, R.; Wang, Z.; Geramifard, A. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084* **2026**.
81. Luo, F.; Chuang, Y.N.; Wang, G.; Xu, Z.; Han, X.; Zhang, T.; Braverman, V. Demystifying OPD: Length Inflation and Stabilization Strategies for Large Language Models. *arXiv preprint arXiv:2604.08527* **2026**.
82. Li, J.; Yin, H.; Xu, H.; Xu, B.; Tan, W.; He, Z.; Ju, J.; Luo, Z.; Luan, J. Video-OPD: Efficient Post-Training of Multimodal Large Language Models for Temporal Video Grounding via On-Policy Distillation. *CoRR* **2026**, *abs/2602.02994*.
83. Cao, D.; Fu, D.; Yu, H.; Zheng, S.; Tan, X.; Jin, T. X-OPD: Cross-Modal On-Policy Distillation for Capability Alignment in Speech LLMs. *CoRR* **2026**, *abs/2603.24596*.
84. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.P.; et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In Proceedings of the Advances in Neural Information Processing Systems, Datasets and Benchmarks Track, 2023.
85. Wang, H.; Long, X.; Li, Z.; Xu, Y.; Li, T.; Tang, Y. To Mix or To Merge: Toward Multi-Domain Reinforcement Learning for Large Language Models. *CoRR* **2026**, *abs/2602.12566*.

 doi:10.20944/preprints202605.2007.v1

86. Zhou, L.; Zhao, B.; Yu, R.; Rodola, E. Demystifying Mergeability: Interpretable Properties to Predict Model Merging Success. *CoRR* **2026**, *abs/2601.22285*.
87. Fu, Y.; Huang, H.; Jiang, K.; Liu, J.; Jiang, Z.; Zhu, Y.; Zhao, D. Revisiting On-Policy Distillation: Empirical Failure Modes and Simple Fixes. *arXiv preprint arXiv:2603.25562* **2026**.
88. Wang, W. Knowledge Distillation Must Account for What It Loses. *arXiv preprint arXiv:2604.25110* **2026**.
89. Liu, H.; Liu, Z.; Tang, R.; Yuan, J.; Zhong, S.; Chuang, Y.N.; Li, L.; Chen, R.; Hu, X. LoRA-as-an-Attack! Piercing LLM Safety Under The Share-and-Play Scenario. *arXiv preprint arXiv:2403.00108* **2024**.
90. Yuan, Z.; Xu, Y.; Shi, J.; Zhou, P.; Sun, L. Merge Hijacking: Backdoor Attacks to Model Merging of Large Language Models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, 2025. https://doi.org/10.18653/v1/2025.acl-long.1571.
91. Lu, L.; Zuo, Z.; Sheng, Z.; Zhou, P. Merger-as-a-Stealer: Stealing Targeted PII from Aligned LLMs with Model Merging. In Proceedings of the Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, 2025.
92. Cong, T.; Ran, D.; Liu, Z.; He, X.; Liu, J.; Gong, Y.; Li, Q.; Wang, A.; Wang, X. Have You Merged My Model? On The Robustness of Large Language Model IP Protection Methods Against Model Merging. In Proceedings of the Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis, 2024, pp. 69–76. https://doi.org/10.1145/3689217.3690614.
93. Yang, Z.; Ding, W.; Feng, S.; Tsvetkov, Y. Among Us: Measuring and Mitigating Malicious Contributions in Model Collaboration Systems. *CoRR* **2026**, *abs/2602.05176*.
94. Khadem, F.; Mousavi, S.; Fang, Y.; Liu, Y. DP-OPD: Differentially Private On-Policy Distillation for Language Models. *arXiv preprint arXiv:2604.04461* **2026**.
95. Shenaj, D.; Bohdal, O.; Ceritli, T.; Ozay, M.; Zanuttigh, P.; Michieli, U. K-Merge: Online Continual Merging of Adapters for On-device Large Language Models. *CoRR* **2025**, *abs/2510.13537*.
96. Shenfeld, I.; Damani, M.; Hübotter, J.; Agrawal, P. Self-Distillation Enables Continual Learning. In Proceedings of the ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving, 2026.
97. Dmonte, A.; Gupta, V.; Perry, D.J.; Arehart, M. Improving Training Efficiency and Reducing Maintenance Costs via Language Specific Model Merging. In Proceedings of the CoRR, 2026.
98. Djuhera, A.; Kadhe, S.R.; Ahmed, F.; Zawad, S.; Boche, H. SafeMERGE: Preserving Safety Alignment in Fine-Tuned Large Language Models via Selective Layer-Wise Model Merging. *CoRR* **2025**, *abs/2503.17239*, [2503.17239]. https://doi.org/10.48550/ARXIV.2503.17239.
99. Zaman, K.; Choshen, L.; Srivastava, S. Fuse to Forget: Bias Reduction and Selective Memorization through Model Fusion. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024. https://doi.org/10.18653/v1/2024.emnlp-main.1045.
100. Yi, X.; Zheng, S.; Wang, L.; Wang, X.; He, L. A safety realignment framework via subspace-oriented model fusion for large language models. *arXiv preprint arXiv:2405.09055* **2024**.
101. Qin, R.; Wang, Q.; Liu, D.; Li, Q.; Wei, Z.; Shen, W. Multilingual Safety Alignment via Self-Distillation. *arXiv preprint arXiv:2605.02971* **2026**.
102. Hammoud, H.A.A.K.; Michieli, U.; Pizzati, F.; Torr, P.; Bibi, A.; Ghanem, B.; Ozay, M. Model Merging and Safety Alignment: One Bad Model Spoils the Bunch. In Proceedings of the Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2024.
103. DeepSeek-AI.; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Wang, P.; Zhu, Q.; Xu, R.; Zhang, R.; Ma, S.; et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR* **2025**, *abs/2501.12948*.
104. Biggs, B.; Seshadri, A.; Zou, Y.; Jain, A.; Golatkar, A.; Xie, Y.; Achille, A.; Swaminathan, A.; Soatto, S. Diffusion Soup: Model Merging for Text-to-Image Diffusion Models. In Proceedings of the Computer Vision – ECCV 2024, 2024, Vol. 15121, *Lecture Notes in Computer Science*, pp. 257–274. https://doi.org/10.1007/978-3-031-73036-8_15.
105. Ceritli, T.; Bohdal, O.; Ozay, M.; Moon, J.; Lee, K.; Ko, H.; Michieli, U. HydraOpt: Navigating the Efficiency-Performance Trade-off of Adapter Merging. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 2025; pp. 26887–26909. https://doi.org/10.18653/v1/2025.emnlp-main.1365.
106. Wang, H.; Ping, B.; Wang, S.; Han, X.; Chen, Y.; Liu, Z.; Sun, M. LoRA-Flow: Dynamic LoRA Fusion for Large Language Models in Generative Tasks. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024, 2024.

 doi:10.20944/preprints202605.2007.v1

107. Yao, Y.; Liu, S.; Liu, Z.; Li, Q.; Liu, M.; Han, X.; Guo, Z.; Wu, H.; Song, L. Activation-Guided Consensus Merging for Large Language Models. *CoRR* **2025**, *abs/2505.14009*.
108. Zhao, X.; Sun, G.; Cai, R.; Zhou, Y.; Li, P.; Wang, P.; Tan, B.; He, Y.; Chen, L.; Liang, Y.; et al. Model-GLUE: Democratized LLM Scaling for A Large Model Zoo in the Wild. In Proceedings of the Advances in Neural Information Processing Systems 37, NeurIPS 2024, 2024, pp. 13349–13371.
109. Goddard, C.; Siriwardhana, S.; Ehghaghi, M.; Meyers, L.; Karpukhin, V.; Benedict, B.; McQuade, M.; Solawetz, J. Arcee's MergeKit: A Toolkit for Merging Large Language Models. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2024. https://doi.org/10.18653/v1/2024.emnlp-industry.36.
110. Huang, C.; Ye, P.; Chen, T.; He, T.; Yue, X.; Ouyang, W. EMR-Merging: Tuning-Free High-Performance Model Merging. In Proceedings of the Advances in Neural Information Processing Systems 37, NeurIPS 2024, 2024.
111. Khalifa, M.; Tan, Y.C.; Ahmadian, A.; Hosking, T.; Lee, H.; Wang, L.; Ustun, A.; Sherborne, T.; Galle, M. If You Can't Use Them, Recycle Them: Optimizing Merging at Scale Mitigates Performance Tradeoffs. *CoRR* **2024**, *abs/2412.04144*.
112. Tekin, S.F.; Ilhan, F.; Hu, S.; Huang, T.; Xu, Y.; Yahn, Z.; Liu, L. H3Fusion: Helpful, Harmless, Honest Fusion of Aligned LLMs. In Proceedings of the Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2026, pp. 6993–7013.
113. Akizuki, R.; Kudo, Y.; Yoshinari, N.; Hirose, Y.; Nishimoto, T.; Uchida, K.; Shirakawa, S. Surrogate Benchmarks for Model Merging Optimization. In Proceedings of the AutoML 2025 Non-Archival Content Track, 2025.
114. Hitit, O.K.; Girrbach, L.; Akata, Z. A Systematic Study of In-the-Wild Model Merging for Large Language Models. *Transactions on Machine Learning Research* **2026**.
115. Minut, A.R.; Mencattini, T.; Santilli, A.; Crisostomi, D.; Rodolà, E. Mergenetic: a Simple Evolutionary Model Merging Library. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations); Mishra, P.; Muresan, S.; Yu, T., Eds., Vienna, Austria, 2025; pp. 572–582. https://doi.org/10.18653/v1/2025.acl-demo.55.
116. Wei, Y.; Cheng, R.; Jin, W.; Yang, E.; Shen, L.; Hou, L.; Du, S.; Yuan, C.; Cao, X.; Tao, D. OptMerge: Unifying Multimodal LLM Capabilities And Modalities Via Model Merging. *CoRR* **2025**, *abs/2505.19892*.
117. Wang, Y.; Gu, Y.; Zhang, Y.; Zhou, Q.; Yan, Z.; Xie, C.; Wang, X.; Yuan, J.; Yang, H. Model Merging Scaling Laws in Large Language Models. *CoRR* **2025**, *abs/2509.24244*.
118. Wang, H.; Long, X.; Li, Z.; Xu, Y.; Li, T.; Tang, Y. To Mix or To Merge: Toward Multi-Domain Reinforcement Learning for Large Language Models, 2026, [arXiv:cs.AI/2602.12566].
119. Wang, Y.; Gu, Y.; Wang, Z.; Li, K.; Yang, Y.; Yan, Z.; Xie, C.; Wu, J.; Yang, H. MergePipe: A Budget-Aware Parameter Management System for Scalable LLM Merging. *CoRR* **2026**, *abs/2602.13273*.