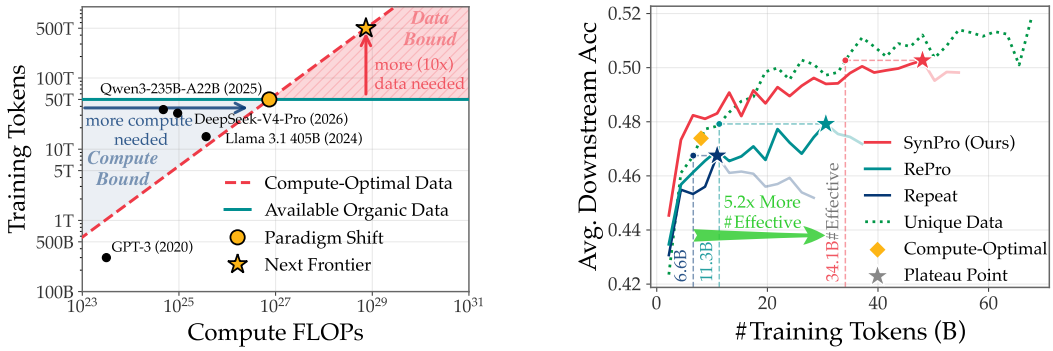
Generating Pretraining Tokens from Organic Data for Data-Bound Scaling

Zichun Yu[1], Chenyan Xiong[1,2]
[1]Language Technologies Institute, Carnegie Mellon University [2]Xlue
{zichunyu,cx}@andrew.cmu.edu

Abstract

LLM pretraining is shifting from a compute-bound to a data-bound regime, where available human (organic) text falls far short of scaling demands. However, reaching the data-bound regime does not mean the model has fully utilized its organic corpus. In this paper, we introduce SynPro, a synthetic data generation framework that helps LLMs more thoroughly learn from limited organic data. SynPro applies two operations, rephrasing and reformatting, that present the same organic source in diverse forms to facilitate deeper learning without introducing external information. Both generators are optimized via reinforcement learning with quality, faithfulness, and data influence rewards, and are continuously updated as pretraining plateaus to target content the model has yet to absorb. We pretrain 400M, 1.1B, and 2B models with 10% of their Chinchilla-optimal tokens (0.8B, 2.2B, and 4B) from DCLM-Baseline, reflecting a realistic data-bound regime in frontier pretraining. Our results reveal that organic data is significantly underutilized by standard repetition: SynPro unlocks 3.4–5.2× the effective tokens of repetition, even surpassing the non-data-bound oracle that trains on equivalent unique data at the 1.1B and 2B scales. Analyses confirm that faithful, model-aware synthesis sustains data-bound scaling without causing distribution collapse. We open-source our code at https://github.com/cxcscmu/SynPro.

1 Introduction



(a) Paradigm shift in frontier model pretraining (b) 400M model & 0.8B unique organic tokens

Figure 1: (a) Paradigm shift in frontier pretraining from compute-bound to data-bound. (b) Typical data-bound setup (400M model, 1/10 compute-optimal data); 1.1B in Figure 6.

"We've achieved peak data and there'll be no more."[1] As shown in Figure 1a, frontier model pretraining is undergoing a paradigm shift from compute-bound to data-bound scaling: while the compute-optimal data requirement (Hoffmann et al., 2022) increases steadily with surging compute, the growth of high-quality human (organic) text can no longer keep

[1]Attributed to Ilya Sutskever in public remarks discussing the limits of available pretraining data.

pace (Villalobos & Ho, 2022; Maini et al., 2025). Beyond the transition point, the next frontier of scaling demands an order of magnitude more data than is currently available (Villalobos et al., 2024). In this data-bound regime, training with repeated passes over the available corpus often yields diminishing returns and rapid saturation (Muennighoff et al., 2023).

To further scale pretraining beyond the organic data limit, synthetic data emerges as a practical path (Maini et al., 2024; Ben Allal et al., 2024; Maini et al., 2025). However, unconstrained generation can lead to distribution collapse (Shumailov et al., 2024; Dohmatob et al., 2025) or distill the generator's parametric knowledge in ways that hurt generalization (Chen et al., 2024). At the same time, prior work (Frank, 2023; Warstadt et al., 2023) suggests that effective learning may require much less data than current pretraining practice, implying that available data may still be underutilized. These motivate a more constrained use of synthetic data: *Can we generate synthetic data grounded in organic data to help LLMs learn more?*

In this paper, we introduce SynPro, a synthetic data generation framework that helps pretraining models learn more thoroughly from limited organic data. SynPro generates data through two operations that facilitate model learning: *rephrasing* (Yu & Xiong, 2025), which introduces lexical and syntactic diversity while preserving core semantics, and *reformatting* (Su et al., 2025), which converts source content into task-oriented forms. Both operations are optimized via reinforcement learning with a *quality* reward that ensures coherent text, a *faithfulness* reward that grounds outputs in the source document, and a *data influence* reward (Yu et al., 2024) that steers generation toward content the current pretraining model has yet to absorb. SynPro continuously updates the generator to produce informative yet grounded data that helps the model continue improving.

We pretrain 400M, 1.1B, and 2B models with 10% of their Chinchilla-optimal tokens from the English corpus DCLM-Baseline (Li et al., 2024), reflecting a realistic data-bound regime in frontier pretraining. As shown in Figure 1b, our results reveal that organic pretraining data is significantly underutilized by standard repetition: SynPro unlocks 5.2× the *effective tokens* (equivalent unique data yielding the same performance) of simple repetition and 3.0× those of RePro (Yu & Xiong, 2025), the state-of-the-art web rephrasing baseline. SynPro stays above repetition at every point of the curve, while repetition quickly degrades once it plateaus. At the 1.1B and 2B scales, SynPro even surpasses training on the same amount of unique organic data, demonstrating that faithful synthesis can unlock significantly more value from limited data for LLM pretraining.

To better understand why these gains arise, we first show that our synthetic data preserves both pointwise and distributional properties of the organic corpus rather than collapsing toward a narrow mode, confirming the value of grounded generation for sustained pretraining gains. Furthermore, our generator adaptively shifts its output toward content the current model has yet to absorb, producing more informative data throughout training where static approaches decay. These results highlight that faithful, model-aware synthesis can sustain data-bound scaling without causing distribution collapse or relying on distillation.

We summarize our contributions as follows:

1. We propose SynPro, a model-aware synthetic data generation framework that helps pretraining models more thoroughly utilize a limited organic corpus.
2. We systematically define and study the data-bound regime, where SynPro achieves up to 5.2× the effective tokens over repetition, approaching the unique data oracle.
3. SynPro reveals that organic pretraining data is underutilized rather than exhausted, and faithful synthesis can unlock more value from it without distribution collapse.

2 Related work

LLM scaling and data wall. Progress in large language models has been driven by jointly scaling parameters, computation, and training data (Kaplan et al., 2020; Hoffmann et al., 2022). Compute is no longer the primary bottleneck thanks to hardware improvements and architectural innovations (Shazeer et al., 2017; Liu et al., 2024); instead, projections suggest that publicly available human-written text will be insufficient to sustain current

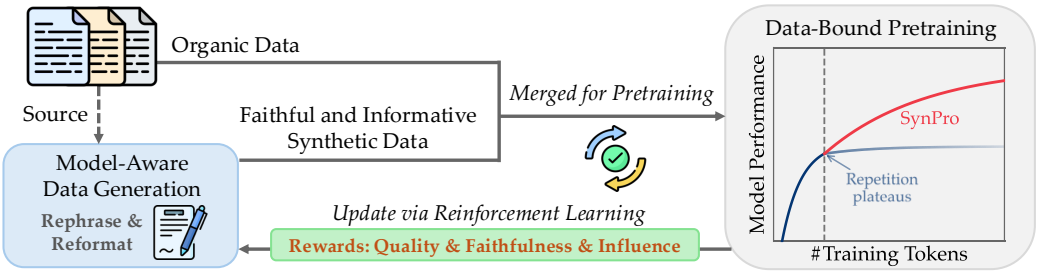


Figure 2: Overview of SynPro. We train generators to provide faithful and informative synthetic data from organic source, enabling sustained improvement for data-bound scaling.

scaling trajectories (Villalobos & Ho, 2022; Villalobos et al., 2024; Shen et al., 2025). When the available data falls well below the compute-optimal requirement (Hoffmann et al., 2022), practitioners often resort to repeated passes over the same corpus, which yields diminishing gains after only a few (typically 4) epochs (Muennighoff et al., 2023). This phenomenon, known as the data wall, leads to a plateau in performance despite increased training time.

Synthetic data for pretraining. Generating synthetic text is a natural strategy to augment a limited organic corpus (Havrilla et al., 2024). Effective methods include document-level paraphrasing (Maini et al., 2024), guided rewriting (Nguyen et al., 2025), and textbook-style generation (Li et al., 2023; Ben Allal et al., 2024; Hao et al., 2025), which improve pretraining data volume and quality (Su et al., 2025; Abdin et al., 2024; Maini et al., 2025). We follow the prompt design of Nemotron-CC for reformatting, but *train the generator with explicit rewards* instead of relying on static prompts. However, unconstrained synthesis risks model collapse (Dohmatob et al., 2024; 2025): successive training on synthetic data erodes the tail distribution (Shumailov et al., 2024), and negative effects can propagate to post-training (Chen et al., 2024). These findings highlight that grounding synthetic data in the source content is essential. To keep such faithfulness, ProX (Zhou et al., 2025) and RefineX (Bi et al., 2025) restrict editing to conservative operations such as deletion and normalization, while RePro (Yu & Xiong, 2025) optimizes a rephraser via quality and faithfulness rewards to produce high-quality yet grounded data. Unlike RePro that recycles low-quality web data, we target the *data-bound scaling* regime, which motivates a *model-aware influence reward* to keep mining new signal from the same organic corpus.

Model-aware data curation. A line of work on curriculum learning (Zhang et al., 2026) tailors data strategies to the model's evolving needs. On the selection side, DsDm (Engstrom et al., 2024), MATES (Yu et al., 2024) and GREATS (Wang et al., 2024) leverage data influence (Koh & Liang, 2017) to prioritize informative samples, while CLIMB (Diao et al., 2025) tunes domain proportions through iterative search guided by a proxy performance predictor. On the generation side, Montessori-Instruct (Li et al., 2025) trains a teacher to produce high-influence instruction-tuning examples for a target student and both models are updated in tandem. These advances demonstrate the strong potential of model-aware optimization to enhance data curation. Our work extends such model-aware optimization to *generating pretraining tokens* for data-bound scaling.

3 Method

This section presents SynPro (Figure 2), an effective and faithful synthetic data generation framework that helps the pretraining model better utilize a limited organic corpus.

3.1 Data-bound scaling regime

We assume access to an organic corpus $\mathcal{D}_{\text{org}}$, comprising all web-sourced data available for pretraining. Given a compute budget C, the compute-optimal data requirement is $D^*(C)$

tokens (Hoffmann et al., 2022). We define the *available data ratio* α as:

$$\alpha = \frac{|\mathcal{D}_{\text{org}}|}{D^*(C)}. \tag{1}$$

In the early stages of LLM development, data was abundant relative to compute C, so $\alpha > 1$ and scaling was *compute-bound*. As illustrated in Figure 1a, frontier pretraining is now shifting to a *data-bound* regime where publicly available text is approaching exhaustion (Villalobos et al., 2024). Meanwhile, Hoffmann et al. (2022) predict that the next frontier requires roughly $100\times$ more compute and $10\times$ more data, which places α at around **10%**, meaning the available organic data covers only a fraction of compute-optimal requirements. The standard practice in this regime is to repeatedly train the language model $\mathcal{M}$ on $\mathcal{D}_{\text{org}}$, but this yields diminishing returns after only a few epochs (Muennighoff et al., 2023).

3.2 Model-aware synthetic data generation

To overcome data limitations, SynPro enables a better utilization of the limited organic corpus by synthesizing grounded and informative data $\mathcal{D}_{\text{syn}}$ across three repeating stages: (1) LM pretraining, (2) generation policy update, and (3) generation of new synthetic data.

To start the process, we apply an initial generation policy π_0 to each organic sample $x \in \mathcal{D}_{\text{org}}$, conditioned on a prompt p, to produce an initial synthetic corpus and training set:

$$\mathcal{D}^0_{\text{syn}} = \{\pi_0(p, x) \mid x \in \mathcal{D}_{\text{org}}\}, \tag{2}$$

$$\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{org}} \cup \mathcal{D}^0_{\text{syn}}. \tag{3}$$

Stage 1: LM pretraining. At each iteration i, we continue pretraining on $\mathcal{D}_{\text{train}}$ from the previous checkpoint $\mathcal{M}^*_{i-1}$ ($\mathcal{M}^*_0$ is randomly initialized) until the reference loss $\mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M})$ saturates. Here, the reference set $\mathcal{D}_{\text{ref}}$ serves as a proxy for the model's generalization and does not overlap with the downstream evaluation. We define saturation as the point where the reference loss fails to improve over the best of the previous *two epochs*, formally:

$$\mathcal{M}_i = \mathcal{M}^*_{i-1},\ \mathcal{L}_0 = \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}^*_{i-1}), \tag{4}$$

$$\text{For } t = 1, 2, \ldots : \tag{5}$$

$$\mathcal{M}'_i \leftarrow \mathcal{A}(\mathcal{M}_i,\ \mathcal{D}_{\text{train}}),\ \mathcal{L}_t = \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}'_i), \tag{6}$$

$$\text{If } t > 1 \text{ and } \mathcal{L}_t \geq \min(\mathcal{L}_{t-1},\ \mathcal{L}_{t-2}) \text{: } \textbf{break} \tag{7}$$

$$\mathcal{M}_i = \mathcal{M}'_i, \tag{8}$$

$$\mathcal{M}^*_i = \mathcal{M}_i, \tag{9}$$

where $\mathcal{A}(\mathcal{M},\ \mathcal{D}_{\text{train}})$ denotes one epoch of training on $\mathcal{D}_{\text{train}}$ starting from checkpoint $\mathcal{M}$.

Stage 2: Policy update. When model pretraining saturates, we update the generation policy to produce synthetic data that is more informative for the current saturated model $\mathcal{M}^*_i$. The policy is optimized under a composite reward:

$$r_i(x, \tilde{x}) = \lambda_{\text{quality}}\, r_{\text{quality}}(\tilde{x}) + \lambda_{\text{faithful}}\, r_{\text{faithful}}(x, \tilde{x}) + \lambda_{\text{influence}}\, r_{\text{influence}}(\tilde{x} \mid \mathcal{M}^*_i), \tag{10}$$

where λ_{quality}, $\lambda_{\text{faithful}}$, and $\lambda_{\text{influence}}$ control the relative weight of each reward component. The quality, faithfulness, and data influence rewards ensure that generated synthetic data is written in high-quality language, grounded in the source document, and targeted at what the current model has yet to learn (detailed in §3.3). We update the policy with standard reinforcement learning to maximize the expected reward:

$$\pi_i = \arg\max_{\pi}\ \mathbb{E}_{x \sim \mathcal{D}_{\text{org}},\, \tilde{x} \sim \pi(\cdot \mid p, x)} \left[r_i(x, \tilde{x}) \right]. \tag{11}$$

Stage 3: Generation of new synthetic data. The updated policy π_i generates a fresh set of synthetic data from the organic corpus, which is appended to the training set:

$$\mathcal{D}^i_{\text{syn}} = \{\pi_i(p, x) \mid x \in \mathcal{D}_{\text{org}}\},\ \mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}^i_{\text{syn}}. \tag{12}$$

Operation	Organic source (excerpt)	Synthetic output (excerpt)
Rephrasing	"The acquisition of embryogenic competence has been achieved on a few occasions and with low induction rates 2,3-Butanediol [2,3,12]."	"Obtaining embryogenic competence has been achieved infrequently and with limited induction rates."
Reformatting	"The Delphi method was developed in the United States in the 1950s and 1960s in the military domain."	"*Question:* What is the origin of the Delphi method? *Answer:* Military in the United States in the 1950s and 1960s."

Table 1: Examples of the two operations. Rephrasing preserves the semantics while polishing the surface form, including removing the noisy spans injected by web scraping (marked in red); reformatting restructures the same source content. Full cases are in Appendix F.

We then return to Stage 1 until the reference loss fails to improve in the next iteration, i.e., $\mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}_i^*) \geq \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}_{i-1}^*)$. Together, SynPro continuously provides effective synthetic data for data-bound pretraining. Algorithm 1 summarizes the entire pipeline.

3.3 Synthetic data operations and reward design

SynPro employs two complementary operations (prompts) to synthesize helpful data from $\mathcal{D}_{\text{org}}$, *rephrasing* and *reformatting*. *Rephrasing* rewrites each source document to diversify surface form (word choice, grammar fix, clause ordering) while preserving the core semantics (Yu & Xiong, 2025). *Reformatting* transforms each document into a task-oriented form, such as a comparative analysis, a knowledge highlight, or a reasoning trace, allowing one source document to yield multiple distinct yet grounded outputs (Su et al., 2025). Table 1 shows an example of each operation.

Both operations share the quality and data influence rewards, which apply uniformly regardless of the output format; faithfulness is defined differently for each due to their distinct structural relationships to the source. We detail the reward design below.

Quality (r_{quality}). We adopt DataMan (Peng et al., 2025), a tuned small LM that evaluates text across 13 quality criteria (e.g., coherence, topic focus) and gives an overall score:

$$r_{\text{quality}}(\tilde{x}) = \text{DataMan}(\tilde{x}). \tag{13}$$

This reward ensures the generator produces high-quality text that is well-formed, coherent, and informative, which is crucial for effective pretraining.

Data influence ($r_{\text{influence}}$). Following MATES (Yu et al., 2024) and Forward-INF (Ko et al., 2024), we efficiently compute data influence as the loss reduction on a synthetic sample $\tilde{x}$ after the current model is updated on the reference set $\mathcal{D}_{\text{ref}}$:

$$r_{\text{influence}}(\tilde{x} \mid \mathcal{M}_i^*) = \mathcal{L}(\tilde{x} \mid \mathcal{M}_i^*) - \mathcal{L}(\tilde{x} \mid \mathcal{A}(\mathcal{M}_i^*, \mathcal{D}_{\text{ref}})), \tag{14}$$

where $\mathcal{A}(\mathcal{M}_i^*, \mathcal{D}_{\text{ref}})$ denotes training on $\mathcal{D}_{\text{ref}}$. This reverses the oracle computation, so scoring each sample requires only *two forward passes* instead of a per-sample backward pass, with a detailed derivation provided in Appendix C. This reward steers the generator toward outputs useful to the current model, making the synthetic data model-aware.

Faithfulness (r_{faithful}). For rephrasing, following RePro (Yu & Xiong, 2025), we combine three binary rewards: semantic similarity via BERTScore (Zhang et al., 2020), structural preservation via LLM-as-a-judge, and a length constraint to penalize free-form generation:

$$r_{\text{faithful}}(x, \tilde{x}) = \mathbf{1}[\text{BERTScore}(x, \tilde{x}) \geq \tau_{\text{sem}}] \cdot \mathbf{1}[\text{Structure}(x, \tilde{x})] \cdot \mathbf{1}\left[\frac{\text{Len}(\tilde{x})}{\text{Len}(x)} \leq \tau_{\text{len}}\right]. \tag{15}$$

For reformatting, where surface form changes substantially, we instead train a small reward model distilled from an LLM to classify whether the output is faithful to the source, yielding a binary reward $r_{\text{faithful}}(x, \tilde{x}) \in \{0, 1\}$. The faithfulness reward ensures synthetic data reflect the source document rather than the generator's distilled knowledge.

4 Experimental setup

Pretraining model and data. We pretrain decoder-only Transformers (Vaswani et al., 2017) from scratch at three scales: a 1.1B model uses the OLMo2-1B (Walsh et al., 2025) architecture, together with a 400M scaled-down variant and a 2B scaled-up variant. We randomly sample from DCLM-Baseline (Li et al., 2024) as the organic corpus $\mathcal{D}_{\text{org}}$, the state-of-the-art open-source pretraining dataset. The compute-optimal data requirements are 8B/22B/40B for the 400M/1.1B/2B model (Hoffmann et al., 2022). To study different degrees of data limitation, we set $\alpha \in \{5\%, 10\%, 15\%\}$ for the 400M model (corresponding to 0.4B, 0.8B, and 1.2B organic tokens) and $\alpha = 10\%$ for the 1.1B and 2B models (2.2B and 4B organic tokens, respectively). 10% reflects a typical bottleneck in frontier pretraining based on the current scaling trend (Villalobos et al., 2024). We use warmup and stable phases of the WSD scheduler (Hu et al., 2024). More details are provided in Appendix D.

Baselines. We compare SynPro against four baselines:

1. *Repeat*: repeatedly training on the full organic data $\mathcal{D}_{\text{org}}$ until saturation.
2. *QBSR* (Quality-Based Selective Repetition) (Luo et al., 2025): a static curriculum that, after full repetition saturates, continues training by repeating only the top 80%, 60%, 40%, and 20% of data ranked by quality scores (we choose DataMan here).
3. *MATES* (Yu et al., 2024): an adaptive curriculum that, after full repetition saturates, selects data with positive influence for the next epoch and iterates until saturation.
4. *RePro* (Yu & Xiong, 2025): augmenting the organic corpus with rephrased data generated by OLMo2-1B-Instruct (Walsh et al., 2025) trained with quality and faithfulness rewards, which has been shown to outperform other web rephrasing methods such as WRAP (Maini et al., 2024) and ReWire (Nguyen et al., 2025).

We also report *Unique Data*, which trains on unique organic data from DCLM-Baseline, as a non-data-bound oracle. This comparison is not apples-to-apples to our method.

Evaluation. Following Walsh et al. (2025), we report *zero-shot accuracy* on 9 downstream tasks: ARC-Easy, ARC-Challenge (Clark et al., 2018), SciQ (Welbl et al., 2017), OpenBookQA (Mihaylov et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), CommonsenseQA (Talmor et al., 2019), and SIQA (Sap et al., 2019). These tasks provide a comprehensive assessment of commonsense reasoning, language understanding, and knowledge. We further define *effective tokens* as the amount of unique organic data that, trained for a single epoch, yields the same performance, to quantify data efficiency. We also report *recovery ratio*, the fraction of the performance gap from Repeat to Unique Data:

$$\text{Recovery Ratio} = \frac{\text{Method Acc} - \text{Repeat Acc}}{\text{Unique Acc} - \text{Repeat Acc}}. \tag{16}$$

Implementation details. We use FLAN (Wei et al., 2022) as the held-out reference set $\mathcal{D}_{\text{ref}}$ for computing data influence following Yu et al. (2025). We initialize both generators with OLMo2-1B-Instruct and train them with quality and faithfulness rewards to serve as π_0. For rephrasing, we prompt Qwen3-1.7B (Yang et al., 2025) for structural faithfulness, and set $\tau_{\text{sem}} = 0.65$; $\tau_{\text{len}} = 1.25$ following RePro. For reformatting, we fine-tune Qwen3-1.7B on 10k labels from Gemini 3.1 Flash-Lite as the faithful judge. The val. accuracy of our judge is 95%. Reward coefficients are $\lambda_{\text{quality}} = 1$, $\lambda_{\text{faithful}} = 1$, and $\lambda_{\text{influence}} = 3$. Empirically, SynPro converges in three iterations. More details and prompts are provided in Appendix D and G.

5 Evaluation results

In this section, we present main results (§5.1), conduct ablations (§5.2), and analyze pointwise faithfulness (§5.3), distribution preservation (§5.4), and model-awareness (§5.5). Additional results and compute details are provided in the Appendix E.

Method	#Training Tokens	#Effective Tokens	*Commonsense Reasoning* CSQA	OBQA	PIQA	SIQA	*Language Understanding* HellaSwag	WinoG	*World Knowledge* ARC-e	ARC-c	SciQ	Avg	Rec. Ratio
400M model, 8B Chinchilla-optimal tokens, α=5% (0.4B available organic tokens)													
Unique	*13.1B*	*13.1B*	*0.3415*	*0.3340*	*0.6480*	*0.4145*	*0.3731*	*0.5130*	*0.6070*	*0.2876*	*0.8330*	*0.4835*	*100%*
Repeat	5.5B	2.9B $_{1.0\times}$	0.3006	0.2920	0.6045	0.4110	0.3193	0.5083	0.5088	0.2575	0.7230	0.4361	0%
QBSR	6.3B	3.2B $_{1.1\times}$	0.3071	0.3040	0.6251	0.4120	0.3225	0.5122	0.5386	0.2341	0.7210	0.4418	12%
MATES	6.0B	3.0B $_{1.0\times}$	0.3112	0.3060	0.5990	0.4033	0.3209	0.5193	0.5105	0.2742	0.7010	0.4384	5%
RePro	7.6B	4.0B $_{1.4\times}$	0.3055	**0.3200**	0.6322	0.4023	0.3384	**0.5320**	0.5474	0.2709	0.7510	0.4555	41%
SynPro	13.1B	**9.9B** $_{\mathbf{3.4\times}}$	**0.3604**	0.3160	**0.6393**	**0.4156**	**0.3485**	0.5249	**0.6053**	**0.2843**	**0.8070**	**0.4779**	**88%**
400M model, 8B Chinchilla-optimal tokens, α=10% (0.8B available organic tokens)													
Unique	*48.0B*	*48.0B*	*0.3686*	*0.3360*	*0.6790*	*0.4253*	*0.4303*	*0.5399*	*0.6579*	*0.3211*	*0.8510*	*0.5121*	*100%*
Repeat	10.9B	6.6B $_{1.0\times}$	0.3358	0.3140	0.6442	0.4135	0.3519	0.5304	0.5579	0.2676	0.7920	0.4675	0%
QBSR	12.5B	6.9B $_{1.0\times}$	0.3342	0.3160	0.6333	0.4140	0.3533	0.5209	0.5754	0.2742	0.8000	0.4690	3%
MATES	12.1B	7.5B $_{1.1\times}$	0.3415	0.3200	0.6436	0.4150	0.3602	0.5012	0.5684	0.2709	0.8220	0.4714	9%
RePro	30.6B	11.3B $_{1.7\times}$	0.3227	**0.3420**	0.6540	0.4197	0.3783	**0.5312**	0.5947	0.2742	0.7960	0.4792	26%
SynPro	48.0B	**34.1B** $_{\mathbf{5.2\times}}$	**0.3882**	**0.3420**	**0.6649**	**0.4284**	**0.3868**	0.5201	**0.6123**	**0.3278**	**0.8540**	**0.5027**	**79%**
400M model, 8B Chinchilla-optimal tokens, α=15% (1.2B available organic tokens)													
Unique	*61.2B*	*61.2B*	*0.3636*	*0.3600*	*0.6817*	*0.4284*	*0.4327*	*0.5351*	*0.6351*	*0.3211*	*0.8620*	*0.5133*	*100%*
Repeat	17.5B	7.9B $_{1.0\times}$	0.3309	0.3360	0.6420	0.4222	0.3750	0.5249	0.5456	0.2876	0.7960	0.4734	0%
QBSR	19.9B	8.6B $_{1.1\times}$	0.3268	0.3120	0.6474	0.4232	0.3806	0.5185	0.5825	0.3010	0.7980	0.4767	8%
MATES	19.7B	8.7B $_{1.1\times}$	0.3366	0.3260	0.6453	0.4263	0.3766	0.5288	0.5667	0.2910	0.7970	0.4771	9%
RePro	39.3B	17.9B $_{2.3\times}$	0.3497	0.3260	0.6561	0.4273	0.3998	0.5099	0.6263	0.3077	0.8170	0.4911	44%
SynPro	61.2B	**67.6B** $_{\mathbf{8.6\times}}$	**0.4054**	**0.3500**	**0.6687**	**0.4365**	**0.4167**	**0.5304**	**0.6596**	**0.3211**	**0.8690**	**0.5175**	**111%**
1.1B model, 22B Chinchilla-optimal tokens, α=10% (2.2B available organic tokens)													
Unique	*56.8B*	*56.8B*	*0.4062*	*0.3720*	*0.7095*	*0.4427*	*0.5261*	*0.5564*	*0.7193*	*0.3813*	*0.8910*	*0.5561*	*100%*
Repeat	39.3B	15.5B $_{1.0\times}$	0.3694	0.3420	0.6926	0.4304	0.4589	0.5304	0.6526	0.3344	0.8450	0.5173	0%
QBSR	43.7B	13.7B $_{0.9\times}$	0.3784	0.3340	0.6828	0.4268	0.4450	0.5233	0.6561	0.3311	0.8590	0.5152	−5%
MATES	42.6B	15.9B $_{1.0\times}$	0.3669	0.3500	0.6910	0.4340	0.4577	0.5367	0.6509	0.3278	0.8520	0.5185	3%
RePro	45.9B	21.7B $_{1.4\times}$	0.4029	0.3460	**0.6942**	0.4371	0.4893	**0.5446**	0.6649	0.3311	0.8770	0.5319	38%
SynPro	56.8B	**57.4B** $_{\mathbf{3.7\times}}$	**0.4586**	**0.3520**	0.6910	**0.4678**	**0.4917**	0.5391	**0.7018**	**0.4147**	**0.9090**	**0.5584**	**106%**
2B model, 40B Chinchilla-optimal tokens, α=10% (4B available organic tokens)													
Unique	*69.9B*	*69.9B*	*0.4169*	*0.3940*	*0.7345*	*0.4468*	*0.5648*	*0.5912*	*0.7368*	*0.4080*	*0.9070*	*0.5778*	*100%*
Repeat	52.4B	21.8B $_{1.0\times}$	0.3980	0.3700	0.7013	0.4268	0.5208	0.5430	0.6947	0.3746	0.8730	0.5447	0%
SynPro	69.9B	**73.9B** $_{\mathbf{3.4\times}}$	**0.4726**	**0.4160**	**0.7111**	**0.4570**	**0.5391**	**0.5699**	**0.7439**	**0.4415**	**0.9220**	**0.5859**	**124%**

Table 2: Data-bound pretraining results. Every method is trained to its own saturation point (§3.2), so **#Training Tokens** differ across methods. *Unique* denotes the oracle performance obtained by training on all unique tokens, i.e., non-data-bound regime. **Bold** and underline indicate the best and second-best results among methods with the same available organic corpus, so the *Unique* oracle is excluded.

5.1 Main results

Table 2 compares all methods across scales and data regimes. SynPro consistently outperforms all baselines. At 400M, it improves over Repeat by 9.6%, 7.5%, and 9.3% under α=5%, 10%, and 15%, respectively. As α increases, the gap between SynPro and Unique Data narrows; at α=15%, SynPro even surpasses Unique Data (0.5175 vs. 0.5133). The gains remain notable at the 1.1B scale, where SynPro outperforms Repeat by 7.9% and exceeds Unique Data (0.5584 vs. 0.5561) as well. The same pattern holds when scaling up to 2B, where SynPro improves over Repeat by 7.6% and again surpasses Unique Data (0.5859 vs. 0.5778), indicating that the benefits persist rather than diminish as model scale grows. Compared with RePro, SynPro is especially strong on world knowledge, as reformatting exposes factual content in structurally diverse forms and the influence reward prioritizes knowledge the model has not yet absorbed. These gains further extend beyond our evaluation suite, to held-out validation corpora as well as continuation and more challenging reasoning benchmarks (Appendix E.1). By contrast, QBSR and MATES yield modest gains, showing that selective repetition alone may not be particularly helpful in data-bound regimes.

Method	#Effective Tokens	Commonsense Reasoning				Language Understanding		World Knowledge				Rec.
		CSQA	OBQA	PIQA	SIQA	HellaSwag	WinoG	ARC-e	ARC-c	SciQ	Avg	Ratio
Unique	*48.0B*	*0.3686*	*0.3360*	*0.6790*	*0.4253*	*0.4303*	*0.5399*	*0.6579*	*0.3211*	*0.8510*	*0.5121*	*100%*
SynPro	**34.1B** $_{5.2\times}$	**0.3882**	0.3420	**0.6649**	**0.4284**	**0.3868**	0.5201	0.6123	**0.3278**	0.8540	**0.5027**	**79%**
w/o Rephrasing	14.3B $_{2.2\times}$	0.3759	0.3280	0.6551	0.4197	0.3518	0.5280	0.6123	0.2742	0.8270	0.4858	41%
w/o Reformatting	11.8B $_{1.8\times}$	0.3423	0.3200	0.6523	0.4191	0.3825	0.5272	0.5930	0.2809	0.8070	0.4805	29%
w/o Quality Reward	11.3B $_{1.7\times}$	0.3202	0.3220	0.6529	0.4089	0.3761	0.5036	0.5860	0.3110	0.8320	0.4792	26%
w/o Influence Reward	17.8B $_{2.7\times}$	0.3702	0.3340	0.6474	0.4253	0.3650	0.5091	0.6035	0.3043	**0.8580**	0.4908	52%
w/o Faithfulness Reward	13.8B $_{2.1\times}$	0.3202	0.3280	0.6415	0.4202	0.3595	0.5272	0.6158	0.2943	0.8570	0.4849	39%
w/o RL Training	13.8B $_{2.1\times}$	0.3260	**0.3440**	0.6404	0.4191	0.3537	0.5233	0.6105	0.3144	0.8320	0.4848	39%
w/o Data Merge	18.3B $_{2.8\times}$	0.3726	0.3300	0.6605	0.4227	0.3840	0.5107	0.6035	0.3110	0.8390	0.4927	57%
Nemotron-CC-HQ Prompt	15.1B $_{2.3\times}$	0.3292	0.3320	0.6211	0.4268	0.3682	0.5138	**0.6306**	0.3144	0.8490	0.4872	44%
Repeat	6.6B $_{1.0\times}$	0.3358	0.3140	0.6442	0.4135	0.3519	**0.5304**	0.5579	0.2676	0.7920	0.4675	0%

Table 3: Ablation study on the 400M/α=10% setting (0.8B organic tokens).

Furthermore, SynPro delivers substantially stronger data efficiency than all baselines. At 400M, the effective token multiplier grows from 3.4× at α=5% to 5.2× at α=10% and 8.6× at α=15%, where the effective token count even exceeds the actual training tokens. At 1.1B and 2B, the gain remains significant, with SynPro reaching 3.7× and 3.4× the effective tokens of Repeat, respectively. Across all settings, the effective tokens of SynPro are consistently around 3× those of RePro. These results show that SynPro generates more effective pretraining tokens from a limited organic corpus, enabling more efficient utilization of available data. These gains also persist after post-training (Appendix E.2), and the additional cost of our generation pipeline is analyzed in Appendix E.3.

5.2 Ablation studies

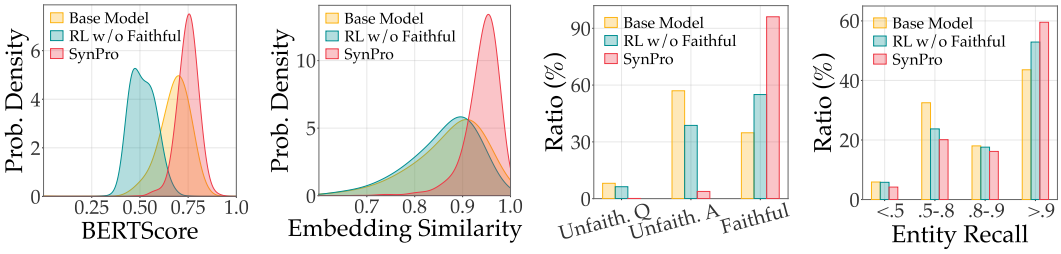
We perform ablation studies in the 400M setting. Removing reformatting causes significant drops in both commonsense reasoning (−2.6%) and world knowledge (−4.6%), as reformatted outputs may expose factual content in structured forms that reinforce both reasoning and knowledge learning. Removing rephrasing causes a notable drop in language understanding (−4.0% in HellaSwag), indicating that contextual comprehension benefits primarily from the lexical and structural diversity that rephrasings provide.

Without the quality reward, the commonsense reasoning drops the most (−3.7%), suggesting that it encourages coherent generation useful for commonsense tasks. Removing the influence reward produces a smaller but uniform drop across all categories, confirming its role in pushing generation toward content the model has yet to learn. We also apply the Nemotron-CC-HQ (Su et al., 2025) prompt to OLMo2-1B-Instruct. Despite its improvements on knowledge tasks, the overall performance remains below SynPro and the lack of faithfulness guarantees may introduce *distillation effects* where outputs reflect the generator's knowledge rather than the organic content (see §5.3). "w/o RL Training" prompts the base generator with the same two operations directly, which lowers the recovery ratio from 79% to 39%, showing that the two operations alone are insufficient and RL is essential to make the generator produce high-quality and faithful data. "w/o Data Merge" discards previous synthetic data and trains only on the latest generation, resulting in a 22% recovery ratio drop, suggesting that accumulating data across iterations is more beneficial. We further ablate the two models adopted in our generation pipeline, the semantic faithfulness reward and the generator initialization, in Appendix E.4.

5.3 Pointwise faithfulness

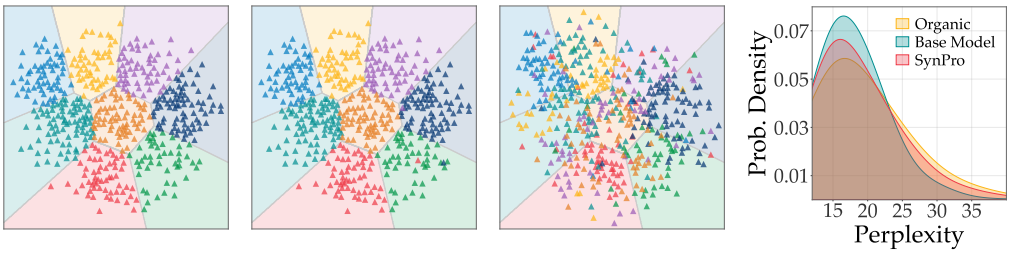
In this analysis, we validate whether SynPro preserves pointwise faithfulness to the organic data, which is critical for avoiding hallucinated content or distilled knowledge that undermines generalization (Yu & Xiong, 2025). We randomly sample 1,000 organic documents not used in RL training and apply each operation with three generators: the base model (OLMo2-1B-Instruct), RL without the faithfulness reward, and SynPro.

For rephrasing, we measure the semantic similarity between each organic document and the rephrasing generated from it. As shown in Figure 3a, SynPro achieves a mean BERTScore



(a) BERTScore (b) jina-embeddings (c) Classification (d) Named Entity Recall

Figure 3: Faithfulness analysis on 1,000 randomly sampled organic documents not seen in RL. For rephrasing: (a) BERTScore and (b) jina-embeddings similarity between original and rephrased text. For reformatting: (c) faithfulness classification and (d) named entity recall.



(a) Organic (b) SynPro (c) Base Model (d) Perplexity

Figure 4: Distribution preservation analysis. t-SNE illustration of Voronoi clusters, where each △ denotes (a) one organic source, (b) SynPro rephrasing, and (c) base-model rephrasing. (d) Perplexity distributions from the 400M model trained on 61.2B unique tokens.

of **0.75** with a tighter distribution, compared to 0.68 for the base model. Notably, RL without the faithfulness reward drops to 0.52, confirming that optimizing quality alone may harm faithfulness (Yu & Xiong, 2025). To verify generalization beyond BERTScore, we compute embedding similarity using *jina-embeddings-v5-text* (Akram et al., 2026) (Figure 3b). Our generator achieves the highest mean similarity (**0.94**) with the lowest variance among all baselines, confirming genuine semantic preservation rather than reward overfitting.

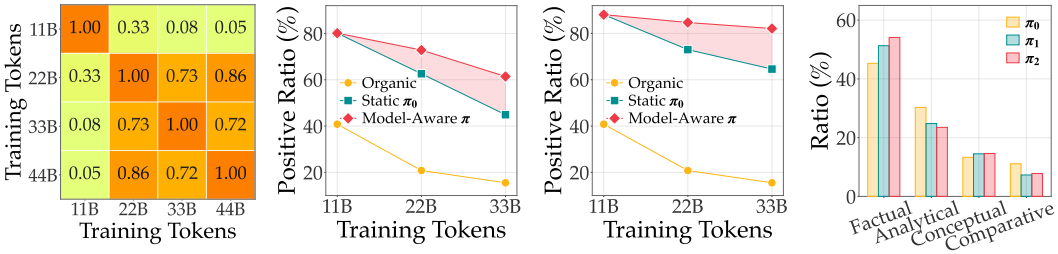
For reformatting, our fine-tuned faithfulness judge (§4) categorizes each output against its organic source as faithful (on-topic question with a correct answer), unfaithful (incorrect or unsupported) answer, or unfaithful (off-topic) question. As shown in Figure 3c, SynPro achieves a **96.1%** faithfulness rate, compared to only 34.8% for the base model and 55.0% for RL without the faithfulness reward. The base model produces unfaithful answers 57.0% of the time, indicating that without targeted training, the generator frequently hallucinates content beyond the source document. As an independent check, we extract named entities from the reformatted text with *BERT-base-NER* (Devlin et al., 2019) and compute recall against the original. As shown in Figure 3d, **76%** of our samples exceed 80% recall versus 62% for the base model, confirming that faithfulness generalizes beyond the reward metric.

In summary, the faithfulness reward improves reward-aligned metrics and generalizes to independent evaluations, confirming that SynPro produces faithful synthetic data.

5.4 Distribution preservation

Beyond pointwise faithfulness, we examine whether the overall distribution of synthetic data preserves the characteristics of the organic corpus, which is critical for preventing model collapse and ensuring meaningful performance gains (Shumailov et al., 2024).

We first examine the preservation of the semantic distribution. We embed 500 organic and rephrased texts with *jina-embeddings-v5-text*, apply k-means (k=8), and visualize the resulting Voronoi regions on the t-SNE projections. SynPro (Figure 4b) closely resembles the organic distribution (Figure 4a) and retains 99.2% of cluster assignments, confirming that



(a) Influence Correlation (b) Influence: Rephrase (c) Influence: Reformat (d) Reformat Type

Figure 5: Model-awareness analysis on the 1.1B model. (a) Influence correlation and (b, c) positive influence ratio over pretraining. (d) Reformat type distribution across iterations.

our rephrasings preserve the semantics well. By contrast, the base model (Figure 4c) retains only 42.2%, with triangles scattered across mismatched regions, showing that without the faithfulness constraint, the generator shifts the semantic distribution substantially.

We then focus on diversity preservation. Following Shumailov et al. (2024), we compute the perplexity of organic and synthetic text from the 400M oracle model trained on 61.2B unique tokens. As shown in Figure 4d, SynPro stays much closer to the organic perplexity distribution than the base model, which produces a tighter, lower-perplexity shape that does not capture the long tail of organic data. This indicates that SynPro better preserves the diversity of the organic corpus rather than collapsing toward more predictable text. Nevertheless, a smaller gap to the organic distribution still remains, as model-generated text may carry intrinsic characteristics that narrow diversity, and we leave further exploration of this phenomenon to future work.

5.5 Model-awareness analysis

Finally, we analyze how the influence reward shapes the generator's output across iterations. First, we compute influence on the initial training data $\mathcal{D}_{\text{org}} \cup \mathcal{D}^0_{\text{syn}}$. As shown in Figure 5a, the model's preferences shift substantially early in pretraining, as reflected by the low influence correlation between 11B and later checkpoints. 44B has a high influence correlation with 22B, reflecting that our model has largely plateaued. We further track the positive influence ratio (fraction of samples with positive data influence, Eq. 14) for both the static (π_0) and model-aware (π_1, π_2) policies. Figures 5b and 5c show that both organic and static synthetic data's positive influence ratios drop rapidly as the model memorizes the repeated corpus. By contrast, our model-aware policy maintains a consistently higher ratio after each update, as the refreshed generator targets content the model has yet to learn, explaining the sustained performance gains of SynPro over static approaches.

To examine how the generator adapts its outputs across iterations, we classify reformatting types via Gemini 3.1 Flash-Lite: *factual*, *analytical*, *conceptual*, and *comparative*. Figure 5d shows that from π_0 to π_2, the factual proportion increases from 45.3% to 54.1%, while analytical and comparative outputs decrease. This suggests the influence reward steers the generator toward the model's factual gaps. Appendix F shows cases to support this claim.

6 Conclusion

In this paper, we introduce SynPro, an effective synthetic data generation framework for the data-bound scaling regime. Our results highlight two key insights. First, organic data is underutilized, not exhausted. SynPro helps the pretraining model more thoroughly learn from a limited organic corpus, matching performance achieved with much more unique data. Second, faithfulness is essential for synthetic pretraining data, as grounding outputs in the organic source enriches rather than distorts the training distribution, while unconstrained generation risks collapse and distillation. We hope SynPro motivates future work to break the data wall and sustain LLM scaling in the data-bound regime.

Acknowledgments

We thank Amazon for funding Zichun Yu through the Amazon AI Ph.D. Fellowship Program. We thank CMU Foundation and Language Model (FLAME) Center for providing support of computational resources.

Ethics statement

We use publicly available data and models, and we do not foresee significant ethical concerns specific to this work beyond those already associated with language model pretraining and synthetic data generation. However, as with other forms of model-generated content, synthetic data may reflect biases introduced by the generator itself. In our approach, we explicitly incorporate faithfulness objectives into the reward design to mitigate such effects and encourage generated data to closely preserve the distribution of the original data. While this does not fully eliminate all risks associated with model-generated text, it provides a principled mechanism for reducing unintended distortions during synthetic data generation and helps maintain consistency between synthetic and original data distributions.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *ArXiv preprint*, 2024.

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. jina-embeddings-v5-text: Task-targeted embedding distillation. *ArXiv preprint*, 2026.

Loubna Ben Allal, Anton Lozhkov, and Daniel van Strien. Cosmopedia: How to create large-scale synthetic data for pre-training. Hugging Face Blog, 2024.

Baolong Bi, Shenghua Liu, Xingzhang Ren, Dayiheng Liu, Junyang Lin, Yiwei Wang, Lingrui Mei, Junfeng Fang, Jiafeng Guo, and Xueqi Cheng. RefineX: Learning to refine pre-training data at scale from expert-guided programs. *ArXiv preprint*, 2025.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proc. of AAAI*, 2020.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abdin. On the diversity of synthetic data and its impact on training large language models. *ArXiv preprint*, 2024.

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*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, 2019.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan SU, Markus Kliegl, ZIJIA CHEN, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. Nemotron-CLIMB: Clustering-based iterative data mixture bootstrapping for language model pre-training. In *Proc. of NeurIPS*, 2025.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *Proc. of ICML*, 2024.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong model collapse. In *Proc. of ICLR*, 2025.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. DsDm: Model-aware dataset selection with datamodels. In *Proc. of ICML*, 2024.

Michael C Frank. Bridging the data gap between children and large language models. *Trends in Cognitive Sciences*, 2023.

Xintong Hao, Ruijie Zhu, Ge Zhang, Ke Shen, and Chenggang Li. Reformulation for pretraining data augmentation. *ArXiv preprint*, 2025.

Alex Havrilla, Andrew Dai, Laura O'Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, Fabrizio Milo, Sharath Chandra Raparthy, Kanishk Gandhi, Baber Abbasi, et al. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models. *ArXiv preprint*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proc. of ICLR*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Proc. of NeurIPS*, 2022.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *Proc. of COLM*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv preprint*, 2020.

Myeongseob Ko, Feiyang Kang, Weiyan Shi, Ming Jin, Zhou Yu, and Ruoxi Jia. The mirrored influence hypothesis: Efficient data influence estimation by harnessing forward passes. In *Proc. of CVPR*, 2024.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proc. of ICML*, 2017.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proc. of SOSP*, 2023.

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. In *Proc. of NeurIPS*, 2024.

Xiaochuan Li, Zichun Yu, and Chenyan Xiong. Montessori-Instruct: Generate influential training data tailored for student learning. In *Proc. of ICLR*, 2025.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *ArXiv preprint*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *ArXiv preprint*, 2024.

Kairong Luo, Zhenbo Sun, Xinyu Shi, Shengqi Chen, Bowen Yu, Yunyi Chen, Chenyi Dang, Hengtao Tao, Hui Wang, Fangming Liu, et al. PCMind-2.1-Kaiyuan-2B technical report. *ArXiv preprint*, 2025.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proc. of ACL*, 2024.

Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, et al. BeyondWeb: Lessons from scaling synthetic data for trillion-scale pretraining. *ArXiv preprint*, 2025.

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 *Proc. of EMNLP*, 2018.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. In *Proc. of NeurIPS*, 2023.

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. Recycling the web: A method to enhance pre-training data quality and quantity for language models. In *Proc. of COLM*, 2025.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proc. of ACL*, 2016.

Ru Peng, Kexin Yang, Yawen Zeng, Junyang Lin, Dayiheng Liu, and Junbo Zhao. DataMan: Data manager for pre-training large language models. In *Proc. of ICLR*, 2025.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proc. of AAAI*, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proc. of AAAI*, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proc. of EMNLP*, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *ArXiv preprint*, 2024.

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

Tao Shen, Didi Zhu, Ziyu Zhao, Zexi Li, Chao Wu, and Fei Wu. Will LLMs scaling hit the wall? breaking barriers via distributed resources on massive edge devices. *ArXiv preprint*, 2025.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 2024.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CC: Transforming common crawl into a refined long-horizon pretraining dataset. In *Proc. of ACL*, 2025.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proc. of NAACL-HLT*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NeurIPS*, 2017.

Pablo Villalobos and Anson Ho. Trends in training dataset sizes. Epoch AI Blog, 2022.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data. In *Proc. of ICML*, 2024.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, et al. 2 OLMo 2 furious. In *Proc. of COLM*, 2025.

Jiachen T Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for llm training in every iteration. In *Proc. of NeurIPS*, 2024.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell (eds.), *Proc. of the BabyLM / CoNLL*, 2023.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *Proc. of ICLR*, 2022.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Workshop on Noisy User-generated Text*, 2017.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *ArXiv preprint*, 2025.

Zichun Yu and Chenyan Xiong. RePro: Training language models to faithfully recycle the web for pretraining. *ArXiv preprint*, 2025.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *Proc. of NeurIPS*, 2024.

Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wen-tau Yih, and Chenyan Xiong. Group-level data selection for efficient pretraining. In *Proc. of NeurIPS*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proc. of ACL*, 2019.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with bert. In *Proc. of ICLR*, 2020.

Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. Beyond random sampling: Efficient language model pretraining via curriculum learning. In *Proc. of EACL*, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *ArXiv preprint*, 2025.

Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. Programming every example: Lifting pre-training data quality like experts at scale. In *Proc. of ICML*, 2025.

Appendix Table of Contents

A Disclosure of LLM usage **15**

B SynPro algorithm **15**

C Derivation of the influence approximation **15**

D Experimental details **16**

E Additional results **17**

E.1 Generalization results 17

E.2 Compatibility with post-training 19

E.3 Compute cost 19

E.4 Ablation of models in the generation pipeline 19

E.5 High quality of generated synthetic data 20

E.6 RL training dynamics 20

F Case study **21**

G Prompts **26**

A Disclosure of LLM usage

We truthfully disclose the following use of LLMs in this work. First, LLMs are used as reward models in our method, including quality- and faithfulness-related scoring components. Second, LLM assistance was used for code implementation and figure scripting. Third, LLM assistance was used to support literature search and review, as well as to help draft parts of the paper text; however, references were added manually by the authors. All experiments were run manually by the authors, and all analyses were performed and verified manually. LLMs were not used to originate the core research ideas.

B SynPro algorithm

SynPro algorithm is summarized in Algorithm 1.

C Derivation of the influence approximation

Following MATES (Yu et al., 2024), we first formulate the oracle data influence of a sample x by the change in reference loss after training on x:

$$\mathcal{I}(x \mid \mathcal{M}_i^*) = \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{A}(\mathcal{M}_i^*, x)) - \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}_i^*). \tag{17}$$

Define

$$g_x = \nabla_{\mathcal{M}} \mathcal{L}(x \mid \mathcal{M}_i^*), \tag{18}$$

$$g_{\text{ref}} = \nabla_{\mathcal{M}} \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}_i^*). \tag{19}$$

Approximating one update on x and one update on $\mathcal{D}_{\text{ref}}$ by gradient steps with step size η,

$$\mathcal{A}(\mathcal{M}_i^*, x) \approx \mathcal{M}_i^* - \eta g_x, \tag{20}$$

$$\mathcal{A}(\mathcal{M}_i^*, \mathcal{D}_{\text{ref}}) \approx \mathcal{M}_i^* - \eta g_{\text{ref}}. \tag{21}$$

Algorithm 1 SynPro: Model-Aware Synthetic Data Generation

Require: Organic corpus $\mathcal{D}_{\text{org}}$, reference set $\mathcal{D}_{\text{ref}}$, initial generation policy π_0

// Initialization

$\mathcal{D}^0_{\text{syn}} \leftarrow \{\pi_0(p, x) \mid x \in \mathcal{D}_{\text{org}}\}$

$\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{org}} \cup \mathcal{D}^0_{\text{syn}}$

$\mathcal{M}^*_0 \leftarrow$ random initialization

for $i = 1, 2, \ldots$ **do**

  // Stage 1: LM Pretraining

  $\mathcal{M}_i \leftarrow \mathcal{M}^*_{i-1}$, $\mathcal{L}_0 \leftarrow \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}^*_{i-1})$

  for $t = 1, 2, \ldots$ **do**

    $\mathcal{M}'_i \leftarrow \mathcal{A}(\mathcal{M}_i, \mathcal{D}_{\text{train}})$

    $\mathcal{L}_t \leftarrow \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}'_i)$

    if $t > 1$ and $\mathcal{L}_t \geq \min(\mathcal{L}_{t-1}, \mathcal{L}_{t-2})$ **then**

      break

    end if

    $\mathcal{M}_i \leftarrow \mathcal{M}'_i$

  end for

  $\mathcal{M}^*_i \leftarrow \mathcal{M}_i$

  if $\mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}^*_i) \geq \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}^*_{i-1})$ **then**

    break

  end if

  // Stage 2: Policy update

  $r_i(x, \tilde{x}) \leftarrow \lambda_{\text{quality}}\, r_{\text{quality}}(\tilde{x}) + \lambda_{\text{faithful}}\, r_{\text{faithful}}(x, \tilde{x}) + \lambda_{\text{influence}}\, r_{\text{influence}}(\tilde{x} \mid \mathcal{M}^*_i)$

  $\pi_i \leftarrow \arg\max_\pi \ \mathbb{E}_{x \sim \mathcal{D}_{\text{org}}, \tilde{x} \sim \pi(\cdot \mid p, x)}[r_i(x, \tilde{x})]$

  // Stage 3: Generation of new synthetic data

  $\mathcal{D}^i_{\text{syn}} \leftarrow \{\pi_i(p, x) \mid x \in \mathcal{D}_{\text{org}}\}$

  $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}^i_{\text{syn}}$

end for

and using first-order Taylor expansion around $\mathcal{M}^*_i$,

$$\begin{aligned}
\mathcal{I}(x \mid \mathcal{M}^*_i) &= \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{A}(\mathcal{M}^*_i, x)) - \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}^*_i) \\
&\approx \nabla_{\mathcal{M}} \mathcal{L}(\mathcal{D}_{\text{ref}} \mid \mathcal{M}^*_i)^\top \left(\mathcal{A}(\mathcal{M}^*_i, x) - \mathcal{M}^*_i\right) \\
&\approx -\eta\, g_x^\top g_{\text{ref}} \\
&\approx -\nabla_{\mathcal{M}} \mathcal{L}(x \mid \mathcal{M}^*_i)^\top \left(\mathcal{M}^*_i - \mathcal{A}(\mathcal{M}^*_i, \mathcal{D}_{\text{ref}})\right) \\
&\approx \mathcal{L}(x \mid \mathcal{A}(\mathcal{M}^*_i, \mathcal{D}_{\text{ref}})) - \mathcal{L}(x \mid \mathcal{M}^*_i).
\end{aligned} \tag{22}$$

which matches the mirrored influence view of Forward-INF (Ko et al., 2024). The practical benefit is that we only need one update on $\mathcal{D}_{\text{ref}}$ to form $\mathcal{A}(\mathcal{M}^*_i, \mathcal{D}_{\text{ref}})$, after which scoring each candidate sample only requires evaluating $\mathcal{L}(x \mid \mathcal{M}^*_i)$ and $\mathcal{L}(x \mid \mathcal{A}(\mathcal{M}^*_i, \mathcal{D}_{\text{ref}}))$, i.e., forward inference only. where we take the negative oracle as the influence reward used in §3.3 since higher reward should denote better data:

$$r_{\text{influence}}(x \mid \mathcal{M}^*_i) = \mathcal{L}(x \mid \mathcal{M}^*_i) - \mathcal{L}(x \mid \mathcal{A}(\mathcal{M}^*_i, \mathcal{D}_{\text{ref}})) \approx -\mathcal{I}(x \mid \mathcal{M}^*_i). \tag{23}$$

D Experimental details

We provide training details in Table 4 and implementation details of SynPro in Table 5.

Hyperparameter	400M LM	1.1B LM	Synthetic Data Generator
Steps	-	-	600
Batch size	512	512	24
Sequence length	2048	2048	4096 (2048i + 2048o)
Max learning rate	6e-4	4e-4	1e-6
Optimizer	AdamW	AdamW	AdamW
Scheduler	WSD	WSD	Cosine

Table 4: Training details.

Component	Setting
Reference set $\mathcal{D}_{\text{ref}}$	FLAN (Wei et al., 2022)
Initial policy π_0	OLMo2-1B-Instruct
Quality reward model	DataMan (Peng et al., 2025)
Rephrasing faithfulness model	Qwen3-1.7B (Yang et al., 2025)
Rephrasing faithfulness method	Few-shot prompting for structural faithfulness
Reformat faithfulness model	Qwen3-1.7B
Reformat faithfulness method	Fine-tuned on 10,000 distilled from Gemini 3.1 Flash-Lite
τ_{sem}	0.65
τ_{len}	1.25
λ_{quality}	1
$\lambda_{\text{faithful}}$	1
$\lambda_{\text{influence}}$	3
Iterations to converge	3
Generator training algorithm	GRPO (Shao et al., 2024)
GRPO clipping ϵ	0.2
GRPO β	0.005
GRPO rollouts per input	8
Generator inference framework	vllm (Kwon et al., 2023)
Decoding temperature	1.0
Decoding top-p	0.9

Table 5: Implementation details of SynPro.

E Additional results

In this section, we report generalization results (§E.1), verify compatibility with post-training (§E.2), discuss compute cost (§E.3), ablate the models used in our generation pipeline (§E.4), analyze the quality of our synthetic data (§E.5), and examine RL training dynamics (§E.6).

E.1 Generalization results

To verify that the gains of SynPro come from genuinely better learning, we further evaluate the 1.1B models on held-out validation corpora and more downstream benchmarks (Table 6).

We first report validation loss on three widely used corpora, DCLM-val (in-distribution), C4-val and Pile-val (out-of-distribution). SynPro achieves the lowest loss on all three corpora, reducing DCLM-val from 2.855 to **2.771**, C4-val from 3.359 to **3.242**, and Pile-val from 2.923 to **2.772** relative to Repeat. These results indicate that SynPro helps the model better learn and generalize from limited organic data.

Method	Validation Corpora			Generalization Tasks			
	DCLM-val (CE)↓	C4-val (CE)↓	Pile-val (CE)↓	LAMBADA (CE)↓	LAMBADA (Acc)↑	COPA↑	MMLU↑
Repeat	2.855	3.359	2.923	0.6645	0.3577	0.6300	0.3049
RePro	2.815	3.299	2.849	0.6136	0.3771	0.7200	0.3157
SynPro	**2.771**	**3.242**	**2.772**	**0.5823**	**0.4007**	**0.7300**	**0.3326**

Table 6: Generalization results on the 1.1B/α=10% setting. CE denotes cross-entropy loss (lower is better) and Acc denotes accuracy (higher is better). MMLU is evaluated with 5 shots. DCLM-val is in-distribution, while C4-val and Pile-val are out-of-distribution.

Method	*Commonsense Reasoning*				*Language Understanding*		*World Knowledge*			Avg
	CSQA	OBQA	PIQA	SIQA	HellaSwag	WinoG	ARC-e	ARC-c	SciQ	
Repeat	0.369	0.342	0.693	0.430	0.459	0.530	0.653	0.334	0.845	0.517
+ SFT	0.417	0.346	0.693	0.451	0.453	0.534	0.665	0.341	0.864	0.529
RePro	0.403	0.346	0.694	0.437	0.489	0.545	0.665	0.331	0.877	0.532
+ SFT	0.433	0.334	0.699	0.468	0.481	0.559	0.668	0.348	0.892	0.542
SynPro	0.459	**0.352**	0.691	0.468	**0.492**	0.539	0.702	**0.415**	0.909	0.558
+ SFT	**0.493**	0.350	**0.700**	**0.472**	0.491	**0.563**	**0.714**	0.391	**0.923**	**0.566**

Table 7: Downstream accuracy before and after supervised fine-tuning (SFT) on FLAN (Wei et al., 2022), on the 1.1B/α=10% setting.

The gains of SynPro also generalize well to continuation tasks such as LAMBADA (Paperno et al., 2016) and COPA (Roemmele et al., 2011), even though these are not directly targeted by our synthetic data operation. SynPro consistently improves over Repeat and RePro on both tasks, reducing LAMBADA cross-entropy from 0.6645 to **0.5823** and improving COPA accuracy from 63.0% to **73.0%** relative to Repeat. These gains indicate that the benefits of SynPro extend beyond the evaluation tasks in Table 2, improving broader contextual prediction and causal reasoning rather than merely fitting the specific formats used during synthetic data generation.

We finally evaluate our 1.1B checkpoints on 5-shot MMLU (Hendrycks et al., 2021), a widely used and challenging knowledge-reasoning benchmark. SynPro improves over Repeat by **2.8%** absolute, doubling the gain of RePro, our strongest baseline. These gains indicate that SynPro further strengthens the model's capability on more complex reasoning tasks.

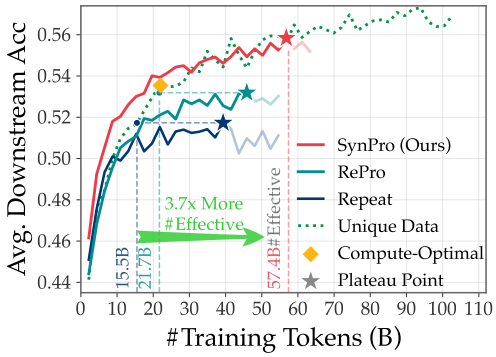


Figure 6: 1B model & 2.2B unique organic tokens

E.2 Compatibility with post-training

To examine whether the benefits of better pretraining data persist after post-training, we fine-tune the 1.1B checkpoints of Repeat, RePro, and SynPro on FLAN, a standard open-source SFT dataset. As shown in Table 7, SynPro retains a **3.7%** absolute gain over Repeat after SFT, close to the **4.1%** gain before SFT, and more than doubles the gain of RePro (1.3%). These results indicate that the advantage of our synthetic data is not washed out by post-training, and that a stronger pretrained backbone remains valuable after SFT.

E.3 Compute cost

Table 8 reports the H100 GPU hours for each method under the same number of training tokens. Compared with simple repetition, SynPro incurs additional synthesis cost, but the *relative* overhead decreases as the pretraining model scales: the total cost is $13.6/2.7 \approx 5.0\times$ Repeat at 400M, but only $49.0/19.2 \approx 2.6\times$ at 1.1B. This trend is expected because pretraining becomes increasingly dominant at larger model sizes, while our synthesis pipeline relies on a relatively lightweight 1B model, so the extra cost of our method is progressively amortized.

Table 9 further decomposes the total cost of SynPro over the whole pipeline into generator RL training, generator inference, and language model pretraining. As the model size scales up, pretraining increasingly dominates the total cost (from 73.3% at 400M to 80.8% at 1.1B), so our entire data generation pipeline takes only 26.7% of the total cost at 400M and 19.2% at 1.1B. More importantly, our focus is data-bound pretraining, where naively spending compute on repeated passes over the same small organic corpus quickly hits the data wall and yields diminishing returns. In this regime, paying moderate additional compute to generate grounded and model-aware synthetic data is acceptable, since it unlocks substantially better potential of the limited organic data and enables significant gains over traditional repetition.

	400M Model (Per 0.8B Organic Tokens)				1.1B Model (Per 2.2B Organic Tokens)			
Method	**Synthesis**	**Selection**	**Pretraining**	**Total**	**Synthesis**	**Selection**	**Pretraining**	**Total**
Repeat	–	–	2.7	2.7	–	–	19.2	19.2
QBSR	–	0.2	2.7	2.9	–	0.6	19.2	19.8
MATES	–	0.3	2.7	3.0	–	1.2	19.2	20.4
RePro	7.8	–	2.7	10.5	21.3	–	19.2	40.5
SynPro	10.9	–	2.7	13.6	29.8	–	19.2	49.0

Table 8: H100 GPU hours breakdown by method. Hours are computed per unit of organic data (0.8B tokens for 400M, 2.2B tokens for 1.1B).

Setup	Generator Training	Generator Inference	Pretraining	Total
400M/α=10%	26 (11.8%)	33 (14.9%)	162 (73.3%)	221 (100%)
1.1B/α=10%	29 (4.7%)	89 (14.5%)	496 (80.8%)	614 (100%)

Table 9: Decomposition of the total H100 GPU hours of SynPro, summed over all three iterations. Percentages denote the share of the total cost.

E.4 Ablation of models in the generation pipeline

Our generation pipeline involves two key models, BERTScore (Zhang et al., 2020) as the semantic faithfulness reward and OLMo2-1B-Instruct as the initialization of the generator. We ablate both choices in Table 10.

Method	#Effective Tokens	Commonsense Reasoning				Language Understanding		World Knowledge				Rec.
		CSQA	OBQA	PIQA	SIQA	HellaSwag	WinoG	ARC-e	ARC-c	SciQ	Avg	Ratio
SynPro	**34.1B** 5.2×	0.3882	**0.3420**	**0.6649**	0.4284	0.3868	0.5201	0.6123	**0.3278**	**0.8540**	**0.5027**	**79%**
w/ Qwen3-Embedding	33.8B 5.1×	**0.3939**	0.3360	0.6610	0.4278	**0.3941**	0.5264	0.6070	0.3211	0.8480	0.5017	77%
w/ Self-Generation	32.0B 4.8×	0.3776	0.3400	0.6420	**0.4289**	0.3726	**0.5304**	**0.6404**	0.3043	0.8450	0.4979	68%
Repeat	6.6B 1.0×	0.3358	0.3140	0.6442	0.4135	0.3519	**0.5304**	0.5579	0.2676	0.7920	0.4675	0%

Table 10: Ablation of the models adopted in our generation pipeline, on the 400M/α=10% setting. "w/ Qwen3-Embedding" replaces BERTScore with Qwen3-Embedding-0.6B similarity as the semantic faithfulness reward, and "w/ Self-Generation" initializes the generator from our own pretrained checkpoint instead of OLMo2-1B-Instruct.

Replacing BERTScore with the embedding similarity from Qwen3-Embedding-0.6B (Zhang et al., 2025) yields nearly identical performance (0.5017 vs. **0.5027**), indicating that our faithfulness reward is robust to the choice of embedding backbone.

We further train the generator from our own pretrained model, by instruction-tuning the 400M checkpoint of the Repeat baseline on FLAN (Wei et al., 2022) and applying the same RL pipeline, which forms a closed-loop "self-generation" setup. This variant still substantially outperforms Repeat (**0.4979** vs. 0.4675) and approaches the generator initialized from OLMo2, confirming that the gains of SynPro still hold without an external generator.

E.5 High quality of generated synthetic data

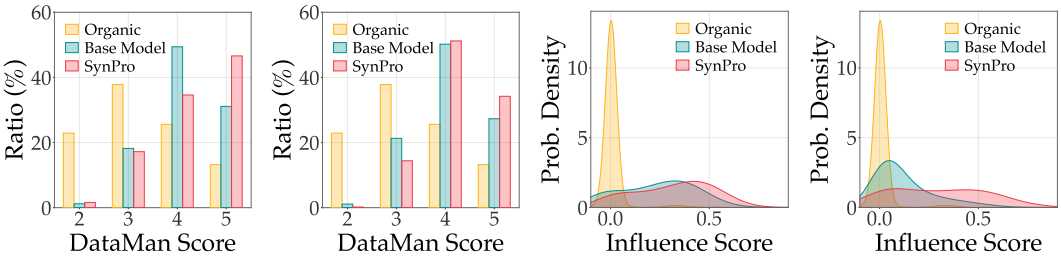
We evaluate the quality of generated synthetic data along two dimensions: intrinsic text quality via DataMan (Peng et al., 2025) scores and benefits to the pretraining model via data influence scores. We use the same 1,000 sampled documents as in §5.3 and compare three conditions: organic data, the base generator, and SynPro (using the π_1 checkpoint from the 400M setting at 21.8B training tokens). Influence scores are computed following Eq. 14, using pretraining checkpoints at 21.8B training tokens as well.

DataMan score. Figures 7a and 7b show the DataMan score distributions for rephrasing and reformatting, respectively. SynPro substantially improves over the organic data (mean 3.28), which achieves a mean score of **4.26** for rephrasing and **4.19** for reformatting. In comparison, the base generator achieves a mean score of 4.10 for rephrasing and 4.04 for reformatting. The shift toward higher scores (particularly 5) confirms that our quality reward drives the generator to produce more coherent and well-structured text while maintaining faithfulness.

Data influence. Figures 7c and 7d show the influence score distributions. Organic data has near-zero mean influence (0.01), reflecting that the pretraining model has already absorbed most of organic content through repeated exposure. For rephrasing, SynPro achieves a mean influence of **0.32**, a moderate improvement compared to 0.28 from the base generator since rephrasings largely preserve the original content. The effect is more pronounced for reformatting, where SynPro reaches a mean of **0.29**, doubling the base generator's 0.12. This gap arises as the reformatting operation produces structurally novel outputs (e.g., QA pairs, reasoning traces) that present familiar content in forms the model has not yet seen, and the influence reward further steers generation toward content the model finds most informative. We also find the influence score to be nearly uncorrelated with the semantic faithfulness score on these documents (Pearson -0.09, Spearman -0.09), indicating that the two rewards capture complementary rather than redundant signals.

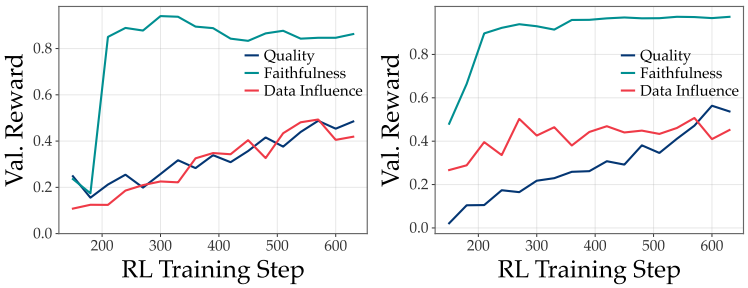
E.6 RL training dynamics

Figure 8 shows example validation reward curves during RL training for the π_1 generator in the 400M setting. A first observation is that the three rewards can be improved jointly rather than trading off sharply against one another. For both rephrasing and reformatting, the quality, faithfulness, and data influence rewards all rise substantially over training and



(a) Rephrase:DataMan (b) Reformat:DataMan (c) Rephrase:Influence (d) Reformat:Influence

Figure 7: Quality analysis on 1,000 randomly sampled organic documents not seen in RL. (a, b) DataMan score and (c, d) data influence score for rephrasing and reformatting, respectively. SynPro uses π_1 checkpoint from the 400M setting at 22B training tokens.



(a) Rephrasing reward curves (b) Reformat reward curves

Figure 8: Example validation reward (400M pretraining model, π_1 generator) curves during RL training for (a) rephrasing and (b) reformatting.

remain high near convergence, indicating that the generator can simultaneously become more coherent, more source-grounded, and more useful to the current pretraining model.

Second, the faithfulness reward increases especially quickly in the early stage of training and reaches a high level well before convergence. Intuitively, faithfulness is a relatively easy signal to optimize early on, as the generator quickly learns to stay close to the source document and avoid unsupported generations, after which later training focuses more on improving quality and informativeness.

Finally, our reward coefficients achieve a balance among these objectives. As described in §4, we set $\lambda_{\text{quality}} = 1$, $\lambda_{\text{faithful}} = 1$, and $\lambda_{\text{influence}} = 3$, so that the combined contribution of the quality and data influence rewards is on a similar scale to the faithfulness reward. This setting prevents the policy from over-optimizing faithfulness alone, achieving a practical balance between faithfulness and informative synthetic data generation. We tuned these coefficients by validation trends, but this is a common one-time overhead in RL. For other RL hyperparameters, we follow best practices in RePro (Yu & Xiong, 2025).

F Case study

In this section, we present representative examples showing how SynPro transforms organic data into faithful and model-aware synthetic data.

The rephrasing comparison highlights the effect of the faithfulness reward more directly. Relative to the rephrasing of base model (OLMo2-1B-Instruct), both π_0 and π_1 stay much closer to the source content and preserve the original instructional structure, instead of drifting into generic advice or summary-style text. Among them, π_1 is more factually abundant than π_0, consistent with the additional effect of the data influence reward.

We next turn to reformatting cases, where the blue spans mark source-grounded evidence that directly supports the reformatted outputs. In both examples, all shown outputs remain

grounded in the source, but the initial policy π_0 tends to produce broader and more surface-level reformats, while later model-aware policies place greater emphasis on advanced factual content that helps the model improve. The intermediate policy π_1 already moves in this direction, but π_2 makes the shift more pronounced.

Specifically, in Case 1, π_0 mostly asks local identification questions (e.g., "saturation," "gaze confirmation," and "Fitts Law"), whereas π_2 shifts toward more concrete takesaways about VR stress, trauma, and ethical concerns. In Case 2, π_0 already extracts several factual items, but they remain relatively shallow and loosely organized. π_1 increases specificity, while π_2 further concentrates on explicit factual content about the Delphi method, including its origin, purpose, and questionnaire-based aggregation. Importantly, Case 2 contains two largely independent pieces of information, one about the Delphi method and one about the militia system, yet the generator is able to reformat salient content from both sources rather than collapsing onto only one topic. Together, these cases strongly support the trend in Figure 5d, where the generator shifts away from broader, surface-level reformats and toward more factual, source-grounded outputs across policy updates.

SynPro: Rephrasing Case

Organic (truncated)

you want to get into the action and show up during my ACT. But you need to leave for a while. We can talk later." By naming the monster, you can help tame the monster. Creating some cognitive distance from the anxious thoughts allows you to achieve a measure of control over them. 13. Imagined practice makes perfect: If you have had many experiences of anxiety during tests, it may help to visualize yourself taking a test without suffering the effects of anxiety. It's a practice known as cognitive rehearsal, or guided imagery, taking a mental walk-through in advance of a performance event. Athletes do it all the time, imagining themselves performing at their peak level, in advance of the high-stakes event. Imagine yourself walking through the test, missing problems and staying calm and centered: Lay down a new template of you as a peak-tester, and make the images as vivid and sensorily rich as possible, so your mind believes them. Imagined practice can be as powerful as actual practice. 14. Regulate your breathing: Deep diaphragmatic breathing is a powerful anxiety-reducing technique because it activates the body's relaxation response. Breathing from the diaphragm, in a slow measured way, filling the stomach then the chest, stimulates the vagus nerve, which controls the parasympathetic (calming) nervous system. Stimulating the vagus nerve leads to a decrease in heart rate, blood pressure and other sympathetic responses. Practice breathing deeply and consciously, multiple times per day, and this technique will be available to you on test day. 15. Imagine you are breathing into your heart center: There's a simple technique I use when I'm nervous, where I place my hand over my heart, and take deep breaths, focusing my energy on my heart. This is a technique I learned from an organization called Heartmath. The act of combining slow breathing, a nurturing gesture, and redirecting my consciousness to my heart, helps restore a sense of calm.

OLMo2-1B-Instruct Rephrasing (truncated)

In order to take part in the ACT, you must first excuse yourself temporarily. Discussing it later, however, will be beneficial. Through naming the monster, one can control it effectively. Preparing mentally for the exam without being overwhelmed by anxiety is similar to imagining yourself performing well and staying composed during the test. Athletes regularly employ such practices, visualizing themselves succeeding. This mental rehearsal can be just as effective as actual study. Here is a paraphrased version: Maintain a balance in your anxiety before the ACT, by stepping away and discussing later. Dealing with anxiety mentally can make you more in control of it. By acknowledging the presence of the monster, you're more likely to manage it. Imagining yourself as calm and successful during the test can prepare you effectively. Regular exercises like deep breathing, focusing on your heart, visu-

alizing in a safe way, regular physical activity, ensuring you sleep well, practicing relaxation techniques, and engaging in conversations or even meditating can all help reduce anxiety. These strategies are applicable across various aspects of life and education, emphasizing the importance of self-regulation. It's important to note that the listed techniques cover various facets of mental health management, from physical relaxation to psychological coping strategies, and their general applicability extends far beyond the context of an ACT test. The goal is to build resilience and emotional regulation abilities which are universally useful.

π_0 **Rephrasing (truncated)**

You want to engage actively and participate during my ACT. However, you must temporarily disconnect for discussion. We can discuss later. By labeling the monster, you can help mitigate its effects. Creating some mental distance from anxious thoughts assists in gaining control over them. 13. Imagined rehearsal brings perfection practice: If you have frequently experienced anxiety during exams, envisioning yourself performing without experiencing anxiety-induced discomfort can serve as a practice known as cognitive rehearsal or guided imagery. Athletes regularly engage in this technique by visualizing themselves at their peak levels before performance events. Imagine walking through the exam, missing questions and maintaining composure and centering: Develop a new template of yourself as a high-performing test-taker, enhancing your mental imagery vividly and sensorially. 14. Control breathing: Deep diaphragmatic breathing is a potent anxiety-reducing method due to its stimulation of the body's relaxation response. Breathing from the diaphragm, in a measured slow manner, fills the stomach then the chest, activates the vagus nerve, and reduces heart rate, blood pressure, and other sympathetic reactions. Practicing deep breathing multiple times per day and employing this technique during test days will be beneficial. 15. Visualize breathing into your heart: I employ a simple technique when feeling anxious where I place my hand over my heart and engage in slow breathing, focusing my consciousness on my heart. This technique was learned from an organization called Heartmath. The action of combining slow breathing, a nurturing gesture, and redirecting mental focus to your heart, helps restore a sense of calm.

π_1 **Rephrasing (truncated)**

you might want to dive into action and showcase up during my ACT, but you need to take a break beforehand. We can discuss this further later. By naming the monster, you can help calm it down. Creating some cognitive distance from anxious thoughts allows you to gain a sense of control over them. 13. Imagined practice makes perfect: If you have experienced many instances of anxiety during tests, visualize yourself taking a test without letting anxiety interfere. This is known as cognitive rehearsal, or guided imagery, taking a mental walk-through before a performance. Athletes do this all the time, imagining themselves performing at their best, in advance of a crucial event. Visualize yourself walking through the test, bypassing difficult problems and staying calm and centered: Lay down a new template of yourself as a peak-tester, and fill it with as many sensory details as possible so your mind believes it. Imagined practice can be as effective as actual practice. 14. regulate your breathing: Deep, diaphragmatic breathing is a powerful technique for reducing anxiety, as it activates the body's relaxation response. Breathing from the diaphragm, in a slow and measured way, filling the stomach then the chest, stimulates the vagus nerve, which controls the parasympathetic (calming) nervous system. Stimulation of the vagus nerve reduces heart rate, lowers blood pressure, and diminishes other sympathetic responses. Practice breathing deeply and consciously, multiple times a day, and this technique will be ready for you on test day. 15. Imagine breathing into your heart center: I have a simple technique I use when feeling nervous, where I place my hand on my heart and take deep breaths, focusing my energy on my heart. This technique, which I learned from an organization called Heartmath, combines

slow breathing, a nurturing gesture, and redirecting my consciousness to my heart, helps restore a sense of calm.

SynPro: Reformat Case 1

Organic (truncated)

The more saturated an object or image compared to its surroundings, the more likely it is to draw attention. ...Again, saturation can be used in VR to direct user's attention in the same way it has been used in media previously. ...This author suggests using desaturation to keep user's focused in 360-degree VR environments. ...In VR, gaze input is becoming common practice. Often, a means of confirmation is to have the user gaze at an object or image for a set amount of time to confirm critical actions. ...Most Real-Time Strategy games use a positive feedback loop by encouraging the player to gather resources to build factories or other structures. These new structures then produce even more resources. ...The time required to move to a target is a function of the target size and distance to the target (Lidwell, 2010). ...In VR Fitts law is still relevant. ...VR experiences have the same opportunity to induce stress in the user. ...A user in VR is completely immersed in the experience, and traumatic events in VR can seem real to the user.

π_1 **Reformat (truncated)**

- Question: What is mentioned as an advantage of using desaturation? Answer: To keep user's focused in 360-degree VR environments.
- Question: How does VR confirmation work? Answer: Confirmation is used to confirm "critical actions, inputs, or commands."
- Question: What caution is given regarding VR experiences? Answer: A designer must question the ethics of creating obscenely graphic, disturbing, and horrifying VR experiences.

π_0 **Reformat (truncated)**

- Question: What design principle is discussed that involves directing user attention? Answer: Saturation
- Question: What type of interaction does the text discuss that relies on user gaze input? Answer: Gaze confirmation
- Question: How does feedback in video games positively affect continued play? Answer: Encourage player gathering resources and expanding structures
- Question: What law was discussed regarding mouse movement in computing? Answer: Fitts Law
- Question: What does this law predict? Answer: The time it takes for an individual to move a cursor from one point to another
- Question: VR experiences can induce a similar stress response. What might this include? Answer: Traumatic events

---

π_2 **Reformat (truncated)**

- Question: What does the text discuss?
- Answer: Fitts Law principle in VR as described through anecdotal examples.
- Question: What factor does the text emphasize about immersive VR experiences?
- Answer: Extreme immersion can induce severe stress and potential lasting mental effects.
- Question: What form of stress is mentioned in the text?
- Answer: Stress induced through intense, realistic VR environments that can create psychologically impactful experiences.
- Question: What consideration for VR interaction is highlighted?
- Answer: The importance of ethical considerations for content designers due to the heightened susceptibility to emotional and psychological effects of VR experiences.

SynPro: Reformat Case 2

Organic (truncated)

The Delphi method was developed in the United States in the 1950s and 1960s in the military domain. It has been considered particularly useful in helping researchers determine the range of opinions which exist on a particular subject, in investigating issues of policy or clinical relevance and in trying to come to a consensus on controversial issues. The objectives can be roughly divided into those which aim to measure diversity and those which aim to reach consensus. ...Different ways to employ this method have been devised ...comment on a questionnaire .....The militia system was treasured by many people because the existence of a well-regulated militia, composed of civilians readily available for emergency military service, tended to deprive the government of an excuse for maintaining standing armies. ...Alexander Hamilton, for one, complained that the militia system violated the economic principle of division of labor. ...Citizens were always going to resist unpaid military training ...

π_1 **Reformat (truncated)**

- Question: What is the Delphi method? Answer: Developed in the United States in the 1950s and 1960s in the military domain.
- Question: What is its primary use? Answer: Determining the range of opinions on a particular subject, investigating issues of policy or clinical relevance, and coming to a consensus on controversial issues.
- Question: What aim does the Delphi method have? Answer: Measure diversity and consensus on contentious subjects.
- Question: Who complained that the militia system violated economic principles? Answer: Alexander Hamilton.
- Question: What was the issue identified as being problematic about the militia system? Answer: The militia system was noted as being ineffective due to citizen resistance to unpaid military service.

π_0 **Reformat (truncated)**

- Question: What kind of a method was developed in the US in the 50s and 60s? Answer: Delphi method
- Question: In what domain did the Delphi method emerge? Answer: Military
- Question: What was the method meant to do? Answer: Measure diversity
- Question: What system was criticized for depriving the government of an excuse to maintain a standing army? Answer: The militia system
- Question: Why did some comment about the militia system? Answer: Because it served the interests of a government needing less risk of using permanent armies.
- Question: What concerns were voiced regarding the militia? Answer: The necessity of citizens resisting unpaid military service

---

π_2 **Reformat (truncated)**

- Question: What is the origin of the Delphi method?
- Answer: Military in the United States in the 1950s and 1960s
- Question: What is the purpose of the Delphi method?
- Answer: Measuring diversity and developing consensus
- Question: How are experts' opinions aggregated in Delphi method sessions?
- Answer: Through structured questionnaires
- Question: How was the risk of federal oppression addressed by the drafters of the Constitution?
- Answer: The problem of oppression is mitigated by allowing a standing army with the right of Congress to call forth the militia.
- Question: Why might the government prefer to maintain a well-trained militia rather than a standing army?
- Answer: A militia that is disciplined and reliable is more effective in protecting liberty than an army that might turn against its citizens.

G Prompts

This section provides the detailed prompts used for each evaluation in our paper. We adapt and modify prompts from prior works such as RePro (Yu & Xiong, 2025) and Nemotron-CC (Su et al., 2025), and also design new prompts for specific evaluations. The prompts are designed to be clear, specific, and aligned with the goal we want to achieve.

Rephrasing Prompt (adapted from RePro)

Your task is to read and paraphrase the provided text following these instructions:
- Delete clearly irrelevant content:
  - Website headers, navigation bars, or menu items (e.g., "Home — About — Contact")
  - Unrelated HTTP links (e.g., ads, trackers, developer tools)
  - Generic footers (e.g., contact info, privacy policies, unsubscribe links)
  - Empty lines or decorative elements (e.g., "—")
- Preserve all content that is relevant and meaningful:
  - Informative or independently useful
  - Related to the topic, even tangentially
  - Provides context, background, or supporting value
  - Includes technical terms, key concepts, factual details, reasoning, and examples
- Handle mixed-relevance sentences carefully:
  - Remove only the irrelevant fragment if the rest remains coherent
  - Delete the whole sentence if the remainder loses meaning
- Do not alter meaningful content unnecessarily:
  - Only delete or modify when content is clearly meaningless or off-topic
  - Preserve the original structure, logic, and depth of the text
- Do not add explanations, notes, assumptions, or claims not found in the original text

Here is the text:
{Organic Text}
Task:
After thoroughly reading the above text, paraphrase it in high-quality and clear English following the instructions.
Start your response immediately with "Here is a paraphrased version:" and then provide the paraphrased text.

Reformat Prompt (adapted from Nemotron-CC)

Task: Read the text and convert it into a different format.
Follow these instructions:
1. Ask diverse questions that require different cognitive skills or cover different aspects of the text.
2. Ask questions in various forms such as:
- Yes/No questions that require determining whether a statement is true or false.
- Open-ended questions that begin with words like what, how, when, where, why and who.
- Multi-choice questions that offers two or more options to choose from. Include the options in the question.
- Comparison questions that compare two quantities or objects and determine the relationship between them.
- Reading comprehension questions that test the ability to understand and analyze the text.
- Problem-solving questions that test the ability to solve mathematical, physical, or logical problems.
3. Focus on asking questions about factual information, important knowledge, or concrete details in the text.
4. Write questions and answers using clear and concise language.
5. Use plain text. Do not use Markdown.
6. Each question and answer pair should be on a separate line. Tag the question with "Question:" and the answer with "Answer:".
Text:
{Organic Text}
Task:
After reading the above text, ask up to 8 questions and provide the correct answers following the instructions. Give your response in this format:
Here are the questions and answers based on the provided text:
- Question: [first question] Answer: [first answer]
- Question: [second question] Answer: [second answer]
....

DataMan Prompt

Please score the text on fourteen evaluation criteria and specify its domain:
Text: {Text}
Domain:_
[1]Accuracy:_/5
[2]Coherence:_/5
[3]Language Consistency:_/5
[4]Semantic Density:_/5
[5]Knowledge Novelty:_/5
[6]Topic Focus:_/5
[7]Creativity:_/5
[8]Professionalism:_/5
[9]Style Consistency:_/5
[10]Grammatical Diversity:_/5
[11]Structural Standardization:_/5
[12]Originality:_/5
[13]Sensitivity:_/5
[14]Overall Score:_/5

Structural Faithfulness Prompt (adapted from RePro)

[Instruction]
You are given two pieces of text: an original pretraining data sample and a rephrased version.
Your task is to judge if the rephrased version preserves the **structure** of the original sample.
- By "structure", we mean formatting, style, and presentation (e.g., paragraphing, JSON, list format, code blocks, markdown usage, plain text style).
- Do NOT consider semantic meaning. Ignore whether the words are the same or the content is equivalent.
- Focus only on whether the rephrased sample follows the same textual structure as the original (e.g., if the original is plain text paragraphs, the rephrased should also be plain text; if the original has bullet lists, the rephrased should also have bullet lists).
[Output]
Output **only** '1' if the structure is preserved.
Output **only** '0' if the structure is not preserved.
[Examples]
Example 1:
Original:
This is a paragraph.
This is another line.
Rephrased:
Here is a rewritten paragraph.
Here is another line of text.
Explanation: Both are plain text paragraphs, no special formatting. Structure preserved. Output: 1
—
Example 2:
Original:
- Item one
- Item two
Rephrased:
First item. Second item.
Explanation: The original uses a bullet list, while the rephrased is plain sentences. Structure not preserved. Output: 0
—
Example 3:
Original:
{{"name": "Alice", "age": 30}}
Rephrased: {{"person": "A.", "years": 30}}
Explanation: Both are JSON objects with the same structured format. Structure preserved. Output: 1
Explanation: The original is plain code with no markdown fences, while the rephrased introduces code fences. Structure not preserved. Output: 0
[Original]
{Organic Text}
[Rephrased]
{Rephrased Text}

Reformat Faithfulness Prompt

You are given an original text passage and a list of topic-content pairs generated from it. For each pair, determine whether it is faithful to the original text.
A faithful reformat pair satisfies ALL of the following:
1. The topic is about a subject covered or clearly implied by the original text.
2. The content is correct and supported by the original text.
3. The content does not contradict the text.
Labels:
- "Faithful": the topic is relevant and the content is correct.
- "Unfaithful_Topic": the topic covers something not covered or implied by the text.
- "Unfaithful_Content": the topic is valid but the content is wrong or unsupported.
Original Text:
{Organic Text}
Reformat Pairs:
{Reformat Text}
Respond with exactly one label per line in this format (no extra text):
1. [label]
2. [label]
...

Reformat Type Classification Prompt

Classify each question into exactly one of the following four categories:
Categories:
- "Factual": Questions asking for a specific fact such as who, what, when, where, or which. The answer is a concrete entity, date, number, or name directly stated in the text.
- "Analytical": Questions asking for reasoning such as why, how, cause-effect, or mechanisms. The answer explains a process, reason, or causal relationship.
- "Conceptual": Questions asking for definitions, explanations, or meaning such as what does X mean, what is X, or describe X. The answer clarifies a concept or term.
- "Integrative": Questions requiring combining multiple pieces of information such as summarize, what is the overall takeaway, or how do X and Y relate. The answer synthesizes across the text.
Text:
{Reformat Text}
Respond with exactly one category per line in this format (no extra text):
1. [category]
2. [category]
...