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: 4 June 2026

doi: 10.20944/preprints202605.2007.v2

Keywords: model fusion; large language models; knowledge transfer

 doi:10.20944/preprints202605.2007.v2

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 [4], Guanghao Zhu [1], Sirui Huang [1], Ming Li [1,5,*] 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] InfiX.ai
[5] PolyU-Daya Bay Technology and Innovation Research Institute
* Correspondence: ming.li@polyu.edu.hk (M.L.); hongxia.yang@polyu.edu.hk (H.Y.)
† Equal contribution.

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. A comprehensive list of papers about model fusion is available at https://github.com/Baicaihaochi/Awesome-Model-Fusion-Survey.

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 (Li et al. 2026b; Zheng et al. 2025).

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 (Li et al. 2026b; Song and Zheng 2026a; Yadav et al. 2025; Yang et al. 2026a).

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 (Agarwal et al. 2024; Wan et al. 2024; Yadav et al. 2023; Yang et al. 2024a). 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 (DeepSeek-AI 2026), NVIDIA's Nemotron-Cascade 2 (Yang et al. 2026c), and GLM-5 (GLM-5 Team 2026) adopt on-policy distillation to integrate or recover model capabilities. Second, model fusion supports continual learning by absorbing

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

 doi:10.20944/preprints202605.2007.v2

new task signals while preserving earlier capabilities. AIMMerging (Feng et al. 2025), RECALL (Wang et al. 2025a), and NUFILT (Qiu et al. 2026) show how fusion can add new knowledge while reducing forgetting.

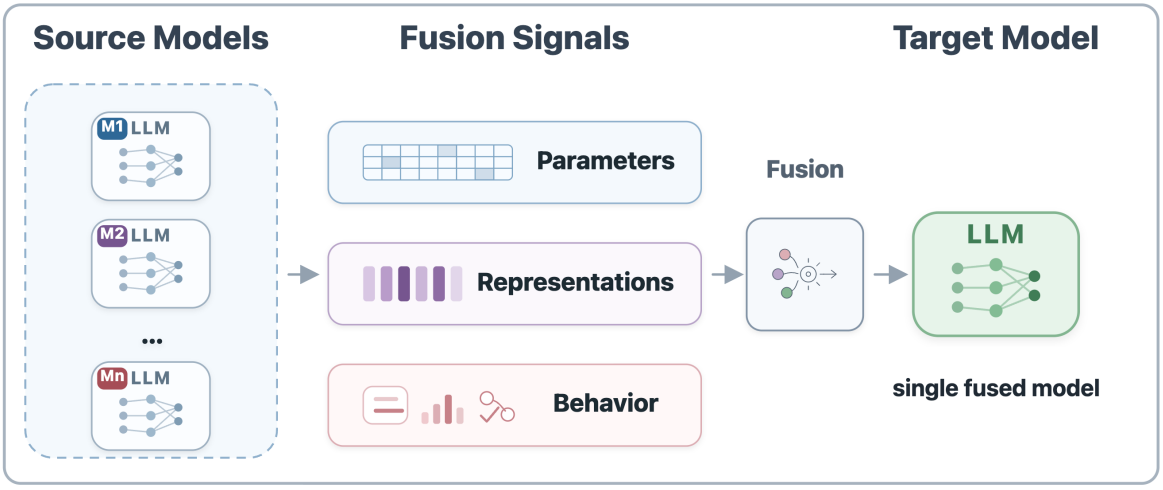

Figure 1. Model fusion overview. Source models provide different signals to form a single target model.

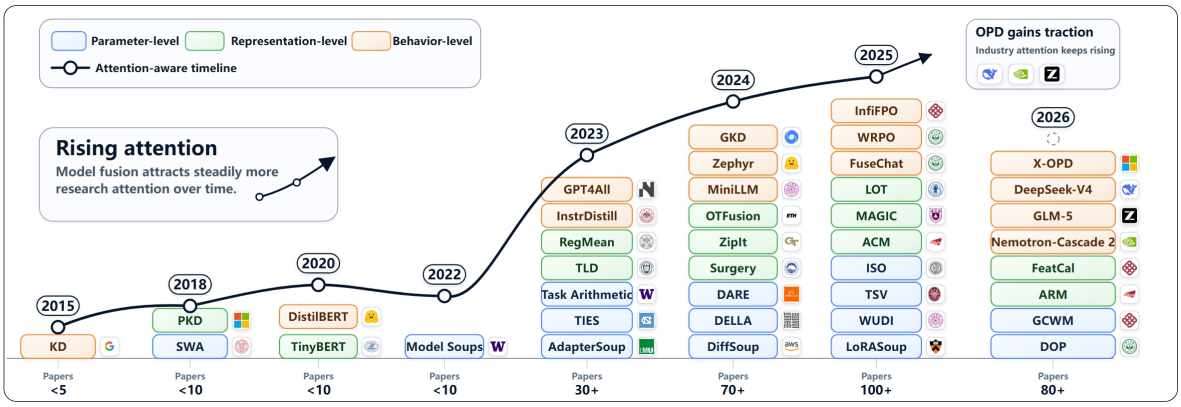

Figure 2. Progress timeline of model fusion.

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 (Ainsworth et al. 2023; Cao et al. 2026b; Wortsman et al. 2022). Moreover, fusion becomes harder when source models differ in architecture, tokenizer, or modality, because parameter and representation alignment can be unstable (Cui et al. 2026; Du et al. 2025; Sung et al. 2023). In evaluation, recent benchmarks improve standardization, but average scores can still hide local degradation and cross-capability interference (Cao et al. 2026b; He et al. 2025; Tam et al. 2024; Tang et al. 2025).

As shown in Table 1, existing surveys mainly systematize model merging or knowledge transfer as separate topics (Fang et al. 2026; Gou et al. 2021; Qin et al. 2025; Song and Zheng 2026b; Xu et al. 2024; Yang et al. 2025). 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.v2

Table 1. Coverage of related surveys.

Survey	Venue & Year	Param. level	Repre. level	Behav. level
Gou et al. (2021)	IJCV'21		✓	✓
Xu et al. (2024)	arXiv'24		✓	✓
Yadav et al. (2025)	TMLR'25	✓		
Yang et al. (2025)	TIST'25		✓	✓
Qin et al. (2025)	IJIS'25	✓		✓
Yang et al. (2026a)	CSUR'26	✓	✓	
Song and Zheng (2026a)	arXiv'26	✓	✓	
Li et al. (2026b)	TNNLS'26	✓	✓	
Song and Zheng (2026b)	arXiv'26			✓
Fang et al. (2026)	AIR'26		✓	✓
Ours		✓	✓	✓

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:

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

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.v2

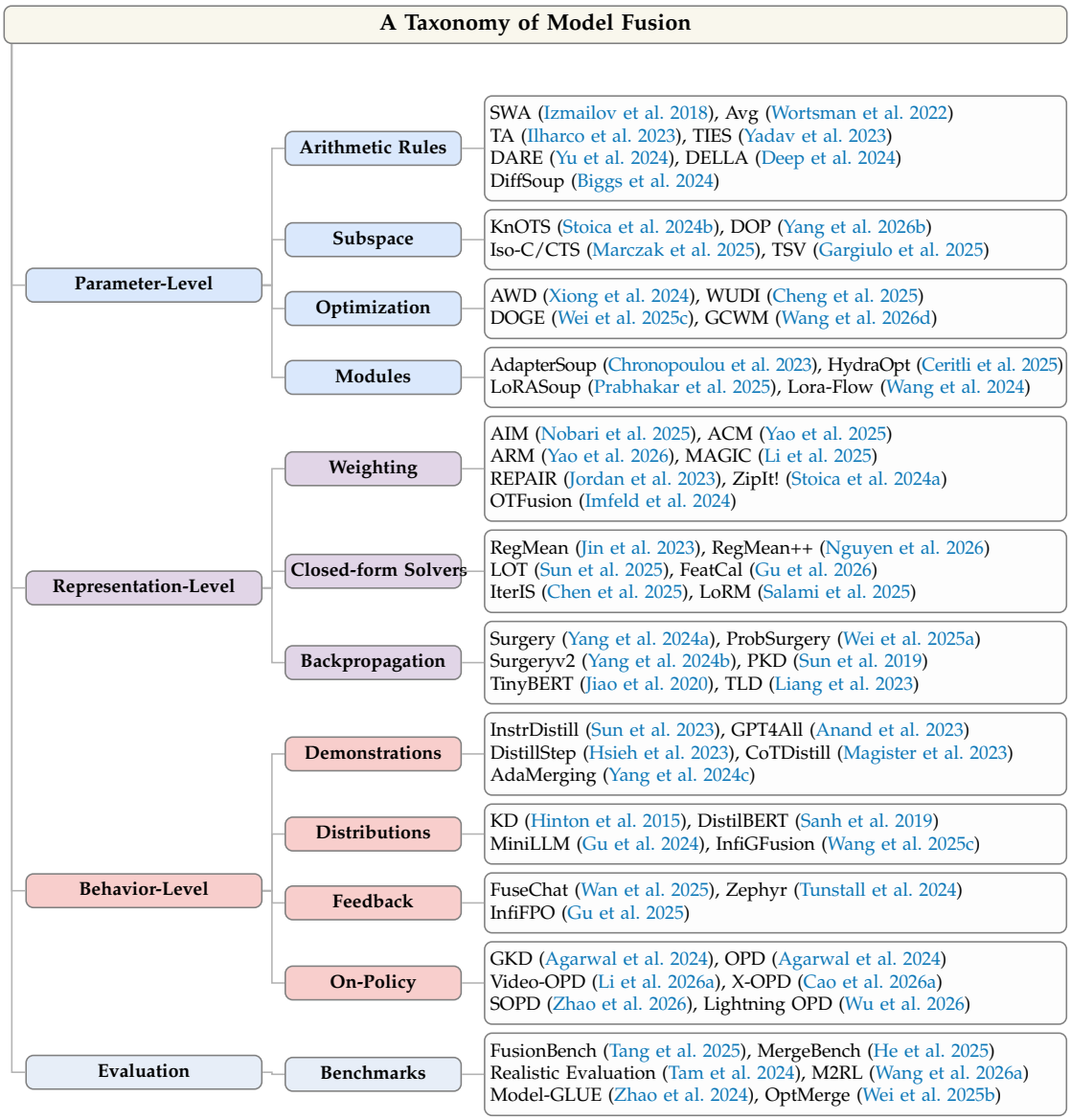

Figure 3. A taxonomy of model fusion for LLMs and MLLMs. The method branches are organized by the main object being fused or aligned: parameters, representations, or behaviors. The evaluation branch summarizes representative benchmark resources. Representative methods and resources are illustrative rather 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 (Izmailov et al. 2018) 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 (Wortsman et al. 2022) show that multiple fine-tuned models can be averaged when they lie in a nearby parameter basin, while task arithmetic (Ilharco et al. 2023) 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

 doi:10.20944/preprints202605.2007.v2

deltas. For example, TIES-Merging, DARE, and DELLA-Merging reduce interference through sign consistency, random dropping or rescaling (Deep et al. 2024; Yadav et al. 2023; Yu et al. 2024).

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 (Gargiulo et al. 2025; Marczak et al. 2025; Stoica et al. 2024b). For continual fusion, DOP (Yang et al. 2026b) 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 (Xiong et al. 2024) optimizes a decomposition of task vectors into redundant and disentangled components, improving orthogonality while preserving task-specific performance. WUDI (Cheng et al. 2025) 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 (Wang et al. 2026d) and DOGE (Wei et al. 2025c) 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 (Chronopoulou et al. 2023), which averages selected domain adapters, and LoRA soups (Hu et al. 2022; Prabhakar et al. 2025), 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.

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 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 (Nobari et al. 2025) estimates weight saliency from activation magnitudes on a task-agnostic calibration set. MAGIC (Li et al. 2025) calibrates representation and weight magnitudes, while Merging Beyond (Yao et al. 2026) uses activation subspaces to form rotation-aware updates. Related alignment methods compute correspondence from representations before fusion: REPAIR (Jordan et al. 2023) rescales preactivations, ZipIt! (Stoica et al. 2024a) matches units by activation similarity, and Transformer Fusion (Imfeld et al. 2024) 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 (Jin et al. 2023) uses input covariance to merge each linear module so that its output matches source-module outputs. RegMean++ (Nguyen

 doi:10.20944/preprints202605.2007.v2

et al. 2026) improves this local view by adding intra-layer and cross-layer dependencies. LOT-Merging (Sun et al. 2025) and FeatCal (Gu et al. 2026) 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 (Salami et al. 2025) applies output matching to low-rank modules, and IterIS (Chen et al. 2025) 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 (Sun et al. 2019) matches hidden states from selected source layers, while TinyBERT (Jiao et al. 2020) extends the signal to embeddings, attention maps, hidden states, and predictions. Task-aware layer-wise distillation (Liang et al. 2023) 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 (Yang et al. 2024a) learns a lightweight module to correct final-layer representation bias. Surgeryv2 (Yang et al. 2024b) extends this repair across multiple layers. ProbSurgery (Wei et al. 2025a) 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.

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 (Hinton et al. 2015), while DistilBERT shows its effectiveness for language model compression (Sanh et al. 2019). 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 (Wang et al. 2025c). 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 (Yang et al. 2024c); MWA weights checkpoints by training

 doi:10.20944/preprints202605.2007.v2

metrics such as loss or training step (Yu and Choi 2025); and Fisher Merging uses sample-estimated Fisher information to approximate posterior precision for parameter-wise averaging (Matena and Raffel 2022).

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 (Anand et al. 2023; Sun et al. 2023), while rationale or step-level distillation transfers intermediate reasoning processes (Hsieh et al. 2023; Magister et al. 2023). 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 (Tunstall et al. 2024; Wan et al. 2025). InfiFPO further formulates fusion as implicit preference optimization, absorbing source-model advantages without direct pivot model access (Gu et al. 2025).

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 (Ross et al. 2011); in LLMs, GKD instantiates it by distilling from teacher feedback on student-generated sequences (Agarwal et al. 2024). Recent OPD variants study self-distillation, black-box or semi-on-policy supervision, offline logit reuse, token-efficient supervision, and stabilization (Chen et al. 2026; Luo et al. 2026; Wu et al. 2026; Xu et al. 2026; Zhao et al. 2026), and extend OPD to multimodal trajectories such as video grounding and speech LLM alignment (Cao et al. 2026a; Li et al. 2026a). 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 (He et al. 2025). This is important because a target model can improve the average score while losing one source capability. Other metrics can examine interference, 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 (Tang et al. 2025) gives unified settings for comparing many parameter-level fusion methods across model and task pools. MergeBench (He et al. 2025) 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

 doi:10.20944/preprints202605.2007.v2

from distillation studies (Song and Zheng 2026b; Xu et al. 2024). Future directions include shared source settings and clearer reports of representation drift, behavior transfer, judge settings, and total fusion cost (Agarwal et al. 2024; Gu et al. 2026; Wan et al. 2024; Yang et al. 2024a; Zheng et al. 2023).

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- 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.

Table 2. Quantitative analysis for Takeaway 2, using results from M2RL (Wang et al. 2026b) and MergeBench (He et al. 2025). 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

 doi:10.20944/preprints202605.2007.v2

Table 3. Quantitative analysis for Takeaway 3, using results from FeatCal (Gu et al. 2026) and TinyBERT (Jiao et al. 2020).

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

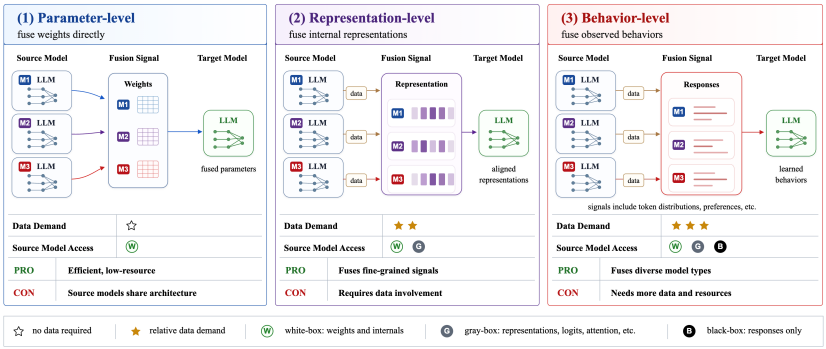

Figure 4. Three levels of model fusion and their practical trade-offs.

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 (Ainsworth et al. 2023; Wortsman et al. 2022; Zhou et al. 2026). 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 (Gu et al. 2026; Yang et al. 2024a). 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 (Agarwal et al. 2024; Song and Zheng 2026b). 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 (Cui et al. 2026) uses optimal transport for cross-architecture LLM fusion, while AdaMMS (Du et al. 2025) 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 (Tang et al. 2025) and MergeBench (He et al. 2025) begin to fix source pools and task settings, while Realistic Evaluation (Tam et al. 2024) 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

 doi:10.20944/preprints202605.2007.v2

(Fu et al. 2026; Song and Zheng 2026b; Wang 2026). 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 (Liu et al. 2024) and Merge Hijacking (Yuan et al. 2025) show that harmful updates can survive fusion. Merger-as-a-Stealer (Lu et al. 2025) studies private information leakage, while MergeGuard (Cong et al. 2024) studies fingerprints and IP protection. Among Us (Yang et al. 2026d) further studies malicious contributions in model collaboration. Future work can combine source screening, provenance, contribution attribution, and privacy-aware fusion (Khadem et al. 2026).

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

6. Conclusion

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.

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.

 doi:10.20944/preprints202605.2007.v2

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.

Model Fusion in Continual Learning.

In continual learning, model fusion can add new task or domain knowledge while limiting forgetting. AIMMerging (Feng et al. 2025) and NUFILT (Qiu et al. 2026) apply parameter-level fusion to add new task updates while reducing forgetting and interference. K-Merge (Shenaj et al. 2025) extends this setting to online LoRA fusion for on-device LLMs. RECALL (Wang et al. 2025a) uses hidden representations for hierarchical fusion without historical data. SDFT (Shenfeld et al. 2026) 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 (Jin et al. 2023; Yang et al. 2024c). Language Specific Model Merging (Dmonte et al. 2026) fuses language-specific models to lower multilingual training and update costs. SurgeryV2 (Yang et al. 2024b) and FeatCal (Gu et al. 2026) repair representation drift after fusion. FuseLLM (Wan et al. 2024) and DeepSeek-V4 (DeepSeek-AI 2026) 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 (Djuhera et al. 2025) and Fuse to Forget (Zaman et al. 2024) use parameter-level fusion to preserve safety or reduce unwanted behavior. Safety Realignment (Yi et al. 2024) uses subspace-oriented model fusion to realign unsafe models. Multilingual Safety Alignment via Self-Distillation (Qin et al. 2026) transfers safety behavior across languages through behavior-level fusion. However, unsafe source models can also propagate misalignment during fusion (Hammoud et al. 2024).

Model Compression.

Model compression uses source models to build smaller target models with similar capabilities. LoRA soups (Prabhakar et al. 2025) and LoRM (Salami et al. 2025) can fold several lightweight modules into one target module. DeepSeek-R1 (DeepSeek-AI et al. 2025) transfers reasoning patterns into six dense models with 1.5B to 70B parameters. Nemotron-Cascade 2 (Yang et al. 2026c) 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.v2

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 (Izmailov et al. 2018)	UAI'18	C checkpoint trajectory	W checkpoints	CV CNN/CV classifiers
Model Soups / Avg (Wortsman et al. 2022)	ICML'22	S same-base fine-tuned sources	W full weights	CV CLIP/ViT
Task Arithmetic (Ilharco et al. 2023)	ICLR'23	S same-base task vectors	T per-task vectors	CV LLM ED CLIP, GPT-2, T5
TIES-Merging (Yadav et al. 2023)	NeurIPS'23	S same-base task vectors	T per-task deltas	CV ED ViT and T5
DARE (Yu et al. 2024)	ICML'24	S homologous same-base models	T per-task deltas	LM LLM BERT/RoBERTa, Llama
DELLA-Merging (Deep et al. 2024)	arXiv'24	S same-base task vectors	T per-task deltas	LLM Llama-2 experts
DiffSoup (Biggs et al. 2024)	ECCV'24	S shared diffusion checkpoint	W diffusion weights	DIF text-to-image diffusion
Subspace-based methods				
KnOTS (Stoica et al. 2024b)	ICLR'25	S same-base LoRA sources	P T LoRA task updates	CV LLM CLIP-ViT, Llama3
DOP (Yang et al. 2026b)	NeurIPS'25	S sequential same-base experts	T A task and merged updates	CV ED ViT, Flan-T5
Iso-C / CTS (Marczak et al. 2025)	ICML'25	S same-base task matrices	A aggregated task matrix	CV LLM CLIP-ViT, LLMs
TSV (Gargiulo et al. 2025)	CVPR'25	S same-base task matrices	T per-task matrices	CV CLIP-ViT
Optimization-based methods				
AWD (Xiong et al. 2024)	arXiv'24	S same-base task vectors	T disentangled task vectors	CV LM ViT, RoBERTa
WUDI (Cheng et al. 2025)	ICML'25	S same-base task vectors	A merged task vector	CV LM LLM ViT, RoBERTa, Llama
DOGE (Wei et al. 2025c)	ICML'25	S same-base task vectors	T A modified merged update	CV LM LLM vision and NLP models
GCWM (Wang et al. 2026d)	arXiv'26	C continual same-backbone updates	A cumulative update state	LLM Qwen3
Module merging				
AdapterSoup (Chronopoulou et al. 2023)	EACL Findings'23	S shared-backbone adapters	P adapters	LLM GPT-2
HydraOpt (Ceritli et al. 2025)	EMNLP'25	S shared-backbone adapters	P low-rank adapters	LLM Llama/Qwen-style LLMs
LoRASoup (Prabhakar et al. 2025)	COLING Industry'25	S shared-backbone LoRAs	P LoRA modules	LLM Llama-7B
Lora-Flow (Wang et al. 2024)	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.v2

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.

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 (Jordan et al. 2023)	ICLR'23	A same architecture after permutation alignment	N	E CNN classifiers
ZipIt! (Stoica et al. 2024a)	ICLR'24	A architecture-compatible sources	L A	E vision Transformers/classifiers
Transformer Fusion (Imfeld et al. 2024)	ICLR'24	A aligned Transformer variants	L A	E T Transformer encoders/enc–decoders
AIM (Nobari et al. 2025)	NeurIPS'25	S same-base LLM checkpoints	F	D decoder-only LLMs
ACM (Yao et al. 2025)	arXiv'25	S same-base LLM checkpoints	F	D decoder-only LLMs
MAGIC (Li et al. 2025)	arXiv'25	S same-base or aligned sources	F N	E D CV and Llama merging
Merging Beyond (Yao et al. 2026)	arXiv'26	S sequential same-backbone updates	F L	D streaming LLM updates
Closed-form representation solvers				
RegMean (Jin et al. 2023)	ICLR'23	S same architecture and tokenizer	L	T language models
RegMean++ (Nguyen et al. 2026)	TMLR'26	A same-family compatible models	L	E T D encoder, enc–dec, decoder-only
LoRM (Salami et al. 2025)	ICLR'25	S PEFT modules over compatible bases	L P	E D LoRA-equipped models
IterIS (Chen et al. 2025)	CVPR'25	S compatible LoRA adapters	P	M text-to-image, VLM, and LLM adapters
LOT-Merging (Sun et al. 2025)	NeurIPS'25	S same-base task-vector checkpoints	L N	D Transformer/LLM checkpoints
FeatCal (Gu et al. 2026)	arXiv'26	H mismatch handled by projection/alignment	L N P	D post-merging decoder-only LLMs
Backpropagation-based representation transfer				
Patient KD (Sun et al. 2019)	EMNLP'19	T teacher–student fusion	F	E BERT-style encoders
TinyBERT (Jiao et al. 2020)	EMNLP Findings'20	T teacher–student fusion	F	E BERT-style encoders
Task-aware LWD (Liang et al. 2023)	ICML'23	T teacher–student fusion	F P	E language-model compression
Representation Surgery (Yang et al. 2024a)	ICML'24	S same-base multi-task merged models	P	E encoder-based multi-task models
Surgeryv2 (Yang et al. 2024b)	arXiv'24	S same-base multi-task merged models	P	E aligned-tokenizer settings
ProbSurgery (Wei et al. 2025a)	ICML'25	S same-base multi-task merged models	P	E multi-task model merging
RECALL (Wang et al. 2025a)	EMNLP'25	S continual same-family checkpoints	L N	C in-domain checkpoint sequences
NUFILT (Qiu et al. 2026)	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.

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.

 doi:10.20944/preprints202605.2007.v2

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 (Hinton et al. 2015)	NeurIPS'15	T teacher–student	D O soft outputs	general neural networks
DistilBERT (Sanh et al. 2019)	NeurIPS'19	T BERT teacher–student	D O token distributions	E BERT encoders
InfiGFusion (Wang et al. 2025c)	NeurIPS'25	M multi-source LLMs	D O logit geometry	D decoder-only LLMs
Demonstration fusion				
Instruction Distillation (Sun et al. 2023)	arXiv'23	T stronger teacher	X O instruction responses	D LLM rankers
GPT4All (Anand et al. 2023)	GitHub'23	B API teacher	X O assistant demos	C chatbot tuning
Distilling Step-by-Step (Hsieh et al. 2023)	ACL'23	T larger LLM teacher	X O rationales and labels	D small reasoning LMs
Teaching Small LMs to Reason (Magister et al. 2023)	ACL'23	T reasoning teacher	X O CoT rationales	D small LMs
Feedback fusion				
FuseChat (Wan et al. 2025)	EMNLP'25	M chat-model sources	X F O responses and preferences	C chat fusion
Zephyr (Tunstall et al. 2024)	COLM'24	T aligned teacher	F O preference data	C chat alignment
InfiFPO (Gu et al. 2025)	NeurIPS'25	M source preferences	F O preference optimization	D LLM fusion
On-policy and trajectory-level fusion				
†DAgger (Ross et al. 2011)	AISTATS'11	T expert policy	R P learner-state actions	A imitation learning
GKD / OPD (Agarwal et al. 2024)	ICLR'24	T teacher on student states	D R P self-generated sequences	D autoregressive LMs
Self-Distilled Reasoner (Zhao et al. 2026)	arXiv'26	S self-distillation	X R P reasoning traces	D reasoning LLMs
SODA (Chen et al. 2026)	arXiv'26	B black-box teacher	X R S semi-on-policy data	D black-box distillation
Lightning OPD (Wu et al. 2026)	arXiv'26	T teacher-logit source	D S offline OPD signals	D reasoning LLMs
TIP (Xu et al. 2026)	arXiv'26	T sampled-token teacher	D R P token-importance signals	D token-efficient OPD
Demystifying OPD (Luo et al. 2026)	arXiv'26	T rollout teacher	D R P stabilized token signals	D OPD stabilization
Video-OPD (Li et al. 2026a)	arXiv'26	H multimodal teacher	R P video trajectories	M video grounding MLLMs
X-OPD (Cao et al. 2026a)	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 LM; 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.v2

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.

Resource	Venue	Vision	Text	LLM	MLLM	Model Pool	Task Pool	Hetero.	Fusion Type	Evaluation Focus	Open
Realistic Evaluation (Tam et al. 2024)	arXiv'24	Y	Y	N	N	Y	Y	P	Vision / text merging	Acc., compositionality	Y
Model-GLUE (Zhao et al. 2024)	NeurIPS D&B'24	N	N	Y	N	Y	Y	Y	Heterogeneous LLM merging	Model selection, aggregation	Y
MergeKit (Goddard et al. 2024)	EMNLP-I'24	N	N	Y	N	P	P	P	Recipe-based merging	Leaderboard perf.	Y
EMR-Merging (Huang et al. 2024)	NeurIPS'24	Y	Y	P	P	Y	Y	P	Tuning-free merging	Acc., scalability	Y
Merging at Scale (Khalifa et al. 2024)	arXiv'24	N	N	Y	N	Y	Y	N	LLM merging	Scaling, expert count	N
H3Fusion (Tekin et al. 2026)	EACL'26	N	N	Y	N	Y	Y	N	Alignment merging	Helpful., honest., harmless.	P
SMM-Bench (Akizuki et al. 2025)	AutoML-N'25	N	N	Y	N	Y	Y	P	Surrogate merge search	Search cost, ranking	Y
Systematic Study (Hitit et al. 2026)	TMLR'26	N	N	Y	N	Y	Y	N	LLM merging study	Method reliability	N
Mergenetic (Minut et al. 2025)	ACL Demo'25	N	N	Y	N	P	P	P	Evolutionary merging	Fitness, search efficiency	Y
FusionBench (Tang et al. 2025)	JMLR'25	Y	Y	Y	P	Y	Y	P	Merging / ensemble / mixing	Acc., robust., OOD	Y
MergeBench (He et al. 2025)	NeurIPS D&B'25	N	N	Y	N	Y	Y	N	Domain LLM merging	Acc., forgetting, runtime	Y
OptMerge (Wei et al. 2025b)	ICLR'26	N	N	N	Y	Y	Y	Y	MLLM merging	VQA, OCR, grounding	Y
Merging Scaling Law (Wang et al. 2025b)	ICML'26	N	N	Y	N	Y	Y	N	Large-scale LLM merging	Scaling law, expert count	Y
M2RL (Wang et al. 2026a)	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 (Goddard et al. 2024) and Mergenetic (Minut et al. 2025), are included when they provide reusable fusion pipelines or practical evaluation settings. They are not treated as fixed benchmark suites.

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.v2

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 (Feng et al. 2025), NUFILT (Qiu et al. 2026), and K-Merge (Shenaj et al. 2025) 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 (Shenfeld et al. 2026). 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 (Goddard et al. 2024) makes LLM fusion easier to run. MergePipe (Wang et al. 2026c) further shows that expert-parameter I/O and repeated scans become key bottlenecks as the source pool grows. For method search, FusionBench (Tang et al. 2025) and MergeBench (He et al. 2025) improve standard comparison, but large models still make candidate evaluation costly. For behavior-level fusion, source feedback can also be expensive. Lightning OPD (Wu et al. 2026) and TIP (Xu et al. 2026) 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

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *Proceedings of ICLR*.

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. 2023. Git Re-Basin: Merging Models modulo Permutation Symmetries. In *Proceedings of the International Conference on Learning Representations, ICLR 2023*.

Rio Akizuki, Yuya Kudo, Nozomu Yoshinari, Yoichi Hirose, Toshiyuki Nishimoto, Kento Uchida, and Shinichi Shirakawa. 2025. Surrogate Benchmarks for Model Merging Optimization. In *AutoML 2025 Non-Archival Content Track*.

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. In *GitHub*.

 doi:10.20944/preprints202605.2007.v2

Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. 2024. Diffusion Soup: Model Merging for Text-to-Image Diffusion Models. In *Computer Vision – ECCV 2024*, volume 15121 of *Lecture Notes in Computer Science*, pages 257–274.

Di Cao, Dongjie Fu, Hai Yu, Siqi Zheng, Xu Tan, and Tao Jin. 2026a. X-OPD: Cross-Modal On-Policy Distillation for Capability Alignment in Speech LLMs. *CoRR*, abs/2603.24596.

Yuan Cao, Dezhi Ran, Yuzhe Guo, Mengzhou Wu, Simin Chen, Linyi Li, Wei Yang, and Tao Xie. 2026b. An Empirical Study and Theoretical Explanation on Task-Level Model-Merging Collapse. *CoRR*, abs/2603.09463.

Taha Ceritli, Ondrej Bohdal, Mete Ozay, Jijoong Moon, Kyenghun Lee, Hyeonmok Ko, and Umberto Michieli. 2025. HydraOpt: Navigating the Efficiency-Performance Trade-off of Adapter Merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26887–26909, Suzhou, China. Association for Computational Linguistics.

Hongxu Chen, Runshi Li, Bowei Zhu, Zhen Wang, and Long Chen. 2025. IterIS: Iterative Inference-Solving Alignment for LoRA Merging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025*.

Xiwen Chen, Jingjing Wang, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hejian Sang, Zhipeng Wang, Alborz Geramifard, and Feng Luo. 2026. SODA: Semi On-Policy Black-Box Distillation for Large Language Models. *arXiv preprint arXiv:2604.03873*.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. 2025. Whoever Started the Interference Should End It: Guiding Data-Free Model Merging via Task Vectors. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 10121–10143. PMLR.

Alexandra Chronopoulou, Matthew E. Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight Averaging to Improve Generalization of Pretrained Language Models. In *Findings of the Association for Computational Linguistics: EACL 2023*.

Tianshuo Cong, Delong Ran, Zesen Liu, Xinlei He, Jinyuan Liu, Yichen Gong, Qi Li, Anyu Wang, and Xiaoyun Wang. 2024. Have You Merged My Model? On The Robustness of Large Language Model IP Protection Methods Against Model Merging. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, pages 69–76.

Chenhang Cui, Binyun Yang, Fei Shen, Yuxin Chen, Jingnan Zheng, Xiang Wang, An Zhang, and Tat-Seng Chua. 2026. Transport and Merge: Cross-Architecture Merging for Large Language Models. *CoRR*, abs/2602.05495.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. 2024. DELLA-Merging: Reducing Interference in Model Merging through Magnitude-Based Sampling. *CoRR*, abs/2406.11617.

DeepSeek-AI. 2026. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. Technical report, DeepSeek-AI.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR*, abs/2501.12948.

Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, Syed Zawad, and Holger Boche. 2025. Safemerge: Preserving safety alignment in fine-tuned large language models via selective layer-wise model merging. *CoRR*, abs/2503.17239.

Alphaeus Dmonte, Vidhi Gupta, Daniel J Perry, and Mark Arehart. 2026. Improving Training Efficiency and Reducing Maintenance Costs via Language Specific Model Merging. In *CoRR*.

Yiyang Du, Xiaochen Wang, Chi Chen, Jiabo Ye, Yiru Wang, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Zhifang Sui, Maosong Sun, and Yang Liu. 2025. AdaMMS: Model Merging for Heterogeneous Multimodal Large Language Models with Unsupervised Coefficient Optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025*.

Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zhengliang Liu, Zhenyuan Yang, Haoran Lu, Xilin Gong, Yufang Liu, Terry Ma, Wei Ruan, Ali Abbasi, Jing Zhang, Tao Wang, Ehsan Latif, Wei Liu, Wei Zhang, Soheil Kolouri, and 5 others. 2026. Knowledge distillation and dataset distillation of large language models: Emerging trends, challenges, and future directions. *Artificial Intelligence Review*, 59(17).

Yujie Feng, Jian Li, Xiaoyu Dong, Pengfei Xu, Xiaohui Zhou, Yujia Zhang, Zexin LU, Yasha Wang, Alan Zhao, Xu Chu, and Xiao-Ming Wu. 2025. AIMMerging: Adaptive Iterative Model Merging Using Training Trajectories for Language Model Continual Learning. In *Proceedings of the 2025 Conference on Empirical*

 doi:10.20944/preprints202605.2007.v2

Methods in Natural Language Processing, pages 13420–13437, Suzhou, China. Association for Computational Linguistics.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. 2026. Revisiting On-Policy Distillation: Empirical Failure Modes and Simple Fixes. *arXiv preprint arXiv:2603.25562*.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. 2025. Task Singular Vectors: Reducing Task Interference in Model Merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

GLM-5 Team. 2026. GLM-5: from Vibe Coding to Agentic Engineering. *CoRR*, abs/2602.15763.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A Toolkit for Merging Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vision*, 129(6):1789–1819.

Yanggan Gu, Shuo Cai, Zihao Wang, Wenjun Wang, Yuanyi Wang, Pengkai Wang, Sirui Huang, Su Lu, Jianmin Wu, and Hongxia Yang. 2026. FeatCal: Feature Calibration for Post-Merging Models. *Preprint*, arXiv:2605.13030.

Yanggan Gu, Zhaoyi Yan, Yuanyi Wang, Yiming Zhang, Qi Zhou, Fei Wu, and Hongxia Yang. 2025. InfiFPO: Implicit Model Fusion via Preference Optimization in Large Language Models. In *Advances in Neural Information Processing Systems*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge Distillation of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Hasan Abed Al Kader Hammoud, Umberto Michieli, Fabio Pizzati, Philip Torr, Adel Bibi, Bernard Ghanem, and Mete Ozay. 2024. Model Merging and Safety Alignment: One Bad Model Spoils the Bunch. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. 2025. MergeBench: A Benchmark for Merging Domain-Specialized LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531.

Oğuz Kağan Hitit, Leander Girrbach, and Zeynep Akata. 2026. A systematic study of in-the-wild model merging for large language models. *Transactions on Machine Learning Research*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In *Findings of ACL*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations, ICLR 2022*.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. 2024. EMR-Merging: Tuning-Free High-Performance Model Merging. In *Advances in Neural Information Processing Systems 37, NeurIPS 2024*.

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

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. 2024. Transformer Fusion with Optimal Transport. In *Proceedings of the International Conference on Learning Representations, ICLR 2024*.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Averaging Weights Leads to Wider Optima and Better Generalization. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI 2018*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless Knowledge Fusion by Merging Weights of Language Models. In *Proceedings of the International Conference on Learning Representations, ICLR 2023*.

 doi:10.20944/preprints202605.2007.v2

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. 2023. REPAIR: Renormalizing Permuted Activations for Interpolation Repair. In *Proceedings of the International Conference on Learning Representations, ICLR 2023*.

Fatemeh Khadem, Sajad Mousavi, Yi Fang, and Yuhong Liu. 2026. DP-OPD: Differentially Private On-Policy Distillation for Language Models. *arXiv preprint arXiv:2604.04461*.

Muhammad Khalifa, Yi-Chern Tan, Arash Ahmadian, Tom Hosking, Honglak Lee, Lu Wang, Ahmet Ustun, Tom Sherborne, and Matthias Galle. 2024. If You Can't Use Them, Recycle Them: Optimizing Merging at Scale Mitigates Performance Tradeoffs. *CoRR*, abs/2412.04144.

Jiaze Li, Hao Yin, Haoran Xu, Boshen Xu, Wenhui Tan, Zewen He, Jianzhong Ju, Zhenbo Luo, and Jian Luan. 2026a. Video-OPD: Efficient Post-Training of Multimodal Large Language Models for Temporal Video Grounding via On-Policy Distillation. *CoRR*, abs/2602.02994.

Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. 2026b. Deep model fusion: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 37(5):2008–2024.

Yayuan Li, Jian Zhang, Jintao Guo, Zihan Cheng, Lei Qi, Yinghuan Shi, and Yang Gao. 2025. MAGIC: achieving superior model merging via magnitude calibration. *CoRR*, abs/2512.19320.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*.

Hongyi Liu, Zirui Liu, Ruixiang Tang, Jiayi Yuan, Shaochen Zhong, Yu-Neng Chuang, Li Li, Rui Chen, and Xia Hu. 2024. LoRA-as-an-Attack! Piercing LLM Safety Under The Share-and-Play Scenario. *arXiv preprint arXiv:2403.00108*.

Lin Lu, Zhigang Zuo, Ziji Sheng, and Pan Zhou. 2025. Merger-as-a-Stealer: Stealing Targeted PII from Aligned LLMs with Model Merging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2025*.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. 2026. Demystifying OPD: Length Inflation and Stabilization Strategies for Large Language Models. *arXiv preprint arXiv:2604.08527*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching Small Language Models to Reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartlomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. 2025. No Task Left Behind: Isotropic Model Merging with Common and Task-Specific Subspaces. In *Proceedings of the International Conference on Machine Learning, ICML 2025*.

Michael Matena and Colin Raffel. 2022. Merging Models with Fisher-Weighted Averaging. In *Advances in Neural Information Processing Systems 35, NeurIPS 2022*.

Adrian Robert Minut, Tommaso Mencattini, Andrea Santilli, Donato Crisostomi, and Emanuele Rodolà. 2025. Mergenetic: a simple evolutionary model merging library. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 572–582, Vienna, Austria. Association for Computational Linguistics.

The-Hai Nguyen, Dang Huu-Tien, Takeshi Suzuki, and Le-Minh Nguyen. 2026. RegMean++: Enhancing Effectiveness and Generalization of Regression Mean for Model Merging. *Transactions on Machine Learning Research*. Expert Certification.

Amin Heyrani Nobari, Kaveh Alimohammadi, Ali ArjomandBigdeli, Akash Srivastava, Faez Ahmed, and Navid Azizan. 2025. Activation-Informed Merging of Large Language Models. In *Advances in Neural Information Processing Systems*.

Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. 2025. LoRA Soups: Merging LoRAs for Practical Skill Composition Tasks. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025*.

Laiqiao Qin, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. 2025. Knowledge distillation in federated learning: A survey on long lasting challenges and new solutions. *International Journal of Intelligent Systems*, 2025(1).

Ruiyang Qin, Qingzhuo Wang, Dongrui Liu, Qiang Li, Zhihua Wei, and Wen Shen. 2026. Multilingual Safety Alignment via Self-Distillation. *arXiv preprint arXiv:2605.02971*.

Zihuan Qiu, Lei Wang, Yang Cao, Runtong Zhang, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. 2026. Null-Space Filtering for Data-Free Continual Model Merging: Preserving Stability, Promoting Plasticity. In *The Fourteenth International Conference on Learning Representations*.

 doi:10.20944/preprints202605.2007.v2

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2011. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*.

Riccardo Salami, Pietro Buzzega, Matteo Mosconi, Jacopo Bonato, Luigi Sabetta, and Simone Calderara. 2025. Closed-Form Merging of Parameter-Efficient Modules for Federated Continual Learning. In *Proceedings of the International Conference on Learning Representations, ICLR 2025*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Donald Shenaj, Ondrej Bohdal, Taha Ceritli, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. 2025. K-Merge: Online Continual Merging of Adapters for On-device Large Language Models. *CoRR*, abs/2510.13537.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. 2026. Self-Distillation Enables Continual Learning. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*.

Mingyang Song and Mao Zheng. 2026a. Model merging in the era of large language models: Methods, applications, and future directions. *Preprint*, arXiv:2603.09938.

Mingyang Song and Mao Zheng. 2026b. A survey of on-policy distillation for large language models. *Preprint*, arXiv:2604.00626.

George Stoica, Daniel Bolya, Jakob Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2024a. ZipIt! Merging Models from Different Tasks without Training. In *Proceedings of the International Conference on Learning Representations, ICLR 2024*.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. 2024b. Model merging with SVD to tie the Knots. In *Proceedings of the International Conference on Learning Representations*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient Knowledge Distillation for BERT Model Compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Weiwei Sun, Zheng Chen, Xinyu Ma, Lingyong Yan, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Instruction Distillation Makes Large Language Models Efficient Zero-shot Rankers. In *2023*.

Wenju Sun, Qingyong Li, Wen Wang, Yang Liu, Yangli-ao Geng, and Boyang Li. 2025. Towards minimizing feature drift in model merging: Layer-wise task vector fusion for adaptive knowledge integration. In *Advances in Neural Information Processing Systems*.

Yi-Lin Sung, Linjie Li, Kevin Lin, Zhe Gan, Mohit Bansal, and Lijuan Wang. 2023. An Empirical Study of Multimodal Model Merging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*.

Derek Tam, Yash Kant, Brian Lester, Igor Gilitschenski, and Colin Raffel. 2024. Realistic Evaluation of Model Merging for Compositional Generalization. *CoRR*, abs/2409.18314.

Anke Tang, Li Shen, Yong Luo, Enneng Yang, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. 2025. Fusionbench: A unified library and comprehensive benchmark for deep model fusion. *Journal of Machine Learning Research*, 26(307):1–38.

Selim Furkan Tekin, Fatih Ilhan, Sihao Hu, Tiansheng Huang, Yichang Xu, Zachary Yahn, and Ling Liu. 2026. H3Fusion: Helpful, Harmless, Honest Fusion of Aligned LLMs. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6993–7013.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clementine Fourrier, Nathan Habib, and 1 others. 2024. Zephyr: Direct Distillation of LM Alignment. In *First Conference on Language Modeling*.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge Fusion of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Fanqi Wan, Ziyi Yang, Longguang Zhong, Xiaojun Quan, Xinting Huang, and Wei Bi. 2025. FuseChat: Knowledge Fusion of Chat Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025*.

Bowen Wang, Haiyuan Wan, Liwen Shi, Chen Yang, Peng He, Yue Ma, Haochen Han, Wenhao Li, Tiao Tan, Yongjian Li, Fangming Liu, Yifan Gong, and Sheng Zhang. 2025a. RECALL: REpresentation-aligned Catastrophic-forgetting ALLeviation via Hierarchical Model Merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16381–16395, Suzhou, China. Association for Computational Linguistics.

 doi:10.20944/preprints202605.2007.v2

Hanqing Wang, Bowen Ping, Shuo Wang, Xu Han, Yun Chen, Zhiyuan Liu, and Maosong Sun. 2024. LoRA-Flow: Dynamic LoRA Fusion for Large Language Models in Generative Tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*.

Haoqing Wang, Xiang Long, Ziheng Li, Yilong Xu, Tingguang Li, and Yehui Tang. 2026a. To mix or to merge: Toward multi-domain reinforcement learning for large language models. *Preprint*, arXiv:2602.12566.

Haoqing Wang, Xiang Long, Ziheng Li, Yilong Xu, Tingguang Li, and Yehui Tang. 2026b. To Mix or To Merge: Toward Multi-Domain Reinforcement Learning for Large Language Models. *CoRR*, abs/2602.12566.

Wenshuo Wang. 2026. Knowledge Distillation Must Account for What It Loses. *arXiv preprint arXiv:2604.25110*.

Yuanyi Wang, Yanggan Gu, Zihao Wang, Kunxi Li, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. 2026c. MergePipe: A Budget-Aware Parameter Management System for Scalable LLM Merging. *CoRR*, abs/2602.13273.

Yuanyi Wang, Yanggan Gu, Yiming Zhang, Qi Zhou, Zhaoyi Yan, Congkai Xie, Xinyao Wang, Jianbo Yuan, and Hongxia Yang. 2025b. Model Merging Scaling Laws in Large Language Models. *CoRR*, abs/2509.24244.

Yuanyi Wang, Zhaoyi Yan, Yiming Zhang, Qi Zhou, Yanggan Gu, Fei Wu, and Hongxia Yang. 2025c. InfiGFusion: Graph-on-Logits Distillation via Efficient Gromov-Wasserstein for Model Fusion. In *Advances in Neural Information Processing Systems*.

Yuanyi Wang, Yifan Yang, Su Lu, Yanggan Gu, Pengkai Wang, Wenjun Wang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, Jialun Cao, Shing-Chi Cheung, and Hongxia Yang. 2026d. Geometry Conflict: Explaining and Controlling Forgetting in LLM Continual Post-Training. *arXiv preprint arXiv:2605.09608*.

Qi Wei, Shuo He, Enneng Yang, Tingcong Liu, Haobo Wang, Lei Feng, and Bo An. 2025a. Representation Surgery in Model Merging with Probabilistic Modeling. In *Proceedings of the International Conference on Machine Learning, ICML 2025*.

Yongxian Wei, Runxi Cheng, Weike Jin, Enneng Yang, Li Shen, Lu Hou, Sinan Du, Chun Yuan, Xiaochun Cao, and Dacheng Tao. 2025b. OptMerge: Unifying Multimodal LLM Capabilities And Modalities Via Model Merging. *CoRR*, abs/2505.19892.

Yongxian Wei, Anke Tang, Li Shen, Zixuan Hu, Chun Yuan, and Xiaochun Cao. 2025c. Modeling Multi-Task Model Merging as Adaptive Projective Gradient Descent. In *Proceedings of the International Conference on Machine Learning, ICML 2025*.

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. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the International Conference on Machine Learning, ICML 2022*.

Yecheng Wu, Song Han, and Han Cai. 2026. Lightning OPD: Efficient Post-Training for Large Reasoning Models with Offline On-Policy Distillation. *arXiv preprint arXiv:2604.13010*.

Feng Xiong, Runxi Cheng, Wang Chen, Zhanqiu Zhang, Yiwen Guo, Chun Yuan, and Ruifeng Xu. 2024. Multi-Task Model Merging via Adaptive Weight Disentanglement. *CoRR*, abs/2411.18729.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A Survey on Knowledge Distillation of Large Language Models. *CoRR*, abs/2402.13116.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. 2026. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084*.

Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. 2025. A Survey on Model MoErging: Recycling and Routing Among Specialized Experts for Collaborative Learning. *Transactions on Machine Learning Research*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. 2023. TIES-Merging: Resolving Interference When Merging Models. In *Advances in Neural Information Processing Systems*.

Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. 2025. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Trans. Intell. Syst. Technol.*, 16(6).

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2026a. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Comput. Surv.*, 58(8).

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. 2024a. Representation Surgery for Multi-Task Model Merging. In *Proceedings of the International Conference on Machine Learning, ICML 2024*.

 doi:10.20944/preprints202605.2007.v2

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xingwei Wang, Xiaocun Cao, Jie Zhang, and Dacheng Tao. 2024b. Surgeryv2: Bridging the gap between model merging and multi-task learning with deep representation surgery. *arXiv preprint arXiv:2410.14389*.

Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie Zhang. 2026b. Continual model merging without data: Dual projections for balancing stability and plasticity. *Advances in Neural Information Processing Systems*, 38:39275–39305.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024c. AdaMerging: Adaptive Model Merging for Multi-Task Learning. In *The Twelfth International Conference on Learning Representations*.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, Renjie Pi, Grace Lam, Nayeon Lee, Alexander Bukharin, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2026c. Nemotron-cascade 2: Post-training llms with cascade RL and multi-domain on-policy distillation. *CoRR*, abs/2603.19220.

Ziyuan Yang, Wenxuan Ding, Shangbin Feng, and Yulia Tsvetkov. 2026d. Among Us: Measuring and Mitigating Malicious Contributions in Model Collaboration Systems. *CoRR*, abs/2602.05176.

Yuxuan Yao, Shuqi Liu, Zehua Liu, Qintong Li, Mingyang Liu, Xiongwei Han, Zhijiang Guo, Han Wu, and Linqi Song. 2025. Activation-Guided Consensus Merging for Large Language Models. *CoRR*, abs/2505.14009.

Yuxuan Yao, Haonan Sheng, Qingsong Lv, Han Wu, Shuqi Liu, Zehua Liu, Zengyan Liu, Jiahui Gao, Haochen Tan, Xiaojin Fu, Haoli Bai, Hing Cheung So, Zhijiang Guo, and Linqi Song. 2026. Merging Beyond: Streaming LLM Updates via Activation-Guided Rotations. *CoRR*, abs/2602.03237.

Xin Yi, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. 2024. A safety realignment framework via subspace-oriented model fusion for large language models. *arXiv preprint arXiv:2405.09055*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57755–57775. PMLR.

Shi Jie Yu and Sehyun Choi. 2025. Parameter-efficient checkpoint merging via metrics-weighted averaging. *CoRR*, abs/2504.18580.

Zenghui Yuan, Yangming Xu, Jiawen Shi, Pan Zhou, and Lichao Sun. 2025. Merge Hijacking: Backdoor Attacks to Model Merging of Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.

Kerem Zaman, Leshem Choshen, and Shashank Srivastava. 2024. Fuse to Forget: Bias Reduction and Selective Memorization through Model Fusion. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *CoRR*, abs/2601.18734.

Xinyu Zhao, Guoheng Sun, Ruisi Cai, Yukun Zhou, Pingzhi Li, Peihao Wang, Bowen Tan, Yexiao He, Li Chen, Yi Liang, and 1 others. 2024. Model-GLUE: Democratized LLM Scaling for A Large Model Zoo in the Wild. In *Advances in Neural Information Processing Systems 37, NeurIPS 2024*, pages 13349–13371.

Hongling Zheng, Li Shen, Anke Tang, Yong Luo, Han Hu, Bo Du, Yonggang Wen, and Dacheng Tao. 2025. Learning from Models Beyond Fine-Tuning. *Nature Machine Intelligence*, 7(1):6–17.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.

Luca Zhou, Bo Zhao, Rose Yu, and Emanuele Rodola. 2026. Demystifying Mergeability: Interpretable Properties to Predict Model Merging Success. *CoRR*, abs/2601.22285.