SemiAdapt-Instruct: Extensible Instruction Tuning via Latent Domain-Specialised Adapters

Anonymous ACL submission

Abstract

Instruction-tuned LLMs are deployed into environments where domains evolve, yet extending a fine-tuned model's capabilities without full retraining remains an unsolved practical challenge. We present SemiAdapt-Instruct, a modular framework that discovers latent instruction domains, trains per-domain LoRA adapters in parallel, and performs parameter-free routing, incorporating new domains via single-adapter training without modifying existing components. SemiAdapt-Instruct outperforms full model fine-tuning across all configurations on both ROUGE-L and LLM-as-a-judge evaluation, while matching single LoRA fine-tuning and delivering extensibility that monolithic approaches cannot provide. We empirically demonstrate this extensibility by showing that updating a single adapter with new domain data outperforms all monolithic baselines. Our study also finds that independent discovery methods converge on the same specialisation-friendly domains. These findings demonstrate that decomposing heterogeneous instruction data into latent domains enables extensible NLP systems where evolving domains require only targeted single-adapter updates, eliminating the need for full model retraining.

1 Introduction

Supervised instruction fine-tuning has become the standard approach for adapting large language models to follow human instructions (Wei et al., 2021; Ouyang et al., 2022), yet the field has optimised for benchmark performance while leaving deployment maintainability largely unsolved (Kiela et al., 2021). Models are evaluated on static benchmarks but deployed into environments where domains evolve and new capabilities are continuously required. However, full retraining on ever-growing combined datasets poses a structural barrier for practitioners without access to large-scale compute (McGiff and Nikolov, 2025).

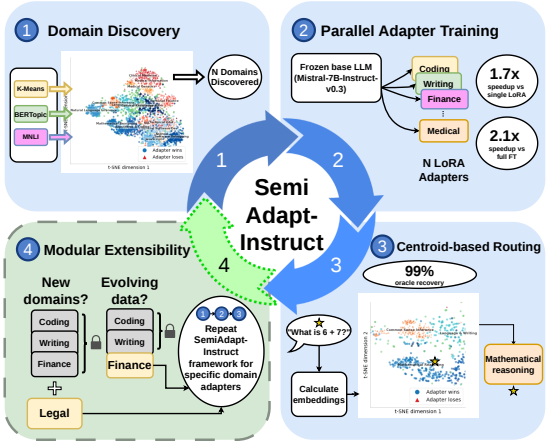

Figure 1: SemiAdapt-Instruct pipeline: (1) automatic domain discovery via unsupervised clustering or zero-shot classification, (2) independent parallel adapter training per domain, (3) parameter-free centroid routing at inference, and (4) modular extensibility via single-adapter updates for new domains or data.

Although parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA) reduce the computational cost of instruction tuning (Hu et al., 2021), a single adapter trained on heterogeneous data inherits two problems. First, heterogeneous training data introduces conflicting gradient signals across domains (Yu et al., 2020; Ling et al., 2025), which limits the degree to which a single model can specialise (Wang et al., 2023). Second, monolithic fine-tuning produces brittle, difficult-to-extend systems (Huang et al., 2026). In such cases, incorporating new domains or updating existing knowledge requires full retraining on the combined dataset, risking catastrophic forgetting of previously learned behaviours (McCloskey and Cohen, 1989; Li et al., 2024; Luo et al., 2025).

Standard Mixture-of-Experts (MoE) architectures address specialisation through jointly trained experts and routing components (Shazeer et al., 2017), but do not resolve the extensibility problem.

Joint training couples all experts together, meaning new domains cannot be added without retraining the full system. Wang et al. (2023) confirm that instruction-tuned models on heterogeneous data fail to achieve consistent cross-domain performance, identifying modularity as a key unmet need. Parameter-efficient methods such as LoRA reduce the cost of retraining but do not eliminate it. Therefore, a single adapter trained on heterogeneous data must still be fully retrained to include new data.

Prior work (Mcgiff and Nikolov, 2026) demonstrates that domain specialisation enables parameter-efficient methods to match or outperform full-model fine-tuning in a low-resource translation. However, that work focused on translation performance within a single fixed MNLI taxonomy and did not address extensibility, automatic domain discovery, or generalisation to heterogeneous instruction tuning.

In this work, we introduce SemiAdapt-Instruct, a modular framework that eliminates the retraining bottleneck for heterogeneous instruction tuning. Rather than assuming a fixed domain taxonomy, SemiAdapt-Instruct automatically discovers latent domains by comparing unsupervised clustering (K-means, BERTopic) and zero-shot classification (MNLI) across a dataset four times larger than prior work. Domain adapters are trained independently and in parallel with routing occurring at inference via parameter-free cosine similarity to domain centroids. Unlike standard MoE or monolithic fine-tuning, new domains are incorporated by training a single additional adapter, while new data within an existing domain requires retraining only the relevant adapter, leaving all other components unchanged. Our contributions are as follows:

- **System.** SemiAdapt-Instruct, a modular instruction tuning framework that matches single LoRA fine-tuning and outperforms full model fine-tuning, while enabling domain updates that monolithic approaches cannot support.
- **Empirical Study.** A systematic cross-method comparison of three automatic domain discovery methods for instruction tuning.
- **Extensibility Demonstration.** A post-deployment domain update simulation showing that single-adapter retraining on new domain data outperforms monolithic baselines.

2 Related Work

2.1 Instruction Tuning and Data Heterogeneity

Instruction tuning has emerged as a fundamental post-training step for aligning large language models with human intent and expectations (Wei et al., 2021; Ouyang et al., 2022). However, ever-expanding instruction datasets contribute to inherent heterogeneity. This presents a central challenge where training on dissimilar tasks introduces conflicting gradient signals and consequently degrades individual task performance (Wang et al., 2023).

Recent work addresses instruction data efficiency through filtering and quality valuation methods rather than structural partitioning. Centroid-based clustering with confidence-guided selection has been used to curate high-quality subsets for efficient training of instruction tuning models (Cai et al., 2025). Similarly, ROSE (Wu et al., 2025) leverages pairwise preference loss as a reward signal to select high-utility samples for task-specific instruction tuning, while LimaCost (Moon et al., 2025) estimates sample value by measuring gradient proximity to the LIMA dataset, selecting data that maximises alignment performance. These approaches improve data efficiency but discard low-scored samples, risking reduced generalisation to out-of-distribution examples. SemiAdapt-Instruct takes an alternative approach, partitioning the full instruction dataset into domain-specific subsets without discarding any examples, with each assigned to exactly one domain adapter.

2.2 Parameter-Efficient Fine-Tuning and Adapter Routing

Parameter-efficient fine-tuning (PEFT) methods reduce the cost of adapting large pre-trained models by updating only a small fraction of parameters. One such method, Low-Rank Adaptation (Hu et al., 2021), achieves competitive performance with full fine-tuning at significantly reduced memory cost by expressing weight updates as low-rank matrix decompositions. However, catastrophic forgetting is a well-documented challenge in heterogeneous instruction tuning settings, whereby updating model parameters on a new task or domain degrades prior capabilities (McCloskey and Cohen, 1989; Li et al., 2024; Luo et al., 2025). Recent work addresses this challenge through constrained LoRA updates. CLoRA (Lu et al., 2025) introduces orthogonal subspace regularisation to preserve general capabil-

ities during sequential fine-tuning of large language models. Alternatively, SLIM (Han et al., 2025) proposes dynamic routing between LoRA adapters and identity layers to suppress forgetting. Pletenev *et al.* (Pletenev et al., 2025) demonstrate that incorporating new knowledge via LoRA risks degrading previously learned capabilities, with bias amplified when training data is skewed towards specific entities.

The proliferation of domain-specific adapters has motivated work on multi-adapter routing. Tian *et al.* (Tian et al., 2025) propose training a dedicated selector adapter to route between domain and task-specific LoRA modules at inference time. Similarly, GLIDER (Li et al., 2025) constructs LLM-generated global routing vectors combined with local task vectors to select from a pool of PEFT experts. Unlike these approaches, which require either a trained routing component or LLM-generated routing signals, our approach extends parameter-free routing to the instruction tuning setting.

2.3 Domain Discovery Methods

Unsupervised discovery of latent structure in text has a long history in NLP. Latent Dirichlet Allocation (Blei et al., 2003) introduced probabilistic topic modelling, representing documents as mixtures of topics and topics as distributions over words. Neural approaches such as BERTopic (Grootendorst, 2022) and Top2Vec (Angelov, 2020) extend this by clustering dense sentence embeddings, producing more semantically coherent topics. K-Means clustering has been used to identify representative subsets of full instruction datasets (Yu et al., 2024). Zero-shot classification via natural language inference enables domain assignment using predefined label categories without task-specific training (Yin et al., 2019). SemiAdapt-Instruct conducts, to our knowledge, the first systematic comparison of clustering, topic modelling, and zero-shot classification, under identical controlled conditions for instruction tuning domain discovery.

3 Methodology

This section outlines our approach for discovering latent domains in raw instruction tuning data, training domain-specific adapters and evaluating our centroid-based routing method.

3.1 Latent Domain Discovery

This section describes three different approaches for discovering latent domains in a heterogeneous instruction dataset.

3.1.1 Unsupervised Clustering (K-Means)

This experiment uses K-means clustering to partition training data into latent domains. Sentence embeddings are generated for each input using the all-MiniLM-L6-v2 model prior to clustering (Wang et al., 2020).

Selecting K. Identifying an appropriate cluster count proceeds in two stages. To avoid the computational cost of running full K-means across the entire dataset for every candidate value, Mini-Batch K-Means is employed during this exploratory phase, using batch sizes of 4,096 and three random initialisations. Inertia values are recorded across $K \in \{1, \ldots, 50\}$, with the KneeLocator algorithm (Arvai, 2026) applied to the resulting elbow curve to mathematically identify the inflection point at $K = 17$.

To validate this, silhouette and Calinski-Harabasz scores are computed for $K \in \{14, 15, 16, 17, 18, 19, 20\}$ on a 20,000-sample subset. Silhouette score peaks at $K = 17$ (0.029), while Calinski-Harabasz peaks earlier at $K = 14$ (212.4). We select $K = 17$ as the point where the elbow criterion and silhouette score agree on an optimum. The low absolute silhouette value reflects the inherently overlapping surface patterns of instruction data rather than poor cluster quality, a dissociation confirmed empirically by centroid routing accuracy of 95.0% (Section 5.4). Full K-means clustering is then run on the complete dataset using $K = 17$.

3.1.2 Topic Modelling (BERTopic)

Traditional topic modelling methods such as Latent Dirichlet Allocation are primarily designed for large text corpora and perform poorly on short texts like individual prompts (Qiang et al., 2020). We therefore adopt BERTopic (Grootendorst, 2022), which generates contextual embeddings and applies context-aware clustering to identify coherent topics, automatically determining the number of domains (Robledo et al., 2022) and yielding 107 topics. Given the identification of outlier examples under this approach, we additionally train adapters on super-groupings ($N = 24$), where coarser topic clusters are derived via intertopic distance merging

with outliers excluded, enabling a direct comparison of topic granularity within the same discovery framework.

3.1.3 Zero-Shot Classification (MNLI)

In this experiment, we apply zero-shot domain classification to the combined instruction dataset using `facebook/bart-large-mnli` (Lewis et al., 2020), a BART-based model fine-tuned on the Multi-Genre Natural Language Inference (MNLI) corpus (Williams et al., 2018). For each instruction, the model performs textual entailment between the instruction text and a set of hypothesis strings of the form *"This text is about [label]"*, returning a probability distribution over the candidate labels. The label with the highest entailment probability is assigned as the domain, along with a confidence score.

The candidate label set is drawn directly from the 24 super topic groups produced by the BERTopic experiment described in Section 3.1.2. This design choice serves two purposes. First, it enables a controlled comparison between a distributional clustering approach (BERTopic) and a classification-based approach (MNLI) over an identical label space, isolating the effect of the domain discovery mechanism rather than the label ontology. Second, it reflects a realistic deployment scenario in which an existing topic taxonomy is used to label new data without requiring model retraining.

3.2 Domain-Specific Adapter Training

Following domain discovery, we train a dedicated LoRA (Hu et al., 2021) adapter for each identified domain across each domain discovery method. LoRA constrains weight updates to a low-rank subspace, preserving general model capabilities while enabling domain-specific specialisation, a property we validate empirically through comparison with full fine-tuning in Section 5.1.

Adapters are trained independently on each domain split, entirely decoupling domain discovery, adapter training, and inference routing. This modularity allows new domains to be incorporated by training a single additional adapter without retraining existing components, preserving existing adapter performance by construction since the base model remains frozen throughout.

Baselines. We fine-tune two baselines on the full training set: a single LoRA adapter providing a domain-agnostic parameter-efficient baseline, and a full fine-tune baseline updating all model parameters to assess whether parameter efficiency is a convenience or a necessity. Hyperparameters for all training runs are reported in Section 4.

3.3 Centroid-Based Inference Routing

Domain specialisation requires identifying the correct domain at test time without ground-truth labels. We employ a parameter-free routing mechanism that requires no additional learned components. For each discovered domain k, a centroid $\mathbf{c}_k$ is precomputed as the mean of all training instruction embeddings assigned to that cluster:

$$\mathbf{c}_k = \frac{1}{|D_k|} \sum_{x_i \in D_k} \phi(x_i) \tag{1}$$

where D_k is the set of training instructions assigned to domain k and $\phi(x_i)$ denotes the sentence embedding of instruction x_i. At inference, a test instruction x is routed to the domain $\hat{k}$ whose centroid has the highest cosine similarity:

$$\hat{k} = \arg\max_k \frac{\phi(x) \cdot \mathbf{c}_k}{\|\phi(x)\| \, \|\mathbf{c}_k\|} \tag{2}$$

The adapter corresponding to domain $\hat{k}$ is then applied to the frozen base model for generation. This approach adds no trainable parameters and requires only a single embedding forward pass per inference call. We evaluate two routing configurations: an oracle setting, where each test instance is routed to its ground-truth domain adapter, and a centroid setting, where instances are routed to their predicted domain adapters.

4 Experimental Setup

We construct a heterogeneous instruction dataset by combining eight publicly available instruction-following corpora spanning diverse domains, yielding 509,174 total examples split into 458,255 training and 50,918 test examples via a 90/10 partition applied consistently across all discovery methods.

Dataset	Domain	#Samples	Avg Tokens
Alpaca-cleaned	General	51,760	∼120
CodeAlpaca-20K	Code	20,022	∼40
WebGPT	QA/Retrieval	18,994	∼169
Databricks Dolly 15K	General	15,011	∼129
Finance-Alpaca	Finance	68,912	∼162
AlpaCare-MedInstruct	Medicine	52,002	∼184
FLAN (CoT)	Reasoning	74,771	∼70
StackOverflow Dialogues	Programming	207,702	∼130
Total		**509,174**	

Table 1: Composition and statistics of the instruction-tuning dataset. Additional dataset information is described in Appendix A

All adapters are trained on Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) with LoRA rank r=16, α=32, dropout=0.05, targeting query, key, value, and output projections. Training uses batch size 16, learning rate 2e-4 with AdamW, for 1 epoch per domain on a single NVIDIA A100 80GB GPU.

We evaluate using ROUGE-L (Lin, 2004) computed via the `rouge-score` package[1] as a universal lexical metric enabling consistent comparison across all domains and discovery methods. To validate automatic metric findings, we additionally conduct blind pairwise preference evaluation using GPT-4o as an independent judge (Zheng et al., 2023) across all domain discovery methods, following established LLM-as-a-judge methodology.

5 Results and Discussion

5.1 Overall Performance

Table 2 presents ROUGE-L scores across all domain discovery methods on matched evaluation splits. Across all four configurations, oracle and centroid-routed adapter systems consistently outperform full model fine-tuning (Δ = +0.013–+0.026 ROUGE-L), demonstrating that modular parameter-efficient adaptation is preferable to monolithic full-model retraining on heterogeneous instruction data regardless of discovery method choice.

Relative to the more competitive single LoRA baseline, K-means (K=17) is the only method where both oracle and centroid routing exceed the matched single LoRA score, achieving ROUGE-L of 0.4341 and 0.4314 respectively against 0.4282 for single LoRA. BERTopic super-24 oracle also outperforms its matched single LoRA baseline (+0.003). Across all methods, oracle ROUGE-L differences relative to single LoRA remain within 0.01, confirming broad performance parity. GPT-4o pairwise evaluation further confirms this parity as K-means and BERTopic-107 achieve near-identical GPT-4o preference rates (48.8% and 48.5%), while MNLI and BERTopic super-24 show modest judge preference for single LoRA (41.0% and 43.0%), consistent with their negative ROUGE-L deltas. Per-domain analysis revealing where specialisation gains concentrate is presented in Section 5.2.

Notably, ROUGE-L and GPT-4o evaluation diverge for BERTopic-107. Despite a negative ROUGE-L delta (-0.006), GPT-4o judges its responses as comparable to single LoRA (48.5%), suggesting that lexical overlap is an imperfect proxy for response quality in latent domain instruction evaluation.

Across all configurations, modular domain specialisation achieves performance parity with single LoRA fine-tuning while outperforming full model fine-tuning, with the added advantage that new domains and data require only single-adapter updates.

5.2 Per-Domain Analysis

Table 3 and Figure 4 in Appendix C show that 9 of 17 K-means domains achieve positive ROUGE-L deltas relative to single LoRA, with winning domains averaging $+0.007$ and losing domains averaging -0.008. GPT-4o pairwise evaluation confirms parity with single LoRA (48.8% average win rate) while showing consistent preference for domain adapters over full fine-tuning across all 17 domains (51.0–74.0%, mean 63.2%). Together, these results demonstrate that domain specialisation matches single LoRA quality while outperforming full model fine-tuning, delivering an extensible system without systematic quality degradation.

From a ROUGE-L perspective, the largest gains concentrate in Mathematical Reasoning (c10, $\Delta = +0.021$), Technical Q&A (c2, $\Delta = +0.012$), and General Software Development (c14, $\Delta = +0.011$). GPT-4o win rates versus single LoRA reveal that ROUGE-L gains do not uniformly reflect preference improvements: Mathematical Reasoning achieves only 44% preference over single LoRA despite a positive ROUGE-L delta ($+0.021$), suggesting adapters learn response structure without improving semantic correctness.

Several domains show agreement between ROUGE-L and GPT-4o evaluation. JavaScript/jQuery, General Knowledge and Natural Language Inference all show consistent positive signals across both metrics, suggesting domain specialisation provides reliable quality improvement in these domains. GPT-4o preference for domain adapters over full fine-tuning is consistent across the top-performing domains, with General Software Dev (74%), Mathematical Reasoning (66%), and Technical Q&A (63%) all showing substantial adapter preference.

The medical domains present an interesting metric divergence. Medical Genetics shows a meaningful ROUGE-L loss ($\Delta = -0.010$) yet GPT-4o preferences tell the opposite story, with Medical Ge-

[1] https://github.com/google-research/rouge

Method	Oracle	Centroid	LoRA	Full-FT	Δ (Ora−LoRA)	Δ (Ora−FT)	GPT-4o vs LoRA	GPT-4o vs FT
K-means (K=17)	**0.4341**	**0.4314**	0.4282	0.4082	**+0.0059**	**+0.0259**	**48.8%**	**63.2%**
BERTopic (N=107)	0.4371	0.4324	0.4435	0.4240	−0.0064	+0.0131	48.5%	58.0%
MNLI (N=24)	0.4166	0.4083	0.4169	0.3995	−0.0003	+0.0171	41.0%	60.5%
BERTopic (N=24)	0.4446	0.4369	0.4412	0.4220	+0.0030	+0.0226	43.0%	62.0%

Table 2: ROUGE-L on matched evaluation splits (500 examples per domain). Single LoRA and Full FT scores computed on same matched subsets. Δ (Ora−LoRA) = oracle minus single LoRA; Δ (Ora−FT) = oracle minus full fine-tuning. GPT-4o Win% computed on 200 pairs per method (1,700 for K-means, see Section 5.2).

Cl.	Domain	Train N	Oracle	LoRA	FT	Δ (O−L)	Δ (O−FT)	vs LoRA	vs FT
c10	Mathematical Reasoning	14,744	0.5249	0.5042	0.5019	+0.021	+0.023	44%	66%
c2	Technical Q&A	34,393	0.4063	0.3939	0.3608	+0.012	+0.045	43%	63%
c14	General SW Dev	48,069	0.4428	0.4316	0.4141	+0.011	+0.029	48%	74%
c5	Language & Writing	33,865	0.4672	0.4594	0.4275	+0.008	+0.040	50%	74%
c11	NL Inference	37,241	0.5909	0.5833	0.5789	+0.008	+0.012	51%	59%
c4	Content Gen	32,297	0.3173	0.3144	0.2952	+0.003	+0.022	45%	62%
c15	Algorithms & Coding	24,790	0.4765	0.4738	0.4700	+0.003	+0.007	44%	64%
c13	JavaScript/jQuery	28,916	0.4264	0.4241	0.4034	+0.002	+0.023	61%	62%
c8	General Knowledge	48,487	0.4684	0.4667	0.4308	+0.002	+0.038	54%	62%
c12	Common Sense Inf.	6,956	0.5591	0.5595	0.5524	−0.000	+0.007	47%	51%
c7	Personal Finance	20,204	0.1956	0.1978	0.1928	−0.002	+0.003	45%	59%
c6	Clinical Medicine	14,515	0.4337	0.4361	0.4182	−0.002	+0.016	58%	61%
c0	Medical Info	20,929	0.4002	0.4039	0.3651	−0.004	+0.035	48%	57%
c16	Medical Genetics	13,357	0.4058	0.4152	0.3739	−0.010	+0.032	53%	64%
c3	SW Debugging	37,935	0.3823	0.3944	0.3719	−0.012	+0.010	43%	65%
c1	Android/Mobile Dev	21,225	0.3972	0.4105	0.3868	−0.013	+0.010	46%	64%
c9	SQL/Databases	20,332	0.4566	0.4702	0.4484	−0.014	+0.008	49%	67%
	Macro Avg		**0.4324**	0.4282	0.4160	**+0.004**	**+0.021**	**48.8%**	**63.2%**

Table 3: Per-domain oracle evaluation for K-means (K=17) on matched 500-example evaluation subsets. Δ (O−L) and Δ (O−FT) report oracle minus single LoRA and oracle minus full fine-tuning ROUGE-L respectively. The *vs LoRA* and *vs FT* columns report GPT-4o pairwise preference proportions from 100 comparisons per domain.

netics achieving 53% and Clinical Medicine 58% win rates despite neutral or negative lexical overlap (-0.010 and $\Delta = -0.002$ respectively). This suggests domain-specialised adapters produce higher-quality medical responses that diverge lexically from reference answers, consistent with known limitations of lexical overlap metrics for paraphrase-rich text (Zhang et al., 2020). Domain specialisation in medical contexts may therefore be underestimated by ROUGE-L alone. GPT-4o preference over full fine-tuning remains positive for both domains, with Clinical Medicine at 61% and Medical Genetics at 64%.

Discovering balanced latent domain groupings appears to be a key factor in adapter performance. The three medical domains (Medical Genetics, Clinical Medicine, and Medical Information) occupy overlapping regions of the embedding space (Figure 5 in Appendix 6), proximate to Language & Writing, Content Generation, and General Knowledge clusters. This proximity suggests medical instruction following requires general language production skills distributed across multiple topical clusters rather than concentrated in domain-specific data. Representation sharing across these related clusters could benefit medical adapter performance, and motivates exploring functional domain discovery as a complement to topical clustering. That is, domain discovery may benefit from dimensions relating to the kind of task an instruction requires, not only its topic. Future work should compare latent topical and functional domain discovery for heterogeneous instruction tuning.

These results collectively demonstrate that evaluation of domain-specialised instruction tuning requires metrics sensitive to semantic equivalence, particularly in open-ended medical and general knowledge domains where lexical variation between valid responses is high. Notably, GPT-4o preference over full fine-tuning remains consistently positive across medical domains despite ROUGE-L losses, confirming that adapter quality in these domains is underestimated by lexical metrics alone.

5.3 Cross-Method Domain Convergence

Figure 2 explores whether domain specialisation benefit is an intrinsic property of instruction content or an artefact of the discovery mechanism. We analyse the overlap between K-means winning domains and their corresponding BERTopic super-24 groupings, with several consistent patterns emerging. Adapter-winning K-means clusters predominantly map to adapter-winning BERTopic super-groups: c10 maps 50.7% of examples to super_14 (BERTopic WIN) and c11 maps 76.0% to super_21

(BERTopic WIN). Conversely, adapter-losing clusters predominantly map to adapter-losing super-groups, with c7 mapping 52.2% to super_07 and c16 and c6 mapping predominantly to super_17 and super_08. This convergence across two structurally independent discovery methods suggests that specialisation benefit is largely determined by instruction content rather than the discovery mechanism itself.

To further validate this finding at a finer granularity, we map K-means cluster assignments to BERTopic-107 topic assignments for the same instructions. The K-means mathematical reasoning cluster maps predominantly to a winning BERTopic-107 topic (27.0% of examples), while natural language inference and general knowledge clusters map to winning BERTopic-107 topics in their top mappings. Conversely, K-means losing domains such as personal finance map consistently to losing BERTopic-107 topics. Alignment is weaker for domains such as JavaScript/jQuery, where content is distributed across multiple fine-grained topics with mixed outcomes. Together, comparing K-means with both BERTopic methods futher indicates that semantic coherence of domain content determines specialisation benefit rather than the discovery mechanism itself.

However, two notable exceptions exist: c9 and c12 are adapter-losing K-means clusters whose examples map predominantly to BERTopic adapter-winning super-groups (super_22 and super_06 respectively, with 49.2% and 79.2% overlap). We investigate whether running BERTopic adapters on these K-means test examples recovers performance. Table 4 shows that neither BERTopic adapter matches single LoRA on these examples. The c9 adapter improves marginally over the K-means adapter (+0.004) but remains below single LoRA, while c12 performs worse than both. GPT-4o evaluation provides further context: c9 and c12 achieve near-parity with single LoRA under preference-based evaluation (49% and 47% respectively), suggesting that ROUGE-L underestimates adapter quality for these domains. Notably, c9 achieves 67% GPT-4o preference over full fine-tuning, consistent with the broader pattern of adapter superiority over monolithic retraining. Common Sense Inference (c12) shows the weakest full fine-tuning preference at 51%, which may reflect its dependence on broad distributional knowledge that monolithic training captures more effectively than domain-specific adaptation. Whether this reflects an intrinsic property of common sense reasoning or a limitation of the current dataset composition remains an open question.

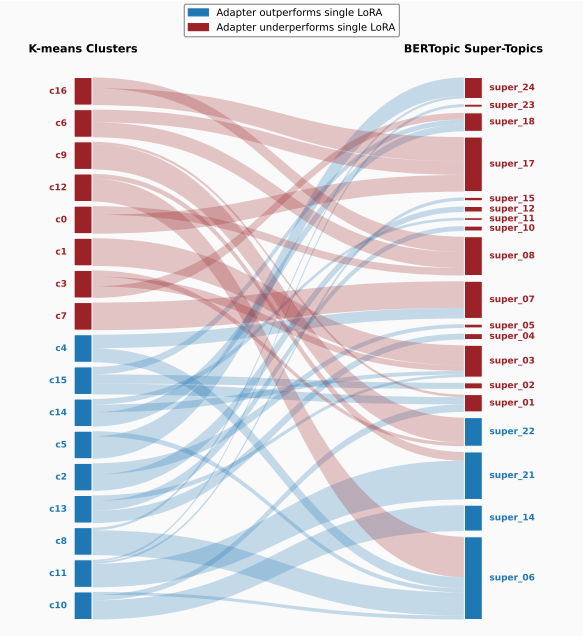

Figure 2: Cross-method domain alignment between K-means clusters (left) and BERTopic super-24 topics (right). Flow thickness indicates the proportion of examples shared between domains. In terms of ROUGE-L, adapter-winning domains are shown in blue and adapter-losing domains in red.

Cluster	Single LoRA	K-means	BERTopic
c9	0.4702	0.4615	0.4603
c12	0.5595	0.5517	0.5541

Table 4: Cross-method adapter evaluation: BERTopic super-24 adapters (super_22 and super_06 respectively) evaluated on K-means c9 and c12 test examples.

5.4 Centroid Routing Retention

Table 5 reports centroid routing accuracy across all domain discovery methods. K-means achieves the highest accuracy at 95.0%, with BERTopic-107 at 85.9% and BERTopic super-24 at 78.6%. MNLI achieves the lowest accuracy at 64.4%, reflecting a fundamental limitation of zero-shot classification as a discovery mechanism. Imposing predefined categories rather than discovering latent distributional structure produces less discriminative centroid representations and correspondingly reduced GPT-4o preference (41.0%).

Despite lower routing accuracy than K-means, BERTopic-107 achieves comparable GPT-4o preference (48.5% vs 48.8%), suggesting routing errors do not proportionally degrade response

Method	Correct	Accuracy
K-means (K=17)	**8073/8500**	**95.0%**
BERTopic (107 topics)	19962/23251	85.9%
BERTopic (super-24)	22771/28985	78.6%
MNLI (24 labels)	4929/7650	64.4%

Table 5: Centroid routing accuracy across domain discovery methods. Accuracy measures the proportion of test examples assigned to the correct domain by cosine similarity to domain centroids, relative to oracle domain labels.

quality. This is explained by the semantic structure of BERTopic-107, where intertopic distance analysis reveals substantial overlap between fine-grained topics, motivating the super-24 groupings described in Section 3.1.2. Routing errors predominantly represent assignments to semantically adjacent topics whose adapters are trained on overlapping content, such as misrouting between *50_jquery_element_div* and *30_div_css_height_width*, rather than cross-domain failures. Quality is preserved because the assigned adapter remains domain-appropriate at a functional level. Parameter-free centroid routing therefore removes the need for a trained routing classifier, requiring only recomputation of the mean embedding of a new domain's training instructions and making domain extension lightweight by design.

5.5 Modular Extensibility and Efficiency

Domain extensibility is a structural property of the SemiAdapt-Instruct architecture. Existing domains are updated by retraining only the relevant adapter on new data, while new domains are incorporated by training a single additional adapter, in both cases leaving all other components unchanged. To empirically validate this claim, we simulate a post-deployment domain update. That is, a 10,000-example subset of Magicoder (Wei et al., 2024) is assigned to K-means domains via existing centroid routing, with 4,199 examples mapped to the Algorithms & Coding cluster (c15). Only this adapter is retrained on the combined original and new data, while all 16 remaining adapters are unchanged. The updated adapter achieves ROUGE-L of 0.553 and BERTScore F1 of 0.934 on 200 held-out Magicoder examples, substantially outperforming all frozen baselines (Table 6). GPT-4o pairwise evaluation confirms strong preference for the updated adapter over the original frozen adapter (65.0%), single LoRA (66.5%), and full fine-tuning (81.5%). This demonstrates that targeted single-adapter updates incorporating new domain data outperform frozen monolithic baselines that cannot adapt without full retraining.

Method	ROUGE-L	BERTScore F1	Updated Preferred
Updated adapter (c15)	**0.553**	**0.934**	—
Original adapter (c15)	0.454	0.912	65.0%
Single LoRA	0.442	0.909	66.5%
Full Fine-Tuning	0.392	0.903	**81.5%**

Table 6: Extensibility experiment: updating only the Algorithms & Coding adapter (c15) with 4,199 new Magicoder examples while leaving all other adapters unchanged. *Updated Preferred* reports the proportion of 200 GPT-4o pairwise comparisons where the updated adapter is preferred over each frozen baseline.

Decomposing heterogeneous data into independent domains enables parallel adapter training, with each adapter training on its own data partition with no inter-adapter dependencies. In our experiments, three adapters were trained simultaneously on a single A100 80GB GPU, reducing wall-clock time by 1.7× relative to single LoRA (10h 8min vs 17h 0min) and 2.0× relative to full model fine-tuning (10h 8min vs 20h 22min). At inference time, parameter-free centroid routing adds negligible latency, with the base model remaining resident in GPU memory while only the active 52MB adapter is swapped per request. Full training and storage statistics are reported in Table 7.

Property	SemiAdapt-Instruct	Single LoRA
Base model params	7,248M	7,248M
Adapter params	13.6M (0.19%)	13.6M (0.19%)
Params per domain	13.6M	N/A
Total stored params	884MB (17 adapters)	52MB
Parallel training	3–4 domains	N/A
Initial training	10h 8min (3 processes)	17h 0min
Speedup vs single LoRA	1.7x	1x
Speedup vs full FT	2.0x	1.2x
New domain addition	1 adapter only	Full retrain
Domain update	1 adapter only	Full retrain

Table 7: Efficiency comparison between SemiAdapt-Instruct and single LoRA fine-tuning.

6 Conclusion

SemiAdapt-Instruct establishes modular domain specialisation as a practical alternative to monolithic instruction tuning. By automatically discovering latent domains, training per-domain LoRA adapters, and parameter-free routing at inference, the framework matches single LoRA fine-tuning and outperforms full model fine-tuning across all configurations. We further demonstrate the extensibility of the framework by retraining a single adapter on new domain data and showing it outperforms all monolithic baselines. Together, these findings demonstrate that modular domain specialisation is a practical and extensible alternative to full retraining in evolving deployment settings.

Limitations

SemiAdapt-Instruct is evaluated on a single heterogeneous English instruction dataset using Mistral-7B-Instruct-v0.3 as the base model. Whether the observed specialisation patterns generalise across languages, dataset compositions, or model families remains an open question, though cross-method convergence across three independent discovery mechanisms provides evidence that findings reflect instruction data structure rather than method or model artefacts.

ROUGE-L provides a universal comparison metric across heterogeneous domains but may underestimate gains for structured output domains where semantic correctness is not fully captured by lexical overlap. LLM-as-a-judge evaluation partially mitigates this limitation, though domain-specific evaluation metrics for automatically discovered domains remain an avenue for future work.

We acknowledge that comparison with learned routing approaches such as SLIM (Han et al., 2025) and GLIDER represents an important direction for future work. The contribution of parameter-free centroid routing is not that it outperforms learned routers, but that it eliminates a trained component that would otherwise require maintenance and retraining as new domains are added. A learned router is structurally coupled to the domain set it was trained on. Centroid routing extends to new domains by recomputing a single mean embedding, with no retraining of any component.

References

Dimo Angelov. 2020. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*.

Kevin Arvai. 2026. kneed.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Hongyi Cai, Jie Li, Mohammad Mahdinur Rahman, and Wenzhen Dong. 2025. Low-confidence gold: Refining low-confidence samples for efficient instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8233–8240, Suzhou, China. Association for Computational Linguistics.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Jiayi Han, Liang Du, Hongwei Du, Xiangguo Zhou, Yiwen Wu, Yuanfang Zhang, Weibo Zheng, and Donghong Han. 2025. SLIM: Let LLM learn more and forget less with soft LoRA and identity mixture. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4792–4804, Albuquerque, New Mexico. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yitong Huang, Ziqi Yang, Zihui Wang, Jianzhong Qi, Rongshan Yu, Xiaoliang Fan, and Cheng Wang. 2026. Hybrid routing for a mixture of lora experts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37):31211–31219.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4297–4308, Miami, Florida, USA. Association for Computational Linguistics.

Pingzhi Li, Prateek Yadav, Jaehong Yoon, Jie Peng, Yi-Lin Sung, Mohit Bansal, and Tianlong Chen. 2025. Glider: Global and local instruction-driven expert router. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6240–6301, Suzhou, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, and 1 others. 2025. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ACM Computing Surveys*, 58(3):1–39.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. 2025. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19165–19181, Vienna, Austria. Association for Computational Linguistics.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

Josh McGiff and Nikola S. Nikolov. 2025. Overcoming data scarcity in generative language modelling for low-resource languages: A systematic review. *Preprint*, arXiv:2505.04531.

Josh Mcgiff and Nikola S. Nikolov. 2026. Semiadapt: Semi-supervised and efficient lora-based domain adaptation for low-resource irish machine translation with transformers. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 10208–10220, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Hyeonseok Moon, Jaehyung Seo, Seonmin Koo, Jinsung Kim, Young-kyoung Ham, Jiwon Moon, and Heuiseok Lim. 2025. LimaCost: Data valuation for instruction tuning of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12841–12854, Suzhou, China. Association for Computational Linguistics.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2022. Webgpt: Browser-assisted question-answering with human feedback, 2022. *URL https://arxiv. org/abs/2112.09332*, 35.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Sergey Pletenev, Maria Marina, Daniil Moskovskiy, Vasily Konovalov, Pavel Braslavski, Alexander Panchenko, and Mikhail Salnikov. 2025. How much knowledge can you pack into a LoRA adapter without harming LLM? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4309–4322, Albuquerque, New Mexico. Association for Computational Linguistics.

Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. 2020. Short text topic modeling techniques, applications, and performance: a survey. *IEEE transactions on knowledge and data engineering*, 34(3):1427–1445.

Sebastian Robledo, Martha Zuluaga, and 1 others. 2022. Topic modeling: Perspectives from a literature review. *IEEE Access*, 11:4066–4078.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Yimin Tian, Bolin Zhang, Zhiying Tu, and Dianhui Chu. 2025. Adapters selector: Cross-domains and multi-tasks LoRA modules integration usage method. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 593–605, Abu Dhabi, UAE. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. In *Advances in Neural Information Processing Systems*, volume 36, pages 74764–74786. Curran Associates, Inc.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. Magicoder: Empowering code generation with oss-instruct. *Preprint*, arXiv:2312.02120.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yang Wu, Huayi Zhang, Yizheng Jiao, Lin Ma, Xiaozhong Liu, Jinhong Yu, Dongyu Zhang, Dezhi Yu, and Wei Xu. 2025. ROSE: A reward-oriented data selection framework for LLM task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13200–13219, Suzhou, China. Association for Computational Linguistics.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 6268–6278.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Simon Yu, Liangyu Chen, Sara Ahmadian, and Marzieh Fadaee. 2024. Diversify and conquer: Diversity-centric data selection with iterative refinement. *Preprint*, arXiv:2409.11378.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Data description

We construct a heterogeneous instruction-tuning dataset by combining multiple publicly available corpora spanning diverse domains and task formats. All datasets are used for research purposes consistent with their respective licenses. Our mixture includes domain-specific datasets such as Finance-Alpaca[2] (financial instructions) and AlpaCare-MedInstruct[3] (medical question answering), alongside general-purpose and reasoning-focused resources. In particular, we incorporate WebGPT (Nakano et al., 2022), a retrieval-based question answering dataset (no explicit license; publicly available for research use), and Chain-of-Thought[4] (CoT) datasets from FLAN to enhance multi-step reasoning capabilities. To capture real-world problem-solving and conversational patterns, we further include StackOverflow dialogues (Xu et al., 2023) (GPL-3.0).

To improve generalisation and coverage, we augment this mixture with additional general and code-oriented instruction datasets, namely Alpaca-cleaned[5], Databricks Dolly[6], and CodeAlpaca[7]. Overall, the resulting dataset balances domain-specific expertise (finance, medicine, programming) with general instruction-following and reasoning abilities, providing a heterogeneous dataset for research training. All trained artifacts derived from this data are intended for research use only, in accordance with the non-commercial restrictions of several constituent datasets (CC BY-NC 4.0).

StackOverflow Dialogues contributes 207,702 examples (40.3% of training data), substantially more than other sources. This imbalance influences K-means cluster boundaries: five of seventeen discovered clusters correspond to programming-related domains (c1, c3, c9, c13, c14), reflecting the dominance of StackOverflow content. While this may distort centroid positions toward programming embedding space, the high routing accuracy (95.0%) and strong per-domain adapter performance for non-programming domains suggest the remaining clusters are sufficiently well-separated for effective specialisation. Future work should examine whether more balanced dataset composition

[2] https://huggingface.co/datasets/gbharti/finance-alpaca (MIT License)

[3] https://huggingface.co/datasets/lavita/AlpaCare-MedInstruct-52k (MIT License)

[4] https://github.com/google-research/FLAN (Apache 2.0)

[5] https://huggingface.co/datasets/yahma/alpaca-cleaned (CC BY-NC 4.0)

[6] https://huggingface.co/datasets/databricks/databricks-dolly-15k (CC BY-SA 3.0)

[7] https://huggingface.co/datasets/sahil2801/CodeAlpaca-20k (CC BY 4.0)

affects domain discovery quality. Dataset composition and token statistics are summarised in Table 1 in the main paper.

B Soft Adapter Merging Ablation

Centroid routing analysis reveals that 21.4% of test examples exhibit routing ambiguity, defined as a cosine similarity gap below 0.05 between the nearest and second-nearest domain centroids, indicating that a considerable subset of instructions are semantically related to multiple domains. Winning K-means domains show considerably lower ambiguity rates than losing domains: c10 (Mathematical Reasoning) and c11 (Natural Language Inference) exhibit ambiguity rates of 4.3% and 2.8% respectively, while adapter-losing domains such as c14 (48.4%) and c4 (38.5%) show substantially higher rates. This inverse relationship between routing confidence and specialisation benefit implies that domain coherence, manifesting in well-separated centroid representations, is necessary for optimal adapter specialisation.

Given the routing ambiguity observed for a subset of examples, we investigate whether soft adapter merging can improve performance for ambiguous examples. We implement this by computing a weighted average of the top-2 adapter LoRA parameters directly, using normalised centroid similarity scores as weights. Contrary to expectation, soft merging consistently underperforms hard routing at both ambiguity thresholds tested: gap < 0.05 ($n = 107$, $\Delta = -0.006$) and gap < 0.15 ($n = 289$, $\Delta = -0.001$). At gap < 0.05, similarity scores are nearly equal, producing approximately 50/50 weight averaging, which dilutes domain-specific specialisation without compensating benefit from the secondary adapter. These results confirm that hard centroid routing is the preferred inference strategy, with its 95% oracle retention rate demonstrating near-optimal performance without merging overhead. The failure of uniform soft merging does not preclude more sophisticated combination strategies, though learned weighting would reintroduce a trained component, trading the extensibility advantage of parameter-free routing for potential quality gains on ambiguous examples.

C Extended Per-Domain Analysis

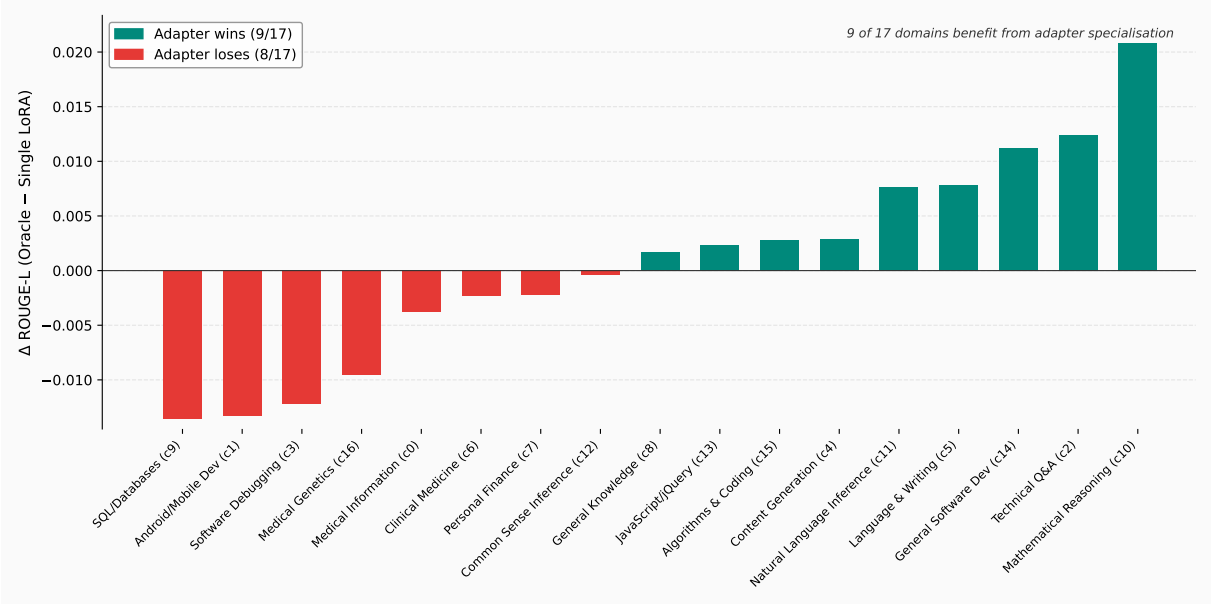

Figure 3: Visualisation of per-domain oracle ROUGE-L evaluation for K-means (K=17).

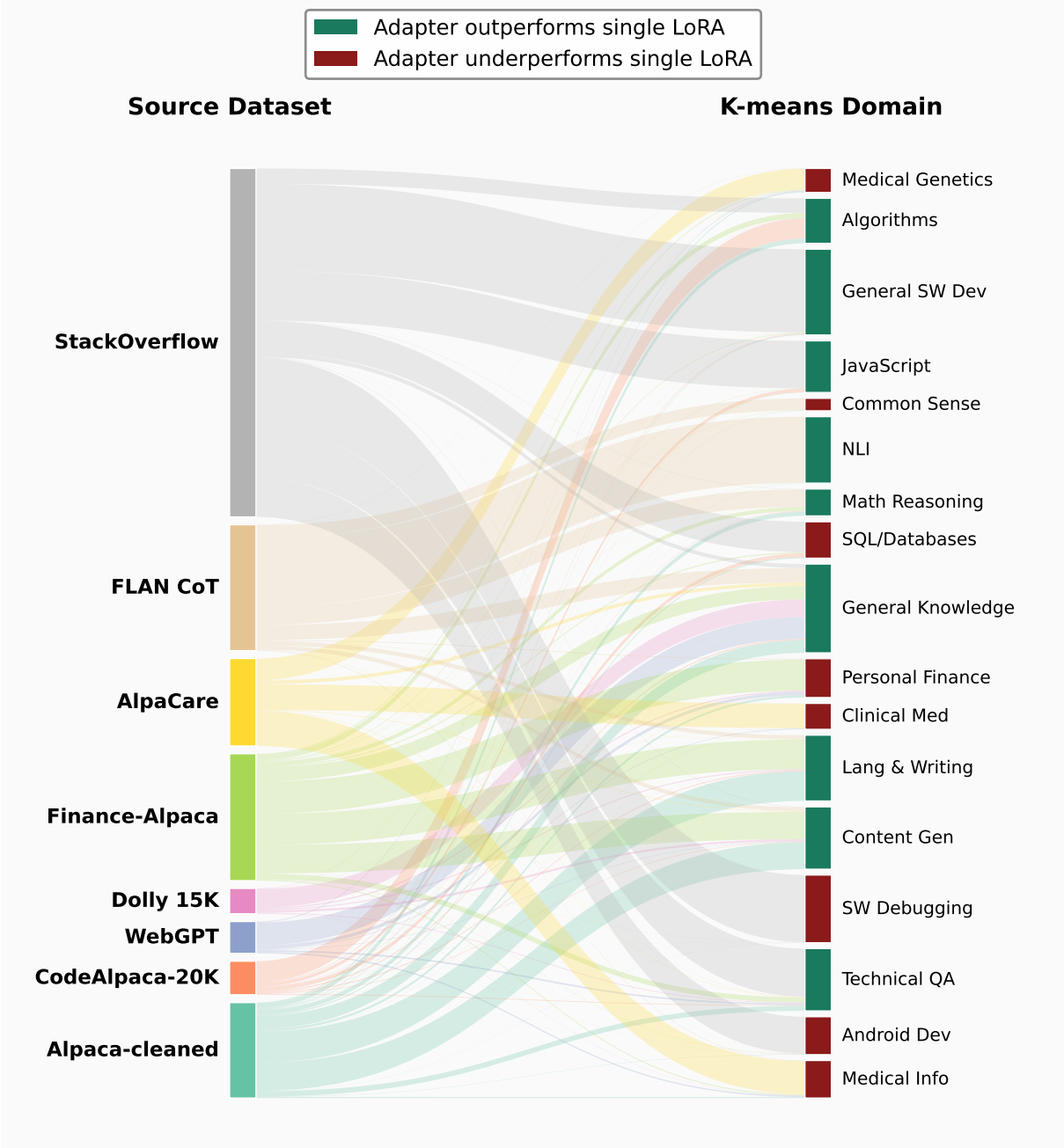

Figure 4: Source dataset composition across K-means (K=17) domain clusters. Flow width is proportional to the number of examples assigned to each cluster. Domain colours indicate adapter-winning (green) and adapter-losing (red) clusters under ROUGE-L evaluation.

ID	N	Name	ID	N	Name
−1	219,139	-1_human_ai_topic_use	53	1,686	53_cancer_chemotherapy_treatment_breast
0	16,064	0_sql_table_query_database	54	1,680	54_surgery_knee_pain_bone
1	15,078	1_title_context_com context_com	55	1,667	55_customer_email_customer service_service
2	14,880	2_step_possible tell_yes possible	56	1,610	56_aws_azure_s3_bucket
3	11,906	3_stock_money_tax_credit	57	1,598	57_xml_xml file_xslt_xml document
4	11,339	4_states_war_title_context	58	1,576	58_numbers_sum_10_print
5	10,808	5_sentence_following sentence_rewrite	59	1,523	59_learning_machine learning_neural
6	7,979	6_think_carefully make_step	60	1,496	60_certificate_ssl_encryption_key
7	7,289	7_claim_noptions yes_following	61	1,473	61_disorder_depression_anxiety_stress
8	7,163	8_covid_covid 19_19_outbreak	62	1,443	62_mei ok_mei_leo_ok let
9	6,356	9_view_android_layout_listview	63	1,356	63_triangle_area_cm_calculate area
10	6,313	10_premise_hypothesis_step	64	1,317	64_maps_google maps_google_map
11	5,990	11_animals_species_dogs_animal	65	1,317	65_spring_spring boot_boot_bean
12	5,568	12_string_regex_regular expression	66	1,243	66_react_component_state_props
13	5,293	13_genetic_gene_mutation_disease	67	1,198	67_healthy_exercise_diet_health
14	5,085	14_request_socket_authentication_http	68	1,187	68_django_model_topic django_field
15	4,962	15_reasoning_noptions_reasoning process	69	1,180	69_sort_sorting_array_order
16	4,737	16_sentence_sense noptions_make sense	70	1,113	70_thread_threads_mutex_threading
17	4,525	17_drug_medication_hypertension_patient	71	1,104	71_random_generate random_random number
18	4,306	18_date_datetime_time_format	72	1,069	72_opencv_image_cv_images
19	4,234	19_android_app_gradle_android studio	73	1,033	73_gridview_datagrid_row_wpf
20	3,650	20_class_pointer_type_function	74	1,020	74_song_album_band_released
21	3,453	21_angular_angularjs_ng_directive	75	1,011	75_qt_creator_topic qt_pyqt
22	3,247	22_directory_command_path_file	76	1,007	76_spark_hadoop_pyspark_dataframe
23	3,145	23_product_marketing_business_iphone	77	1,000	77_error_error message_message_type
24	2,767	24_plot_chart_matplotlib_axis	78	995	78_internet_reddit_title_com
25	2,743	25_patient_year old_symptoms_old	79	978	79_facebook_api_twitter_app
26	2,725	26_test natural_language inference	80	951	80_computer_software_cpu_processor
27	2,698	27_maven_jar_eclipse_java	81	945	81_headaches_patient_headache_symptoms
28	2,689	28_json_object_array_json object	82	902	82_think student_teacher let_premise
29	2,685	29_team_league_football_sports	83	889	83_heart_blood_circulatory_human heart
30	2,640	30_div_css_height_width	84	886	84_email_mail_outlook_emails
31	2,432	31_laravel_topic laravel_codeigniter	85	873	85_stream consciousness_random thoughts
32	2,420	32_artificial intelligence_robots	86	873	86_maximum_array_maximum value
33	2,413	33_visual studio_visual_studio_dll	87	805	87_statement_statements_loop_boolean
34	2,273	34_poem_sky_words_sun	88	794	88_technology_article_impact_essay
35	2,245	35_climate_climate change_global warming	89	775	89_pdf_pdf file_document_pdf files
36	2,190	36_git_branch_repository_commit	90	761	90_rails_gem_ruby_rake
37	2,187	37_story_short story_generate_character	91	748	91_website_search_web_search engine
38	2,043	38_diabetes_type diabetes_insulin	92	735	92_firebase_analytics_authentication
39	2,033	39_video_audio_play_youtube	93	710	93_decimal_round_rounding_number
40	1,995	40_heart_chest_chest pain_myocardial	94	709	94_array_arr_given array_list
41	1,972	41_dataframe_pandas_column_df	95	665	95_average_deviation_standard deviation
42	1,955	42_lung_chest_patient_cough	96	645	96_vue_vue js_component_js
43	1,924	43_equation_2x_3x_calculate	97	616	97_alzheimer_alzheimer disease_amyloid
44	1,880	44_le ok_denny asked_ok answer	98	613	98_salary_employees_sql query
45	1,798	45_mei ok_mei_leo_ok let	99	602	99_class_create class_student_grade
46	1,785	46_medical_medical school_education	100	577	100_tensorflow_tensor_model_keras
47	1,774	47_nonsensical_following sentences	101	567	101_webpack_babel_module_npm
48	1,770	48_le ok_denny asked_ok answer	102	564	102_ethical_medical_ethics_patient
49	1,761	49_series_film_consciousness	103	551	103_utf_file_encoding_text file
50	1,750	50_jquery_element_div_javascript	104	534	104_dog_premise_hypothesis entailed
51	1,721	51_docker_jenkins_container_kubernetes	105	513	105_prime_prime numbers_prime number
52	1,714	52_mongodb_elasticsearch_mongoose			

Table 8: BERTopic 107-topic domain labels and training set counts.

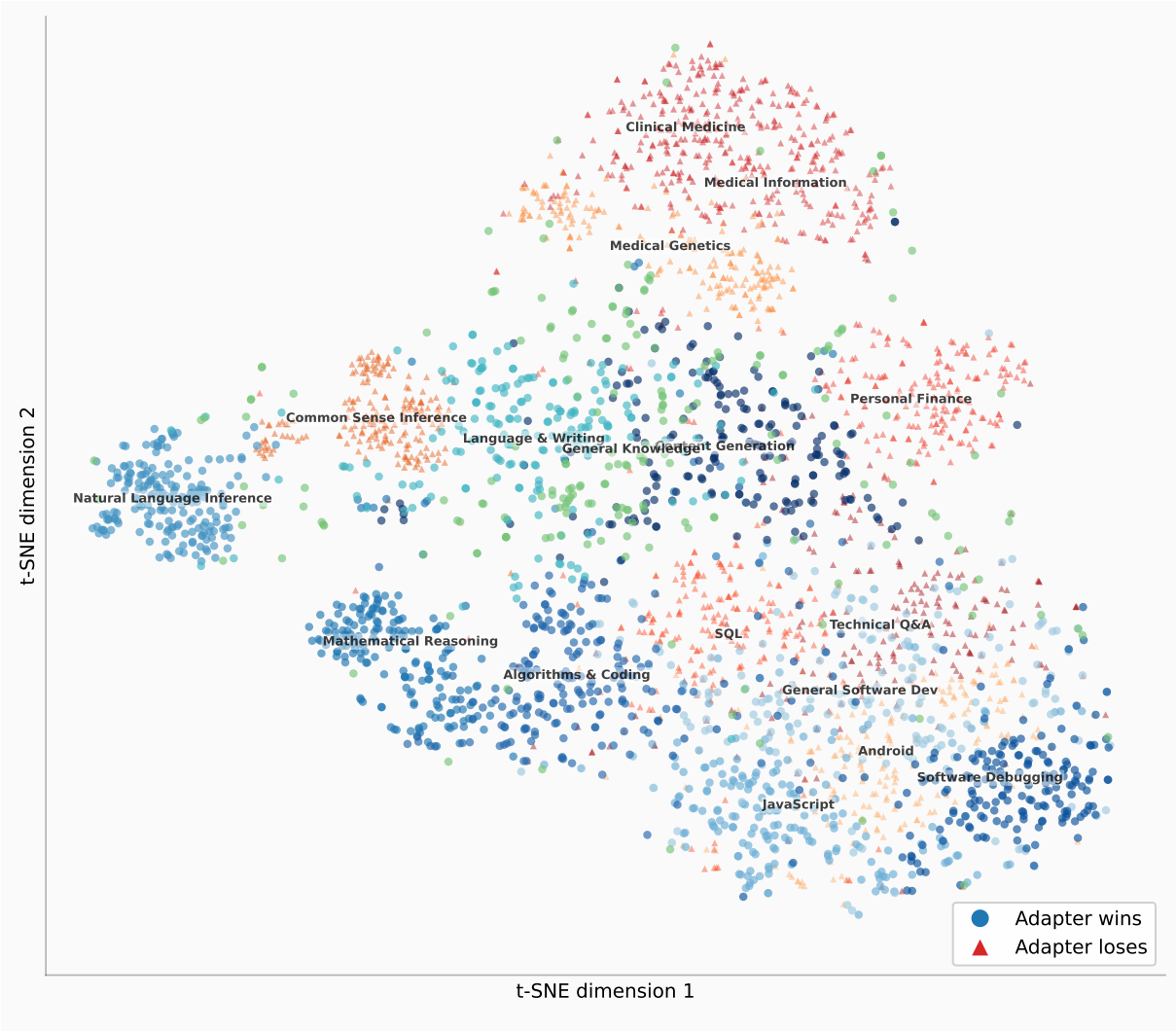

Figure 5: t-SNE projection of K-means (K=17) domain cluster assignments across the training set. Circles indicate adapter-winning clusters and triangles indicate adapter-losing clusters under ROUGE-L evaluation.

Super-24 Topic	BERTopic-107 Topic IDs
super_01	71, 105, 58, 95, 43, 63, 18, 93
super_02	94, 86, 69
super_03	75, 73, 9, 19, 39, 64, 92, 52, 28
super_04	50, 30
super_05	59, 32
super_06	47, 16, 11, 1, 15, 29, 4
super_07	80, 91, 3, 55, 78, 23, 88, 35
super_08	67, 38, 25, 13, 81, 61, 97
super_09	102, 46
super_10	101, 66, 21, 96
super_11	36, 51, 56, 76
super_12	87, 99, 20, 70
super_13	41, 24
super_14	6, 85
super_15	62, 45
super_16	60, 79, 14, 84
super_17	53, 17, 54, 83, 40, 42, 8
super_18	33, 77, 22, 90, 31, 12, 103, 89, 57
super_19	65, 27
super_20	100, 68, 72
super_21	104, 26, 10, 82, 2, 7
super_22	0, 98
super_23	44, 48
super_24	49, 74, 37, 34, 5

Table 9: BERTopic super-24 groupings showing constituent BERTopic-107 topic IDs. Super-groups were derived by merging semantically related topics via intertopic distance mapping.