UPCYCLING THE LOW-QUALITY CORNERS OF THE WEB FOR PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern language model pretraining pipelines collect far more web text than they ultimately use, discarding large fractions through quality filtering. We call this ***token waste***: potentially useful tokens are acquired during data collection but removed before training. Existing synthetic data methods mostly *recycle* data by rewriting high-quality documents that already survive filtering. We instead ask whether data deemed too low-quality for training must actually be discarded; can they instead be ***upcycled*** by transforming low-quality documents into useful pretraining data. Concretely, we train a small language model with preference optimization over learned quality and influence-based utility signals, then use it to upcycle the lowest-quality 14B tokens from C4 (i.e., wasted tokens). Across language models from 19M to 1.4B parameters, upcycled tokens consistently outperform wasted tokens on in-domain and out-of-distribution language modeling, as well as downstream performance. When combined with high-quality data, upcycled tokens recover much of the benefit of adding more high-quality data. Additional experiments show that upcycling preserves factual knowledge lost through filtering, improves existing synthetic data recipes, remains effective with substantially smaller synthesizers, and extends over-training gains beyond what is achieved by repeating high-quality data. Together these results show that *token waste can be prevented*, with benefits extending well beyond recovering pretraining utility.

1 INTRODUCTION

The Internet contains far more data than what is ultimately used to train language models (LMs). A longstanding *garbage in, garbage out* intuition has encouraged removing low-quality data to prevent it from degrading model performance (Brodley & Friedl, 1999). Consistent with this view, modern pretraining pipelines begin with large web crawls and remove substantial amounts of data through language identification and heuristic filtering (Wenzek et al., 2020; Raffel et al., 2020), deduplication (Penedo et al., 2023; 2024), and learned quality filtering (Li et al., 2024). These choices can substantially improve model performance when the training budget permits using only a fraction of the available data. However, the benefits of aggressive filtering are not universal. Quality filtering can hurt performance on some evaluation domains even when improving others (Longpre et al., 2023), while unfiltered data pools can outperform filtered subsets in high-compute regimes as larger models better tolerate lower-quality data (Mohri et al., 2026). Beyond these performance trade-offs, aggressive filtering can also introduce systematic coverage biases (Dodge et al., 2021), exclude useful content that does not match the characteristics favored by quality filters (Nait Saada et al., 2025), and discard rare information in the corpus (Kandpal et al., 2023).

Yet aggressive filtering coincides with growing data demand, as compute-optimal scaling (Hoffmann et al., 2022) and over-training (Gadre et al., 2024) call for ever larger token budgets. This tension motivates the substantial effort toward creating additional training data synthetically. In particular, language models are used to generate new textbook-like documents from curated material (Gunasekar et al., 2023; Ben Allal et al., 2024), or increase the effective size of existing corpora through rephrasing and reformulation (Maini et al., 2024; Hao et al., 2025; Nguyen et al., 2025). However, these approaches largely operate on data that *already survives upstream selection* (i.e., *recycling*),

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

leaving the much larger pool of filtered-out tokens unused.[1] We refer to this phenomenon as ***token waste***: *potentially useful tokens are collected during data acquisition but discarded before training*. We therefore study how synthetic data generation can reduce token waste by *upcycling* the vast pool of wasted tokens to meet growing data demands.

Concretely, we take the lowest-quality 14B tokens from C4 (Dodge et al., 2021) under the DCLM quality classifier (Li et al., 2024) as a proxy for data that modern selection pipelines would discard (App. A), and call them *Wasted Tokens* (§2.1). We then transform these documents with a lightweight synthesizer LM into *Upcycled Tokens* (§2.2), with the goal of recovering useful pretraining signal from data otherwise expected to hurt performance. To calibrate how much utility can be recovered, we use a token-matched DCLM sample as *Retained Tokens*, representing conventionally retained high-quality data and providing a reference ceiling for the utility recoverable from Wasted Tokens. We compare these corpora both in isolation and when supplementing Retained Tokens, evaluating in-distribution and out-of-distribution language modeling and downstream performance.

We make the following contributions:

- **Upcycling recovers substantial pretraining utility from discarded data.** Across 19M–1.4B models, Upcycled Tokens outperform Wasted Tokens and, when mixed with Retained Tokens, recover much of the benefit of additional Retained Tokens (§3).
- **Upcycling preserves factual knowledge lost through filtering.** Wasted Tokens contain factual knowledge that is absent from Retained Tokens and is not recovered by adding more Retained Tokens. Upcycled Tokens preserve this knowledge while improving its utility (§4.1).
- **Upcycling extends the benefits of over-training under data constraints.** When high-quality data must be repeated to reach large training budgets, supplementing it with Upcycled Tokens outperforms repeating the high-quality data alone (§4.4).

2 PRELIMINARIES

We begin by describing the experimental setup and the procedure to train our synthesizer LM.

2.1 EXPERIMENTAL SETUP

In this section, we describe the models we pretrain and the data we use. Our setup isolates the effect of upcycling the Wasted Tokens by comparing token-matched training corpora across model scales.

Models. We train decoder-only language models from scratch at 19M, 55M, 152M, 670M, and 1.4B parameters, using Llama-2 architecture and tokenizer (Touvron et al., 2023) based on the training setup of Kim et al. (2025). We train five seeds at 19M, 55M, and 152M, three at 670M, and one at 1.4B. We provide architecture and training details in App. C.

Data. To model the data discarded by modern pretraining pipelines, we begin with C4 (Raffel et al., 2020), which is substantially less filtered and curated than more recent web corpora such as DCLM (Li et al., 2024), RefinedWeb (Penedo et al., 2023), FineWeb (Penedo et al., 2024), and Nemotron-CC (Su et al., 2025). We sample 200B C4 tokens, tokenized with the Llama-2 tokenizer, and score the source documents using the FastText (Joulin et al., 2016) quality classifier from DCLM. We retain the lowest-scoring 14B tokens as a proxy for data that a modern pipeline would discard (see App. A for construction and justification). We refer to these data as the *Wasted Tokens*. For each source document in the Wasted Tokens, we synthesize one transformed version using the model described in §2.2, which form the *Upcycled Tokens*. Lastly, we use a random 14B tokens subsample of DCLM as our high-quality baseline, which we call the *Retained Tokens*. At each training scale, we construct token-matched subsets of these three corpora.

For each model size, we evaluate six data mixtures spanning two regimes. To assess standalone utility, we train on Wasted, Upcycled, and Retained at a token-to-parameter ratio (TPR) of 10, using 0.19B, 0.55B, 1.5B, 6.7B, and 14B tokens. To assess utility alongside high-quality data, we

[1] For example, Li et al. (2024) report that their final selected dataset constitutes only 1.4% of the initial data pool, with quality selection alone removing 89.8% of the data that survived preceding filters.

train Retained + Wasted, Retained + Upcycled, and Retained + Retained at TPR=20, matching the compute-optimal regime (Hoffmann et al., 2022), with budgets of 0.19B + 0.19B, 0.55B + 0.55B, 1.5B + 1.5B, 6.7B + 6.7B, and 14B + 14B tokens. The mixed data comparisons therefore match both the amount of Retained Tokens and the total token budget. Within each corpus, subsets are nested across scales, with 0.19B $\subset$ 0.55B $\subset$ 1.5B $\subset$ 6.7B $\subset$ 14B tokens. Training documents from §2.2 are excluded from all pretraining corpora.

Evaluation. We evaluate held-out next-token loss on DCLM (Li et al., 2024) and FineWeb-Edu (Penedo et al., 2024), and centered CORE (Li et al., 2024) score for downstream performance. CORE aggregates multiple downstream benchmarks and is centered so that a score of 0 corresponds to random performance. For loss, we report the mean across documents with its standard error. For CORE, we report the mean and sample variance across training seeds.

2.2 Training the Synthesizer Language Model

In this section, we describe how we train our synthesizer LM. We generate multiple rewrites for each source document, rank them using a learned quality score and an influence-based estimate of pretraining utility, and train the synthesizer with DPO (Rafailov et al., 2024).

Candidate generation. We reserve 8000 documents from the same Wasted Tokens distribution used in our pretraining experiments, ensuring that they do not overlap with the pretraining corpora. We use SmolLM2-1.7B-Instruct (Ben Allal et al., 2025) both to generate the preference candidates and as the initialization of the synthesizer model. For each source document, we sample 16 rewrites with temperature 1.0, top-$p = 1.0$, no top-k truncation, and a maximum generation length of 4096 tokens. We remove refusals and duplicate generations. We defer the prompting details to App. F.

Preference construction. Let x_i denote a source document from the Wasted Tokens and $x_{i,j}$ its j-th candidate. We rank candidates using (i) the FastText quality classifier used to construct DCLM (Joulin et al., 2016; Li et al., 2024) and (ii) an influence-based estimate of pretraining utility (Koh & Liang, 2020). We compute influence scores using LogIX and its LoGra gradient projection method (Choe et al., 2024), with SmolLM2-135M (Ben Allal et al., 2025) as a proxy model. We obtain rank-32 LoGra projections of gradients of the summed next token loss with respect to all MLP projection matrices, using sequences of at most 2048 tokens and damping 0.01. Let $\tilde{g}(x)$ denote the projected gradient for document x and C the damped gradient covariance estimate. For $M = 5000$ DCLM target documents $\{z_m\}_{m=1}^{M}$, we compute the ℓ-RelatIF normalized influence score (Barshan et al., 2020), $h(x_{i,j}) = \frac{1}{M}\sum_{m=1}^{M} \frac{\tilde{g}(z_m)^\top C^{-1}\tilde{g}(x_{i,j})}{\sqrt{\tilde{g}(x_{i,j})^\top C^{-1}\tilde{g}(x_{i,j})}}$, where larger values indicate greater predicted utility on the target DCLM distribution.

We treat quality as the primary criterion and use influence to distinguish candidates with similar quality. Specifically, we clip $q(x_{i,j})$ to $[0.01, 0.9]$ and discretize it into bins of width 0.05. We then standardize $h(x_{i,j})$ across the candidates generated from the same source x_i, and rank candidates by quality bin and standardized influence. We retain the 5K chosen–rejected pairs with the largest margins and split them by source document into 90% training and 10% validation sets.

Direct preference optimization. We optimize SmolLM2-1.7B-Instruct (Ben Allal et al., 2025) with DPO (Rafailov et al., 2024), using low-rank adapters (Hu et al., 2021) on all attention and MLP projections with rank 32 and scaling factor 64. We train for three epochs with a learning rate of 3×10^{-5}, using cosine decay, a regularization coefficient $\beta = 0.1$, and a maximum length of 2048.

Generating upcycled tokens. We apply the synthesizer to each source document that forms the Wasted Tokens, using the same prompt and generation hyperparameters as in candidate generation. The resulting generations constitute the Upcycled Tokens and are used directly without any filtering. We provide representative examples of source documents and their upcycled counterparts in App. G.

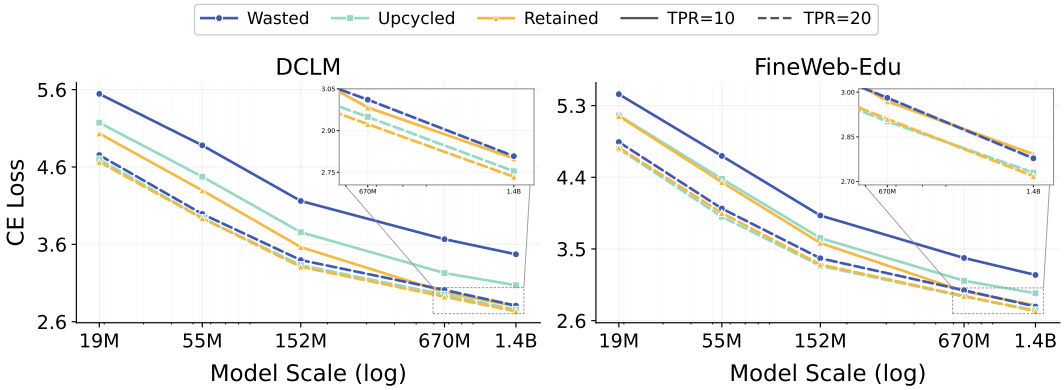

Figure 1: **Upcycling recovers substantial pretraining utility from Wasted Tokens.** Held-out loss on DCLM and FineWeb-Edu, averaged across seeds. Upcycled Tokens consistently outperform Wasted Tokens, while Retained + Upcycled recovers most of the gap to Retained + Retained.

3 Upcycling Wasted Tokens

We now test whether upcycling *Wasted Tokens* can recover their pretraining utility. We first compare *Upcycled Tokens* with *Wasted Tokens* to isolate the effect of transformation, using *Retained Tokens* as a high-quality reference. We then compare *Retained + Upcycled* with *Retained + Wasted* and *Retained + Retained* to evaluate upcycling when supplementing high-quality data. We measure held-out loss on DCLM and FineWeb-Edu together with downstream performance on CORE.

Upcycling substantially improves discarded data in isolation. Across all five model scales, *Upcycled Tokens* achieve markedly lower held-out loss than the corresponding *Wasted Tokens* on both DCLM and FineWeb-Edu (Figure 1). This gap remains large as model size increases: at 1.4B parameters, DCLM loss decreases from 3.472 to 3.072 and FineWeb-Edu loss from 3.174 to 2.944. Thus, the advantage of upcycling Wasted Tokens persists even at the largest model scale. These gains also hold when Upcycled Tokens are combined with Retained Tokens. Across every model scale and both evaluation distributions, *Retained + Upcycled* substantially outperforms *Retained + Wasted*. More strikingly, *Retained + Upcycled* remains consistently close to *Retained + Retained*, despite replacing half of the high-quality training tokens with data synthesized from documents that would otherwise be discarded. At 1.4B parameters, replacing Wasted with Upcycled Tokens closes 73% of the loss gap between *Retained + Wasted* and *Retained + Retained* on DCLM and 80% on FineWeb-Edu. On FineWeb-Edu, *Retained + Upcycled* matches or outperforms *Retained + Retained* from 19M through 670M parameters. Upcycling data from the rejected tail therefore provides nearly the same pretraining utility as allocating the additional token budget to Retained Tokens.

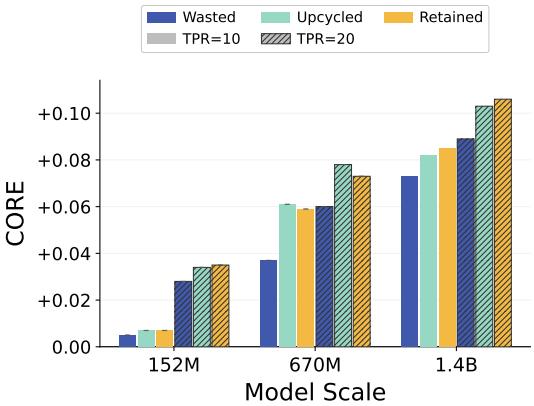

Figure 2: **Utility of upcycling transfers to downstream performance.** Centered CORE scores for models achieving above-random performance.

The benefits of upcycling extend to downstream capabilities as measured on CORE (Figure 2).*Upcycled Tokens* consistently outperform *Wasted Tokens*, with the same pattern holding when each is combined with *Retained Tokens*. At 670M and 1.4B *Retained + Upcycled* remains nearly on par with *Retained + Retained*, scoring 0.078 vs. 0.073 at 670M and 0.103 vs. 0.106 at 1.4B.

Taken together, these results show that low pretraining utility in its original form does not make the underlying data unusable. Upcycling recovers much of the utility lost in Wasted Tokens and, when combined with Retained Tokens, nearly matches allocating the same token budget entirely to high-quality data. We provide further analysis in App. D, showing that upcycling reduces surprisal while largely preserving the semantic content of the source documents.

4 CHARACTERIZING UPCYCLING

We now examine the broader benefits of upcycling beyond recovering pretraining utility.

4.1 UPCYCLING PRESERVES FACTUAL KNOWLEDGE LOST THROUGH FILTERING

Motivation. Pretraining enables models to internalize reusable factual knowledge, making data selection consequential not only for average data quality but also for *knowledge coverage*. This is particularly important because factual information on the web is long-tailed. Models learn frequent facts more reliably (Kandpal et al., 2023), perform worse on less popular entities (Mallen et al., 2023; Sun et al., 2024), and exhibit factual recall that strongly depends on fact frequency (Chang et al., 2024). Aggressive filtering may therefore improve average data utility while removing much of the available evidence for already scarce facts. These losses can also be uneven across domains (Longpre et al., 2023), as illustrated by C4's blocklist filtering disproportionately removing text from and about minority individuals (Dodge et al., 2021).

We therefore introduce NOKLB (**No Knowledge Left Behind**), a grounded multiple-choice benchmark constructed exclusively from documents rejected by quality selection (Li et al., 2024). Accordingly, NOKLB tests whether filtering removes factual information from the training corpus and whether models trained on the resulting data consequently fail to acquire it.

Benchmark construction. We construct NOKLB using a multi-stage curation procedure designed to isolate factual knowledge that would otherwise be excluded from training. We first score documents from the *Wasted Tokens* with the FineWeb-Edu classifier (Penedo et al., 2024), then use Llama-3.1-70B-Instruct (Grattafiori et al., 2024) to select sources containing non-trivial, reusable factual information and generate candidate QA pairs. We then verify each answer against external evidence using GPT-5.6 Luna (OpenAI, 2026) with web search, providing an additional validation signal beyond source-document grounding. This process yields 2050 questions that directly probe factual knowledge removed by quality selection. We provide full construction details in App. E.

Results. We evaluate the 670M and 1.4B models from §3, as they achieve above-random performance on CORE. We use zero-shot multiple-choice scoring with answer text, select the answer with the highest length-normalized log-probability, and report test-set accuracy. Table 1 shows that quality selection removes factual knowledge that is not restored by allocating additional training tokens to the retained data, as *Retained* + *Wasted* consistently outperforms *Retained* + *Retained*. Upcycling preserves this complementary knowledge while improving its utility, with *Retained* + *Upcycled* outperforming both *Retained* + *Wasted* and *Retained* + *Retained* at both model scales. Together, these comparisons show that rejected documents contain useful knowledge absent from the selected corpus and that upcycling preserves this information while making it more useful for pretraining.

Table 1: **Upcycling preserves factual knowledge removed by quality filtering.** NOKLB accuracy. Retained + Upcycled achieves the highest accuracy at both model scales.

Corpus	670M	1.4B
Wasted	$26.91_{0.78}$	$32.05_{\mathrm{N/A}}$
Upcycled	$27.20_{1.06}$	$32.78_{\mathrm{N/A}}$
Retained	$21.85_{0.63}$	$27.61_{\mathrm{N/A}}$
Retained + Wasted	$30.07_{1.69}$	$37.41_{\mathrm{N/A}}$
Retained + Upcycled	$\mathbf{32.05}_{0.86}$	$\mathbf{39.41}_{\mathrm{N/A}}$
Retained + Retained	$25.59_{0.33}$	$33.32_{\mathrm{N/A}}$

To contextualize NOKLB, Table 2 reports performance for DCLM 7B (Li et al., 2024), trained on 2.5T tokens from the DCLM corpus, alongside several large language models. Every reference model except DCLM 7B performs better on NOKLB than on MMLU, whereas DCLM 7B uniquely reverses this ordering. Since NOKLB probes knowledge removed by quality filtering, this reversal provides suggestive evidence that DCLM filtering reduces knowledge coverage.

Table 2: **NOKLB and MMLU accuracy for larger LLMs.**

Model	NOKLB	MMLU
DCLM 7B	62.44	63.72
Olmo-3-7B-Instruct	73.49	69.10
Llama-3.1-8B-Instruct	77.86	69.40
gpt-oss-20b	79.66	74.88
Qwen3-8B	80.93	76.89

4.2 Upcycling Improves Existing Synthetic Data Recipes

Motivation. A growing line of synthetic pretraining work develops explicit recipes for transforming source content into more useful training data (Yang et al., 2024; Ma et al., 2025; Yang et al., 2026). Our method targets a different component of the pipeline, the model that *executes* a chosen recipe. We therefore view recipe design and synthesizer optimization as complementary. To test this, we adopt the *Table* transformation of Niklaus et al. (2026), which restructures a document into a table and generates a grounded question–answer pair. Because Table is their strongest transformation for multiple-choice question answering, it provides a stringent test of whether synthesizer optimization can further improve an already effective recipe.

Setup. We keep the Table transformation fixed and vary only the synthesizer used to realize it. As a baseline, we apply the Table prompt with SmolLM2-1.7B-Instruct. We then specialize the same model using the training procedure from §2.2: candidates are generated with the Table prompt and ranked solely by their estimated influence on 5K MMLU (Hendrycks et al., 2021) training questions. This replaces the objective used in our main experiments with a task-aligned influence signal suited to the question–answer structure targeted by the Table transformation. We otherwise retain the same preference construction and optimization procedure. We apply either the original or specialized synthesizer to the same source documents and evaluate the resulting synthetic data by pretraining 670M and 1.4B models at TPR=10 and 20. We report downstream performance on CORE.

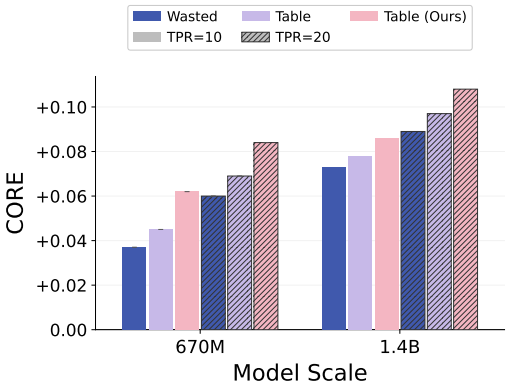

Figure 3: **Synthesizer optimization compounds the gains of an existing synthetic data recipe.** CORE performance for the *Table* transformation of Niklaus et al. (2026).

Results. Optimizing the synthesizer consistently improves the utility of the existing Table recipe across model scales and token budgets (Figure 3). At 670M parameters, specialization increases CORE from 0.045 to 0.062 at TPR=10 and from 0.069 to 0.084 at TPR=20. The same pattern persists at 1.4B parameters, where performance increases from 0.078 to 0.086 at TPR=10 and from 0.097 to 0.108 at TPR=20. Importantly, the unoptimized Table transformation already improves over the corresponding Wasted Tokens baseline in every setting, and specializing the synthesizer provides a further gain. The improvement therefore compounds, rather than substitutes for, the benefit of the existing recipe, supporting our view that recipe design and synthesizer optimization are complementary components.

4.3 Upcycling Remains Effective with Smaller Synthesizers

Motivation. In our main experiments, the *Upcycled Tokens* are generated by a 1.7B language model (§2.2). This raises a natural alternative explanation for the observed improvement: the synthesizer may simply act as a teacher, distilling knowledge from its parameters into the models pretrained on its generations. If this were the primary source of the gains, the effectiveness of upcycling could become increasingly dependent on synthesizer capacity as the pretrained model scales, limiting the scalability of our method. We test this possibility by investigating whether upcycling remains effective when the model consuming the synthetic data is much larger than the model producing it.

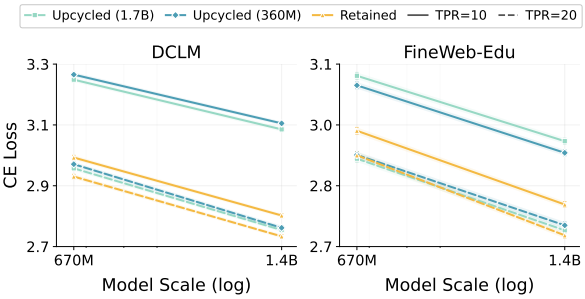

Figure 4: **Upcycling remains effective with a smaller synthesizer.** Held-out DCLM and FineWeb-Edu loss for data generated by 360M and 1.7B synthesizers.

Setup. We repeat our synthesizer training pipeline from §2.2 using SmolLM2-360M-Instruct (Ben Allal et al., 2025), replacing the 1.7B model used in our main experiments. We then use the resulting *Upcycled Tokens* to pretrain 670M and 1.4B models, which are approximately two and four times larger than the synthesizer. Mirroring our main experiments, we pretrain these models both on *Upcycled Tokens* in isolation and on *Upcycled Tokens* together with high-quality *Retained Tokens*. In each setting, we compare against token-matched *Upcycled Tokens* generated by the 1.7B synthesizer and the corresponding high-quality *Retained Tokens* baseline.

Results. Upcycling remains effective even when the synthesizer is smaller than the model being pretrained. Across DCLM loss and FineWeb-Edu loss (Figure 4), and CORE (Figure 5), data produced by the 360M synthesizer remains close to data produced by the 1.7B synthesizer, both in isolation and when combined with *Retained Tokens*. At 1.4B parameters, for example, using the 360M rather than the 1.7B synthesizer changes DCLM loss from 3.072 to 3.091 for *Upcycled Tokens*, and from 2.754 to 2.761 for *Retained + Upcycled*. These differences are small relative to the gains from upcycling itself, since the corresponding *Wasted Tokens* and *Retained + Wasted* baselines obtain losses of 3.472 and 2.808, respectively.

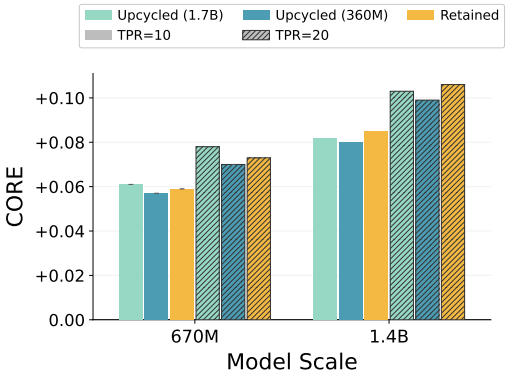

Figure 5: **360M synthesizer retains the downstream benefits of upcycling.** CORE performance for data generated by the 360M and 1.7B synthesizers.

The gains therefore do not require the synthesizer to match or exceed the scale of the model that is trained on its outputs. Accordingly, these experiments make a simple teacher–student distillation account insufficient to explain our results, while not ruling out some transfer of parametric knowledge from the synthesizer. At the same time, they show that upcycling can remain effective as the pretrained model scales beyond the synthesizer, suggesting that effective upcycling need not require correspondingly large synthesizers. Notably, this contrasts prior work such as REWIRE, which uses a 70B rewriter and estimates roughly 88K H100 GPU-hours to generate 100B tokens (Nguyen et al., 2025).

4.4 Upcycling Extends the Benefits of Over-Training

Motivation. Modern language models are increasingly *over-trained* relative to compute-optimal scaling (Gadre et al., 2024). This pushes token budgets far beyond the 20 tokens per parameter prescribed by Hoffmann et al. (2022).[2] With a fixed supply of high-quality data, however, such budgets require increasingly many repetitions, whose marginal utility eventually diminishes (Muennighoff et al., 2023; Kim et al., 2025). Aggressive filtering therefore incurs a second cost beyond discarding useful signal, as it shrinks the distinct data pool available for over-training.

This motivates a practical role for upcycling in data constrained over-training. Recent work shows that, with sufficient compute, training on a larger pool containing lower-quality data can outperform repeatedly training on a smaller, aggressively filtered high-quality data pool (Mohri et al., 2026). We therefore ask whether upcycling offers a better alternative by supplementing Retained Tokens with higher utility Upcycled Tokens rather than repeating retained data or training on lower-quality data.

Setup. We study a deliberately data-constrained setting following Mohri et al. (2026). We fix the available high-quality corpus to 1.5B *Retained Tokens* and compare it against an expanded pool containing the same 1.5B Retained Tokens together with 14B *Upcycled Tokens*. This ratio roughly follows the learned-quality selection stage of the DCLM pipeline, which filters 89.8% of the documents surviving the preceding stages (Li et al., 2024). We pretrain 55M and 152M models on 56B and 150B tokens, corresponding to TPR=1000. We repeat each data pool as necessary to reach the training budget. Unlike Mohri et al. (2026), we use a higher weight decay, motivated by evidence

[2]For example, dense Qwen3 models are trained on 36T tokens, corresponding to over 1,000 TPR even at 32B parameters (Yang et al., 2025), while SmolLM2-1.7B is trained on 11T tokens (Ben Allal et al., 2025).

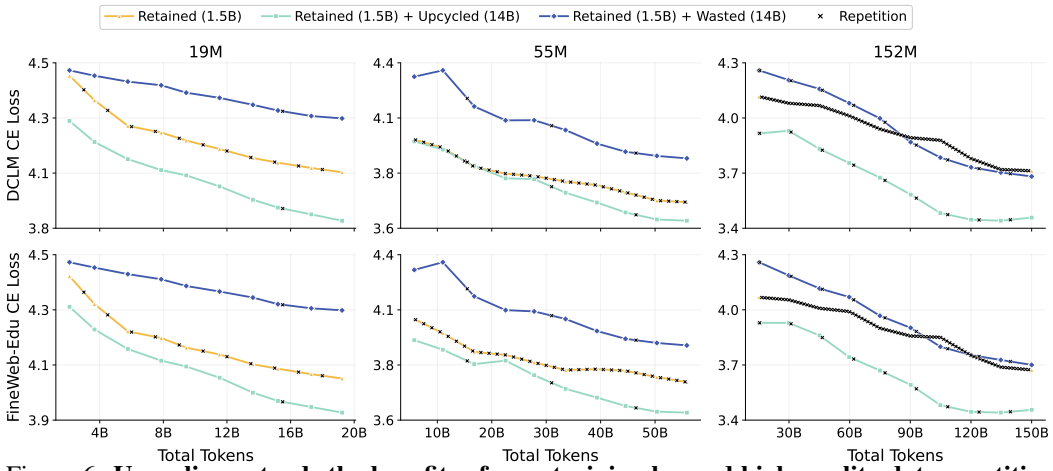

Figure 6: **Upcycling extends the benefits of over-training beyond high-quality data repetition.** Held-out loss on DCLM and FineWeb-Edu throughout training for 55M and 152M models. We denote each repetition of the corpus with a cross.

that the optimal weight decay increases with repetition (Fang et al., 2025), thereby strengthening the repeated high-quality baseline relative to their setting. We evaluate using the same setup as in §3.

Results. The benefits of upcycling persist even in the extreme over-training regime (Figure 6). At TPR=1K, Retained + Upcycled achieves the lowest held-out loss on both DCLM and FineWeb-Edu across both model scales, with corresponding improvements in CORE (Figure 7). Notably, simply expanding the data pool with lower-quality tokens is not always beneficial. Despite repeating the 1.5B Retained Tokens 37 times at 55M and 100 times at 152M parameters, training on Retained Tokens alone outperforms Retained + Wasted at 55M and remains competitive at 152M. This contrasts with Mohri et al. (2026), who find that larger, less-filtered data pools can outperform repeatedly training on aggressively filtered data at sufficiently high compute. One possible explanation is that our stronger weight decay mitigates degradation from repetition (Fang et al., 2025), although setup differences prevent isolating its effect. Crucially, even against this strongly regularized repetition baseline, supplementing high-quality data with Upcycled Tokens yields further improvements in both language modeling and downstream performance. Thus, upcycling extends the benefits of over-training beyond what can be achieved by repeatedly training on high-quality data alone.

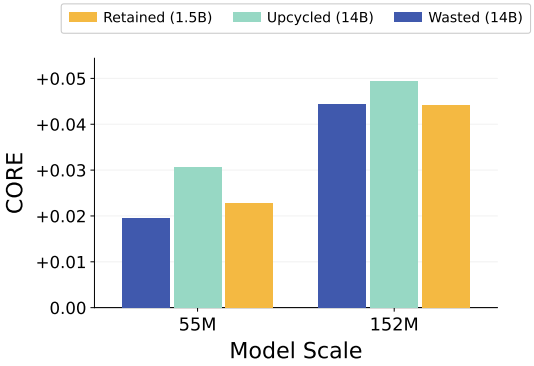

Figure 7: **Upcycling improves downstream performance under over-training.** CORE scores for 55M and 152M models trained at TPR=1K.

5 Related Work

Web data curation. Common Crawl corpora typically apply language identification and heuristic filtering (Wenzek et al., 2020; Raffel et al., 2020; Rae et al., 2022), followed by deduplication (Penedo et al., 2023; Soldaini et al., 2024; Penedo et al., 2024). Some pipelines additionally combine web text with curated sources (Gao et al., 2020; Soldaini et al., 2024) or use learned models to select documents for training (Penedo et al., 2024; Li et al., 2024). Accordingly, prior work largely focuses on selecting or emphasizing the data most useful for training. Existing methods therefore prune data (Marion et al., 2023), resample examples according to importance (Xie et al., 2023b), increase diversity (Tirumala et al., 2023), reweight domains (Xie et al., 2023a), or estimate data quality and utility with learned models (Wettig et al., 2024; Engstrom et al., 2024; Shum et al., 2025). Although these

approaches improve the average utility of the retained data, such filtering may also remove atypical or rare information (Dodge et al., 2021; Longpre et al., 2023; Kandpal et al., 2023), particularly because learned quality scores reflect the reference distributions used to train the scoring models (Nait Saada et al., 2025). Our work complements these selection methods by improving the utility of the low-quality tail, enabling it to contribute alongside the corpus's high-quality data.

Synthetic pretraining data. Scaling trends suggest that the demand for public human text will outpace its supply (Villalobos et al., 2024). This constraint motivates the use of synthetic data to extend finite human-generated corpora. Existing approaches broadly follow two strategies: i) synthesizing new documents from curated topics, educational resources, or structured seeds (Gunasekar et al., 2023; Ben Allal et al., 2024), and ii) transforming documents drawn from web corpora (Maini et al., 2024; Su et al., 2025; Hao et al., 2025; Nguyen et al., 2025). Within the first line, models expand limited seeds into large collections of textbook-style educational documents (Gunasekar et al., 2023; Ben Allal et al., 2024). Along the second line, which effectively *recycles* data that survives filtering, WRAP rephrases C4 passages into Wikipedia articles and question–answer pairs (Maini et al., 2024), Nemotron-CC applies different transformations according to estimated source quality (Su et al., 2025), and MGA reformulates documents for different genres and audiences (Hao et al., 2025), and REWIRE rewrites the broader DCLM-RefinedWeb pool before retaining its highest-scoring generations (Nguyen et al., 2025). Unlike these approaches, which transform broad or quality-stratified source pools, we explicitly condition the source distribution on the lowest-scoring tail. We detail this distinction in App. A.

Notably, synthetic data utility depends strongly on diversity, scale, and generation choices. Greater diversity correlates with downstream performance (Chen et al., 2024), while utility also varies with source quality, output style, and generation diversity (Maini et al., 2025), as well as with prompt design, generator choice, source corpus, and the natural data included in the final mixture (Niklaus et al., 2026). These choices interact with training composition, where rewriting tends to provide smaller gains as source quality rises (Pieler et al., 2024), while mixtures of natural and synthetic data are generally more effective than synthetic data alone (Kang et al., 2025). These findings motivate our focus on the opposite end of the quality distribution, where we *upcycle* the lowest-quality subset of C4 into useful pretraining data that complements the high-quality end of the corpus.

Learned rewriters. Learned rewriters remain underexplored, as most synthetic pretraining pipelines rely on frozen general-purpose models (Maini et al., 2024; Su et al., 2025; Hao et al., 2025; Nguyen et al., 2025; Maini et al., 2025; Niklaus et al., 2026). Among the limited work that explicitly optimizes the rewriter itself, ProX and RefineX learn conservative document editing operations (Zhou et al., 2025; Bi et al., 2025), whereas RePro trains a document rewriter through reinforcement learning with a multi-component objective (Yu & Xiong, 2025). More recently, Thrush et al. (2026) train a synthetic data generator using metagradient-based rewards that reflect the effect of generated examples on a downstream differentiable target. These methods demonstrate that rewriters can themselves be optimized for the utility of their outputs, but they operate on curated source distributions rather than specializing for low-quality tail. We instead train a smaller synthesizer specifically for the lowest-quality portion of C4, using DPO with a simpler reward design. This combination reduces both optimization and inference costs while directly targeting data that existing learned rewriters deprioritize.

6 Conclusion

Modern pretraining pipelines follow the *garbage in, garbage out* intuition, filtering low-quality data prior to training and creating substantial *token waste*. At the same time, growing pretraining token demands have motivated synthetic data methods, which primarily recycle data that already survives filtering. Our results show that garbage in need not mean garbage out when low-quality data can be upcycled. Across model scales, upcycling recovers substantial pretraining utility, preserves complementary factual knowledge lost through filtering, compounds the gains of existing synthetic data recipes, and provides a stronger alternative to reusing high-quality data under over-training.

Upcycling wasted tokens unlocks a vast reserve of data that current pipelines leave unused. Future work should build on this reserve by treating filtered data as a source to recover rather than discard.

AI USE STATEMENT

We used generative AI tools to assist with implementing and debugging code and to polish the writing of this paper. All AI-assisted code and text were reviewed and tested by the authors. The authors take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

Additionally, we used generative AI models as part of the experimental methodology to generate synthetic training data and to assist in the construction and external verification of the NoKLB benchmark, as described in Sections §2.2 and §4.1 and the corresponding appendices.

ETHICS STATEMENT

We do not believe that this work raises specific ethical concerns requiring additional discussion beyond the ICLR Code of Ethics. We build on standard, publicly available datasets, models, and infrastructure.

REPRODUCIBILITY STATEMENT

We build on publicly accessible datasets and models. We provide details necessary to reproduce our experiments throughout the paper and appendix. Section 2.1 describes the pretrained models, data construction, training budgets, and evaluation protocol, while Section 2.2 describes candidate generation, preference construction, synthesizer optimization, and generation of the Upcycled Tokens. Additional details on the construction of the Wasted Tokens, synthesizer development, prompts, and pretraining configuration are provided in the corresponding appendices, including App. A, App. B, App. F, and App. C. Upon acceptance, we will release our code, data and models.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L. Leavitt, and Mansheej Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models, 2024. URL https://arxiv.org/abs/2405.20541.

Elnaz Barshan, Marc-Etienne Brunet, and Gintare Karolina Dziugaite. Relatif: Identifying explanatory training examples via relative influence, 2020. URL https://arxiv.org/abs/2003.11630.

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

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

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 arXiv:2507.03253*, 2025.

C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, August 1999. ISSN 1076-9757. doi: 10.1613/jair.606. URL http://dx.doi.org/10.1613/jair.606.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining?, 2024. URL https://arxiv.org/abs/2406.11813.

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, 2024. URL https://arxiv.org/abs/2410.15226.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Grosse, and Eric Xing. What is your data worth to gpt? llm-scale data valuation with influence functions, 2024. URL https://arxiv.org/abs/2405.13954.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Collaborate instead of concatenate, 2021. URL https://arxiv.org/abs/2006.16362.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, 2021. doi: 10.18653/v1/2021.emnlp-main.98.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. Dsdm: Model-aware dataset selection with datamodels. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 12491–12526. PMLR, 2024.

Alex Fang, Hadi Pouransari, Matt Jordan, Alexander Toshev, Vaishaal Shankar, Ludwig Schmidt, and Tom Gunter. Datasets, documents, and repetitions: The practicalities of unequal data quality, 2025. URL https://arxiv.org/abs/2503.07879.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Jenia Jitsev, Luca Soldaini, Alexandros G. Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks, 2024. URL https://arxiv.org/abs/2403.08540.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi,

Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ra-

maswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL https://arxiv.org/abs/2306.11644.

Xintong Hao, Ruijie Zhu, Ge Zhang, Ke Shen, and Chenggang Li. Reformulation for pretraining data augmentation. *arXiv preprint arXiv:2502.04235*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

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, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification, 2016. URL https://arxiv.org/abs/1607.01759.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 15696–15707. PMLR, 2023.

Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. Demystifying synthetic data in llm pre-training: A systematic study of scaling laws, benefits, and pitfalls, 2025. URL https://arxiv.org/abs/2510.01631.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. Pre-training under infinite compute, 2025. URL https://arxiv.org/abs/2509.14786.

Konwoo Kim, Suhas Kotha, Yejin Choi, Tatsunori Hashimoto, Nick Haber, and Percy Liang. Data-efficient pre-training by scaling synthetic megadocs, 2026. URL https://arxiv.org/abs/2603.18534.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions, 2020. URL https://arxiv.org/abs/1703.04730.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 14200–14282. Curran Associates, Inc., 2024. doi: 10.52202/079017-0455. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/19e4ea30dded58259665db375885e412-Paper-Datasets_and_Benchmarks_Track.pdf.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, and toxicity. *arXiv preprint arXiv:2305.13169*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Shengjie Ma, Xuhui Jiang, Chengjin Xu, Cehao Yang, Liyu Zhang, and Jian Guo. Synthesize-on-graph: Knowledgeable synthetic data generation for continue pre-training of large language models, 2025. URL https://arxiv.org/abs/2505.00979.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 14044–14072, 2024. doi: 10.18653/v1/2024.acl-long.757.

Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, Brett Larsen, Cody Blakeney, Charvi Bannur, Christina Baek, Darren Teh, David Schwab, Haakon Mongstad, Haoli Yin, Josh Wills, Kaleigh Mentzer, Luke Merrick, Ricardo Monti, Rishabh Adiga, Siddharth Joshi, Spandan Das, Zhengping Wang, Bogdan Gaza, Ari Morcos, and Matthew Leavitt. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining, 2025. URL https://arxiv.org/abs/2508.10975.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023. URL https://arxiv.org/abs/2212.10511.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023.

Christopher Mohri, John Duchi, and Tatsunori Hashimoto. A bitter lesson for data filtering, 2026. URL https://arxiv.org/abs/2605.19407.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems*, 2023.

Thiziri Nait Saada, Louis Bethune, Michal Klein, David Grangier, Marco Cuturi, and Pierre Ablin. The data-quality illusion: Rethinking classifier-based quality filtering for llm pretraining. *arXiv preprint arXiv:2510.00866*, 2025.

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 *Conference on Language Modeling*, 2025. arXiv:2506.04689.

Joel Niklaus, Guilherme Penedo, Hynek Kydlicek, Elie Bakouch, Lewis Tunstall, Ed Beeching, Thibaud Frere, Colin Raffel, Leandro von Werra, and Thomas Wolf. The synthetic data playbook: Generating trillions of the finest tokens, 2026.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026. Released July 9, 2026. Accessed: September 26, 2026.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 79155–79172. Curran Associates, Inc., 2023. doi: 10.52202/075280-3464. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/fa3ed726cc5073b9c31e3e49a807789c-Paper-Datasets_and_Benchmarks.pdf.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024.

Ru Peng, Kexin Yang, Yawen Zeng, Junyang Lin, Dayiheng Liu, and Junbo Zhao. Dataman: Data manager for pre-training large language models, 2025. URL https://arxiv.org/abs/2502.19363.

Michael Pieler, Marco Bellagente, Hannah Teufel, Duy Phung, Nathan Cooper, Jonathan Tow, Paulo Rocha, Reshinth Adithyan, Zaid Alyafeai, Nikhil Pinnaparaju, Maksym Zhuravinskyi, and Carlos Riquelme. Rephrasing natural text data with different languages and quality levels for large language model pre-training. In *Proceedings of the 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, volume 262, pp. 491–511. PMLR, 2024.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher, 2022. URL https://arxiv.org/abs/2112.11446.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Kashun Shum, Yuzhen Huang, Hongjian Zou, Qi Ding, Yixuan Liao, Xiaoxin Chen, Qian Liu, and Junxian He. Predictive data selection: The data that predicts is the data that teaches. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 55427–55450. PMLR, 2025.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024. URL https://arxiv.org/abs/2402.00159.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2459–2475, 2025. doi: 10.18653/v1/2025.acl-long.123.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (llms)? a.k.a. will llms replace knowledge graphs?, 2024. URL https://arxiv.org/abs/2308.10168.

Tristan Thrush, Sung Min Park, Herman Brunborg, Luke Bailey, Marcel Roed, Neil Band, Christopher Potts, and Tatsunori Hashimoto. Synthetic data for any differentiable target, 2026. URL https://arxiv.org/abs/2604.08423.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Arda Uzunoglu, Benjamin Van Durme, and Daniel Khashabi. Information abundance paradox: Long-context training undermines parametric knowledge, 2026. URL https://arxiv.org/abs/2608.12218.

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, 2024. URL https://arxiv.org/abs/2211.04325.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4003–4012, 2020.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 52915–52971. PMLR, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, 2023b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining, 2024. URL https://arxiv.org/abs/2409.07431.

Zitong Yang, Aonan Zhang, Hong Liu, Tatsunori Hashimoto, Emmanuel Candès, Chong Wang, and Ruoming Pang. Synthetic bootstrapped pretraining. In *International Conference on Learning Representations*, 2026.

Zichun Yu and Chenyan Xiong. Repro: Training language models to faithfully recycle the web for pretraining. *arXiv preprint arXiv:2510.10681*, 2025.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

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, 2025. URL https://arxiv.org/abs/2506.05176.

Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. Programming every example: Lifting pre-training data quality like experts at scale. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 79281–79327. PMLR, 2025.

A FINDING *Wasted Tokens*

C4 is derived from Common Crawl but applies basic language identification and heuristic cleaning (Dodge et al., 2021), removing trivial crawl artifacts while retaining a broad distribution of web documents. At the same time, it predates the stronger learned quality selection used by more recent pretraining corpora such as DCLM (Li et al., 2024) and FineWeb (Penedo et al., 2024). Hence, its low-quality tail provides a useful proxy for documents that survive basic preprocessing but would likely be removed by modern quality filters, constituting the regime we intend *Wasted Tokens* to capture.

As discussed in §2.1, we select the lowest-scoring 14B tokens from 200B C4 tokens using the DCLM FastText classifier. Our goal is not to reproduce the complete rejection set of DCLM, or of any particular pipeline, but to identify documents that quality-based selection would strongly disfavor. Figure 8 compares 100K document subsets of Wasted Tokens, C4, DCLM-RefinedWeb, and DCLM, showing that 100% of Wasted Tokens fall below the DCLM filtering threshold, compared to 91.13% of C4, 89.13% of DCLM-RefinedWeb, and none of DCLM. Thus, this construction provides a precise and deliberately stringent proxy for web data that would be filtered out by a modern pretraining pipeline.

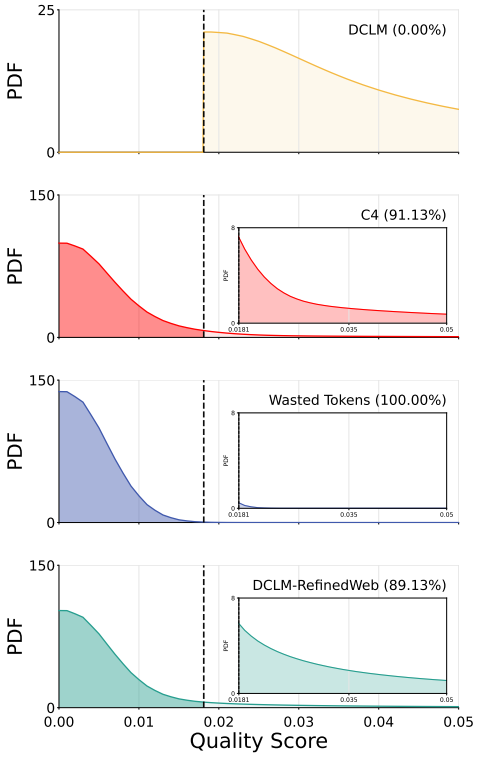

Figure 8: **Wasted Tokens come from the low-quality tail.** Distribution of DCLM FastText quality scores for DCLM, DCLM-RefinedWeb, C4, and Wasted Tokens. Percentages indicate the fraction of documents below the threshold.

Relation to prior recycling methods. This distinguishes our approach from prior recycling work regarding the *source distribution* that is transformed. Much of the synthetic pretraining literature operates on curated or already-filtered data (Gunasekar et al., 2023; Ben Allal et al., 2024; Hao et al., 2025; Niklaus et al., 2026). The closest analogues to our setting are therefore methods that expose rewriting to lower-quality documents. Even these methods, however, use broad or quality-stratified source pools rather than restricting sources to the lowest-quality tail: WRAP transforms C4 without quality conditioning (Maini et al., 2024), REWIRE uses the DCLM-RefinedWeb pool (Nguyen et al., 2025), and Nemotron-CC transforms documents across multiple quality strata (Su et al., 2025). As a result, their rewritten corpora mix source qualities, whereas documents in Upcycled Tokens (§2.1) originates from the lowest-scoring tail (Figure 8). Thus, even the closest prior work does not isolate the extreme low-quality tail as we do.

Importantly, distributional claims about recovering low-quality data should account for the base rate in the source pool. For example, only 81.7% of REWIRE's selected rewrites come from outside its raw top 10%, while that top 10% contributes 18.3% of selected rewrites (Nguyen et al., 2025). This is despite 89.13% of DCLM-RefinedWeb falling below the DCLM threshold in Figure 8, close to the 89.8% rejection rate reported by DCLM (Li et al., 2024). As a result, the raw top 10% contributes about $1.8\times$ its population share. Output filtering thus disproportionately favors higher-quality inputs rather than (preferentially) recovering the low-quality tail. This is consistent with prior evidence that rewriting gains depend on source quality (Pieler et al., 2024) and with Nemotron-CC's use of quality-specific transformations (Su et al., 2025). We instead fix source quality by drawing every source from the extreme low-quality tail and using all Upcycled Tokens without post-generation filtering.

Table 3: **Representative synthesizer development experiments.** Held-out next-token loss after pretraining a 150M proxy language model. Comparisons are most informative within blocks because the proxy pretraining setup changes over the course of development.

Configuration	Data	DCLM	FineWeb-Edu
Optimization			
Base (No Synthesizer Training)	25M×16	4.76	4.59
GRPO	50M×8	4.66	4.46
DPO	50M×8	**4.52**	**4.31**
DPO–RPO	50M×8	4.57	4.39
DPO → GRPO	50M×8	4.58	4.35
DPO–RPO → GRPO	50M×8	4.65	4.42
Preference construction			
5K pairs / 3K sources (160 steps)	50M×8	**4.52**	**4.31**
5K pairs / 3K sources (80 steps)	50M×8	4.59	4.38
8K pairs / 8K sources (160 steps)	50M×8	4.83	4.60
8K pairs / 8K sources (60 steps)	50M×8	4.60	4.36
Generation			
T=1.0**,** p=1.0**, no top-**k	50M×8	**4.52**	**4.31**
T=0.7, p=0.8, k=20	50M×8	5.41	5.01
Greedy	50M×8	5.46	5.06
Reward signal			
PreSelect (Shum et al., 2025)	50M×8	4.67	4.51
FineWeb-Edu Classifier (Penedo et al., 2024)	50M×8	4.57	4.26
DCLM FastText (Li et al., 2024)	50M×8	**4.37**	**4.15**
DataMan (Peng et al., 2025)	50M×8	4.47	4.20
Influence (Choe et al., 2024)	50M×8	4.44	4.25
Influence + DCLM FastText (§2.2)	50M×8	4.42	4.19
Repeated transformation			
Single Pass	50M×8	**4.42**	**4.19**
Double Pass	50M×8	4.73	4.38
Reference			
DCLM	25M×16	**4.16**	**4.22**

B Details for Synthesizer Training

The synthesizer training procedure in §2.2 reflects a broader set of development experiments over optimization algorithms, reward signals, preference construction, and generation hyperparameters. These experiments guide method design rather than constitute controlled ablations, since several components of the development setup change across iterations. We therefore focus on the qualitative trends that inform the final procedure and report representative development runs in Table 3.

Development protocol. We evaluate candidate synthesizers through a small-scale proxy pretraining loop. Unless otherwise noted, each synthesizer generates approximately 50M tokens, which we repeat eight times to train a 150M Llama-2 proxy model on approximately 400M tokens. We evaluate held-out next-token loss on 5K DCLM and 5K FineWeb-Edu documents. This measures the downstream pretraining utility of the generated data rather than the synthesizer's training objective itself. The proxy pretraining configuration also changes during development, so comparisons are most informative within the corresponding blocks of the table.

Optimization. The *Optimization* block of Table 3 compares the base SmolLM2 rewriter, GRPO, DPO, DPO–RPO, and additional GRPO optimization applied after preference optimization. DPO provides the strongest proxy performance among these configurations, while the additional optimization objectives provide no consistent improvement. We therefore use DPO as the synthesizer optimization objective in §2.2.

Preference construction and rewards. The *Preference construction* block of Table 3 shows that increasing the number of preference pairs and source diversity does not necessarily improve synthetic-data utility, motivating our use of a compact set of high-margin preferences. The *Reward signal* block compares several standalone quality and utility signals together with the strongest FastText–influence combination. DCLM FastText provides the strongest standalone signal, while the combined variant remains competitive in these short proxy runs. In longer proxy runs, however, the combined reward consistently outperforms DCLM FastText alone. We therefore use quality as the primary ranking criterion in §2.2 and use influence to distinguish candidates with similar quality.

Generation. The *Generation* block of Table 3 shows that diverse stochastic sampling substantially outperforms restrictive sampling and greedy decoding, which motivates the unconstrained sampling configuration used for the final Upcycled Tokens. The *Repeated transformation* block also provides no benefit from transforming the same transformed documents themselves.

C Pretraining Details

We base our model architectures and optimization setup on Kim et al. (2025). Our 55M, 152M, and 670M models follow their nominal 150M, 300M, and 600M architecture presets, respectively. The lower parameter counts primarily reflect our use of the Llama-2 tokenizer with a 32,000-token vocabulary rather than the larger vocabulary used in their implementation. At the smallest scale, we proportionally reduce the model dimensions following the architectural scaling patterns of the larger configurations. At the largest scale, we retain the hidden size, intermediate size, and attention configuration of the 1.4B preset, but increase the depth from 16 to 24 layers.

Table 4: Architecture, optimization, and parallelism hyperparameters for each model scale.

	19M	**55M**	**152M**	**670M**	**1.4B**
Architecture					
Hidden size	256	512	768	1024	2048
Intermediate size	896	1792	2688	7168	7168
Layers	3	6	12	24	24
Attn. heads (Q)	4	8	12	16	16
Attn. heads (KV)	4	8	12	8	8
Optimization					
Peak LR	2.5×10^{-3}	2.5×10^{-3}	2.5×10^{-3}	1×10^{-3}	3×10^{-4}
Min LR	2×10^{-4}	2×10^{-4}	2×10^{-4}	1×10^{-4}	3×10^{-5}
Weight decay	1.6	1.6	1.6	1.6	3.2
Warmup steps	64	64	64	256	256
Parallelism					
Micro-batch size	16	16	16	8	2
Gradient accumulation	16	4	4	8	32
Data parallelism	1	4	4	4	4

We use the Llama-2 tokenizer with a vocabulary of 32,000 tokens and follow the Llama-2 architecture (Touvron et al., 2023) with SwiGLU activations (Shazeer, 2020), rotary position embeddings (Su et al., 2023), RMSNorm (Zhang & Sennrich, 2019), and untied input and output embeddings. The 19M, 55M, and 152M models use multi-head attention (Cordonnier et al., 2021). The 670M and 1.4B models use grouped-query attention (Ainslie et al., 2023) with two query heads per key-value head. All models use a context length of 4096 tokens, which is near-optimal for parametric knowledge acquisition (Uzunoglu et al., 2026) and is therefore well suited to the knowledge-intensive evaluations in §3, §4.1, and §4.2.

We use the AdamW (Loshchilov & Hutter, 2019) hyperparameters from Kim et al. (2025), including $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and gradient clipping at 1.0. We modify their setup in batch size, learning rate schedule, initialization, and scale-specific weight decay. We use a global batch size of 256 sequences, corresponding to 1.05M tokens, with a fixed warmup followed by cosine decay to a nonzero minimum learning rate. We initialize model parameters with a standard deviation of

0.02. All models are trained in bfloat16 with gradient accumulation in fp32 and use data parallelism without tensor or pipeline parallelism. Table 4 summarizes the complete configuration.

D Analysis

D.1 Upcycling Makes Text More Predictable

We characterize how upcycling changes the statistical structure of text using Qwen3-0.6B-Base (Yang et al., 2025) as an independently pretrained reference model. Importantly, token surprisal is directly aligned with the language modeling objective and has been used to identify more typical or useful pretraining data (Marion et al., 2023; Ankner et al., 2024). For 100K paired Wasted and Upcycled documents, together with 100K Retained documents as a high-quality reference, we therefore measure mean token surprisal, $-\log_2 p_\theta(x_t \mid x_{<t})$, averaging each quantity over tokens within a document.

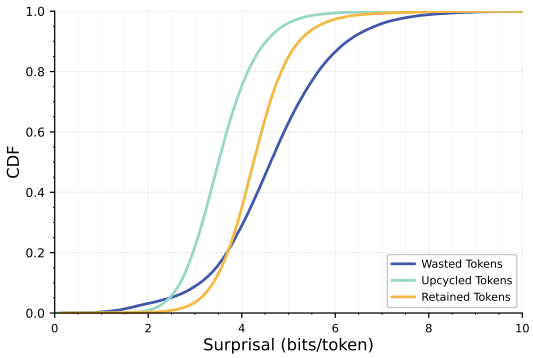

Figure 9: **Upcycling substantially reduces reference-model surprisal.** CDF of mean token surprisal under Qwen3-0.6B-Base for Wasted, Upcycled, and Retained Tokens.

Figure 9 shows a pronounced shift in reference-model surprisal after upcycling, as Upcycled Tokens are substantially less surprising than their Wasted counterparts across most of the distribution and, notably, are also less surprising than Retained Tokens. This suggests that upcycling removes some of the idiosyncratic or difficult-to-model structure in Wasted Tokens, producing text whose realized continuations are more predictable under an independently trained language model.

D.2 Upcycling Preserves the Source

Upcycling substantially changes the predictability of Wasted Tokens (§D.1), raising the question of whether these gains come from preserving and improving the source content or replacing it with largely unrelated text. Following Nguyen et al. (2025), we measure semantic preservation through the cosine similarity between document embeddings. We embed each document from Wasted Tokens and its corresponding Upcycled version using Qwen3-Embedding-8B (Zhang et al., 2025). As reference points, we additionally compute similarities between randomly paired Wasted documents and between randomly paired Wasted and Retained documents.

Figure 10 shows that Upcycled Tokens remain substantially closer to their corresponding Wasted sources than unrelated documents are to one another. The Wasted–Upcycled distribution is concentrated around cosine similarities above 0.6, compared with roughly 0.2 for both random-pair baselines. Thus, despite substantially improving the pretraining utility and predictability of Wasted Tokens, upcycling largely preserves their underlying semantic content. Together with §D.1, these results suggest that upcycling improves how the source information is expressed rather than simply replacing discarded documents with unrelated synthetic content.

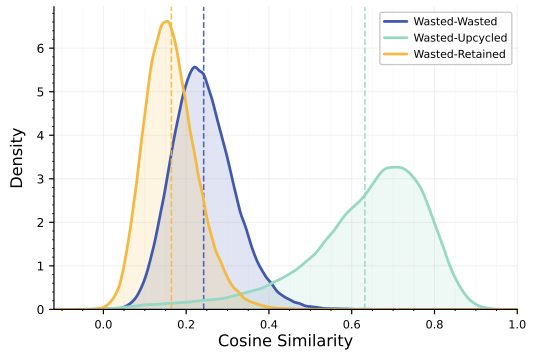

Figure 10: **Upcycling largely preserves the semantic content of the source.** Distribution of cosine similarities between Qwen3-Embedding-8B document embeddings.

E NoKLB Benchmark Construction

This section describes the construction of NoKLB. The pipeline identifies durable, source-grounded factual knowledge in Wasted Tokens and converts it into independently verifiable multiple-choice questions. Table 5 summarizes the number of examples retained at each stage.

Source document selection. We begin with 26,215,415 documents in the Wasted Tokens corpus, consisting of C4 documents rejected by DCLM quality selection. We remove documents shorter than 128 characters and score the remainder with the Fineweb-Edu classifier model.[3] We retain 42,100 documents with FineWeb-Edu scores above 3.0. We then use Llama-3.1-70B-Instruct (Grattafiori et al., 2024) to perform binary classification of whether each document contains valuable factual knowledge, and retain 20,096 documents classified as positive for subsequent benchmark construction.

Knowledge identification and local grounding. The preceding stage retains 20,096 documents that Llama-3.1-70B-Instruct classifies as containing valuable information. We further restrict this set to high-confidence positive predictions, yielding 7,164 documents. These documents contain durable, verifiable, and non-trivial information suitable for reusable factual questions, rather than content dominated by advertising, boilerplate, navigation, resource descriptions, generic marketing, or other source-specific material.

Table 5: **Construction of the NoKLB benchmark.** Number of documents and question–answer pairs remaining after each stage of the curation pipeline. The final benchmark contains 2,050 externally verified four-way multiple-choice questions.

Stage	Remaining
Documents	
Documents in Wasted Tokens	26,215,415
FineWeb-Edu Score > 3.0	42,100
Binary Llama-3.1-70B-Instruct Classification	20,096
QA Pairs	
Grounded QA Pairs	21,770
Encyclopedic Filtering	14,040
Knowledge Depth Filtering	6,813
Natural Language and Mathematics Filtering	6,396
External Verification Filtering	2,051
Final MCQA Questions	2,050

For each retained document, Llama-3.1-70B-Instruct generates candidate question-answer pairs. We then run a separate grounding pass with the same model and retain a pair only when the source document explicitly and completely supports its answer. This produces 21,770 source-grounded question-answer pairs.

Question-answer pair curation. We apply a sequence of pair-level filters to remove questions that test passage comprehension, shallow knowledge, or content outside the intended scope of NoKLB.

The first stage tests whether a question is suitable as an encyclopedic knowledge question. We retain questions that are self-contained, objective, and durable while removing passage-dependent questions, ephemeral claims, opinions, source-specific details, and instance-specific exercises. This criterion yields 14,040 pairs. We then remove introductory, dictionary-like, and otherwise shallow facts. This leaves 6,813 questions containing more specialized or substantive knowledge. Finally, we exclude questions whose primary content concerns natural-language knowledge or mathematics, leaving 6,396 candidate pairs for external verification.

External verification. We independently verify candidate answers using GPT-5.6 Luna (OpenAI, 2026) with web search. The verifier is instructed to consult only reliable external sources and returns one of four structured verdicts, *externally supported*, *contradicted*, *insufficient evidence*, or *time sensitive*.

After retaining only externally supported examples that satisfy our verification criteria, 2,051 question-answer pairs remain. We preserve the complete verification record for every retained example so that the external evidence supporting each decision remains traceable.

[3] https://huggingface.co/HuggingFaceFW/fineweb-edu-classifier

Multiple-choice conversion. For each retained question, Llama-3.1-70B-Instruct generates three plausible incorrect answers. Distractors must match the semantic type and approximate specificity of the correct answer. We reject generations containing duplicate choices, reproducing the correct answer, or violating the required output format and regenerate. One example fails these checks, producing the final 2,050-item four-way NoKLB benchmark.

All prompts used for document-level curation, question–answer generation, local grounding, pair-level filtering, external verification, and distractor generation are provided in our accompanying code repository.

F Prompts

In this section, we provide the prompts used to generate synthetic data in our experiments.

Prompt in §3. For our main experiments, we adopt the *rephrase prompt* from Kim et al. (2026). The prompt is provided below.

Rephrase Prompt

System Prompt
Provide a direct response to the instructions without adding additional notes.

User Prompt
For the following document, regardless of its original content or formatting, write a full article of the same content in high quality English language as in texts on Wikipedia:

Document: {`document`}

Rephrased article:

Prompt in §4.2. For the experiments in §4.2, we use the *Table prompt* from Niklaus et al. (2026). The prompt restructures each source document as a table and generates one question-answer pair grounded in the resulting table. The prompt is provided below.

Table Prompt

User Prompt
Rewrite the document as a structured table that organizes the key information, then generate one question-answer pair based on the table. First extract the main data points and organize them into a clear table format with appropriate headers using markdown table syntax with proper alignment. After the table, generate one insightful question that can be answered using the table data. Provide a clear, concise answer to the question based on the information in the table. Output only the table followed by the question-answer pair, nothing else.

Document: {`document`}

G Representative Upcycling Examples

To qualitatively illustrate the transformations produced by our synthesizer, we show representative pairs of documents from Wasted Tokens and Upcycled Tokens below. We select examples spanning several common forms of low-quality web data, including navigational boilerplate, promotional content, catalog-style text, and repetitive user-generated comments. These examples illustrate how upcycling restructures fragmented or repetitive source material into more coherent text while retaining the central content of the source.

Example 1: Navigational boilerplate.

Wasted Tokens

To Register for a League, simply click the Register button below the League name. For more information on each Sports League, click the League name.
To view the Leagues, click your Sport below.
Be sure to Sign-up for the TRSL Newsletter and receive news on upcoming Leagues, important events and more delivered right to your inbox!
Read the League Rules for Sports Leagues offered by Toronto Recreational Sports League.

Upcycled Tokens

Registration and Access to Sports Leagues
For individuals interested in joining a sports league, the process is relatively straightforward and streamlined. One simple way to navigate this process is by clicking the "Register" button provided for each given league's name, or alternatively, by accessing the league's details by its corresponding sport.
Additionally, users can visit the sports league's homepage to gain a comprehensive understanding of each league's guidelines, including their respective rules and regulations. Moreover, it's beneficial to subscribe to the TRSL newsletter, ensuring that users stay updated with all the latest information on upcoming leagues, significant events, and other noteworthy activities.
The official site of Toronto Recreational Sports League provides a detailed breakdown of all the sports leagues they offer. By registering for a league, one can gain access to competitive and enjoyable sports activities, effectively aligned with personal interests and goals.
In summary, the registration process is intuitive and allows users to effortlessly connect with various sports leagues offered by Toronto Recreational Sports League. From there, they can receive detailed information on each league, including rules and regulations, which aids in informed decisions when choosing a sports league to join.

Example 2: Promotional deal page.

Wasted Tokens

Many Melissa and Doug Toys are 50% Off for TODAY ONLY!
Today only, you can get 50% Off Melissa and Doug Toys!! on Amazon.com!
Melissa & Doug Pattern Blocks and Boards $10!
Melissa & Doug Ice Cream Scoop Set $15!
Melissa & Doug Fishing And Bug Catching Magnetic Game $10!
Click here to see all of the great 50% off Melissa and Doug toys!
Click here for more featured coupons and deals!

Upcycled Tokens

Melissa and Doug, a well-known toy brand, is currently offering a significant discount on a variety of their popular toys. The company's products, including the Pattern Blocks and Boards, the Ice Cream Scoop Set, and the Fishing and Bug Catching Magnetic Game, are priced at 50% off for a limited time. Additionally, Amazon.com has partnered with Melissa and Doug to offer these discounted items.
The discounted prices, ranging from $10 to $15, make these Melissa and Doug toys an attractive option for parents and children looking to buy quality toys at a lower cost. Furthermore, the deal is being promoted on Amazon.com, ensuring maximum visibility and accessibility for potential customers.
In conclusion, Melissa and Doug's significant discount on their popular toys is a great opportunity for families to snag a bargain while ensuring their kids enjoy high-quality play experiences. For the full list of discounted Melissa and Doug toys, you can visit their website or check the coupons and deals section currently available.

Example 3: Catalog-style commercial text.

Wasted Tokens

We carry the largest selection in Arizona of Autographs, Sports Collectibles, Sports Memorabilia, Vintage Sports Cards, Supplies, Display Cases, Sports Gifts, and Novelties.
Hockey Cards (1920–1980), Non-Sports Cards (1880 to 1970), Sports Memorabilia, Sports Collectibles, Game Used Equipment, and Quality Autographs.
Card Issues most wanted are N Cards, T cards, E Cards, Goudey Gum, Play Ball, Topps, Bowman, Fleer, Post Cereal, and many more.
We are always looking to purchase large collections. No collection is too big!

Upcycled Tokens

Our company has a vast and impressive collection of sports memorabilia, including rare and coveted autographs, sports cards, sports equipment, and other unique collectibles. Our extensive range caters to enthusiasts and collectors, ensuring that everyone can find something that interests them.
We specialize in carrying the largest selection of hockey, non-sports, sports memorabilia, and sports collectibles in Arizona. Additionally, our inventory features a diverse array of vintage sports cards, supplies, sports gifts, and sports-themed novelties.
The specific focus of our collection includes 1920–1980 hockey cards, as well as non-sports cards produced from 1880 to 1970. Furthermore, we provide a comprehensive selection of sports memorabilia, ranging from game-used equipment to high-quality autographs.
Our goal is to continually expand our collection, incorporating larger-than-life items and rare artifacts. What's more, our team is always open to potential buyers and collectors, encouraging them to explore our extensive inventory.

Example 4: Repetitive user-generated comments.

Wasted Tokens

your card is absolutely fantastic!! very beautiful! Thanks for playing this weeks Stampin' Royalty Challenge!
Oh, so pretty Narelle. I love that color combo. The cut out tag looks fabulous! Thank you for playing along with us at The Sweet Stop.
What a great card!! Thanks so much for joining us this week at The Sweet Stop.
That is really a great card, I love how you used the sketch and great job with the color challenge. Thanks for playing along with us at Stampin' Royalty!
Great use of teh that sketch and colours. I like the hand cut sentiment! Thanks for playing with Stampin' Royalty this week!
Nice combo of challenges! Love the embossed panel! Thanks for playing along with us at the Sweet Stop this week!

Upcycled Tokens

The card is truly exceptional, with an undeniable sense of beauty. The use of the sketch and the impressive teamwork in the color challenge make it a standout piece. The hand-cut sentiment adds a special touch, emphasizing the unique value of the card.
The challenge was managed exceptionally well by the participant. They used a clever combination of embossed panels and various colors to enhance the card. The result is an exquisite creation that showcases their artistic skills and dedication.
Thanks for participating in the Stampin' Royalty Challenge. We hope you continue to embrace the challenges and enjoy the process of creating beautiful cards. Keep up the fantastic work!

H Complete Results

We report the numerical results underlying the figures throughout the main paper. For held-out loss, values are reported as mean cross-entropy with standard error. For CORE, we report the mean score.

We are unable to report uncertainity of results for 1.4B parameters, since they are based on a single training run.

H.1 Upcycling Wasted Tokens (§3)

Tables 6, 7, and 8 report the results corresponding to §3.

Table 6: Held-out DCLM loss for the main experiments. Values report mean cross-entropy with standard error.

TPR	Corpus	19M	55M	152M	670M	1.4B
10	Wasted	$5.545_{\pm 0.005}$	$4.880_{\pm 0.007}$	$4.161_{\pm 0.007}$	$3.667_{\pm 0.008}$	$3.472_{\pm 0.007}$
10	Upcycled	$5.174_{\pm 0.003}$	$4.475_{\pm 0.007}$	$3.757_{\pm 0.007}$	$3.229_{\pm 0.007}$	$3.072_{\pm 0.007}$
10	Retained	$5.033_{\pm 0.007}$	$4.298_{\pm 0.006}$	$3.565_{\pm 0.006}$	$2.983_{\pm 0.007}$	$2.800_{\pm 0.006}$
20	Retained + Wasted	$4.754_{\pm 0.007}$	$3.993_{\pm 0.006}$	$3.397_{\pm 0.006}$	$3.011_{\pm 0.007}$	$2.808_{\pm 0.006}$
20	Retained + Upcycled	$4.698_{\pm 0.007}$	$3.939_{\pm 0.006}$	$3.332_{\pm 0.007}$	$2.949_{\pm 0.007}$	$2.754_{\pm 0.006}$
20	Retained + Retained	$4.664_{\pm 0.007}$	$3.937_{\pm 0.006}$	$3.310_{\pm 0.006}$	$2.923_{\pm 0.007}$	$2.734_{\pm 0.006}$

Table 7: Held-out FineWeb-Edu loss for the main experiments. Values report mean cross-entropy with standard error.

TPR	Corpus	19M	55M	152M	670M	1.4B
10	Wasted	$5.443_{\pm 0.009}$	$4.668_{\pm 0.007}$	$3.920_{\pm 0.007}$	$3.387_{\pm 0.008}$	$3.174_{\pm 0.008}$
10	Upcycled	$5.176_{\pm 0.003}$	$4.381_{\pm 0.008}$	$3.637_{\pm 0.007}$	$3.101_{\pm 0.008}$	$2.944_{\pm 0.007}$
10	Retained	$5.168_{\pm 0.008}$	$4.339_{\pm 0.007}$	$3.575_{\pm 0.007}$	$2.969_{\pm 0.008}$	$2.792_{\pm 0.007}$
20	Retained + Wasted	$4.843_{\pm 0.006}$	$4.008_{\pm 0.006}$	$3.383_{\pm 0.007}$	$2.981_{\pm 0.008}$	$2.778_{\pm 0.007}$
20	Retained + Upcycled	$4.757_{\pm 0.007}$	$3.905_{\pm 0.007}$	$3.289_{\pm 0.007}$	$2.902_{\pm 0.008}$	$2.730_{\pm 0.007}$
20	Retained + Retained	$4.773_{\pm 0.008}$	$3.951_{\pm 0.007}$	$3.306_{\pm 0.007}$	$2.910_{\pm 0.008}$	$2.718_{\pm 0.007}$

Table 8: Centered CORE performance for the main experiments. Dashes indicate unavailable results.

TPR	Corpus	19M	55M	152M	670M	1.4B
10	Wasted	–	−0.036	0.005	0.037	0.073
10	Upcycled	–	−0.022	0.007	0.061	0.082
10	Retained	–	−0.026	0.007	0.059	0.085
20	Retained + Wasted	–	−0.026	0.028	0.060	0.089
20	Retained + Upcycled	–	−0.022	0.034	0.078	0.103
20	Retained + Retained	–	−0.021	0.035	0.073	0.106

H.2 Upcycling Improves Existing Synthetic Data Recipes (§4.2)

Table 9 reports the numerical results corresponding to §4.2. *Table* denotes the transformation from Niklaus et al. (2026), while *Table (Ours)* uses the same transformation with our specialized synthesizer.

Table 9: Centered CORE performance for the Table transformation with the original and specialized synthesizers.

Corpus	670M	1.4B
Wasted	0.037	0.073
Table	0.045	0.078
Table (Ours)	0.062	0.086
Retained + Wasted	0.060	0.089
Retained + Table	0.069	0.097
Retained + Table (Ours)	0.084	0.108

H.3 Upcycling Remains Effective with Smaller Synthesizers (§4.3)

Tables 10, 11, and 12 report the numerical results corresponding to §4.3. We compare Upcycled Tokens generated by our 1.7B synthesizer with those generated by the 360M synthesizer.

Table 10: Held-out DCLM loss for the 1.7B and 360M synthesizers. Values report mean cross-entropy with standard error.

TPR	Corpus	670M	1.4B
10	Upcycled (1.7B)	$3.229_{\pm 0.007}$	$3.072_{\pm 0.007}$
10	Upcycled (360M)	$3.245_{\pm 0.007}$	$3.091_{\pm 0.006}$
10	Retained	$2.983_{\pm 0.007}$	$2.800_{\pm 0.006}$
20	Retained + Upcycled (1.7B)	$2.949_{\pm 0.007}$	$2.754_{\pm 0.006}$
20	Retained + Upcycled (360M)	$2.962_{\pm 0.007}$	$2.761_{\pm 0.007}$
20	Retained + Retained	$2.923_{\pm 0.007}$	$2.734_{\pm 0.006}$

Table 11: Held-out FineWeb-Edu loss for the 1.7B and 360M synthesizers. Values report mean cross-entropy with standard error.

TPR	Corpus	670M	1.4B
10	Upcycled (1.7B)	$3.101_{\pm 0.008}$	$2.944_{\pm 0.007}$
10	Upcycled (360M)	$3.078_{\pm 0.008}$	$2.916_{\pm 0.007}$
10	Retained	$2.969_{\pm 0.008}$	$2.792_{\pm 0.007}$
20	Retained + Upcycled (1.7B)	$2.902_{\pm 0.008}$	$2.730_{\pm 0.007}$
20	Retained + Upcycled (360M)	$2.911_{\pm 0.008}$	$2.742_{\pm 0.008}$
20	Retained + Retained	$2.910_{\pm 0.008}$	$2.718_{\pm 0.007}$

Table 12: Centered CORE performance for the 1.7B and 360M synthesizers.

TPR	Corpus	670M	1.4B
10	Upcycled (1.7B)	0.061	0.082
10	Upcycled (360M)	0.057	0.080
10	Retained	0.059	0.085
20	Retained + Upcycled (1.7B)	0.078	0.103
20	Retained + Upcycled (360M)	0.070	0.099
20	Retained + Retained	0.073	0.106