GRAFT: GROWING MISSING KNOWLEDGE THROUGH TOPOLOGY-GUIDED DATA SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual pretraining requires selecting data that expands a language model's knowledge without eroding what it has already learned. We introduce a method for constructing latent knowledge as layer-wise directed graphs of concept connections. Building on this construction, we develop GRAFT, a data-selection framework that prioritizes documents with weak concept connectivity in middle layers. Training on the selected data strengthens links among previously isolated concepts and improves downstream performance. Across ten knowledge and reasoning benchmarks and three base models, GRAFT achieves the best overall average rank among the evaluated data-selection methods. In our experiments, GRAFT also achieves higher scoring throughput on long documents than the evaluated model-adaptive baselines. When selecting from a mixture of mathematics and STEM-specialized data, GRAFT improves both general knowledge and mathematical reasoning. Finally, we examine how latent knowledge structures in the middle layers of LLMs relate to reasoning and forgetting: improved reasoning coincides with new connected concepts, whereas forgetting is associated with the loss of established ones.

1 INTRODUCTION

Continual PreTraining(CPT) extends a pretrained language model through additional training on unlabeled text, enabling adaptation to new domains and tasks (Gururangan et al., 2020). For large language models (LLMs) already trained on vast corpora, a central question is what additional data can most effectively complement their existing knowledge. When the original pretraining corpus is unavailable, it is difficult to determine whether a candidate sample provides new learning opportunities or largely repeats known information. Moreover, adapting to a new data distribution can lead to catastrophic forgetting of previously acquired capabilities, making the acquisition of new knowledge and the retention of existing capabilities a joint challenge. (Ibrahim et al., 2024; Jiang et al., 2025). Target-oriented selection can identify data relevant to specified tasks or domains (Wang et al., 2026), but selecting complementary data for broader capability improvement requires criteria that extend beyond a predefined set of targets. These challenges motivate data selection that accounts for both text quality and the knowledge already represented in the model.

Specifically, existing methods mainly seek to improve general model capabilities from two perspectives, as illustrated in Figure 1(a): intrinsic text quality and model adaptivity. Text-quality methods typically rely on human judgments or proxy models to evaluate intrinsic properties of the text, such as its quality and educational value (Wettig et al., 2024). However, because language models are trained on data from diverse sources and time periods, these scores may not generalize across various models, particularly when the target model has already acquired substantial knowledge. Model-adaptive methods instead account for what the model has learned by using its predictive behavior to score candidate data, for example through perplexity (Marion et al., 2023) or the change in prediction loss after masking retrieval heads (Hua et al., 2025). Yet these approaches generally require token-level evaluation and logit computation, which can be expensive for large datasets or long-context documents. This motivates a more efficient adaptive signal for estimating the value of candidate data to a target pretrained model. If a candidate dataset contains knowledge that complements the model's existing knowledge, the model can focus its learning on information it has not yet learned. We therefore introduce **GRAFT**, a CPT data selection method that focuses on the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

model's latent knowledge structure, rather than token-level predictive signals, to identify data that may complement its existing knowledge.

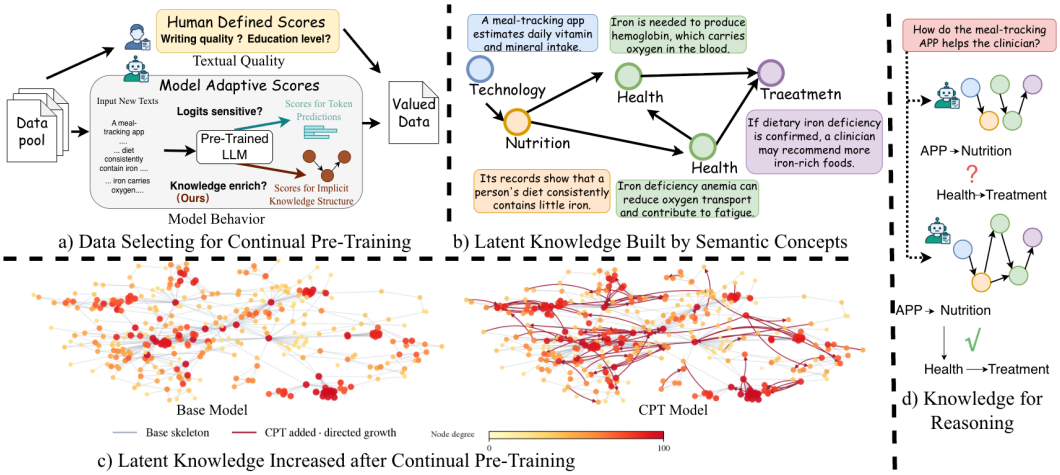

Figure 1: Overview of GRAFT, a knowledge-based data-selection method. **(a)** Existing data-selection methods based on text quality or model behavior, while GRAFT scores the connectivity of internal concepts. **(b)** Conceptual illustration of latent knowledge represented as connections among semantic concepts. **(c)** Visualization of knowledge connections in a middle layer with stronger concept connectivity. **(d)** Increased knowledge connectivity helps LLMs answer downstream tasks more accurately.

Following the illustrations of building knowledge with concepts Quillian (1967); Speer et al. (2017), we model knowledge as a directed graph of semantic concepts and their relationships. Building on this view, we formalize the latent knowledge structure expressed by sentence- and phrase-level meanings in the corpus, as illustrated in Figure 1(b). Given a passage, we identify concepts such as Technology, Nutrition, and Health and infer connections among them from the relationships expressed in the text. For example, a meal-tracking app records iron intake, linking the Technology and Nutrition concepts. As LLMs encounter longer and more diverse texts during training, their knowledge can expand. This knowledge is encoded in the model's internal layers, where CPT can establish additional connections among concepts, as illustrated in Figure 1(c).[1] Finally, the model can use the reshaped knowledge structure to answer downstream questions more accurately because the relevant concepts are connected, as illustrated in Figure 1(d).

We can naturally formalize these evolving concept connections into signal for data selection at CPT stage. We first formalize latent knowledge as layer-wise graphs in Section 2. We introduce the Branch Fraction (BF) score to assess the model's knowledge familiarity with candidate data and present GRAFT, which uses this score to guide continual pretraining. We evaluate GRAFT's performance across models and domains, as well as its scoring efficiency on long documents, in Section 3. Finally, in Section 4, we examine how changes in concept connectivity relate to reasoning performance, the role of middle layers, and catastrophic forgetting. In conclusion, our key contributions are threefold:

1. We represent latent knowledge in LLMs through semantic concepts and their connections. Building on this representation, we propose GRAFT to select data that fills knowledge gaps during continual pretraining.

2. We show that GRAFT efficiently selects data across different model architectures. Compared with other model-adaptive methods, it scores important documents more efficiently and adapts to mixed-domain data.

[1]The model was trained on 3B of Nemotron specialized data using Qwen2.5-3B. The visualization shows increased knowledge in the law domain at layer 21.

3. We analyze model behaviors through the lens of latent knowledge. We examine how changes in concept connectivity relate to reasoning performance and discuss their implications for internal reasoning mechanisms and catastrophic forgetting.

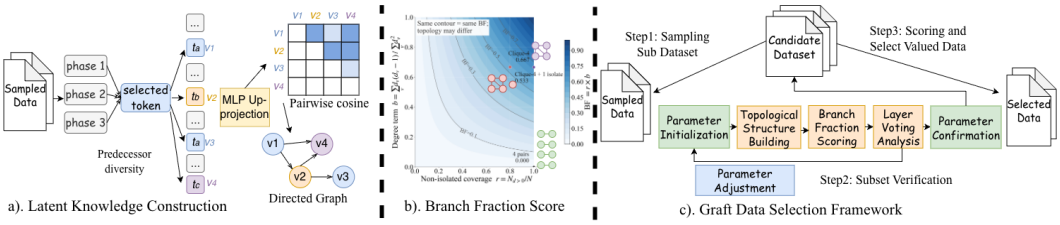

Figure 2: The framework details for GRAFT framework for data selection in CPT. **(a)** Latent knowledge construction for data samples. **(b)** Branch Fraction for evaluating each sample knowledge reachable. **(c)** The whole framework for GRAFT data selection.

2 Latent Knowledge for Data Selection

We introduce a latent-knowledge-based data selection framework, illustrated in Figure 2. Given a data sample, we first construct its latent knowledge structure, as shown in Figure 2(a). Specifically, we use predecessor diversity to identify semantic concept anchors and the MLP up-projection representations of these anchors to characterize connections among concepts. The construction procedure is detailed in Section 2.1. We then define the Branch Fraction score, illustrated in Figure 2(b), to measure the extent to which the LLM has established a connected knowledge structure for the sample, as described in Section 2.2. Finally, Figure 2(c) presents the complete GRAFT framework for data selection, which is described in Section 2.3.

2.1 Latent Knowledge Construction

Following the previous work of building knowledge as graph structures (Quillian, 1967; Speer et al., 2017), we represent latent knowledge at layer l as a directed acyclic graph, $G^{(l)} = (V^{(l)}, E^{(l)})$. Each node $v_i^{(l)} \in V^{(l)}$ denotes a concept embedding extracted from the layer-l MLP up-projection activation at token position i. Each directed edge $(v_i^{(l)}, v_j^{(l)}) \in E^{(l)}$ denotes a connection from an earlier concept to a later one within the same layer. We describe the construction of the nodes and edges below.

Semantic concepts as nodes. LLMs encode concept-level patterns in their token representations (Park et al., 2025; Shani et al., 2026). Beyond individual tokens, higher-level semantic concepts involved in reasoning can be aggregated in MLP up-projection representations at boundary tokens, such as periods and commas (Lindsey et al., 2025; Dai et al., 2025). These boundary tokens are often preceded by diverse sequences of tokens and can therefore summarize information from a wide range of semantic contexts. We quantify this property using predecessor diversity. For a token t, let $\mathcal{P}(t)$ denote the set of tokens that occur immediately before t, and let $p(u \mid t)$ denote the empirical probability that token u precedes t. The predecessor diversity score is defined as

$$D_{\text{prev}}(t) = \exp\left(- \sum_{u \in \mathcal{P}(t)} p(u \mid t) \log p(u \mid t) \right). \tag{1}$$

Intuitively, this score measures the effective number of distinct predecessor tokens while accounting for how evenly they are distributed. A high score indicates that a token occurs after a broad range of preceding contexts, making it a plausible point at which information from those contexts is integrated. We therefore use tokens with high predecessor diversity as candidate semantic anchors.

Following prior work showing that MLP up-projection activations encode semantic concepts (Huben et al., 2024; Sadhukhan et al., 2026), we define the node associated with selected token i at layer l as $\mathbf{v}_i^{(l)} \in \mathbb{R}^d$, where d is the dimension of the up-projection space. Each node captures aggregated semantic cues, while the connections among nodes define the latent knowledge structure. The details of semantic concepts are in Appendix A.1

Concept connections as edges. We build the knowledge by the connection of the concepts from the given context Speer et al. (2017); Hogan et al. (2021), and therefore construct edges between concept representations within each layer. For every pair of selected tokens i and j, we compute the cosine similarity between their MLP up-projection activations, $\mathbf{v}_i^{(l)}$ and $\mathbf{v}_j^{(l)}$. A directed edge is added from node $v_i^{(l)}$ to node $v_j^{(l)}$ when $i < j$ and their similarity exceeds a threshold τ:

$$s_{ij}^{(l)} = \frac{\mathbf{v}_i^{(l)\top}\mathbf{v}_j^{(l)}}{\|\mathbf{v}_i^{(l)}\|_2\|\mathbf{v}_j^{(l)}\|_2}, \qquad (v_i^{(l)}, v_j^{(l)}) \in E^{(l)} \iff i < j \ \wedge \ s_{ij}^{(l)} > \tau. \tag{2}$$

The cosine similarity measures the alignment between two concept representations, while τ determines how much alignment is required to regard them as connected. The ordering constraint $i < j$ directs each edge from an earlier token to a later one, reflecting the model's left-to-right processing and ensuring that the resulting graph is acyclic. Thus, the graph captures layer-wise connections among sufficiently similar concepts as they may share a concept shifting to another semantic concept and utilize the knowledge to the reasoning processing.

2.2 Branch Fraction Score

To quantify how well an LLM has acquired the knowledge expressed in a sample, we introduce the Branch Fraction (BF) score. BF captures two complementary properties of the latent knowledge graph: the proportion of concepts that are connected to at least one other concept and the structural organization of their connections. We expect a well-formed knowledge structure to connect as many concepts as possible while maintaining an efficient and sufficiently compact connectivity pattern. In contrast, fragmented subgraphs or isolated concepts indicate incomplete knowledge integration and lead to lower BF scores. For example, the four-pair structure (green) in Figure 2(b) receives a low score because the concepts are divided into multiple disconnected components. Similarly, a graph containing an isolated concept (red) receives a lower score even when the remaining concepts form a clique comparing with the connected one (purple). Further analysis of BF under different graph structures is provided in Appendix Section A.2. Accordingly, we define BF using the fraction of non-isolated nodes and the degree structure of the graph, as follows:

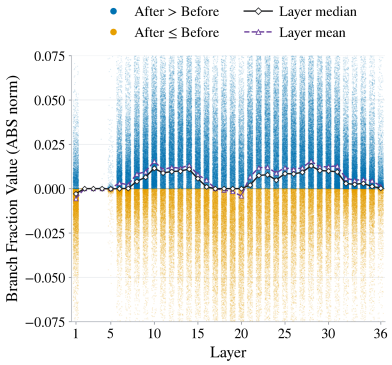

Figure 3: Layer-wise BF before and after continual pretraining.

$$\mathrm{BF}^{(l)} = \mathrm{BF}\left(G^{(l)}\right) = \underbrace{\frac{N_{d>0}^{(l)}}{N^{(l)}}}_{\text{non-isolated nodes}} \cdot \underbrace{\frac{\sum_{v\in V^{(l)}} d_v^{(l)}\left(d_v^{(l)}-1\right)}{\sum_{v\in V^{(l)}}\left(d_v^{(l)}\right)^2}}_{\text{degree structure}}, \tag{3}$$

where $N^{(l)} = |V^{(l)}|$ and $d_v^{(l)} = d_{v,\text{in}}^{(l)} + d_{v,\text{out}}^{(l)}$. $N_{d>0}^{(l)}$ counts nodes incident to at least one edge. The BF scores evaluate LLMs for knowledge at each layer l.

To assess whether BF reflects knowledge familiarity, we continually pretrain Qwen2.5-3B on a randomly sampled 5B-token subset of FineWeb CC-MAIN-2024-10 Penedo et al. (2024), whose data were collected after the model's pretraining cutoff. We then evaluate BF on the same dataset before and after continual pretraining. As shown in Figure 3, the median BF increases in most layers after training, supporting the use of higher BF as an indicator of greater familiarity with the input data. The concept connections may happen in any layers of LLMs, therefore, in data selection step, we use the voting method to evaluate how LLMs adapt to the knowledge. To help LLMs to learn new knowledge smoothly, we first start with a small number of voting layers (eg., 10% of the whole layers), then extend to 90% of layers.

2.3 GRAFT for Data Selection

As BF score evaluating how concepts construct knowledge inside LLMs, we use the BF score as a cue to select the data that has the most potential to enrich LLMs to learn new knowledege. After

ensure the parameters that suitable for for a sampled dataset, we use the series of parameters in the whole datasets, and choose the most suitable data for LLMs to continue training. We first select a small set from candidate dataset (eg., 10k tokens from 40B) and then test the parameters, which include the top token numbers and tau selecting. With the selected parameters, we will have the branch fraction evaluations for the subset and then test the layer voting split the dataset. We provide the details in Appendix Section C.

3 Continue Pre-training for Knowledge Increasing

To study the effectiveness of GRAFT for data selection in continual pretraining, we evaluate it across multiple language models and benchmarks. We first describe the experimental setup in Section 3.1. We then compare GRAFT with several existing data-selection methods in Section 3.2 and evaluate its ability to select data from multiple domains in Section 3.3. Finally, we analyze the impact of the main selection parameters in Section 3.4.

3.1 Experiment Setting

We evaluate GRAFT for knowledge-based continual pretraining. Specifically, we construct the candidate training corpus from NVIDIA's Nemotron-Pretraining-Specialized dataset NVIDIA (2025) and Nemotron-CC-Math-v1 Mahabadi et al. (2025). We use SmolLM2-1.7B Allal et al. (2025), Qwen2.5-3B Team (2024), and Qwen3-8B Team (2025) as the base pretrained language models. For evaluation, we consider ten knowledge and reasoning benchmarks: Massive Multitask Language Understanding (MMLU) Hendrycks et al. (2020), AI2 Reasoning Challenge Easy (ARC-Easy) and Challenge (ARC-Challenge) Clark et al. (2018), CommonsenseQA (CSQA) Talmor et al. (2019), Question Answering via Sentence Composition (QASC) Khot et al. (2020), MMLU-Pro Wang et al. (2024), PIQA Bisk et al. (2020), OpenBookQA Mihaylov et al. (2018), AGIEval Zhong et al. (2024), and HellaSwag Zellers et al. (2019). In mix setting, we also include 7 math tasks as illustrated in Appendix Section B.4.2 We utilize 0-COT to evaluate the models, which ask the answers exactly match the ground truth. We compare GRAFT with several baseline methods, including random selection, proxy-based methods such as DCLM Li et al. (2024) and evaluation based QuRating Wettig et al. (2024), and model-adaptive methods based on perplexity (PPL) Marion et al. (2023) and AttentionInfluence (Attn-Inf) Hua et al. (2025). All of the experiments are trained with 16 H100 GPUs with deepspeed framework. Detailed descriptions of all comparison methods are also provided in the Appendix Section B.

To select a fixed amount of training data from the candidate dataset, we first construct a subsampled subset for continual pretraining. Specifically, we randomly sample 10,000 examples to characterize the score distributions produced by different selection methods. Based on these distributions, we determine a selection threshold and then apply it to the full candidate dataset. Examples whose scores satisfy the threshold are selected for continual pretraining.

3.2 QUALITY DATA FOR CONTINUAL TRAINING

We first evaluate the data-selection methods by measuring their impact on knowledge acquisition during continual pretraining. We inject 3 billion tokens into the SmolLM2-1.7B and Qwen2.5-3B models. For GRAFT, we select examples with a Branch Fraction score below 0.1 and require the score to satisfy the criterion in one to four layers. For the other methods, we select the top 10% of examples according to their respective scores, while keeping the amount of training data fixed across all methods. The results are shown in Table 1. The results show that GRAFT achieves the best overall performance among the evaluated methods, obtaining a macro-average accuracy of 69.65%. This result suggests that topology-guided data selection can provide more effective knowledge acquisition and improve downstream reasoning performance during continual pretraining.

We extend our evaluation to the larger Qwen3-8B model. The increased model scale and the more recently released checkpoint introduce two challenges: efficiently scoring candidate documents and selecting data that can further improve the capabilities of a strong pretrained model. As shown in Figure 4, GRAFT performs feed-forward computations only for selected semantic-anchor tokens. Consequently, its computational efficiency is less sensitive to document length. In contrast, AttentionInfluence requires perturbing attention components and comparing the resulting logits, which

Table 1: Downstream evaluation results for SmolLM2-1.7B and Qwen2.5-3B with 3B-token data injection. Macro denotes the unweighted mean across the ten tasks, and Avg. Rank is computed separately within each model block. Higher task scores and lower average ranks are better. Best and second-best results within each model block are shown in **bold** and underlined, respectively.

Method	MMLU	ARC-E	ARC-C	CSQA	QASC	MMLU-Pro	PIQA	OpenBook	AGIEval	Hella	Macro	Avg. Rank
SmolLM2-1.7B												
Base	20.77	32.37	26.37	17.36	14.47	4.89	26.22	26.40	15.51	8.28	19.26	6.50
Random	35.09	49.41	40.70	24.57	21.27	21.65	45.05	49.20	40.03	35.09	36.21	4.90
DCLM	16.34	19.40	19.54	12.78	9.72	17.92	33.41	36.20	31.65	29.64	22.66	6.50
QuRating	41.95	63.85	52.30	40.13	31.86	24.05	59.41	55.40	43.20	34.60	44.68	3.00
PPL	**44.66**	61.45	53.16	39.64	39.42	**24.14**	50.98	55.00	42.41	35.62	44.65	2.70
Attn-Inf	44.15	67.68	55.97	37.35	35.75	23.30	57.94	**57.20**	**45.89**	**39.98**	46.52	**2.20**
GRAFT	43.74	**70.24**	**57.59**	**49.47**	**40.50**	22.74	**63.82**	53.40	41.46	36.63	**47.96**	**2.20**
Qwen2.5-3B												
Base	43.97	69.11	57.94	48.40	45.25	32.21	67.36	60.80	57.12	57.89	54.01	5.60
Random	55.09	80.60	69.54	65.85	59.50	41.57	76.17	72.60	44.15	53.96	61.90	4.30
DCLM	43.83	64.81	59.22	61.51	52.48	38.71	71.27	59.60	39.40	52.44	54.33	6.50
QuRating	62.09	**92.34**	**81.74**	71.17	**72.25**	42.85	78.35	**81.40**	47.47	57.70	68.73	2.10
PPL	53.71	82.62	74.83	68.55	67.39	42.28	78.45	77.20	41.30	57.55	64.39	3.55
Attn-Inf	51.55	68.77	64.16	65.03	55.62	43.06	78.45	66.40	42.72	56.28	59.21	4.55
GRAFT	**62.68**	91.67	80.72	**71.99**	70.63	**43.65**	**79.54**	78.40	**57.91**	**59.28**	**69.65**	**1.40**

substantially reduces its throughput on long documents to approximately half that of GRAFT. We then select 1 billion tokens for continual pretraining on Qwen3-8B; the results are presented in Figure 5. GRAFT achieves the best overall performance among the evaluated methods, demonstrating that it can identify valuable and complementary data even for a well-trained language model. For example, on AGIEval, GRAFT improves the model's accuracy by approximately 12% relative to the base model, whereas the other evaluated selection methods yield improvements of approximately 7%. These results suggest that GRAFT provides a more effective signal for identifying data that can extend the capabilities of large language models. The complete results are provided in the Appendix Section D.1.

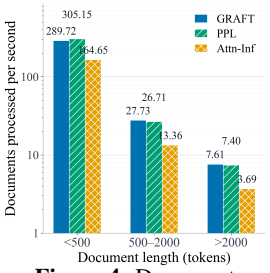

Figure 4: Document scoring evaluation.

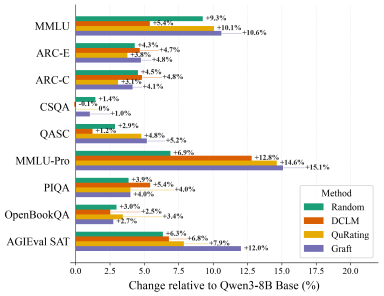

Figure 5: Qwen3-8B continual training via various methods.

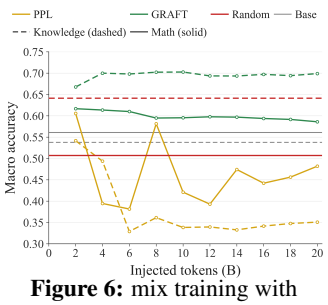

Figure 6: mix training with Specialized and Math datasets

3.3 Cross Domain Knowledge Learning

A key challenge in data selection is that a candidate pool may contain data from multiple domains. We therefore analyze GRAFT's domain-adaptive ability in mixed-domain data selection, focusing on a mixture of STEM-specialized and mathematical data. We report the macro-average performance across ten general knowledge benchmarks and seven mathematical reasoning benchmarks; the complete results are provided in the Appendix Section D.2.

We select 20 billion tokens from the mixed dataset using either perplexity (PPL) or GRAFT and use the selected data for continual pretraining on Qwen2.5-3B. For both methods, we first randomly sample 10,000 documents to estimate the score distributions and determine the selection thresholds. However, PPL is not directly comparable across domains: MATH documents receive substantially lower scores than STEM-specialized documents, with the 10th-percentile scores being 1.73 and 4.17, respectively. As a result, PPL selects approximately 95% mathematical data, producing a highly imbalanced training mixture.

In contrast, GRAFT evaluates the knowledge connectivity with each document. Its selection criterion therefore produces a more balanced mixture, consisting of approximately 55% mathematical

data and 45% STEM-specialized data, as shown in Figure 6. The results further demonstrate the effect of this difference in data composition. When randomly selected data are used for continual pretraining, increasing mathematical knowledge is accompanied by a decline in general knowledge performance, suggesting that the model already possesses relatively strong mathematical capabilities but still has room to acquire additional general knowledge. The imbalanced mixture selected by PPL degrades both general knowledge and mathematical reasoning performance. In contrast, GRAFT selects data that are less familiar to the model across both domains, leading to improved performance on both general knowledge and mathematical reasoning tasks.

3.4 Parameters Studies

The parameters govern two key components of the GRAFT framework: latent knowledge construction and data selection. For latent knowledge construction, we focus on whether the parameter settings produce sufficiently discriminative scores across different data samples. Detailed analysis is provided in Appendix Section C.

For data selection, we analyze the effects of the selection threshold and the number of voting layers, as shown in Figure 7. For each parameter configuration, we select up to 0.1 billion tokens from each data block and run continual pretraining with three random seeds. We report the average performance across the resulting continual-pretraining models. Blocks containing between 0.08 and 0.1 billion tokens are marked as short pools and are evaluated once using all available data. Blocks containing fewer than 0.08 billion tokens are left blank.

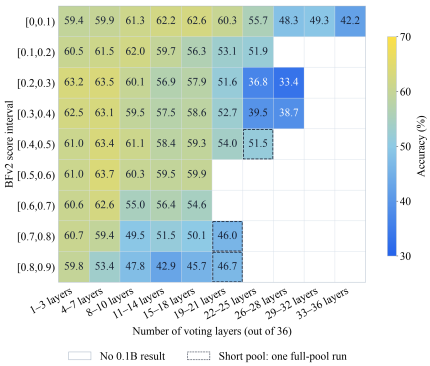

Figure 7: Accuracy across BF thresholds and voting-layer counts.

The results suggest that better performance is generally achieved when using a lower Branch Fraction threshold and fewer voting layers. These findings indicate that effective adaptive data selection should consider two complementary factors. A lower Branch Fraction score suggests that the model has formed fewer connections among the latent concepts in a document, indicating greater potential for knowledge acquisition. Meanwhile, a smaller number of voting layers suggests that the unfamiliar knowledge is concentrated in fewer layers and may therefore be easier to incorporate. Combining these two criteria enables GRAFT to identify data that are particularly suitable for CPT.

4 Discussion

LLMs organize knowledge across their internal layers, with a higher Branch Fraction indicating stronger connectivity among concepts. This observation motivates three questions: (1) How does knowledge propagate across layers and ultimately improve model performance? (2) Which layers play the most important role in connecting knowledge and enhancing the model's capacity for knowledge acquisition? (3) How does the latent knowledge structure change when the model experiences catastrophic forgetting?

Knowledge for Reasoning Increasing We first investigate why continual pretraining improves knowledge-based reasoning. We compare the responses generated by the continually pretrained 3B model and the Qwen2.5-3B base model, focusing on changes in the number of edges in both the question-to-reasoning (Q→R) and reasoning-to-reasoning (R→R) transitions, as shown in Figure 8. The results show that continual pretraining introduces new connections in both transition types, with the number of Q→R edges increasing by more than 25%. This finding suggests that acquiring additional knowledge reshapes the model's latent knowledge topology. Moreover, the newly formed edges connect previously isolated semantic concepts, leading to a more balanced and integrated structure.

Additionally, we visualize knowledge growth in the physics domain using corrected responses from the continually pretrained model, as shown in Figure 9. We use physics questions from the MMLU

and MMLU-Pro datasets as examples to illustrate the model's latent knowledge structure. For each selected token, we identify the top 30 predecessor tokens according to predecessor diversity and collect the preceding 10 tokens associated with each selected token. We merge instances whose preceding-token sets completely overlap and treat the resulting groups as semantic concepts. Edges are constructed using pairwise cosine similarity within each sample, with $\tau = 0.7$. Before CPT, several concepts remain isolated; after training, these concepts become connected to familiar knowledge. For example, the concepts "half Mars" and "Earth's radius" become linked after CPT.

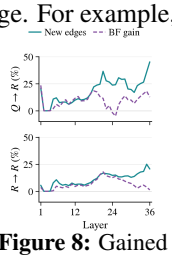

Figure 8: Gained concept connection.

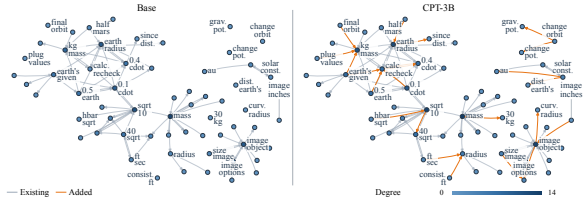

Figure 9: CPT changes latent knowledge structure.

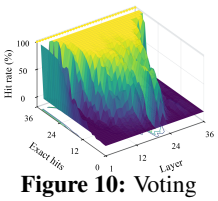

Figure 10: Voting layers

Middle Layers for Reasoning As discussed in Section 3.4, the number of voting layers is a key factor determining how effectively the data-selection procedure adapts to the model. We therefore analyze which layers contribute most to the voting process. Specifically, using a 10k-document subset, we compute the layer-wise hit ratio for samples whose scores fall within $[0, 0.1)$, as shown in Figure 10. The results indicate that the middle layers are more sensitive to data containing less familiar knowledge.

We further investigate whether the middle layers alone can identify beneficial training data, as shown in Figure 11. We continually pretrain Qwen2.5-3B on 3B tokens selected according to BF scores below 0.1 at layers 10, 20, 22, and 30, respectively. The results show that selection based on the middle layers, particularly layers 20 and 22, achieves the strongest downstream performance. Complete results are provided in Appendix Section E.

These suggest that knowledge formation in the middle layers may play an important role in improving reasoning performance, consistent with prior work showing that these layers encode semantic information and support reasoning Chen et al. (2025); Wu et al. (2025); Ben Tamo et al. (2026).

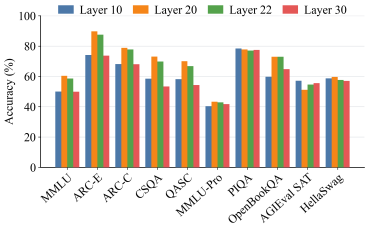

Figure 11: The model performance under various layer-choices.

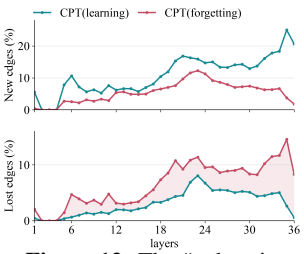

Figure 12: The # edges in Learning and Forgetting.

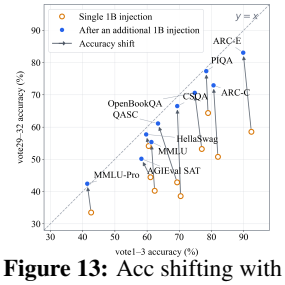

Figure 13: Acc shifting with Forgetting recover.

Catastrophic Forgetting Understanding with Latent Knowledge Latent knowledge provides another perspective for interpreting catastrophic forgetting. Prior work typically characterizes catastrophic forgetting as the degradation that occurs when a model is trained on out-of-domain data that it struggles to integrate. Our results suggest that this degradation is related to the failure to establish latent concept connections. As shown in Section 3.4, when the number of voting layers exceeds 25, more than half of the model's layers fail to construct latent concept connections for a given sample, indicating that the sample is unfamiliar to the model.

We further analyze knowledge connectivity in answers generated after CPT. First, we examine a model trained on beneficial data selected by votes from layers 1–3, labeled *CPT Learning*. This model answers more questions correctly than the base model. We then apply the same knowledge-construction procedure to a model trained on harmful data selected by votes from layers 29–32, labeled *CPT Forgetting*. For questions answered correctly under *CPT Learning*, we construct latent knowledge graphs from each model's responses and count the edges added and lost relative to the base model, as shown in Figure 12. Although the forgetting condition also introduces new

edges, the increase is substantially smaller than under the learning condition. Moreover, it exhibits a much greater reduction in the original connections. These results suggest that catastrophic forgetting involves not only insufficient formation of new knowledge connections but also the disruption of previously established ones.

Meanwhile, we investigate the effect of training order for beneficial and harmful datasets, as shown in Figure 13. We first train the model on either the beneficial or harmful dataset and then continue training it on the opposite dataset. The results show that the two training orders are not equivalent: performance degradation caused by the harmful dataset cannot be fully recovered by subsequent training on the beneficial dataset. The details are shown in Appendix Section E.

5 Related Work

Model inner structure Previous studies have shown that large language models encode relational connections among latent concepts, and that these connections can support the progression of multi-step reasoning (Dai et al., 2025; Shani et al., 2026). Moreover, the internal representations of LLMs exhibit structured semantic geometries, where hierarchical and compositional concepts are organized in a non-random manner across intermediate layers (Park et al., 2025). Recent work has further suggested that the reasoning process of LLMs can be understood through a topological perspective: reasoning trajectories form structured graphs whose connectivity, branching patterns, and recurrent substructures are closely related to the model's reasoning behavior (Minegishi et al., 2026; Zhang et al., 2026). In the post-training setting, graph-based analyses have also revealed that reinforcement learning can reshape the topology of reasoning trajectories, encouraging the model to concentrate on more consistent and successful reasoning paths (Dong et al., 2026; Matsutani et al., 2026). Motivated by these findings, we use LLMs' internal topological structures to guide continual pretraining. We construct a latent knowledge topology and select data that introduces complementary or underrepresented knowledge.

Data selection for training. Data quality is central to efficient domain adaptation (Gururangan et al., 2020). Existing approaches can be broadly grouped into quality-based and model-adaptive selection. Quality-based methods use heuristic filters, human or model judgments, and learned evaluators to rank data by properties such as writing quality, factuality, and educational value (Penedo et al., 2024; Li et al., 2024; Wettig et al., 2024). Model-adaptive methods instead score candidates relative to a target distribution or a pretrained model. Some use importance resampling or reference-model losses to favor data aligned with a target distribution (Xie et al., 2023; Lin et al., 2024); others use model loss or changes in prediction loss under attention-head masking to assess candidate data (Marion et al., 2023; Hua et al., 2025). Neuron-activated graphs further rank candidates by their similarity to target examples (Wang et al., 2026). Interpretability-based methods also use model internals to guide data selection and construction. In particular, sparse autoencoders (SAEs) decompose hidden activations into interpretable semantic features (Huben et al., 2024). Their feature coverage can identify underrepresented features and guide the synthesis of diverse post-training examples (Li et al., 2026; Jing et al., 2026). GRAFT scores candidates through concept connectivity in the target model's own hidden activations. It requires a single forward pass per candidate, without training a proxy or interpretability module or computing token-level prediction losses. This makes the method suitable for selecting data from large continual-pretraining corpora.

6 Conclusion

We introduced Branch Fraction to measure connectivity among latent semantic concepts and used it to develop GRAFT, which selects documents containing weakly connected concepts for continual pretraining. Across three base models and ten knowledge and reasoning benchmarks, GRAFT achieved the best overall average rank among the evaluated data-selection methods. It also scored long documents faster than the compared model-adaptive methods and improved both general knowledge and mathematical reasoning when selecting from mixed-domain data. Further analysis showed that reasoning gains were accompanied by new connections among previously isolated concepts, while forgetting coincided with the loss of established connections. These findings suggest that latent concept connectivity can guide data selection and help explain how continual pretraining changes a model's knowledge.

AI DISCLOSURE SECTION

In this work, we used generative AI tools for assist with translation. We have not used generative AI tools for Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, and discover research topics or identify gaps, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, format references, suggest a structure for a research paper, propose a title or keywords for a research paper are not applicable to this work. Additionally, we used generative AI tools for Formulate questions for surveys or interviews, create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, transcribe recordings of research material, summarize or analyse existing literature. We have reviewed all AI-assisted work. LLM-generated code, draft, figures was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model, 2025.

J Ben Tamo, Daniel Carlander-Reuterfelt, Jonathan Rubin, Oleg Poliannikov, Dezhi Hong, and Mingxian Wang. Linguamap: Which layers of llms speak your language and how to tune them? In *International Conference on Learning Representations*, volume 2026, pp. 33–58, 2026.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Shiqi Chen, Jinghan Zhang, Tongyao Zhu, Wei Liu, Siyang Gao, Miao Xiong, Manling Li, and Junxian He. Bring reason to vision: Understanding perception and reasoning through model merging. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9803–9817. PMLR, 13–19 Jul 2025.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

Xinnan Dai, Xianxuan Long, Chung-Hsiang Lo, Kai Guo, Shenglai Zeng, Dongsheng Luo, and Jiliang Tang. Graphghost: Tracing structures behind large language models. *arXiv preprint arXiv:2510.08613*, 2025.

Yubo Dong, Hehe Fan, Linchao Zhu, and Yi Yang. Structured reasoning for llms: A unified framework for efficiency and explainability. In *The Fourteenth International Conference on Learning Representations*, 2026.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8342–8360, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

Kai Hua, Steven Wu, Ge Zhang, and Ke Shen. Attentioninfluence: Adopting attention head influence for weak-to-strong pretraining data selection. *arXiv preprint arXiv:2505.07293*, 2025.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024.

Jinhao Jiang, Junyi Li, Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. Mix-cpt: A domain adaptation framework via decoupling knowledge learning and format alignment. In *International Conference on Learning Representations*, volume 2025, pp. 82198–82216, 2025.

Yi Jing, Zao Dai, Jinwu Hu, Zijun Yao, Lei Hou, Juanzi Li, and Xiaozhi Wang. Guiding llm post-training data engineering with model internals from sparse autoencoders. *arXiv preprint arXiv:2605.27354*, 2026.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8082–8090, 2020.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps: A math word problem repository, 2016.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

Zhongzhi Li, Xuansheng Wu, Yijiang Li, Lijie Hu, and Ninghao Liu. Less is enough: Synthesizing diverse data in llm feature space with sparse autoencoders. *arXiv preprint arXiv:2602.10388*, 2026.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, et al. Not all tokens are what you need for pretraining. *Advances in Neural Information Processing Systems*, 37:29029–29063, 2024.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL `https://transformer-circuits.pub/2025/attribution-graphs/biology.html`.

Rabeeh Karimi Mahabadi, Sanjeev Satheesh, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc-math: A 133 billion-token-scale high quality math pretraining dataset, 2025.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. 2023.

Kohsei Matsutani, Shota Takashiro, Gouki Minegishi, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Rl squeezes, sft expands: A comparative study of reasoning llms. In *International Conference on Learning Representations*, volume 2026, pp. 141482–141526, 2026.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers, 2020.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Gouki Minegishi, Hiroki Furuta, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Topology of reasoning: Understanding large reasoning models through reasoning graph properties. *Advances in Neural Information Processing Systems*, 38:13907–13938, 2026.

NVIDIA. Nemotron 3 Nano: Open, efficient mixture-of-experts hybrid Mamba-Transformer model for Agentic reasoning, 2025. URL https://research.nvidia.com/labs/nemotron/files/NVIDIA-Nemotron-3-Nano-Technical-Report.pdf. Technical report.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 76441–76463, 2025.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems?, 2021.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. volume 37, pp. 30811–30849, 2024.

M Ross Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral science*, 12(5):410–430, 1967.

Ranajoy Sadhukhan, Sheng Cao, Harry Dong, Changsheng Zhao, Attiano Purpura-Pontoniere, Yuandong Tian, Zechun Liu, and Beidi Chen. Stem: Scaling transformers with embedding modules. 2026.

Chen Shani, Liron Soffer, Dan Jurafsky, Yann LeCun, and Ravid Shwartz-Ziv. From tokens to thoughts: How llms and humans trade compression for meaning. In *International Conference on Learning Representations*, volume 2026, pp. 83019–83055, 2026.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Zijun Wang, Haoqin Tu, Weidong Zhou, Yiyang Zhou, Xiaohuan Zhou, Bingni Zhang, Weiguo Feng, Taifeng Wang, Cihang Xie, and Fengze Liu. Target-oriented pretraining data selection via neuron-activated graph. 2026.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In *International Conference on Machine Learning*, 2024.

Zhaofeng Wu, Xinyan Yu, Dani Yogatama, Jiasen Lu, and Yoon Kim. The semantic hub hypothesis: Language models share semantic representations across languages and modalities. In *International Conference on Learning Representations*, volume 2025, pp. 53705–53723, 2025.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. volume 36, pp. 34201–34227, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Xudong Wang, Zhenzhen Huang, Pengcheng Zheng, Shuai Yuan, Sheng Zheng, Qigan Sun, Jie Zou, et al. Learning global hypothesis space for enhancing synergistic reasoning chain. In *The Fourteenth International Conference on Learning Representations*, 2026.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. pp. 2299–2314, 2024.

A Latent Knowledge Construction

A.1 Semantic Concepts as Nodes

LLMs encode concept-level patterns in their token representations (Park et al., 2025; Shani et al., 2026). Beyond individual tokens, higher-level semantic concepts involved in reasoning can be aggregated in MLP up-projection representations at boundary tokens, such as periods and commas (Lindsey et al., 2025; Dai et al., 2025). These boundary tokens are often preceded by diverse sequences of tokens and can therefore summarize information from a wide range of semantic contexts. We quantify this property using predecessor diversity. For a token t, let $\mathcal{P}(t)$ denote the set of tokens that occur immediately before t, and let $p(u \mid t)$ denote the empirical probability that token u precedes t. Therefore, we give the definition for Predecessor Diversity Score to evaluate the possiblity of tokens that storing previous semantic concepts in Definition 1.

Definition 1 (Predecessor Diversity Score)**.** *For a token t, its predecessor diversity score is defined as*

$$D_{\text{prev}}(t) = \exp\left(- \sum_{u \in \mathcal{P}(t)} p(u \mid t) \log p(u \mid t) \right) \tag{4}$$

Token frequency measures how often a token occurs, but does not indicate the diversity of contexts it follows. Since semantic anchors are intended to aggregate information from varied preceding contexts, we use predecessor diversity to measure the effective number of distinct predecessors. This distinguishes frequently repeated local patterns from tokens that occur after a broader range of contexts

With the up projection of MLP layers, Figure 14 shows the semantic concepts concentration on various tokens and layers. Specifically, a token is able to handle with various of semantic instructions at the same layer, indicating by (a) and (b), which storage technology and health semantic respectively. With the same token, layer 8 in (c) is holding food based health semantic. At the same layer 10, the different token shown in (d), will hold the teatment related semantic.



(a) Token ,, layer 10, cluster 249 (b) Token ,, layer 10, cluster 322 (c) Token ,, layer 8, cluster 222 (d) Token \n, layer 10, cluster 118

Figure 14: Word clouds for semantic clusters.

A.2 Brach Fraction illustration

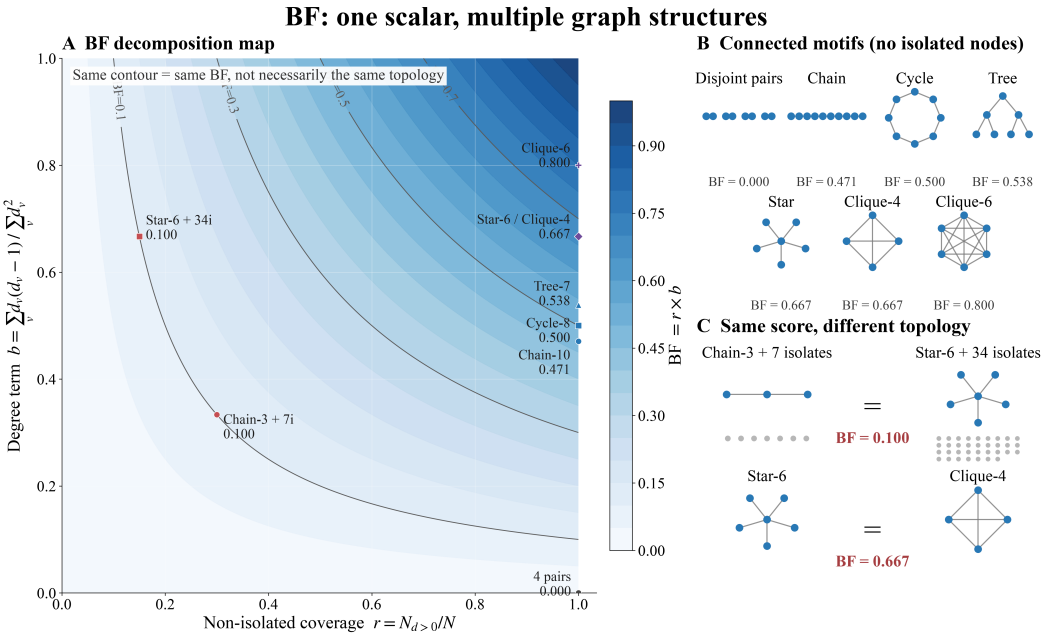

Figure 15: Brach Fraction Illustration

We use BF score to summarize two aspects of graph structure: the fraction of nodes participating in at least one edge and the extent to which edges meet at nodes with degree greater than one. A score of zero indicates that no node has degree greater than one, as in a graph of isolated nodes or disjoint pairs. Chains, cycles, branches, and denser motifs produce different scores as their degree patterns change. BF scores therefore captures meaningful differences in node connectivity, although it does not uniquely identify a graph's topology or determine whether the entire graph is connected.

In Figure 15 (A) BF is the product of non-isolated node coverage r and a degree-based term b; contours indicate equal scores. (B) Representative motifs illustrate how different connection patterns affect the score. (C) Distinct graphs can share a score: a three-node chain with seven isolated nodes and a six-node star with 34 isolated nodes both score 0.100, while a six-node star and a four-node clique both score 0.667. Thus, BF distinguishes aspects of connectivity but is not a unique descriptor of topology.

B Experimental Settings

We document the public sources of the candidate training corpora, the FineWeb subset used for knowledge-familiarity analysis, pretrained checkpoints, and evaluation benchmarks below. Bibliographic citations credit the original releases and publications.

B.1 Continual Pretraining Data

Our data-selection experiments use two publicly available candidate corpora: specialized knowledge and mathematical text. The mixed-domain experiment combines them. We use a separate FineWeb subset for the knowledge-familiarity analysis in Section 2.2.

Nemotron Specialized. We use NVIDIA's Nemotron-Pretraining-Specialized-v1 collection NVIDIA (2025). It contains synthetic material for STEM reasoning, factual knowledge, and scientific coding. Its released subsets include reasoning question–answer pairs, cross-domain reasoning problems, rewritten Wikipedia articles, mathematical textbook material, scientific coding, and STEM instruction data. This collection provides a source of specialized content for studying knowledge enrichment through data selection.

Nemotron Math. We use Nemotron-CC-Math-v1 Mahabadi et al. (2025), a mathematical pretraining corpus derived from Common Crawl. Its construction combines structured web-page extraction with language-model cleaning to preserve mathematical expressions and standardize their formatting, followed by quality filtering. The release includes the `3plus` and `4plus` quality subsets. It provides mathematical and scientific text for evaluating domain-specific data selection.

FineWeb. For the knowledge-familiarity analysis, we randomly sample 5B tokens from the FineWeb `CC-MAIN-2024-10` crawl Penedo et al. (2024) and continually pretrain Qwen2.5-3B on this subset. This subset is separate from the candidate pool used in the data-selection experiments.

B.2 Models

We consider three publicly available pretrained language models: SmolLM2-1.7B, Qwen2.5-3B, and Qwen3-8B-Base. These dense Transformer models span multiple model families and parameter scales.

SmolLM2-1.7B. We use the pretrained checkpoint `HuggingFaceTB/SmolLM2-1.7B` Allal et al. (2025), a dense 1.7B-parameter model. We evaluate it alongside Qwen2.5-3B in the 3B-token continual-pretraining comparison.

Qwen2.5-3B. We use the pretrained checkpoint `Qwen/Qwen2.5-3B` Team (2024), a dense model with approximately 3B parameters. It also serves as the model for the layer-wise knowledge-familiarity analysis in Section 2.2.

Qwen3-8B-Base. We use `Qwen/Qwen3-8B-Base` Team (2025), a dense 8B-parameter pretrained model, to study data selection at a larger model scale.

B.3 Comparison methods

We compare GRAFT against random selection and four representative data selection baselines. All methods select the same amount of continual pretraining data from the same candidate pool.

Random. This baseline samples documents uniformly at random and controls for the effect of performing additional continual pretraining without an informed selection criterion.

DCLM. The DataComp-LM baseline applies model-based quality filtering to identify high-quality documents from a large web corpus Li et al. (2024). It represents proxy-based selection using a learned text-quality classifier.

QuRating. QuRating trains a document scorer from pairwise LLM judgments of writing style, required expertise, factual content, and educational value, and uses the resulting quality ratings for data selection Wettig et al. (2024).

Perplexity (PPL). The perplexity baseline scores candidate documents with the current language model and ranks them using sequence-level perplexity, a simple model-dependent signal that has been shown effective for pretraining-data pruning Marion et al. (2023).

AttentionInfluence (Attn-Inf). AttentionInfluence identifies retrieval attention heads and measures the loss difference caused by masking them. Documents with larger loss differences are treated as more reasoning-intensive, yielding a training-free and model-adaptive selector Hua et al. (2025).

B.4 Evaluation Datasets

Our evaluation suite covers general knowledge, commonsense reasoning, and mathematical problem solving. The following descriptions identify the benchmark sources and their scope.

B.4.1 General Knowledge and Commonsense Reasoning

We evaluate the continually pretrained models on ten multiple-choice benchmarks spanning academic knowledge, scientific reasoning, and commonsense reasoning. We report accuracy (%) on each benchmark.

MMLU. Massive Multitask Language Understanding (MMLU) evaluates broad academic and professional knowledge across 57 subjects, ranging from elementary mathematics and history to computer science and law Hendrycks et al. (2020). It jointly tests factual knowledge and problem-solving ability.

ARC-Easy (ARC-E). The Easy split of the AI2 Reasoning Challenge contains grade-school science questions that can be answered by at least one of the retrieval or word co-occurrence baselines used during dataset construction Clark et al. (2018). It primarily measures general scientific knowledge and basic reasoning.

ARC-Challenge (ARC-C). The Challenge split contains the ARC questions that are answered incorrectly by both construction-time baselines Clark et al. (2018). Consequently, it places greater emphasis on combining scientific knowledge with multi-step reasoning than ARC-E.

CommonsenseQA (CSQA). CommonsenseQA consists of five-way multiple-choice questions constructed from relations in ConceptNet Talmor et al. (2019). Answering these questions requires background commonsense knowledge beyond simple lexical associations.

QASC. Question Answering via Sentence Composition (QASC) is an eight-way multiple-choice science benchmark in which answering a question generally requires retrieving and composing two supporting facts Khot et al. (2020). It evaluates knowledge retrieval, multi-hop composition, and commonsense reasoning.

MMLU-Pro. MMLU-Pro extends MMLU with more challenging, reasoning-focused questions, removes trivial or noisy items, and expands the answer space from four to ten options Wang et al. (2024). It is designed to provide a more discriminative and prompt-stable evaluation of multi-domain reasoning.

PIQA. Physical Interaction: Question Answering (PIQA) presents a goal and two candidate solutions, of which the model must select the more physically plausible one Bisk et al. (2020). The benchmark targets everyday physical commonsense about objects, materials, and actions.

OpenBookQA (OpenBook). OpenBookQA contains four-way elementary-science questions associated with an "open book" of 1,326 core science facts Mihaylov et al. (2018). Questions often require combining one of these facts with additional common knowledge and applying the result to a novel situation.

AGIEval-SAT (AGIEval). AGIEval is built from human standardized examinations Zhong et al. (2024). We use its SAT tasks, which assess mathematical problem solving and English language reasoning in the setting of a U.S. college-admission examination.

HellaSwag (Hella). HellaSwag is a four-way commonsense inference benchmark that asks the model to select the most plausible continuation of an event description or a sequence of instructions Zellers et al. (2019). Its incorrect candidates are machine-generated and adversarially filtered to be difficult for language models while remaining easy for humans.

B.4.2 Mathematical Reasoning

The mathematical evaluation comprises the seven benchmarks reported in Table 4, covering arithmetic word problems, competition mathematics, and theorem application.

GSM8K. GSM8K contains linguistically diverse grade-school mathematics word problems requiring multi-step arithmetic reasoning Cobbe et al. (2021). It evaluates the ability to translate a natural-language problem into a sequence of calculations.

MATH. MATH contains competition-level mathematics problems with worked solutions Hendrycks et al. (2021). Its coverage includes algebra, geometry, number theory, counting and probability, and precalculus, providing a more challenging complement to grade-school arithmetic tasks.

SVAMP. SVAMP introduces variations of elementary mathematics word problems to test whether models understand the problem rather than rely on superficial patterns Patel et al. (2021). The variations modify aspects such as wording and question structure while retaining the arithmetic problem-solving setting.

MAWPS. MAWPS is a repository of mathematics word problems assembled from multiple existing collections Koncel-Kedziorski et al. (2016). It supports evaluation of arithmetic and algebraic problem solving across varied linguistic forms.

TheoremQA. TheoremQA contains expert-curated questions requiring the application of mathematical and scientific theorems Chen et al. (2023). Its coverage extends to mathematics, physics, electrical engineering and computer science, and finance, testing whether a model can apply relevant theoretical knowledge.

GSM-Hard. GSM-Hard is a variant of GSM8K introduced with PAL in which numbers in the questions are replaced by larger, less common values Gao et al. (2023). It tests robustness to increased numerical difficulty within familiar word-problem structures.

ASDiv. ASDiv contains English mathematics word problems designed to provide diversity in both linguistic expression and problem types Miao et al. (2020). It complements the other arithmetic benchmarks by testing generalization across varied descriptions of mathematical relationships.

C Parameter study for Knowledge Construction

C.1 Score Uniformity for Dataset Distinguishability

We evaluate each parameter configuration by measuring the uniformity of data volume across a joint partition of BF scores and layer-wise vote counts. Let $\mathcal{D}$ denote a fixed sample of documents, e.g., 10,000 documents, L the number of model layers, and M the partition resolution, where $2 \leq M \leq L$. For a configuration $\theta = (K, \tau)$, we retain documents whose graphs contain at least two selected token positions:

$$\mathcal{D}_\theta = \left\{ d \in \mathcal{D} : \left| V_d^{(K)} \right| \geq 2 \right\}. \tag{5}$$

Graphs with a BF score of zero remain eligible. Documents with fewer than two nodes are excluded from the metric and counted separately.

BF score bins and layer-wise voting. For each eligible document d, let $s^{\theta}_{d,\ell} \in [0,1]$ denote its BF score at layer ℓ. We partition the score range into M equal-width bins:

$$I_i = \begin{cases} [(i-1)/M,\ i/M), & i = 1, \ldots, M-1, \\ [(M-1)/M,\ 1], & i = M. \end{cases} \tag{6}$$

Each layer casts one vote for the bin containing its score. The number of votes received by bin i is

$$v^{\theta}_{d,i} = \sum_{\ell=1}^{L} \mathbf{1}\left[s^{\theta}_{d,\ell} \in I_i\right], \qquad \sum_{i=1}^{M} v^{\theta}_{d,i} = L, \tag{7}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function.

Vote-count discretization. Positive vote counts are divided into M bands, each spanning a fraction $1/M$ of the model depth. For $v \geq 1$, the vote-band index is

$$g_{L,M}(v) = \left\lceil \frac{Mv}{L} \right\rceil. \tag{8}$$

Equivalently, band j contains integer vote counts satisfying

$$\frac{(j-1)L}{M} < v \leq \frac{jL}{M}, \qquad j = 1, \ldots, M. \tag{9}$$

Zero-vote bins are omitted. This construction yields an $M \times M$ partition indexed by BF score bin and vote-count band.

Data-volume aggregation. Let T_d be the full, untruncated token count of document d, measured using the evaluated model's tokenizer. The data volume assigned to cell (i, j), expressed in billions of tokens, is

$$B^{\theta}_{ij} = 10^{-9} \sum_{d \in \mathcal{D}_{\theta}} T_d\, \mathbf{1}\left[\frac{(j-1)L}{M} < v^{\theta}_{d,i} \leq \frac{jL}{M}\right]. \tag{10}$$

Each document contributes its full token count once to each occupied BF-bin cell. Contributions are neither multiplied by the number of votes nor divided among occupied bins. Consequently, a document may contribute to multiple cells, and the summed cell volume may exceed the unique corpus volume.

Uniformity metrics. For a positive total cell volume, we normalize cell volumes to obtain

$$p^{\theta}_{ij} = \frac{B^{\theta}_{ij}}{\sum_{a=1}^{M} \sum_{b=1}^{M} B^{\theta}_{ab}}. \tag{11}$$

All M^2 cells are included, with empty cells assigned zero probability. The probability range is

$$R(\theta) = \max_{i,j} p^{\theta}_{ij} - \min_{i,j} p^{\theta}_{ij}. \tag{12}$$

Definition 2 (Total Variation Distance from Uniformity)**.** *Let $P_{\theta} = (p^{\theta}_{ij})$ be the normalized data-volume distribution over the $M \times M$ partition, and let U be the uniform distribution with $u_{ij} = 1/M^2$. The total variation distance from uniformity is*

$$D_{\mathrm{TV}}(\theta) = \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left| p^{\theta}_{ij} - \frac{1}{M^2} \right|. \tag{13}$$

It lies in $[0, 1 - 1/M^2]$ and equals zero if and only if $p^{\theta}_{ij} = 1/M^2$ for every cell. Smaller values indicate a more uniform distribution of data volume.

Each metric is computed once per configuration (K, τ) from the joint cell-volume distribution, rather than computed independently at each layer and subsequently averaged. These metrics characterize partition uniformity and do not directly measure data quality or downstream model performance. In our experiments, we set $M = 10$, yielding ten BF score bins and ten positive-vote bands, for a total of 100 cells.

Table 2: Downstream evaluation results for Qwen3-8B. Macro denotes the unweighted mean across the ten tasks, and Δ Base denotes the absolute Macro improvement over the base model. Higher task scores and lower average ranks are better. Best and second-best results are shown in **bold** and underlined, respectively.

Method	MMLU	ARC-E	ARC-C	CSQA	QASC	MMLU-Pro	PIQA	OpenBook	AGIEval	Hella	Macro	Avg. Rank	Δ Base
Base	69.14	92.93	88.48	79.52	79.16	56.17	83.35	87.40	72.31	**77.39**	78.59	4.40	–
Random	75.53	96.93	92.49	**80.67**	81.43	60.04	86.56	**90.00**	76.90	69.26	80.98	2.90	+2.40
DCLM	72.87	97.26	**92.75**	79.44	80.13	63.36	**87.87**	89.60	77.22	68.83	80.93	3.20	+2.35
QuRating	76.09	96.42	91.21	79.52	82.94	64.39	86.67	90.40	78.01	69.31	81.50	2.75	+2.91
GRAFT	**76.46**	**97.35**	92.15	80.34	**83.26**	**64.64**	86.67	89.80	**81.01**	74.14	**82.58**	**1.75**	**+4.00**

C.2 Parameter Studies for Latent Knowledge Construction

During knowledge construction, the number of top-ranked tokens (K) and the similarity threshold τ determine how distinguishable the resulting Branch Fraction scores are across documents. We use Total Variation Entropy to assess whether a given parameter configuration produces a balanced score distribution for the target dataset. A higher Total Variation Entropy indicates a less discriminative score distribution, as shown in Figure 16.

The results show that the number of top-ranked tokens has only a limited effect on the score distribution: even a small value of K provides coverage for more than 99% of the documents.

In contrast, increasing the similarity threshold τ leads to a sharp increase in the proportion of documents assigned low Branch Fraction scores. Therefore, we primarily tune τ during parameter selection and keep K fixed, avoiding parameter settings that produce an excessively concentrated or poorly discriminative score distribution.

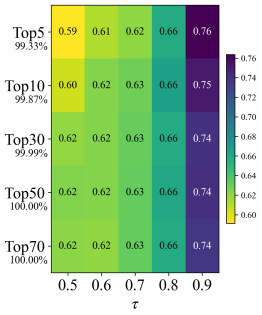

Figure 16: Data distribution.

D Experiments Result Details

D.1 Qwen3-8B Results

Here we show the whole results on Qwen3-8B model in Table 2.

D.2 Mixtured Training on STEM knowledge and MATH

Here we show the whole results on the mix setting in Table 3 and Table 4.

E Additional Results

Table 3: Downstream performance at different continual-pretraining token budgets. Higher values are better.

Tokens	Method	MMLU	ARC-E	ARC-C	CSQA	QASC	MMLU-Pro	PIQA	OpenBook	AGIEval	Hella	Macro
0B	Base	0.4365	0.6709	0.5785	0.4930	0.4525	0.3237	0.6676	0.6020	0.5791	0.5741	0.5378
2B	PPL	0.4380	0.7029	0.6510	0.4357	0.4914	0.3843	0.6039	0.6800	0.5063	0.5227	0.5416
2B	GRAFT	0.5359	0.8199	0.7526	0.5659	0.6263	0.3876	0.7709	0.6240	0.5285	0.5960	0.6208
4B	PPL	0.4069	0.6864	0.6118	0.3726	0.4428	0.4122	0.5223	0.6160	0.4715	0.3944	0.4937
4B	GRAFT	0.5315	0.8350	0.7500	0.6134	0.6242	0.4094	0.7748	0.6960	0.5158	0.5900	0.6340
6B	PPL	0.2551	0.3367	0.3055	0.2310	0.1987	0.3990	0.5185	0.3620	0.3481	0.3346	0.3289
6B	GRAFT	0.5139	0.7875	0.7355	0.5250	0.5940	0.4137	0.7682	0.6580	0.5063	0.6045	0.6107
8B	PPL	0.2683	0.3641	0.3285	0.2531	0.2009	0.4139	0.5381	0.4700	0.3892	0.3867	0.3613
8B	GRAFT	0.5424	0.8114	0.7457	0.5676	0.6285	0.4158	0.7818	0.6220	0.5127	0.5884	0.6216
10B	PPL	0.2716	0.3801	0.3370	0.2514	0.2063	0.4130	0.5076	0.3420	0.3639	0.3100	0.3383
10B	GRAFT	0.5484	0.8321	0.7747	0.5766	0.6382	0.4174	0.7699	0.6180	0.5364	0.5967	0.6308
12B	PPL	0.2554	0.3451	0.3003	0.2555	0.1685	0.4181	0.5174	0.4140	0.3623	0.3577	0.3394
12B	GRAFT	0.5581	0.8670	0.7756	0.6085	0.6739	0.4161	0.7644	0.6540	0.5538	0.6003	0.6472
14B	PPL	0.2670	0.3775	0.3234	0.2375	0.2300	0.3770	0.5141	0.3340	0.3750	0.2910	0.3327
14B	GRAFT	0.5678	0.8695	0.7901	0.5938	0.6425	0.4116	0.7851	0.6860	0.5870	0.6095	0.6543
16B	PPL	0.2662	0.3666	0.3370	0.2596	0.2030	0.3747	0.5212	0.3660	0.3750	0.3452	0.3415
16B	GRAFT	0.5718	0.8620	0.7807	0.5839	0.6382	0.4077	0.7780	0.6960	0.5522	0.6001	0.6471
18B	PPL	0.2693	0.3851	0.3362	0.2645	0.2214	0.3707	0.5223	0.3660	0.3750	0.3652	0.3476
18B	GRAFT	0.5677	0.8540	0.7850	0.5962	0.6415	0.4152	0.7769	0.6640	0.5538	0.6099	0.6464
20B	PPL	0.2684	0.3733	0.3251	0.2523	0.2084	0.3772	0.5174	0.4180	0.3782	0.3896	0.3508
20B	GRAFT	0.5662	0.8544	0.7756	0.5733	0.6328	0.4131	0.7873	0.6560	0.5348	0.6069	0.6400

Table 4: Mathematical reasoning performance at different continual-pretraining token budgets. Higher values are better.

Tokens	Method	GSM8K	MATH	SVAMP	MAWPS	TheoremQA	GSM-Hard	ASDiv	Macro
0B	Base	0.7544	0.4226	0.7520	0.5808	0.2263	0.3851	0.8056	0.5610
2B	PPL	0.7483	0.4770	0.8320	0.6923	0.3000	0.3639	0.8256	0.6056
2B	GRAFT	0.7528	0.4844	0.8290	0.7231	0.3050	0.3776	0.8052	0.6110
4B	PPL	0.4549	0.4124	0.5050	0.3346	0.2875	0.2411	0.5258	0.3945
4B	GRAFT	0.7248	0.4860	0.8250	0.7327	0.2913	0.3836	0.8074	0.6072
6B	PPL	0.3260	0.4166	0.3280	0.4404	0.3175	0.2441	0.5978	0.3815
6B	GRAFT	0.7316	0.4776	0.8040	0.7500	0.2875	0.3829	0.7922	0.6037
8B	PPL	0.6247	0.5090	0.7200	0.6788	0.3300	0.3851	0.8217	0.5813
8B	GRAFT	0.6641	0.4684	0.7570	0.6962	0.2888	0.3571	0.7384	0.5671
10B	PPL	0.4276	0.4942	0.4620	0.4038	0.3175	0.2654	0.5761	0.4209
10B	GRAFT	0.7309	0.4738	0.8010	0.7000	0.3075	0.3942	0.7857	0.5990
12B	PPL	0.2593	0.4944	0.3050	0.4019	0.3025	0.3480	0.6395	0.3929
12B	GRAFT	0.7271	0.4732	0.7980	0.7154	0.2925	0.3851	0.7831	0.5963
14B	PPL	0.3920	0.5040	0.3620	0.5788	0.3213	0.3927	0.7679	0.4741
14B	GRAFT	0.7096	0.4738	0.8010	0.6788	0.3025	0.3760	0.7761	0.5883
16B	PPL	0.4663	0.5108	0.4580	0.4192	0.3213	0.3313	0.5874	0.4420
16B	GRAFT	0.7119	0.4848	0.7840	0.6615	0.3263	0.3738	0.7740	0.5880
18B	PPL	0.5011	0.5250	0.4710	0.4365	0.3438	0.3351	0.5813	0.4563
18B	GRAFT	0.7225	0.4750	0.7990	0.6712	0.3125	0.3760	0.7727	0.5898
20B	PPL	0.5292	0.5162	0.5190	0.4865	0.3238	0.3533	0.6451	0.4819
20B	GRAFT	0.7339	0.4806	0.7970	0.6827	0.3013	0.3685	0.7792	0.5919

Table 5: Performance comparison of four data-injection strategies on ten benchmarks. All results are reported as percentages using 0-shot CoT evaluation. The best result in each column is highlighted in bold.

Training Strategy	MMLU	ARC-E	ARC-C	CSQA	QASC	MMLU-Pro	PIQA	OpenBookQA	AGIEval SAT	HellaSwag	Macro
vote1to3 (1B)	**62.36**	**92.38**	**82.00**	**70.43**	**69.33**	**42.63**	**78.94**	**77.00**	**61.08**	**60.50**	**69.66**
vote29to32 (1B)	40.24	58.54	50.77	38.57	42.87	33.50	64.36	53.20	44.46	54.14	48.07
vote1to3 → vote29to32 (2B)	61.33	89.86	80.63	69.37	63.39	41.42	78.35	74.80	58.23	59.73	67.71
vote29to32 → vote1to3 (2B)	55.36	83.08	72.95	66.50	61.12	42.44	77.37	70.60	50.16	57.73	63.73

Table 6: Performance comparison across different single-layer injection positions. All models are trained with 3B injected tokens and evaluated using the same ten-task 0-shot CoT protocol. Results are reported as percentages. The best result in each column is highlighted in bold.

Layer	MMLU	ARC-E	ARC-C	CSQA	QASC	MMLU-Pro	PIQA	OpenBookQA	AGIEval SAT	HellaSwag	Macro
10	50.04	74.07	68.09	58.48	58.21	40.40	**78.40**	59.80	**57.12**	58.73	60.33
20	**60.38**	**89.73**	**78.84**	**73.05**	**69.98**	**43.31**	77.75	**73.00**	51.11	**59.60**	**67.68**
22	58.61	87.50	77.82	69.78	66.74	42.83	77.09	**73.00**	54.59	57.68	66.56
30	49.89	73.70	68.00	53.32	54.32	41.71	77.42	64.80	55.54	57.05	59.57