Exploring Information Seeking Agent Consolidation

Anonymous ACL submission

Abstract

Information-seeking agents have emerged as a powerful paradigm for knowledge-intensive tasks, yet today's systems remain specialized for the open web, documents, or local knowledge bases, hindering scalable and cross-domain deployment. We present the first systematic empirical study of consolidating these information-seeking agents into a single foundation agentic model. We compare two paradigms—*data-level mixing*, which trains a unified model on a mixture of datasets, and *parameter-level merging*, which merges independently trained experts in parameter space—across 3 training scenarios, evaluating **26** representative parameter-level methods on **10** benchmarks. To compare across heterogeneous benchmarks, we introduce a geometric Composite Score and an Imbalance Score that describe overall performance and task skew. Our analysis shows that (i) well-designed parameter-level merging attains parity with data mixing at a fraction of its training cost and is order-agnostic; (ii) parameter-level merging structurally preserves out-of-domain capabilities that data mixing universally forgets; and (iii) cross-scenario stability is strongly tied to consolidation quality. We distil our observations into a method-selection guide and design principles for next-generation merging operators.

1 Introduction

Information-seeking agents that interleave retrieval, reasoning, and tool use have become the dominant paradigm for knowledge-intensive tasks. The literature has converged on three specialised classes, each tied to its native environment: *Web agents* interacting with search engines and live pages (Wu et al., 2025; Li et al., 2025b,a); *Document agents* reasoning over multi-modal documents (Zhang et al., 2026b; Sun et al., 2025a; Zhu et al., 2025); and *RAG agents* operating over local knowledge bases (Jin et al., 2025a; Tao et al., 2026). Real-world deployments increasingly demand all three capabilities in a single model, since maintaining a separate expert per domain forces ad hoc routing, multiplies serving costs, and prevents capabilities from compounding. Consolidating the three experts into a unified *foundation agentic model* (Figure 1) is therefore both timely and underexplored.

Agent consolidation is more challenging than the model-merging settings studied in prior work (Yang et al., 2026a; Yadav et al., 2024): a consolidated agent must interact with heterogeneous tool APIs, preserve multi-step planning across drastically different action distributions, and avoid forgetting any individual expert. Two paradigms currently compete. *Data-level consolidation* jointly trains on a mixture of all task corpora. *Parameter-level consolidation* merges independently trained experts in weight space (Wang et al., 2025a; Team et al., 2025a). The former is empirically strong but data- and compute-hungry, while the agentic effectiveness, robustness, and trade-offs of the latter remain poorly characterised.

To bridge the gap, we present the first systematic empirical study of information-seeking agent consolidation, organised around four research questions. **(RQ1) Performance**: can parameter-level merging match data-level mixing's performance? **(RQ2) Generalization**: does either paradigm retain out-of-domain capability? **(RQ3) Failure modes**: which scenarios break which methods? **(RQ4) Mechanisms**: which design factors govern merging effectiveness? We evaluate data-mixing baseline and **26** parameter-level methods, varying backbone scale and fine-tuning regime. For effective comparison, we introduce a geometric *Composite Score* and an *Imbalance Score* that describe overall quality and task imbalance, respectively.

Five headline findings emerge in our study. (F1) Well-designed parameter-level consolidation can attain parity with data mixing. (F2) Parameter-level merging better preserves general capabilities. (F3) Stability is structurally correlated to

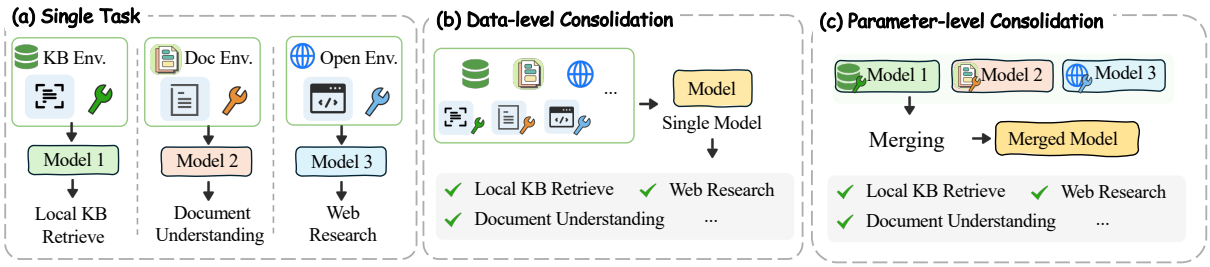

Figure 1: Comparison of three information-seeking agent consolidation paradigms. **(a) Single-task training**, where separate agents are independently trained for local knowledge-base retrieval $\mathcal{D}_{\text{rag}}$, document understanding $\mathcal{D}_{\text{doc}}$, and open-web search $\mathcal{D}_{\text{web}}$, in their respective environments. **(b) Data-level consolidation**, which unifies heterogeneous agent trajectories into a single training set $\mathcal{D}_{\text{all}}$ and learns a single model via joint multi-task training. **(c) Parameter-level consolidation**, which first trains environment-specific expert models as in (a) and then merges them in parameter space to obtain a unified agent without joint retraining.

quality. (F4) Three design factors govern merging effectiveness: consolidation granularity, consensus-indicator informativeness, and expert task affinity. (F5) Parameter-level methods preserve the behavior of original expert agents. We distil these into a deployment guide and four design principles.

In summary, our contributions are as follows:

- **Unified information-seeking agent paradigm.** We formulate web search, document-grounded reasoning, and knowledge-base retrieval under a single paradigm, enabling a consistent abstraction across heterogeneous environments. (§3, §4).
- **Largest-scale systematic evaluation.** We benchmark **26** parameter-level consolidation methods against data mixing on 8 in-domain and 2 OOD benchmarks, with metrics (G, Imb) that separate overall quality from imbalance (§4).
- **Novel empirical findings.** We identify a co-champion pair across 3 scenarios, OOD-retention advantage of parameter-level methods, stability-quality correlation, design factors governing merging effectiveness, and behavior analysis (§5).
- **Design principles.** Based on our empirical study, we distil a constraint-driven method-selection guide and four design principles (§6).

2 Related Works

Information Seeking Agents. RAG has emerged as the primary paradigm for grounding LLMs in external knowledge bases (Wang et al., 2024b), with recent advances in multi-hop reasoning (Li et al., 2024) and adaptive memory-based optimization (Qin et al., 2025). Document-centric agents tackle visually-rich documents such as PDFs and slides through vision-language models (Faysse et al., 2024) and multi-modal retrieval frameworks (Tanaka et al., 2025; Han et al., 2025), enabling precise extraction from complex layouts and multi-page contexts (Ma et al., 2024; Jin et al., 2025b). Web agents navigate dynamic websites to gather open-domain information (Deng et al., 2023; Wu et al., 2025), relying on planning and rollback mechanisms for robustness in open-web environments (Wei et al., 2025; Zhang et al., 2025).

Model Consolidation. Specialized capabilities can be consolidated into unified systems through two paradigms: *data-level mixture* and *parameter-level merging*. Data-level methods build on multi-task instruction tuning, where training on massive task mixtures yields broad generalization (Chung et al., 2024; Longpre et al., 2023). Recent work adapts this paradigm to agentic settings by scaling across heterogeneous environments (Fang et al., 2025) and applying reinforcement learning (Xi et al., 2025). Parameter-level consolidation, in parallel, builds on classical parameter merging (Shoemake, 1985; Utans, 1996), originally used to combine distinct NLP abilities (Yu et al., 2024b; Wan et al., 2025; Yang et al., 2026a) into foundation models without the computational cost of joint retraining. Recent work extends this paradigm to agentic settings (Liao et al., 2025; Maiti et al., 2025). Our work systematically evaluates both strategies to overcome the fragmentation of existing information-seeking agents and enable cross-domain scalability.

3 Preliminaries and Setup

3.1 Task Definition

We study information-seeking agents that answer a natural-language query by retrieving and reasoning over external knowledge sources. Formally, let $q \in \mathcal{Q}$ denote a user query and $y \in \mathcal{Y}$ the target

answer; the agent is a parameterized policy π_θ that induces a mapping

$$\pi_\theta : \mathcal{Q} \times \mathcal{K} \to \mathcal{Y}, \tag{1}$$

where $\mathcal{K}$ denotes the accessible knowledge environment and θ the learnable parameters. We categorize information-seeking tasks into three classes by the structure of $\mathcal{K}$ and the corresponding *retrieval interfaces*, which together define the agent's action space $\mathcal{A}$. Across all settings the agent follows the same `ReAct` paradigm (Yao et al., 2022), differing only in how knowledge is accessed and observed; the resulting T-iteration interaction history is

$$\mathcal{H}_T = (\tau_0, a_0, o_0, \ldots, \tau_i, a_i, o_i, \ldots, \tau_T, a_T), \tag{2}$$

where at each step t the agent generates an internal thought τ_t, selects an action $a_t \in \mathcal{A}$, and the environment returns an observation o_t.

Online Open-ended Web Search. The agent interacts with a dynamic, partially observable environment $\mathcal{E}_{\text{web}}$ via two actions: *search*, parameterized by a query and returning a list of titles and snippets, and *visit*, parameterized by a goal and a URL and returning evidence extracted from the corresponding webpage. Web agents reduce problem uncertainty by leveraging information from the web, which requires strong problem decomposition and associative reasoning.

Document-Grounded Understanding. The agent is provided with a document that may contain multimodal elements such as text, tables, figures, or images, and alternates between *Search* actions, which provide global textual signals over the document collection, and *Read* actions, which perform fine-grained, localized extraction from selected sections. Doc agents therefore focus on localized information extraction, long-context understanding, and coherent reasoning within a single source, rather than retrieval or external exploration.

Local Knowledge Base Retrieval and Generation. The agent operates over a fixed and curated corpus $\mathcal{K}_{\text{rag}} = \{d_1, d_2, \ldots, d_N\}$ whose documents are static and known a priori (e.g., Wikipedia); at each time step it accesses $\mathcal{K}_{\text{rag}}$ through a dense retrieval interface implemented as an embedding index. RAG agents couple retrieval with generation, emphasizing faithful grounding and synthesis of retrieved content.

Table 1 summarizes these distinctions; per-agent implementation details are deferred to Appendix E. Prior work typically trains an agent separately on each dataset, producing parameters θ_{web}, θ_{doc}, and θ_{rag} on $\mathcal{D}_{\text{web}}$, $\mathcal{D}_{\text{doc}}$, and $\mathcal{D}_{\text{rag}}$ respectively. In this work, we investigate how these three information-seeking agents can be consolidated into a unified framework (Figure 1). Existing approaches to consolidation operate at either the data level or the parameter level.

Table 1: Comparison of three different information-seeking agents.

Agent Model π_θ	Environment $\mathcal{K}$	Tool Set $\mathcal{A}$
Web	Internet	*Search*, *Visit*
Doc	Document	*Search*, *Read*
RAG	Wikipedia	*Dense Retrieval*

Data-level Consolidation. Data-level consolidation operates directly on the training data: we merge the ReAct-style trajectory datasets collected from the three environments into a single consolidated dataset

$$\mathcal{D}_{\text{all}} = \mathcal{D}_{\text{web}} \cup \mathcal{D}_{\text{doc}} \cup \mathcal{D}_{\text{rag}}, \tag{3}$$

and then fine-tune a unified agent on $\mathcal{D}_{\text{all}}$, enabling it to handle multiple information environments within a single set of parameters.

Parameter-level Consolidation. Parameter-level consolidation instead merges multiple domain-specialized models in the parameter space: we first train a set of models independently, each specialized for one environment, and then consolidate them by merging their parameters into a generalized model

$$\theta^{(\text{merge})} = \text{merge}\big(\theta_{\text{web}}, \theta_{\text{doc}}, \theta_{\text{rag}}\big)\,,$$

where $\text{merge}(\cdot)$ denotes a generic parameter merging operator. One straightforward instantiation is *linear average* (Wortsman et al., 2022), merging parameters as a convex combination

$$\begin{aligned} \theta^{(\text{merge})} &= \alpha_{\text{web}}\theta_{\text{web}} + \alpha_{\text{doc}}\theta_{\text{doc}} + \alpha_{\text{rag}}\theta_{\text{rag}} \\ \text{s.t.} \quad & \alpha_{\text{web}} + \alpha_{\text{doc}} + \alpha_{\text{rag}} = 1, \quad \alpha_{\cdot} \geq 0. \end{aligned} \tag{4}$$

More advanced merging operators that go beyond convex combinations are presented in §4.

4 Experimental Setup

Training Datasets. Following (Zhang et al., 2026b; Tao et al., 2026; Wu et al., 2025), we construct training data for the three corresponding agents, collecting 4,500 ReAct-style agent trajectories per setting. As shown in Figure 2, the three settings exhibit substantially different tool-use and response

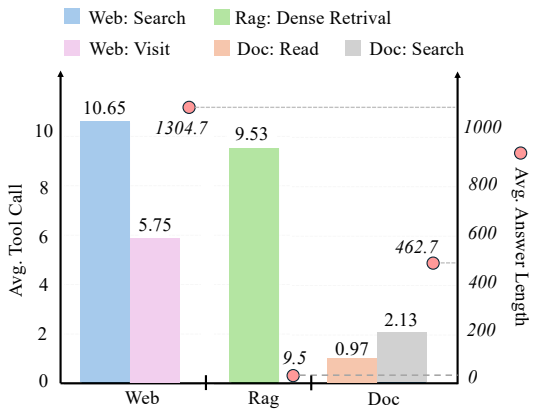

Figure 2: Average tool usage frequency and average answer length across different information-seeking settings in training data.

profiles: Web has the highest tool-call frequency (reflecting iterative search and page visits), Document has the lowest (consistent with its localized search-and-read process), and RAG primarily relies on dense retrieval with fewer tool invocations and substantially shorter responses dictated by its short-form evaluation format. The information-seeking strategies, interaction dynamics, and response length vary significantly across settings. We train on Qwen3-30B-A3B-Think and Qwen3-4B-Think with $128k$ context length. Implementation details are in Appendix B.

Benchmarks. For Web agent evaluation, we use **GAIA** (Mialon et al., 2023) (on its 103-instance text-only subset, following (Li et al., 2025c; Wu et al., 2025)), together with BrowseComp (**BC**) (Wei et al., 2025) and BrowseComp-zh (**BC-zh**) (Zhou et al., 2025), randomly sampling 200 and 100 instances from BC and BC-zh, respectively, due to their high evaluation cost. All are scored by accuracy. For Doc agent evaluation, we use two multimodal long-context document question answering benchmarks, MMLongBench-Doc (**MMBD**) (Ma et al., 2024) and DocBench (**DocB**) (Zou et al., 2025), scored by LLM-as-Judge accuracy. For RAG agent evaluation we use **HotPotQA** (Yang et al., 2018), **AmbigQA** (Min et al., 2020), and **Bamboogle** (Press et al., 2023), reporting the standard Exact Match (EM) and F1 score. To additionally probe out-of-domain (OOD) capability retention beyond the three agentic tasks, we evaluate models on **AIME 2025** (Mathematical Association of America, 2025) for competition mathematics and **LiveCodeBench** (**LCB**, `2408-2502`) (Jain et al., 2024) for code generation, both reported using accuracy. Further benchmark details are in Appendix D.

Table 2: Taxonomy and comparison of parameter-level consolidation methods. We highlight three categories: Basic Interpolation, Interference Resolution, and Data-Driven Optimization. Consensus Strategy specifies the mechanism employed to mitigate interference and harmonize conflicting parameters.

Method	Sparsification	Granularity	Consensus Strategy	Data-Free
Average	–	Model	–	✓
SLERP	–	Model	–	✓
MetaGPT	–	Model	–	✓
LiNeS	–	Layer	–	✓
DARE	✓	Model	Random Dropout	✓
Breadcrumbs	✓	Model	Random Dropout	✓
TIES	✓	Matrix	Sign Consistency	✓
Consensus TA	✓	Matrix	Target Alignment	✓
TADrop	✓	Matrix	Adaptive Dropout	✓
CABS	✓	Matrix	Disjoint Aggregation	✓
PCB Merging	✓	Matrix	Sign Consistency	✓
DELLA	✓	Matrix	Sign Consistency	✓
SCE	✓	Matrix	Sign Consistency	✓
WIDEN	–	Matrix	Magnitude–Direction Disentangle	✓
RAM+	–	Layer	Permutation Alignment	✓
TSV	–	Matrix	Subspace Denoising	✓
ISO-CTS	–	Matrix	Spectral Isotropy	✓
IMPART	✓	Matrix	Spectral Filtering	✓
WUDI	–	Matrix	Subspace Interference Min.	✓
DC-Merge	–	Matrix	Subspace Decomposition	✓
OrthoMerge-C	–	Matrix	Common-Subspace Orthogonalization	✓
OrthoMerge-G	–	Matrix	Global Orthogonalization	✓
AdaMerging	–	Layer	Entropy Minimization	✗
RegMean++	–	Matrix	Activation Matching	✗
CAT Merging	–	Matrix	Feature-Aware Trimming	✗
SLTA	–	Matrix	Submodule Linearity	✗

Aggregated Metrics. Since the benchmarks span heterogeneous scales and sizes, we aggregate them into two overall metrics in three stages: *(i)* a sigmoid-squashed per-benchmark z-score, $s_{i,b} = 1/(1+e^{-z_{i,b}})$ with $z_{i,b} = (x_{i,b} - \mu_b)/\sigma_b$, which removes scale differences and bounds pathological methods near 0; *(ii)* within-task arithmetic means $S_{i,\text{web}}, S_{i,\text{doc}}, S_{i,\text{rag}}$; *(iii)* the **Composite Score** $G(i) = \sqrt[3]{S_{i,\text{web}}\, S_{i,\text{doc}}\, S_{i,\text{rag}}}$, the cross-task geometric mean, which (unlike an arithmetic mean) penalises any single-task collapse. To quantify across-task skew, we additionally report the **Imbalance Score** $\text{Imb}(i) = \max\!\big(0,\, 1 - G(i)/\bar{S}_i\big)$ where $\bar{S}_i$ is the arithmetic mean of the three task scores. It equals 0 for balanced profiles, tends to 1 as performance concentrates on a single task, and is scale-invariant and outlier-robust by construction. All quantities are linearly mapped to $[0, 100]$.

Parameter-level Consolidation Methods. We evaluate **26 representative parameter-level consolidation methods**, organised into three categories in Table 2: *(i)* Basic Interpolation (Average, SLERP, MetaGPT, LiNeS), which linearly or spherically averages weights; *(ii)* Interference Resolution (DARE, Breadcrumbs, TIES, Consensus TA, TADrop, CABS,

PCB Merging, DELLA, SCE, WIDEN, RAM+, TSV, ISO-CTS, IMPART, WUDI, DC-Merge, OrthoMerge-C, OrthoMerge-G), which mitigates cross-task conflicts via sparsification, sign reconciliation, or subspace alignment; and *(iii)* Data-Driven Optimization (AdaMerging, RegMean++, CAT Merging, SLTA), which learns merging coefficients or weights from a few input data. Detailed descriptions are in Appendix C.

5 Empirical Study

We organize our empirical study around four research questions. **(1) Performance**: can parameter-level consolidation match data-level mixing on aggregated in-domain performance? **(2) Generalization**: how well do they retain out-of-domain capabilities such as math and code? **(3) Failure Modes**: which training-scenario factors break which merging methods, and through what mechanism? **(4) Mechanisms**: which design factors govern parameter-level merging's effectiveness, and how does consolidation reshape agent behavior?

5.1 Performance: In-Domain Performance and Cross-Scenario Stability

Research Question 1. *How do parameter-level methods compare to data-level mixing in aggregated performance and cross-scenario stability?*

Results. ❶ Aggregated co-champions. Across the three training scenarios (Qwen3-30B-A3B-Think full-parameter, Qwen3-4B-Think full-parameter, and LoRA-trained Qwen3-30B-A3B-Think), data-level mixing remains a strong baseline. However, RegMean++ is closing the gap once we aggregate to the Composite Score G. RegMean++ and Data Mixing are the *only* two methods ranking in the top-2 on all three scenarios, and they attain the lowest cross-scenario standard deviation. RegMean++ in particular uses a small amount of unlabelled data to obtain per-matrix output features and resolves both inter-model discrepancy and intra-model dependency in a closed form, reaching parity with data mixing. **Well-designed parameter-level consolidation can be comparable to data mixing. ❷ Parameter-level merging is order-agnostic.** Parameter-level consolidation is invariant to the order in which experts are acquired. Data-level training, in contrast, depends critically on joint randomised mixing: applied sequentially on $\mathcal{D}_{\text{web}}$, $\mathcal{D}_{\text{doc}}$, and $\mathcal{D}_{\text{rag}}$, it suffers severe catastrophic forgetting—under a Web→Doc→RAG schedule,

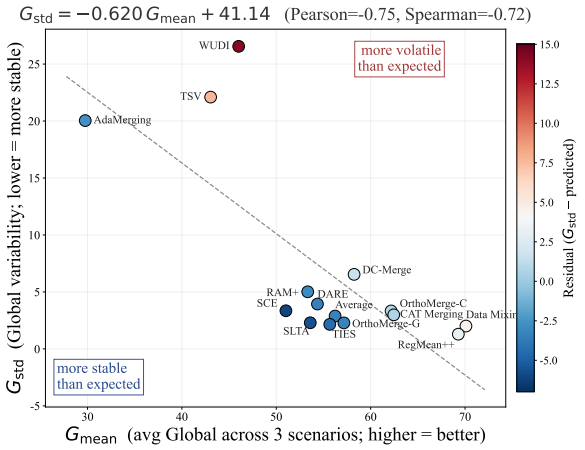

Figure 3: Cross-scenario quality-stability of representative methods. The x-axis is the mean Composite Score across the three training scenarios (Qwen3-30B-A3B-Think, Qwen3-4B-Think, and LoRA-trained Qwen3-30B-A3B-Think); the y-axis is its standard deviation (lower = more stable). The dashed line is the OLS fit.

Web benchmark accuracy drops by 74.38% relative to the post-Web stage and Doc benchmark accuracy by 59.27% relative to the post-Doc stage (Appendix G). This is a practical advantage of parameter-level merging when experts arrive incrementally. **❸ Stability is structurally tied to quality.** Across all three scenarios, G_{mean} and G_{std} are strongly negatively correlated (Pearson $r = -0.75$, Spearman -0.72, Figure 3). Stability is therefore not an independent desideratum. It is a near-necessary condition for trustworthy merging, and new methods should be reported across ≥ 3 scenarios to expose this axis. Appendix J additionally classifies all methods into four categories.

5.2 Generalization: Out-of-Domain Capability Retention

Research Question 2. *Do data-level and parameter-level methods preserve general capabilities outside the training distribution?*

Results. ❶ Data Mixing universally forgets OOD. Data Mixing shows consistently negative ΔAIME and ΔLCB across all three scenarios. Moreover, the magnitude grows monotonically as the training setting becomes more constrained (full 30B-A3B $\rightarrow$ full 4B $\rightarrow$ LoRA 30B-A3B, combined Δ of $-3.48 \rightarrow -13.73 \rightarrow -20.67$). This reflects a direct trade of general capability for in-domain accuracy that is intrinsic to data-level multi-task fine-tuning, and it is amplified whenever the model has less spare capacity to absorb additional knowledge. **❷ A class of parameter-level**

Table 3: Performance comparison using Qwen3-30B-A3B-Think as the backbone model. The performance of Qwen3-4B-Think is reported in Table 8. **G** and **Imb.** are the Composite Score and Imbalance Score describe the overall performance and task imbalance on three tasks. For data- and parameter-level methods, the best, second-best, and third-best values are highlighted with darker, medium, and lighter amber cells, respectively.

Model	Web			Doc		RAG						Overall		OOD	
	GAIA	BC	BC-zh	MMBD	DocB	HotPotQA		AmbigQA		Bamboogle		G↑	Imb.↓	AIME25	LCB
	Acc.	*Acc.*	*Acc.*	*Acc.*	*Acc.*	*EM*	*F1*	*EM*	*F1*	*EM*	*F1*			*Acc.*	*Acc.*
Expert Agent															
Web Agent ($\mathcal{D}_{\text{web}}$)	62.14	21.00	24.00	55.82	79.58	34.10	50.09	44.40	60.87	44.00	61.60	62.96	0.56	81.24	54.10
Doc Agent ($\mathcal{D}_{\text{doc}}$)	68.93	13.50	27.00	64.18	81.79	28.30	46.21	28.90	50.80	41.60	60.98	61.73	0.03	85.00	60.49
RAG Agent ($\mathcal{D}_{\text{rag}}$)	23.30	3.50	11.20	30.03	46.01	48.00	60.98	62.60	73.74	59.20	71.00	30.77	25.32	83.34	59.57
Data-level Consolidation															
Data Mixing ($\mathcal{D}_{\text{all}}$)	64.08	28.00	34.00	63.59	83.29	38.00	42.53	49.50	58.84	53.10	60.20	71.06	0.37	80.21	55.32
Parameter-level Consolidation															
Average	60.19	18.50	29.00	63.40	81.40	18.60	28.93	26.60	38.23	28.00	38.82	53.03	3.29	83.75	61.40
SLERP	60.19	18.00	25.00	62.66	81.67	14.80	26.52	21.60	34.79	18.40	32.96	47.63	5.99	82.51	57.75
MetaGPT	56.31	22.00	33.00	63.40	81.85	13.20	25.65	18.90	33.55	19.20	32.19	48.91	7.96	84.38	61.70
LiNeS	58.25	18.00	33.00	60.35	80.22	22.90	36.20	25.50	41.63	20.00	35.83	53.12	2.86	81.26	55.32
DARE	62.14	19.50	24.00	63.31	80.58	18.70	29.56	25.00	37.66	23.20	37.83	51.21	3.63	84.60	58.05
Breadcrumbs	55.34	10.00	19.00	58.68	77.77	26.10	39.14	28.10	42.25	23.20	41.29	47.86	0.62	78.34	44.68
TIES	62.14	14.50	24.00	60.26	79.76	25.00	38.48	29.60	44.53	32.00	47.22	54.62	0.42	80.01	50.46
Consensus TA	13.59	0.50	6.00	24.49	31.58	13.50	19.77	19.60	30.21	20.00	28.94	14.64	7.05	54.16	24.01
TADrop	54.37	12.00	18.00	59.61	77.77	23.50	37.12	25.60	39.70	22.40	37.84	46.74	1.16	79.58	50.15
CABS	64.08	19.50	29.00	61.92	80.76	24.10	42.89	24.20	46.45	32.80	55.95	59.50	0.76	79.59	51.98
PCB Merging	64.07	23.00	29.00	64.33	81.31	21.40	32.20	31.20	42.00	20.80	33.96	55.22	3.65	83.54	57.45
DELLA	24.27	0.50	9.00	34.94	45.83	18.70	28.99	26.00	37.41	31.20	42.96	23.11	6.53	69.79	37.08
SCE	39.81	2.50	13.00	66.79	54.45	43.10	62.86	59.90	71.26	49.60	62.86	47.04	9.75	83.54	58.66
WIDEN	16.50	2.00	7.00	64.60	81.13	18.70	28.26	26.20	37.13	23.20	36.66	31.82	14.72	67.29	32.83
RAM+	58.25	12.50	7.00	61.92	79.49	22.20	34.30	28.00	41.51	25.60	42.53	46.29	1.67	79.37	50.46
TSV	55.34	16.00	27.00	62.01	79.76	35.70	46.17	50.00	59.06	43.20	55.88	62.74	0.25	82.49	54.41
ISO-CTS	0.97	0.00	0.00	0.00	0.18	0.00	0.00	0.00	0.00	0.00	0.00	4.77	16.68	0.00	0.00
IMPART	62.14	18.00	28.00	62.38	80.58	19.40	29.63	30.60	42.60	24.80	38.47	53.32	2.68	83.54	62.31
WUDI	60.19	19.00	26.00	60.44	78.31	43.30	54.69	55.00	65.87	48.00	62.48	66.23	0.85	80.63	55.62
DC-Merge	57.84	16.00	26.00	64.88	61.62	24.90	34.18	26.30	36.92	25.60	40.50	49.15	1.38	83.75	58.05
OrthoMerge-C	58.25	16.50	23.00	62.57	79.04	29.10	40.34	40.80	52.38	36.00	50.65	58.43	0.02	83.96	59.27
OrthoMerge-G	40.78	18.00	25.00	62.38	81.03	27.40	38.16	37.20	47.50	37.60	50.84	56.07	0.19	86.66	57.45
AdaMerging	60.19	10.00	20.00	58.78	77.31	34.40	45.33	43.50	57.73	49.60	62.29	57.38	1.11	81.66	41.03
RegMean++	61.17	19.50	22.00	64.66	78.95	44.30	56.38	58.50	68.23	50.40	65.34	67.46	0.91	83.96	58.05
CAT Merging	61.17	14.00	24.00	63.59	79.58	25.60	37.36	36.00	48.32	38.40	53.53	57.51	0.12	85.21	58.05
SLTA	60.19	21.00	31.00	62.38	80.31	16.60	27.21	23.00	34.81	20.80	33.98	50.36	5.96	82.91	57.75

methods structurally improves OOD. CAT Merging, DC-Merge, and SCE achieve positive ΔAIME and ΔLCB on *all* three scenarios. They operate at the task-vector level with explicit interference control via null-space projection, component decomposition, or sign consensus, suggesting a structural OOD-friendly inductive bias. RegMean++, although still net-negative on OOD, forgets substantially less than Data Mixing in every scenario, making it a more practical starting point for continual training or general-purpose deployment.

5.3 Failure Modes Across Training Scenarios

Research Question 3. *Which training-scenario factors break which merging methods, and through what mechanism?*

Results. ❶ **Subspace methods collapse under LoRA.** TSV and WUDI are strong on full-parameter trained Qwen3-30B-A3B-Think, but drop on LoRA scenarios. The mechanism is structural: LoRA updates are low-rank and reside in mutually orthogonal subspaces due to random initialization, lacking the spectral overlap that subspace alignment assumes. Instead of filtering noise, these methods truncate the distinct task-specific principal components essential for each domain, producing catastrophic information loss. ISO-CTS already fails for the same reason: its isotropic scaling suppresses the dominant principal encoding critical task knowledge while amplifying noise in the tail directions. ❷ **Small-model fragility.** The 4B backbone exposes a different failure axis. *(a) AdaMerging collapses*: it uses training (rather than test, which is not available) samples as a proxy, causing unstable optimization and over-fitting. *(b) Single-domain experts erase out-of-domain capabilities at smaller models*: On Qwen3-4B-Think, Web and Doc Agents lose RAG ability. The ef-

Table 4: Performance comparison on the LoRA-trained Qwen3-30B-A3B-Think model. We select top-performing parameter-level methods to evaluate. For data- and parameter-level consolidation methods, the best, second-best, and third-best values are highlighted with darker, medium, and lighter amber cells, respectively.

Model	GAIA	Web BC	Web BC-zh	Doc MMBD	Doc DocB	RAG HotPotQA		RAG AmbigQA		RAG Bamboogle		Overall		OOD AIME25	OOD LCB
	Acc.	*Acc.*	*Acc.*	*Acc.*	*Acc.*	*EM*	*F1*	*EM*	*F1*	*EM*	*F1*	G↑	Imb.↓	*Acc.*	*Acc.*
Expert Agent															
Web Agent ($\mathcal{D}_{\text{web}}$)	57.28	13.00	16.00	61.46	78.22	32.20	40.99	32.60	38.15	32.80	40.84	54.80	0.46	72.91	43.47
Doc Agent ($\mathcal{D}_{\text{doc}}$)	60.19	13.00	20.00	59.42	81.76	30.70	47.63	39.50	57.86	38.40	56.48	61.00	0.02	78.54	45.59
RAG Agent ($\mathcal{D}_{\text{rag}}$)	35.92	9.00	17.00	48.96	72.95	44.30	56.27	58.50	69.18	56.00	69.31	50.10	4.47	81.87	51.67
Data-level Consolidation															
Data Mixing ($\mathcal{D}_{\text{all}}$)	57.28	16.00	21.00	62.11	80.49	44.30	55.77	57.30	66.16	50.40	63.68	67.30	0.16	73.96	44.38
Parameter-level Consolidation															
Average	61.17	19.50	25.00	60.81	80.85	15.20	29.36	18.40	35.17	20.30	34.99	55.64	4.58	79.59	50.15
DARE	59.22	15.50	23.00	63.40	80.49	13.60	28.01	16.70	34.30	8.80	27.48	51.96	5.58	78.56	48.94
TIES	49.51	11.50	19.00	57.21	76.23	27.50	41.05	34.40	48.22	36.80	52.05	53.71	0.07	75.63	45.59
SCE	48.54	7.50	20.00	56.03	68.35	43.60	55.33	61.70	72.53	50.40	63.85	55.23	2.27	83.13	58.66
RAM+	56.86	14.50	22.00	62.38	80.85	22.80	39.40	19.70	33.53	25.60	42.40	56.14	1.83	78.12	49.54
TSV	12.62	0.50	6.00	33.36	40.72	1.00	1.22	1.10	1.70	2.40	3.70	12.18	1.01	69.38	41.03
WUDI	9.70	0.50	5.00	23.30	23.87	0.10	0.29	0.30	0.54	1.60	1.67	8.50	9.09	39.80	25.40
DC-Merge	53.40	15.50	21.00	59.70	74.14	37.50	49.02	48.40	60.33	42.40	56.36	61.32	0.13	81.88	59.57
OrthoMerge-C	60.19	13.50	21.00	62.11	76.50	35.60	48.85	45.90	59.14	47.20	61.71	63.30	0.03	81.66	55.93
OrthoMerge-G	53.40	18.50	24.00	63.21	79.04	15.80	29.13	25.60	40.24	16.80	32.50	55.08	3.63	77.91	49.54
AdaMerging	25.24	4.50	14.00	48.71	48.64	1.00	1.23	2.10	2.90	4.80	5.50	21.44	2.73	75.20	47.72
RegMean++	59.22	17.50	21.00	64.60	82.03	44.90	56.02	58.70	68.53	56.80	66.45	70.19	0.12	79.39	49.85
CAT Merging	52.43	16.00	20.00	60.07	75.76	43.40	56.22	55.00	66.89	52.00	63.62	63.86	0.45	82.72	60.49
SLTA	58.25	17.00	23.00	62.85	79.58	17.80	33.90	21.10	41.66	16.00	32.72	55.35	3.33	79.79	48.02

Table 5: Out-of-domain capability retention across the three training scenarios. Each cell reports the *absolute delta* accuracy vs. the backbone (Qwen3-30B-A3B-Think: AIME25= 81.87, LCB= 57.14; Qwen3-4B-Think: AIME25= 75.84, LCB= 48.02). A representative subset of methods is shown.

Method	30B-A3B ΔAIME25	30B-A3B ΔLCB	4B ΔAIME25	4B ΔLCB	30B-A3B (LoRA) ΔAIME25	30B-A3B (LoRA) ΔLCB
Data-level Consolidation						
Data Mixing	−1.66	−1.82	−1.88	−11.85	−7.91	−12.76
Parameter-level Consolidation (representative)						
Average	+1.88	+4.26	+1.04	−3.95	−2.28	−6.99
SCE	+1.67	+1.52	+3.74	−1.21	+1.26	+1.52
DC-Merge	+1.88	+0.91	+4.78	+0.31	+0.01	+2.43
OrthoMerge-C	+2.09	+2.13	+3.11	−0.30	−0.21	−1.21
OrthoMerge-G	+4.79	+0.31	−0.84	+1.52	−3.96	−7.60
WUDI	−1.24	−1.52	−1.47	−7.90	−42.07	−31.74
CAT Merging	+3.34	+0.91	+4.17	+0.92	+0.85	+3.35
RegMean++	+2.09	+0.91	−1.68	−2.43	−2.48	−7.29
AdaMerging	−0.21	−16.11	−75.84	−36.77	−6.67	−9.42

fect largely disappears on Qwen3-30B-A3B-Think models, confirming that catastrophic forgetting in small models is exacerbated. Basic interpolation methods likewise exhibit larger 4B-vs-30B gaps on GAIA, consistent with the hypothesis that larger parameter spaces better absorb merging-induced interference (Yadav et al., 2024).

5.4 Mechanisms: Design Factors and Behavioral Reshaping

Research Question 4. *Which design factors govern parameter-level merging's effectiveness, and how does consolidation reshape agent behavior?*

Results. ❶ **Granularity.** Matrix-level methods consistently outperform model-level interpolation: expert agents exhibit layer-localised capabilities and heterogeneous parameter-update norms (Figure 14), so coarse model-wise averaging dilutes them, whereas fine-grained matrix-wise optimisation preserves them (Luo et al., 2025). ❷ **Consensus-indicator informativeness.** The choice of consensus signal explains most of the spread among parameter-level methods. *(a) Basic interpolation* (Average, SLERP, MetaGPT) succeeds on homogeneous tasks with sphere/linear connectivity but drops sharply on heterogeneous domains (e.g., RAG). *(b) Heuristic sign consensus* (TIES, DELLA, Consensus TA, SCE) alleviates interference but is fragile. Consensus TA still lags substantially on RAG. DELLA and SCE substantially underperform on Web, and an inappropriate sign rule can trigger severe model collapse. *(c) Subspace alignment* (TSV, WUDI) is balanced on full-parameter settings but breaks on LoRA (§5.3). *(d) Data-dependent calibration* (RegMean++, CAT Merging) harmonises inter-model discrepancy and intra-model dependency, improving stability at the cost of forfeiting data independence. More informative consensus indicators yield more robust merges. ❸ **Task affinity.** Web and Doc agents

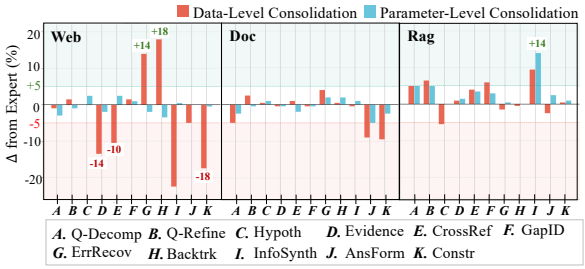

Figure 4: Differences of information-seeking behavior between consolidation methods and the expert agent across benchmarks on Qwen3-30B-A3B-Think. We select the top-performing parameter-level method, RegMean++, as representative. Results are reported across 11 information-seeking categories defined in Table 9.

share tool-use patterns and information-seeking behaviors (Appendix E.4) and can interchange under cross-domain evaluation, whereas RAG operates in a structurally distinct retrieval space. Merging agents with similar environments and behaviors is therefore more likely to recover expert-level performance (Wang et al., 2025a; Team et al., 2025b).

❹ **Behavior preservation.** Beyond accuracy, we annotate the information-seeking behavior of consolidated models against the expert agent on an 11-category schema (A–K) using Claude-Sonnet-4 (per-category deltas in Figure 4). Both data- and parameter-level consolidation alter agent behavior and diversity, but *parameter-level method (RegMean++) preserves the expert agent's behavioral diversity substantially more than data-level mixing*. The tool-call analysis in Appendix H corroborates this via tool-call distributions.

6 Future Outlook

Practical Recommendation. Table 6 summarizes a condensation method selection guide for common deployment constraints.

Design Principles. No existing method dominates across domains, scales, and training regimes. We distil four forward-looking principles for next-generation merging operators.

1. Operate at matrix granularity. Agentic capabilities are layer-localised with heterogeneous update norms (Figure 14). Future operators should default to per-matrix or per-module weighting rather than a single model-wise coefficient.

2. Choose consensus indicators by informativeness, not convenience. Heuristic sign consensus is fragile, subspace alignment is regime-fragile, and full data calibration is robust but data-hungry. We advocate a *cost-controlled middle ground*: lightweight statistics from a few-shot proxy set, with calibration cost (e.g., compute, privacy) treated as a design budget.

3. Account for task affinity via adaptive coefficients. Web and Doc agents share tool-use schemas, whereas RAG is structurally distinct (Appendix E.4), and uniform coefficients ignore this. Future methods should learn *adaptive coefficients* that up-weight synergistic experts and balance orthogonal ones to maximise positive transfer.

4. Respect the intrinsic geometry of parameter updates. Unlike fully fine-tuning, where task vectors often share overlapping singular value spectra, LoRA updates exhibit disjoint spectral signatures constrained by their low-rank initialization. Robust consolidation designs on LoRA must be adaptive to the updates that naturally reside in orthogonal subspaces.

Table 6: Method selection guide by constraint.

Constraint	First choice	Backup	Avoid
Robust across all scenarios	RegMean++, Data Mixing	OrthoMerge-C, CAT Merging	WUDI/TSV (LoRA), AdaMerging (4B)
LoRA-trained experts	RegMean++, Data Mixing, CAT Merging, OrthoMerge-C, DC-Merge	Doc Agent, RAM+	WUDI, TSV, AdaMerging
Small (4B) backbone	Data Mixing, RegMean++, OrthoMerge-C, CAT Merging, DC-Merge	WUDI, LiNeS, PCB Merging, OrthoMerge-G	Web/Doc Agent, CABS, AdaMerging
30B-A3B full-parameter	RegMean++, Data Mixing, WUDI, TSV	Web/Doc Agent, OrthoMerge-C, CAT Merging	RAG Agent, WIDEN, SCE, DELLA, ISO-CTS, Consensus TA
Preserve OOD math/code	CAT Merging, DC-Merge, SCE	OrthoMerge-C	Data Mixing

7 Conclusion

We present the first systematic study of consolidating heterogeneous information-seeking agents across web, document, and knowledge-base environments, benchmarking 26 parameter-level methods against data mixing across three training scenarios. Well-designed parameter-level merging (e.g., RegMean++) matches data mixing at a fraction of its training cost and, unlike data mixing, structurally preserves out-of-domain capabilities. Effective merging requires finer granularity, informative consensus indicators, task-affinity-aware coefficients, and regime-adaptive geometry under LoRA. We hope this work provides foundation for future research on scalable and unified agentic models.

Limitations

The training data of each expert are deliberately moderate in scale and held symmetric across the three domains to isolate consolidation effects from data-quantity confounds. Substantially skewed corpora could reveal regimes not visible in our experiments.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Arash Ahmadian, Seraphina Goldfarb-Tarrant, Beyza Ermis, Marzieh Fadaee, Sara Hooker, and 1 others. 2024. Mix data or merge models? optimizing for diverse multi-task learning. *arXiv preprint arXiv:2410.10801*.

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 *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Rui Dai, Sile Hu, Xu Shen, Yonggang Zhang, Xinmei Tian, and Jieping Ye. 2025. Leveraging submodule linearity enhances task arithmetic performance in llms. *arXiv preprint arXiv:2504.10902*.

MohammadReza Davari and Eugene Belilovsky. 2024. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pages 270–287. Springer.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. 2024. Della-merging: Reducing interference in model merging through magnitude-based sampling. *arXiv preprint arXiv:2406.11617*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim K Goh, Ho-Kin Tang, Daojing He, and 1 others. 2024. Parameter competition balancing for model merging. *Advances in Neural Information Processing Systems*, 37:84746–84776.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, and 1 others. 2025. Towards general agentic intelligence via environment scaling. *arXiv preprint arXiv:2509.13311*.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*.

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 Computer Vision and Pattern Recognition Conference*, pages 18695–18705.

Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. 2025. Mdocagent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025a. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Yiqiao Jin, Rachneet Kaur, Zhen Zeng, Sumitra Ganesh, and Srijan Kumar. 2025b. Slideagent: Hierarchical agentic framework for multi-page visual document understanding. *arXiv preprint arXiv:2510.26615*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, and 1 others. 2025a. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *arXiv preprint arXiv:2509.13305*.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan

Li, Zhengwei Tao, Xinyu Wang, and 1 others. 2025b. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025c. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. 2024. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. *arXiv preprint arXiv:2410.08815*.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. 2025. Marft: Multi-agent reinforcement fine-tuning. *arXiv preprint arXiv:2504.16129*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and 1 others. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Yingfeng Luo, Dingyang Lin, Junxin Wang, Ziqiang Xu, Kaiyan Chang, Tong Zheng, Bei Li, Anxiang Ma, Tong Xiao, Zhengtao Yu, and Jingbo Zhu. 2025. One size does not fit all: A distribution-aware sparsification for more precise model merging. *CoRR*, abs/2508.06163.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, and 1 others. 2024. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010.

Shalini Maiti, Amar Budhiraja, Bhavul Gauri, Gaurav Chaurasia, Anton Protopopov, Alexis Audran-Reiss, Michael Slater, Despoina Magka, Tatiana Shavrina, Roberta Raileanu, and 1 others. 2025. Souper-model: How simple arithmetic unlocks state-of-the-art llm performance. *arXiv preprint arXiv:2511.13254*.

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 *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Mathematical Association of America. 2025. AIME 2025: American invitational mathematics examination. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.

The-Hai Nguyen, Dang Huu-Tien, Takeshi Suzuki, and Le-Minh Nguyen. 2025. Regmean++: Enhancing effectiveness and generalization of regression mean for model merging. *CoRR*, abs/2508.03121.

Ofir Press, Shikhar Murty, Srinivasan Iyer, Mike Lewis, Wen-tau Yih, and Omer Levy. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Qitao Qin, Yucong Luo, Yihang Lu, Zhibo Chu, Xiaoman Liu, and Xianwei Meng. 2025. Towards adaptive memory-based optimization for enhanced retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7991–8004.

Ken Shoemake. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254.

Li Sun, Liu He, Shuyue Jia, Yangfan He, and Chenyu You. 2025a. Docagent: An agentic framework for multi-modal long-context document understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17712–17727.

Wenju Sun, Qingyong Li, Yangliao Geng, and Boyang Li. 2025b. CAT merging: A training-free approach for resolving conflicts in model merging. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. 2025. Vdocrag: Retrieval-augmented generation over visually-rich documents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24827–24837.

Zhengwei Tao, Bo Li, Jialong Wu, Guochen Yan, Huanyao Zhang, Jiahao Xu, Haitao Mi, and Wentao Zhang. 2026. Ragshaper: Eliciting sophisticated agentic rag skills via automated data synthesis. *arXiv preprint arXiv:2601.08699*.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, and 1 others. 2025. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*.

Meituan LongCat Team, Anchun Gui, Bei Li, Bingyang Tao, Bole Zhou, Borun Chen, Chao Zhang, Chengcheng Han, Chenhui Yang, Chi Zhang, and 1 others. 2025a. Introducing longcat-flash-thinking: A technical report. *arXiv preprint arXiv:2509.18883*.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, and 1 others. 2025b. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.

Joachim Utans. 1996. Weight averaging for neural networks and local resampling schemes. In *Proc. AAAI-96 Workshop on Integrating Multiple Learned Models. AAAI Press*, pages 133–138. Citeseer.

Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. 2025. Fusechat: Knowledge fusion of chat models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21629–21653.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, and 1 others. 2025a. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*.

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. 2025b. Lines: Post-training layer scaling prevents forgetting and enhances model merging. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. 2024a. Localizing task information for improved model merging and compression. In *Proceedings of the 41st International Conference on Machine Learning*, pages 50268–50287.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, and 1 others. 2024b. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17716–17736.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, and 1 others. 2025. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, and 1 others. 2025. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115.

Prateek Yadav, Tu Vu, Jonathan Lai, Alexandra Chronopoulou, Manaal Faruqui, Mohit Bansal, and Tsendsuren Munkhdalai. 2024. What matters for model merging at scale? *arXiv preprint arXiv:2410.03617*.

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 Computing Surveys*, 58(8):1–41.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Sihan Yang, Kexuan Shi, and Weiyang Liu. 2026b. Orthogonal model merging. *arXiv preprint arXiv:2602.05943*.

Yan Yang, Yixia Li, Hongru Wang, Xuetao Wei, James Jianqiao Yu, Yun Chen, and Guanhua Chen. 2025a. Impart: Importance-aware delta-sparsification for improved model compression and merging in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18817–18829.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Zongzhen Yang, Binhang Qi, Hailong Sun, Wenrui Long, Ruobing Zhao, and Xiang Gao. 2025b. CABS:

conflict-aware and balanced sparsification for enhancing model merging. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Extend model merging from fine-tuned to pre-trained large language models via weight disentanglement. *arXiv preprint arXiv:2408.03092*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024b. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Xiangchi Yuan, Dachuan Shi, Chunhui Zhang, Zheyuan Liu, Shenglong Yao, Soroush Vosoughi, and Wenke Lee. 2026. Behavior knowledge merge in reinforced agentic models. *arXiv preprint arXiv:2601.13572*.

Han-Chen Zhang, Zi-Hao Zhou, Mao-Lin Luo, Shimin Di, Min-Ling Zhang, and Tong Wei. 2026a. Dcmerge: Improving model merging with directional consistency. *arXiv preprint arXiv:2603.06242*.

Qintong Zhang, Xinjie Lv, Jialong Wu, Baixuan Li, Zhengwei Tao, Guochen Yan, Huanyao Zhang, Bin Wang, Jiahao Xu, Haitao Mi, and 1 others. 2026b. Docdancer: Towards agentic document-grounded information seeking. *arXiv preprint arXiv:2601.05163*.

Zhisong Zhang, Tianqing Fang, Kaixin Ma, Wenhao Yu, Hongming Zhang, Haitao Mi, and Dong Yu. 2025. Enhancing web agents with explicit rollback mechanisms. *arXiv preprint arXiv:2504.11788*.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, and 1 others. 2025. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. 2024. Metagpt: Merging large language models using model exclusive task arithmetic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1724.

Dawei Zhu, Rui Meng, Jiefeng Chen, Sujian Li, Tomas Pfister, and Jinsung Yoon. 2025. Doclens: A tool-augmented multi-agent framework for long visual document understanding. *arXiv preprint arXiv:2511.11552*.

Anni Zou, Wenhao Yu, Hongming Zhang, Kaixin Ma, Deng Cai, Zhuosheng Zhang, Hai Zhao, and Dong Yu. 2025. Docbench: A benchmark for evaluating llm-based document reading systems. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 359–373.

A Discussion

Why agent consolidation is crucial? Recent progress in agent reinforcement learning has largely focused on optimizing agents within individual task verticals, yet practical agentic systems must integrate behaviors learned under heterogeneous environments. While reinforcement learning improves vertical performance, it also induces strong environment-specific policy entanglement, making naive integration unstable and interference-prone. We view this work as an initial exploration of the agent consolidation problem, and hope that the observations and insights derived from our study can help inform and inspire subsequent work in this direction. As a natural next step, we will extend this line of work to agentic reinforcement learning settings, where the interaction between policy optimization and consolidation, as well as the resulting increase in behavioral diversity, is expected to play a central role.

B Implementation Details

We use Megatron-LM to finetune Qwen3-30B-A3B-Think[1] and Qwen3-4B-Think[2]. For LoRA training, we set LoRA rank to 8 and apply LoRA modules to all linear layers. The training loss curves and benchmark performance of trained Qwen3-30B-A3B-Think, Qwen3-4B-Think, and Qwen3-30B-A3B-Think with LoRA models are shown in Figures 5, 6, and 7, respectively.

We use the *vLLM* framework (Kwon et al., 2023) for inference, with the temperature set to 0.6, the top_p parameter set to 0.95, and the presence penalty set to 1.1.

C Detailed Descriptions of Parameter-level Consolidation Methods

Below, we provide brief descriptions of the 26 parameter-level consolidation methods evaluated in

[1] https://huggingface.co/Qwen/Qwen3-30B-A3B-Thinking-2507

[2] https://huggingface.co/Qwen/Qwen3-4B-Thinking-2507

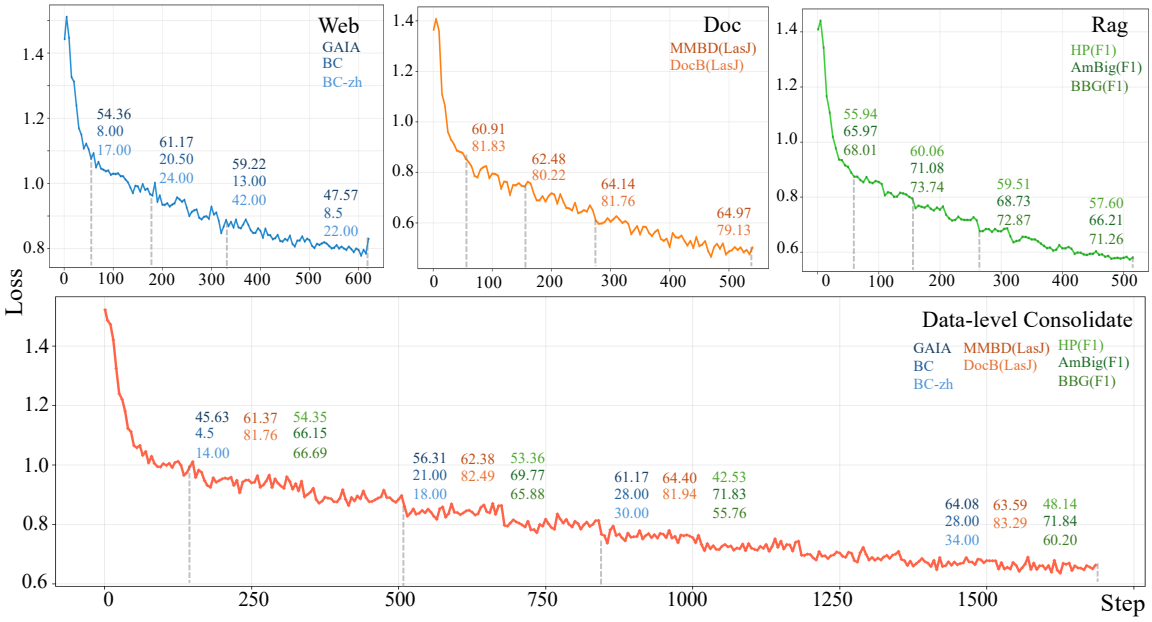

Figure 5: Training loss curves and benchmark performance during training of Qwen3-30B-A3B-Think.

this work. The order is consistent with Table 3 and groups methods by the three categories in Table 2.

Average (Wortsman et al., 2022) performs simple parameter averaging: $\theta^{(\text{merge})} = \sum_{i \in \mathcal{S}} \alpha^{(i)} \theta^{(i)}$. It assumes linear connectivity and aims to aggregate weights into a single centroid solution to improve generalization and robustness without increasing inference cost.

SLERP (Shoemake, 1985; Ahmadian et al., 2024) merges models by interpolating parameters along a spherical path: $\theta^{(merge)} = \frac{\sin((1-t)\Omega)}{\sin(\Omega)}\theta_1 + \frac{\sin(t\Omega)}{\sin(\Omega)}\theta_2$. This accounts for the geometric structure of the high-dimensional parameter space.

MetaGPT (Zhou et al., 2024) solves for a regularized optimization problem for the scaling coefficients to merge.

LiNeS (Wang et al., 2025b) applies depth-dependent scaling to parameter updates. It scales updates in deeper layers more aggressively while keeping shallow layers closer to pre-trained weights, reducing forgetting and interference.

DARE (Yu et al., 2024b) randomly drops $p\%$ parameter updates ($\boldsymbol{T} = \theta_{Expert} - \theta_{Base}$) on each expert model and rescales the remaining ones to reduce redundancy of task vectors and minimize interference between models.

Breadcrumbs (Davari and Belilovsky, 2024) constructs sparse masks by filtering out both insignificantly small perturbations and excessively large outliers from task vectors, retaining only the effective breadcrumbs.

TIES (Yadav et al., 2023) includes three steps: *Trimming* insignificant redundant parameters, *Electing* a unified sign direction based on majority voting (i.e., $s = \text{sign}\left(\sum_{i \in \mathcal{S}} \text{sign}(\theta^{(i)})\right)$), and *Merging* only the values that align with the elected direction.

Consensus TA (Wang et al., 2024a) constructs per-task masks to eliminate selfish weights. It filters these conflicting parameters to seek a consensus among task vectors.

TADrop (Luo et al., 2025) is a distribution-aware sparsification method. It analyzes the weight distribution of each tensor to adaptively determine local sparsity ratios, preserving complex update patterns.

CABS (Yang et al., 2025b) performs Conflict-Aware sparsification to prune overlapping and Balanced Sparsification with n:m pruning. This ensures a uniform distribution of retained weights across layers and prevents interference.

PCB Merging (Du et al., 2024) scores parameter importance in task vectors based on intra- and inter-task balancing, and discards the bottom-ranked parameters.

DELLA (Deep et al., 2024) drops parameter updates $\boldsymbol{T}$ based on magnitude and rescales the remaining ones. It prioritizes high-magnitude updates with task-critical information while randomly discarding low-magnitude updates.

SCE (Wan et al., 2025) includes three stages: *Selecting* top $p\%$ elements with high variance; *Cal-*

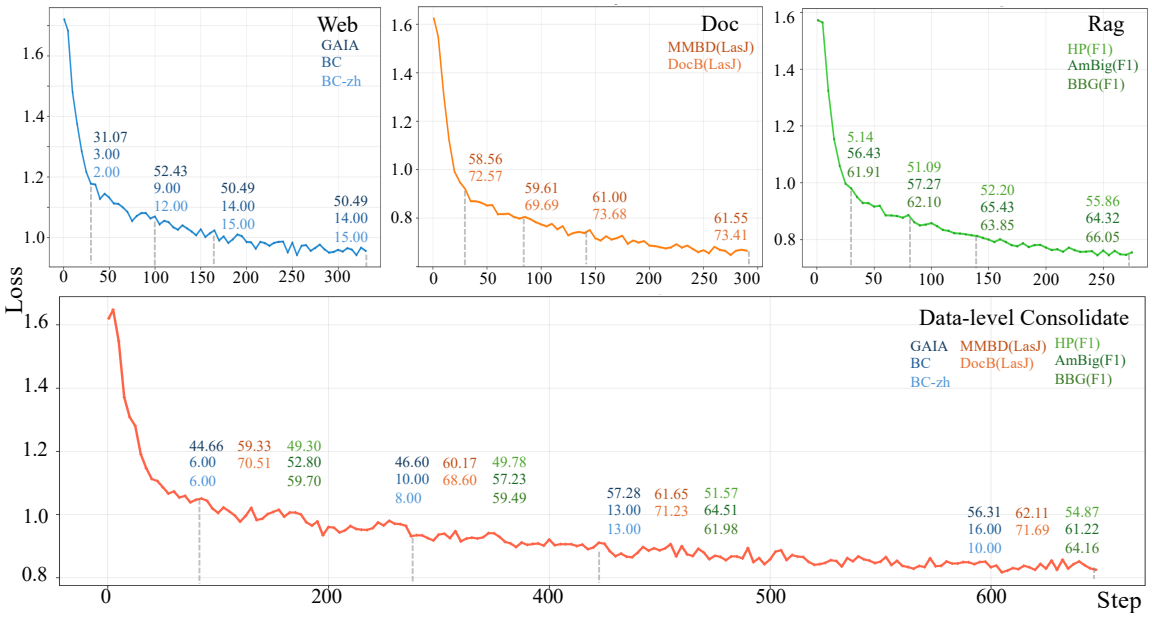

Figure 6: Training loss curves and benchmark performance during training of Qwen3-4B-Think.

culating matrix-level merging coefficients based on the sum of squares of these selected elements; *Erasing* parameter updates with conflicting signs to eliminate interference.

WIDEN (Yu et al., 2024a) disentangles each task vector into magnitude and direction components and re-weights them by per-parameter importance before fusion, alleviating magnitude conflicts that arise when task vectors carry highly heterogeneous scales.

RAM+ (Yuan et al., 2026) extends activation-aware re-basin merging with adaptive component selection: it first permutation-aligns expert models into a shared loss basin, then adaptively re-weights aligned components according to per-task activation statistics to mitigate residual interference.

TSV (Gargiulo et al., 2025) leverages Singular Value Decomposition (SVD) to compress task vectors. It identifies the principal directions of weight updates and merges models in a low-rank subspace, retaining the most expressive components while reducing noise.

ISO-CTS (Marczak et al., 2025) aligns task vectors by flattening their singular value spectrum. It explicitly models both a shared common subspace and task-specific subspaces to harmonize conflicts in the parameter space.

IMPART (Yang et al., 2025a) utilizes SVD for importance-aware delta-sparsification. It dynamically adjusts the sparsity ratio for different singular vectors based on their contribution to the task.

WUDI (Cheng et al., 2025) directly minimizes a layer-wise interference objective. It identifies and mitigates specific components of task vectors that cause destructive interference, optimizing the merge without additional data.

DC-Merge (Zhang et al., 2026a) decomposes each task vector into a shared common component and a task-specific residual component, and applies differential scaling to the two parts before aggregation, preserving consensus directions while attenuating task-specific noise.

OrthoMerge-C (Yang et al., 2026b) performs orthogonalization-based merging with a *common-subspace* constraint: it projects each task vector onto a shared common subspace estimated from cross-task task vectors, suppressing components that lie outside the consensus direction.

OrthoMerge-G (Yang et al., 2026b) is the *global* variant of OrthoMerge-C: it orthogonalizes each task vector against the union of all other task-vector directions, yielding stronger interference removal at the cost of more aggressive truncation of task-specific information.

AdaMerging (Yang et al., 2024) automatically learns layer-wise merging coefficients by optimizing the entropy minimization objective $\min_{\boldsymbol{\lambda}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{\text{test}}}[\mathcal{H}(f(\boldsymbol{x};\theta^{(base)} + \sum_i \boldsymbol{\lambda}_i \odot \boldsymbol{T}^{(i)}))]$ on unlabeled test data.

RegMean++ (Nguyen et al., 2025) enhances generalization by explicitly modeling intra- and cross-layer dependencies. It derives the closed-form solution $\boldsymbol{W}^{(merge)} = (\sum_i \hat{\boldsymbol{G}}_i)^{-1} \sum_i \hat{\boldsymbol{G}}_i \boldsymbol{W}_i$,

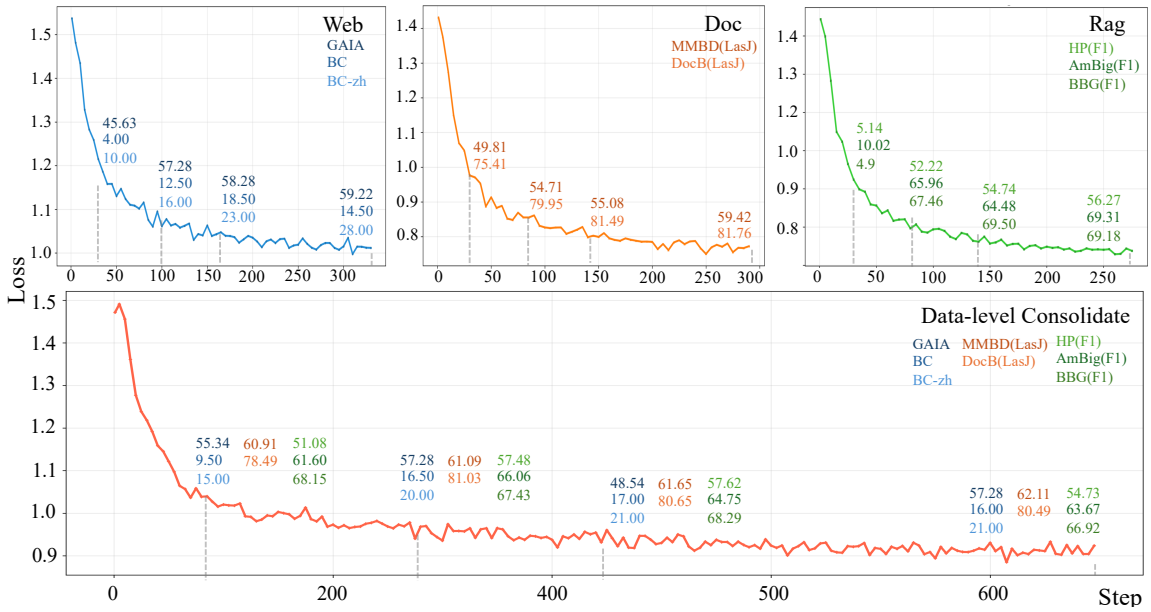

Figure 7: Training loss curves and benchmark performance during training of Qwen3-30B-A3B-Think with LoRA.

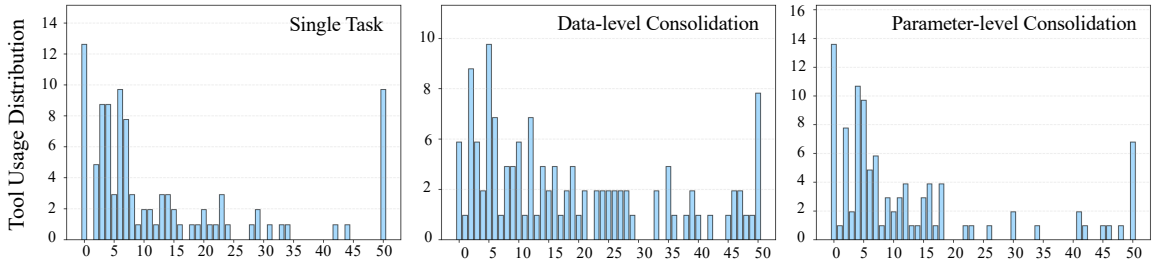

Figure 8: Tool call distributions of different consolidation strategies across Web benchmarks.

where $\hat{\boldsymbol{G}}_i$ represents the regularized inner-product matrix of input features propagated through the merged model.

CAT Merging (Sun et al., 2025b) resolves feature-level interference by projecting task vectors onto the null space of conflicting activations via the transformation $\boldsymbol{T}_i - \boldsymbol{T}_i \boldsymbol{B}_k \boldsymbol{B}_k^\top$. It selectively removes components aligned with the removal basis $\boldsymbol{B}_k$ that disrupt shared feature representations while preserving task-specific knowledge.

SLTA (Dai et al., 2025) performs *Subspace Linear Task Arithmetic*: instead of operating in the full parameter space, it restricts task-vector arithmetic to a low-rank subspace identified from base-model activations, focusing the merge on representation-relevant directions and reducing off-manifold interference.

D Benchmarks

This section provides a detailed overview of the benchmarks employed for Web, Document, and RAG agents.

D.1 Web Agent Benchmarks

GAIA (Mialon et al., 2023). General AI Assistants (GAIA) is a benchmark designed to evaluate general-purpose AI systems on questions that are conceptually simple for humans but challenging for models due to the requirement for complex reasoning, tool usage, and multi-modality. Following the protocols in prior studies (Li et al., 2025c; Wu et al., 2025), we focus on the text-only validation subset, comprising 103 instances. This subset isolates the agent's reasoning and browsing capabilities from visual processing factors. Performance is reported using accuracy.

BrowseComp (Wei et al., 2025). BrowseComp evaluates web agents on realistic user tasks that require interacting with real-world websites. The dataset emphasizes "entangled information," where answers cannot be retrieved via simple queries and necessitate persistent navigation and multi-page information integration. Given the significant time and computational cost associated with live web

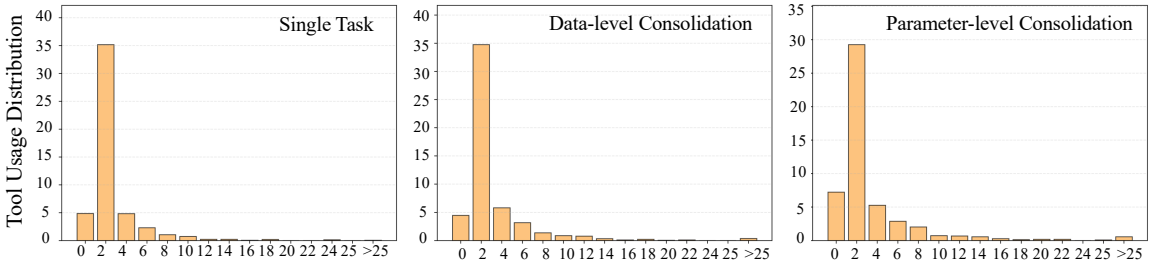

Figure 9: Tool call distributions of different consolidation strategies across Doc benchmarks.

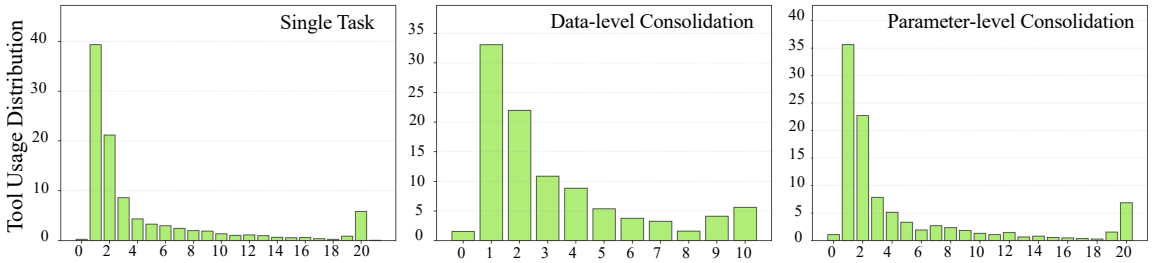

Figure 10: Tool call distributions of different consolidation strategies across RAG benchmarks.

browsing evaluation, we evaluate our method on a randomly sampled subset of 200 instances. We report success rates based on answer accuracy.

BrowseComp-zh (Zhou et al., 2025). As the Chinese counterpart to BrowseComp, this dataset is constructed to reflect the unique characteristics of the Chinese internet ecosystem (e.g., distinct platform ecosystems and search behaviors). It tests the agent's robustness in non-English environments. Similar to BC, we utilize a randomly sampled subset of 100 instances for evaluation and report accuracy.

D.2 Document Agent Benchmarks

MMLongBenchDoc (Ma et al., 2024). MMLongBenchDoc focuses on multimodal long-context document understanding. It features lengthy documents (averaging approximately 20k tokens, 47.5 pages) rich in layout elements such as charts, tables, and images across seven diverse domains.. A significant portion of the questions requires cross-page reasoning, testing the agent's ability to maintain long-term dependency and fuse multimodal information across extensive contexts.

DocBench (Zou et al., 2025). DocBench consists of 229 real-world documents and 1,082 questions, designed to assess the robustness of document reading systems across five domains and four major question types, providing a holistic view of an agent's document processing capabilities.

D.3 RAG Agent Benchmarks

HotPotQA (Yang et al., 2018). HotPotQA is specifically designed to test multi-hop reasoning. Answering questions requires the agent to find and reason over multiple supporting documents to derive the correct answer, challenging the agent's ability to decompose complex queries.

AmbigQA (Min et al., 2020). AmbigQA addresses the challenge of ambiguity in open-domain questions. The agent must handle queries with multiple potential answers by retrieving diverse evidence or disambiguating the context, thereby testing the precision and coverage of the retrieval process.

Bamboogle (Press et al., 2023). Bamboogle consists of questions where the answer cannot be found on the first page of standard search engine results. This dataset evaluates the agent's resilience and its ability to perform multi-step retrieval when direct search fails.

D.4 Out-of-Domain Benchmarks

To assess whether the agent-consolidated model still retains the broader reasoning capabilities of its backbone, we evaluate it on two benchmarks that lie completely outside the Web, Document, and RAG training distributions. Both probe long-horizon symbolic reasoning, one in natural language with formal math, and one in executable code, which our training data never directly targets.

AIME 2025 (Mathematical Association of America, 2025). The American Invitational Mathematics Examination is a long-running selective high-school olympiad administered by the Mathematical Association of America. The 2025 edition consists of two contests (AIME I and AIME II), each comprising 15 problems whose answers are integers in $[0, 999]$, totalling 30 problems that cover algebra, combinatorics, number theory, and geometry. The integer-answer format eliminates judging ambiguity, allowing strict exact-match scoring; problems require multi-step symbolic manipulation rather than rote computation, making AIME a clean probe of out-of-distribution mathematical reasoning. We evaluate under chain-of-thought decoding and report accuracy (percentage of correctly answered problems averaged over both contests).

LiveCodeBench (Jain et al., 2024). LiveCodeBench (LCB) is a holistic, contamination-free coding benchmark that continuously aggregates new problems from competitive-programming platforms (LeetCode, AtCoder, and Codeforces) and re-releases the test suite as periodically updated time slices. Each problem ships with a hidden test bank that judges Pass@1 by executing the generated code against unseen inputs. We adopt the `2408-2502` time slice, which restricts evaluation to problems released between August 2024 and February 2025; this time-windowed protocol is designed to minimise overlap with model pre-training corpora and provides a tighter measurement of genuine code-generation skill than static benchmarks. We report Pass@1 accuracy under greedy decoding.

E Agents

E.1 Web Agent

The web agent employs two types of tools, following previous work (Wu et al., 2025; Tao et al., 2025): *Search* and *Visit*:

- ***Search*** is used to retrieve information via the Google search engine. Its inputs are search queries, and it supports issuing multiple queries in parallel. For each query, the tool returns the top-10 results, where each result includes a title, a snippet, and the corresponding URL.
- ***Visit*** is used to access and process specific web pages. The input consists of a set of urls, each paired with a dedicated visit goal. The full content of each page is first retrieved using Jina, after which a summarization model extracts goal-relevant information. In this work, we use gpt-oss-120b (Agarwal et al., 2025) as the summarization model.

E.2 Doc Agent

The document agent employs two types of tools, following previous work (Zhang et al., 2026b):

- ***Search.*** is used to conduct keyword-based full-text search over the given documents. Its inputs are search keywords, and it returns the corresponding section IDs, page numbers, and surrounding text snippets for each match. A visible window is used to constrain the snippet length for efficient localization. This tool provides the agent with global textual signals for guiding subsequent information access.
- ***Read.*** is used to access and process specific document sections. The input consists of a goal and a set of section IDs. For each section, the tool first retrieves local textual information, consisting of all text within the section, and local visual information, consisting of images and tables within the section, together with a page-level screenshot that captures the full layout of the page containing the section. Subsequently, a multimodal summarization model M_m is used as an auxiliary reader to jointly integrate textual and visual inputs and return consolidated goal-relevant content.

E.3 RAG Agent

The RAG agent employs a dense retrieval tool:

- ***Dense Retrieval.*** is used to retrieve relevant documents from a knowledge base, Wikipedia. The parameters include a query and a top-k value, representing the search string and the maximum number of relevant documents to return, respectively. The tool encodes the query using a pretrained text embedding model[3] (Karpukhin et al., 2020) and computes similarity scores between the query and documents indexed in the KB. It returns documents whose similarity scores exceed a threshold τ, while ensuring that the number of returned documents does not exceed k.

E.4 Tool Schema

This section details the tool schemas provided to the agent. The specific JSON structures defining

[3] https://github.com/facebookresearch/DPR

the tools for the Web agent, Doc agent, and RAG agent are shown in Figures 11, 12, and 13.

F Results on Qwen3-4B-Think

We report the expert agent, data-level consolidation, and parameter-level consolidation performance comparison results on Qwen3-4B-Think model in Table 8.

G Practical Advantages of Parameter-level Consolidation

Although both data-level mixing and parameter-level consolidation can produce a single multi-capability agent, parameter-level merging offers three deployment-oriented advantages that are difficult to recover with data-level mixing. We summarise them here as a complement to the in-domain and OOD comparisons in §5.1-§5.2.

1. Modular continual learning. Data-level consolidation requires re-training on the union of all task corpora whenever a new domain arrives—approximately $1{,}960$ A100 GPU h for our three-task setting (Web, Doc, RAG), and the cost scales linearly with the number of incorporated tasks. Parameter-level consolidation instead maintains an independent expert per task and combines arbitrary subsets on demand. In our pipeline, most merges take < 20 GPU h, a $\sim 10^2\times$ speed-up. When experts must be acquired sequentially (rather than mixed jointly), data-level fine-tuning further suffers from catastrophic forgetting. We quantify this effect in Table 7 below. We sequentially fine-tune the Qwen3-30B-A3B-Think backbone in the order Web $\rightarrow$ Doc $\rightarrow$ RAG. Because Web and Doc share tool-use patterns and information-seeking behaviors (Appendix E.4), training on Doc after Web yields a mild positive transfer on the Web benchmarks. The subsequent RAG stage, however, severely erases the previously acquired Web and Doc capabilities: averaged across the three Web benchmarks, accuracy collapses by 74.38% relative to the post-Web stage; averaged across the two Doc benchmarks, by 59.27% relative to the post-Doc stage. The final model essentially regresses to a single-domain RAG expert. In contrast, parameter-level merging is order-agnostic by construction (§5.1, ❷): the consolidated weights are a symmetric function of $\{\theta_{\text{web}}, \theta_{\text{doc}}, \theta_{\text{rag}}\}$ and exhibit no schedule sensitivity, which justifies its preference whenever experts arrive incrementally or training cannot be repeated.

2. Data sovereignty. Parameter-level merging is the only viable option when source datasets cannot be pooled because of privacy regulations (e.g., medical or enterprise records), licensing restrictions, or jurisdictional constraints. It is also the natural integration path when only *pre-trained expert checkpoints* are available—e.g., when consolidating community-released agents whose original training data are no longer accessible.

3. Better OOD knowledge retention. As established in §5.2 and Table 5, several parameter-level methods achieve positive ΔAIME and ΔLCB on *all* three training scenarios, whereas Data Mixing forgets math and code capabilities universally, with the magnitude growing as the training budget tightens (full 30B-A3B $\rightarrow$ full 4B $\rightarrow$ LoRA 30B-A3B, combined Δ of $-3.48 \rightarrow -13.73 \rightarrow -20.67$).

Together, these three properties enable concrete deployment scenarios that are otherwise impractical: *(i)* privacy-preserving or federated consolidation across distributed expert holders; *(ii)* integration of pre-existing community agents without re-collecting their training data; and *(iii)* rapid prototyping by merging a curated subset of experts on demand.

H Comparison on Tool Call

Figures 8, 9, and 10 respectively illustrate the tool call distributions of the expert agent, data-level consolidation, and top-performing parameter-level consolidation on Web benchmarks, Doc benchmarks, and RAG benchmarks. Parameter-level consolidation exhibits a tool-call distribution that is closer to data-level consolidation and the expert agent.

I Single-Capability Champions Across Scenarios

Figure 15 visualises per-capability rankings of all merging methods across the three training scenarios. We sort methods within each capability panel by their *mean rank* across scenarios; cell colour encodes the rank in each individual scenario (greener = better). Three observations stand out: **(i)** the Web capability champion shifts substantially across scenarios—*Average* ascends to #1 in the 4B and LoRA settings while TIES/CABS lead on 30B-A3B; **(ii)** Doc is dominated by data-dependent methods, with *RegMean++* ranking first or near-first in every scenario; **(iii)** RAG is essentially monopolised by the *RAG Agent* expert, the only entity that ranks #1 on RAG in all three scenarios. These

Table 7: Sequential fine-tuning of Qwen3-30B-A3B-Think under the Web → Doc → RAG schedule. Each row reports per-benchmark accuracy at the end of the corresponding training stage (RAG benchmarks are EM / F1). Continual training on the next domain steadily erases previously acquired capabilities, with the final RAG stage causing a 74.38% Web drop and a 59.27% Doc drop relative to the prior stage-empirical evidence of catastrophic forgetting in data-level continual fine-tuning.

Stage	Web			Doc		RAG (EM / F1)		
	GAIA	BC	BC-zh	MMBD	DocB	HotPotQA	AmbigQA	Bamboogle
Web	62.14	21.00	24.00	55.82	79.58	34.10 / 50.09	44.40 / 60.87	44.00 / 61.60
Web → Doc	65.64	22.00	30.00	64.42	80.76	35.20 / 51.45	48.00 / 64.83	40.10 / 58.53
Web → Doc → RAG	19.41	1.00	7.00	27.91	31.22	43.60 / 54.77	54.40 / 65.42	59.10 / 68.50

patterns complement the aggregated views in Tables 3, 4, and 8, and motivate the cross-scenario stability analysis in §5 (Figure 3).

J Four-Class Method Taxonomy via $G \times \mathrm{Imb}$

Jointly inspecting the Composite Score G and the Imbalance Score Imb decomposes the evaluated methods into four engineering classes with distinct remediation strategies (using the 30B-A3B numbers from Table 3 as the reference scenario; the same partitions hold qualitatively on 4B and LoRA, see Tables 8 and 4):

All-rounders (G high, Imb low). Methods that balance the three task domains. Representatives: Data Mixing ($G = 71.06$, $\mathrm{Imb} = 0.37$), RegMean++ ($G = 67.46$, $\mathrm{Imb} = 0.91$), OrthoMerge-C ($G = 58.43$, $\mathrm{Imb} = 0.02$), CAT Merging ($G = 57.51$, $\mathrm{Imb} = 0.12$), and OrthoMerge-G ($G = 56.07$, $\mathrm{Imb} = 0.19$). *Engineering action:* directly deployable.

Specialists (G moderate, Imb high). Methods that excel on a single domain at the cost of the others. Representatives: RAG Agent ($G = 30.77$, $\mathrm{Imb} = 25.32$, RAG Score ≈ 87 vs. Web Score ≈ 20); also Web Agent and Doc Agent on the 4B scenario after RAG erasure. *Engineering action:* retain for the dominant capability; pair with complementary components for missing ones.

Collapsed (G low, Imb low). Methods that fail uniformly across domains. Representatives: ISO-CTS ($G = 4.77$, all three task scores below 10) and Consensus TA ($G = 14.64$, all three task scores below 25). *Engineering action:* structural failure; usually unrecoverable without redesigning the merging operator.

Imbalanced underperformers (G low, Imb high). Methods that fail on one or two domains while remaining acceptable on another. Representatives: SCE on 30B-A3B ($G = 47.04$, $\mathrm{Imb} = 9.75$, Web Score ≈ 27) and AdaMerging on 4B ($G = 10.46$, with collapsed Doc and Web). *Engineering action:* targeted intervention—fix the missing capability via auxiliary data or a complementary expert.

The cross-scenario migrations of class memberships further reveal robustness profiles: DC-Merge moves from *Collapsed* at 30B to *All-rounder* at 4B and LoRA; WUDI moves from *All-rounder* at 30B to *Imbalanced* (in fact almost *Collapsed*) under LoRA; and AdaMerging is *Imbalanced* at 30B but *Collapsed* at 4B. These migrations align with the cross-scenario failure analysis in §5.3.

K Potential Risks

A stronger and more general information-seeking agent would be constructed, which inherits and may amplify the risks of their per-domain components, or synthesize some unexpected behavior. Unified web, document, and knowledge-base capabilities lower the cost of large-scale information gathering and could be misused for surveillance, targeted phishing, or automated extraction of private content. They may also compound hallucinations across heterogeneous sources when retrieval grounding is weak.

Table 8: Performance comparison with Qwen3-4B-Think model as backbone. **G** and **Imb.** are the Composite Score and Imbalance Score describe the overall performance and task imbalance on three tasks. For data- and parameter-level consolidation, the best, second-best, and third-best values are highlighted with darker, medium, and lighter amber cells, respectively.

Model	Web			Doc		RAG						Overall		OOD	
	GAIA	BC	BC-zh	MMBD	DocB	HotPotQA		AmbigQA		Bamboogle		G↑	Imb.↓	AIME25	LCB
	Acc.	*Acc.*	*Acc.*	*Acc.*	*Acc.*	*EM*	*F1*	*EM*	*F1*	*EM*	*F1*			*Acc.*	*Acc.*
Expert Agent															
Web Agent ($\mathcal{D}_{\text{web}}$)	50.49	14.00	15.00	57.39	69.98	1.60	2.80	4.60	7.43	3.20	5.15	43.55	11.68	72.08	39.82
Doc Agent ($\mathcal{D}_{\text{doc}}$)	46.60	7.50	15.00	61.00	73.68	1.40	3.10	2.80	4.00	2.40	3.30	40.88	12.02	70.83	45.59
RAG Agent ($\mathcal{D}_{\text{rag}}$)	41.75	11.00	11.00	45.01	52.00	42.70	55.86	55.50	66.05	54.40	64.32	56.06	5.94	75.84	37.39
Data-level Consolidation															
Data Mixing ($\mathcal{D}_{\text{all}}$)	56.31	16.00	10.00	62.11	71.69	43.50	54.87	54.40	64.16	50.40	61.22	71.92	0.99	73.96	36.17
Parameter-level Consolidation															
Average	55.34	13.00	21.00	55.91	68.69	14.80	28.76	20.80	38.70	15.20	32.62	60.02	1.37	76.88	44.07
SLERP	49.51	10.00	16.00	58.78	71.96	11.40	24.38	17.70	36.42	12.80	30.78	55.66	1.22	74.80	44.68
MetaGPT	46.60	13.00	13.00	58.13	70.05	16.60	26.12	23.70	35.31	20.00	29.81	57.13	0.44	72.30	40.73
LiNeS	46.60	11.00	16.00	53.33	74.29	24.60	39.10	30.70	47.85	27.20	44.71	62.60	0.09	68.55	39.21
DARE	52.43	9.00	14.00	62.29	71.51	15.10	29.20	21.80	40.54	27.20	43.54	59.91	0.26	74.59	39.51
Breadcrumbs	19.42	1.50	3.00	45.66	62.34	14.80	22.39	15.40	23.57	20.80	30.20	33.17	7.75	75.64	47.42
TIES	38.83	13.50	20.00	59.52	70.24	11.10	29.83	16.10	36.65	17.60	37.02	58.67	0.90	70.64	40.12
Consensus TA	8.74	1.50	1.00	31.79	30.04	6.50	13.84	6.50	10.05	7.20	16.28	18.82	4.05	23.74	8.51
TADrop	45.63	10.50	16.00	56.28	70.60	16.10	34.02	19.40	41.52	14.40	36.32	57.46	0.23	73.33	41.95
CABS	44.66	10.00	14.00	56.19	71.23	0.50	0.85	0.50	0.00	0.00	0.62	38.59	13.40	67.50	33.74
PCB Merging	50.49	12.00	16.00	60.07	74.32	18.80	28.50	22.80	34.81	20.00	30.78	60.36	0.64	78.96	45.59
DELLA	6.80	1.00	2.00	30.04	29.58	6.40	13.03	6.10	13.81	8.00	17.19	18.45	4.82	34.38	11.25
SCE	27.18	3.00	18.00	52.49	55.93	25.90	35.17	37.40	49.72	28.80	37.96	50.79	2.02	79.58	46.81
WIDEN	17.48	2.50	2.00	34.56	41.47	11.90	21.41	10.30	20.57	12.80	24.67	25.26	4.42	42.92	16.41
RAM+	50.49	12.00	17.00	56.93	71.05	10.90	28.80	19.10	39.25	12.80	32.30	57.56	1.04	73.74	41.33
TSV	47.57	13.00	21.00	53.88	51.91	13.50	27.74	17.70	33.95	20.80	34.83	54.21	1.81	75.23	41.64
ISO-CTS	0.97	0.00	0.00	18.67	9.00	0.00	0.00	0.00	0.00	0.00	0.00	10.73	5.58	0.00	0.00
IMPART	53.39	10.50	15.00	58.78	65.06	10.30	26.25	18.80	37.02	17.60	36.46	56.03	0.63	70.41	39.01
WUDI	52.43	12.00	19.00	58.96	71.05	20.70	32.56	30.00	43.15	20.80	34.19	63.32	0.26	74.37	40.12
DC-Merge	49.51	8.00	13.00	58.50	70.69	35.10	46.45	41.20	52.68	45.60	54.71	64.26	1.38	80.62	48.33
OrthoMerge-C	47.57	12.50	12.00	59.06	70.87	30.30	42.83	38.50	51.58	40.80	54.43	65.60	0.63	78.95	47.72
OrthoMerge-G	55.34	12.00	15.00	59.33	71.60	16.40	29.12	19.90	38.92	21.60	35.81	60.34	0.48	75.00	49.54
AdaMerging	6.80	0.00	7.00	0.00	9.17	0.00	0.00	0.00	0.00	0.00	0.00	10.46	16.72	0.00	11.25
RegMean++	48.54	12.50	16.00	58.96	76.31	35.50	46.12	49.10	59.40	40.80	50.76	70.21	0.49	74.16	45.59
CAT Merging	51.46	12.50	15.00	61.83	71.87	24.60	34.58	37.00	47.65	25.60	38.12	65.12	0.01	80.01	48.94
SLTA	47.57	12.50	13.00	54.16	76.68	11.00	23.88	16.30	34.27	16.00	32.15	55.09	1.09	74.79	45.28

Table 9: Definitions of the 11 information-seeking behavior categories used in Figure 4.

ID	Category	Definition
A	Q-Decomp	**Query decomposition**: breaking a complex information need into multiple sub-questions or sub-goals that can be solved sequentially.
B	Q-Refine	**Query refinement**: modifying or reformulating the query based on intermediate results (e.g., adding constraints, using synonyms, changing phrasing) to improve retrieval relevance.
C	Hypoth	**Hypothesis generation**: forming a tentative answer or explanation before full verification, typically as a working theory to guide further search.
D	EvidenceV	**Evidence verification**: checking whether retrieved information supports (or contradicts) the current hypothesis/answer, including consistency checks against the original question.
E	CrossRef	**Cross-reference validation**: comparing or triangulating information across different sources, sections, or passages to confirm correctness or resolve conflicts.
F	GapID	**Gap identification**: explicitly recognizing missing, unclear, or insufficient information needed to answer the question, and stating what is still required.
G	ErrRecov	**Error recovery**: handling failed, irrelevant, or noisy retrieval/search attempts by applying corrective actions (e.g., retrying with new terms, switching strategy).
H	Backtrck	**Backtracking**: abandoning an unproductive line of search and returning to an earlier step, alternative hypothesis, or previous plan to continue progress.
I	InfoSynth	**Information synthesis**: integrating evidence from multiple retrieved pieces into a coherent understanding, combining complementary facts rather than relying on a single snippet.
J	AnsForm	**Answer formulation**: organizing and presenting the final response clearly (e.g., structured reasoning, ordered steps, concise conclusion) based on accumulated evidence.
K	Constr	**Constraint satisfaction**: ensuring the final answer satisfies explicit requirements and constraints (format, scope, conditions, units, time range, etc.) stated by the user/task.

Tool Schemas for Web Agent

Search

```
{
  "type": "function",
  "function": {
    "name": "search",
    "description": "Performs batched web searches: supply an array 'query'; the tool retrieves
          the top 10 results for each query in one call.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "array",
          "items": {
            "type": "string"
          },
          "description": "Array of query strings. Include multiple complementary search queries
                in a single call."
        }
      },
      "required": ["query"]
    }
  }
}
```

Visit

```
{
  "type": "function",
  "function": {
    "name": "visit",
    "description": "Visit webpage(s) and return the summary of the content.",
    "parameters": {
      "type": "object",
      "properties": {
        "url": {
          "type": ["string", "array"],
          "items": {
            "type": "string"
          },
          "minItems": 1,
          "description": "The URL(s) of the webpage(s) to visit. Can be a single URL or an
              array of URLs."
        },
        "goal": {
          "type": "string",
          "description": "The goal of the visit for webpage(s)."
        }
      },
      "required": ["url", "goal"]
    }
  }
}
```

Figure 11: Tool schema for web agent: *Search* and *Visit*.

Tool Schemas for Doc Agent

Search

```
{
    "type": "function",
    "function": {
        "name": "search",
        "description": "Find and extract all paragraphs and sections where any of the provided
            search terms appear",
        "parameters": {
            "type": "object",
            "properties": {
                "keywords": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    },
                    "description": "A list of query keywords for searching"
                }
            },
            "required": ["keywords"]
        }
    }
}
```

Read

```
{
    "type": "function",
    "function": {
        "name": "read",
        "description": "Read multiple sections by section IDs and extract useful information
            from all content contained in those sections, including both visual elements and
            textual elements.",
        "parameters": {
            "type": "object",
            "properties": {
                "section_ids": {
                    "type": "array",
                    "items": {
                        "type": "string"
                    },
                    "description": "A list of section IDs to read from the document"
                },
                "goal": {
                    "type": "string",
                    "description": "The user goal that guides what useful information should be
                        extracted from the selected sections"
                }
            },
            "required": ["section_ids", "goal"]
        }
    }
}
```

Figure 12: Tool schema for doc agent: *Search* and *Read*.

Tool Schema for RAG Agent

Dense Retrieval

```
{
  "type": "function",
  "function": {
    "name": "dense_retrival",
    "description": "Semantic vector search over the knowledge base. Falls back to configured
        top_k or 5.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
          "minLength": 1,
          "description": "Natural language question or statement to retrieve against the
              knowledge base."
        },
        "top_k": {
          "type": "integer",
          "minimum": 1,
          "description": "Optional override for number of retrieved results."
        }
      },
      "required": ["query"]
    }
  }
}
```

Figure 13: Tool schema for RAG agent: dense semantic retrieval over a vectorized knowledge base.

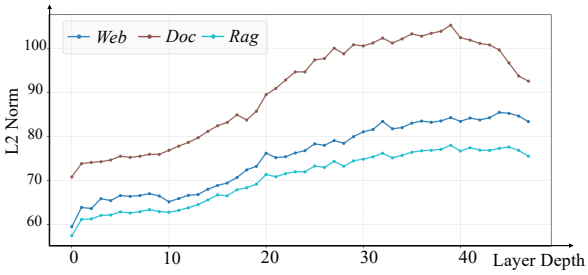

Figure 14: Layer-wise L2 norm of parameter updates of expert agents for Web, Doc, and RAG agents across model depth.

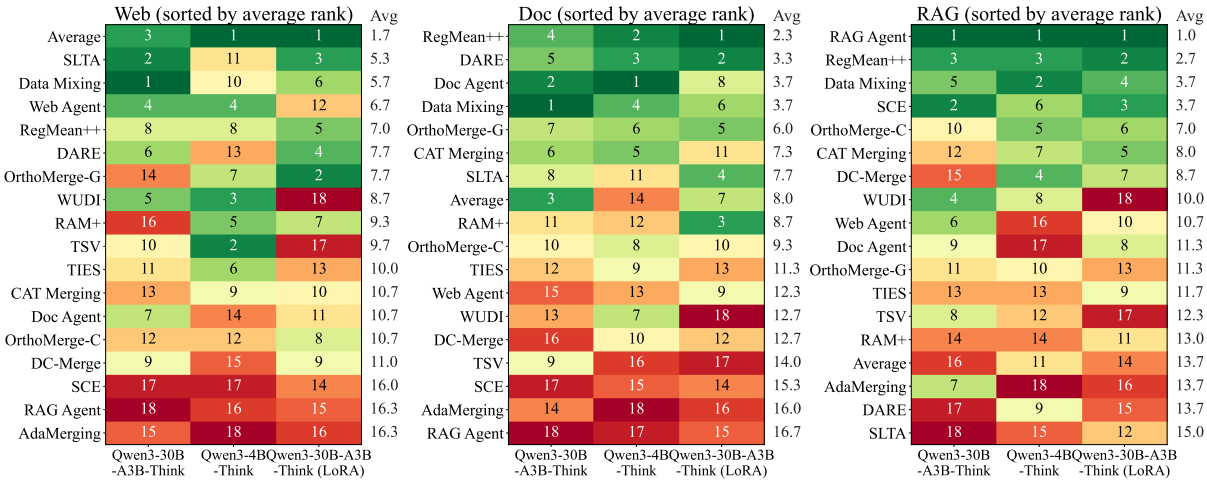

Figure 15: Per-capability rankings of merging methods across the three training scenarios (Qwen3-30B-A3B-Think, Qwen3-4B-Think, and LoRA-trained Qwen3-30B-A3B-Think). Each cell encodes the method's rank for that capability in that scenario (greener = better rank). Left: Web; centre: Doc; right: RAG. Methods within each panel are sorted by their mean rank across scenarios.