# Lightweight Surrogate-Assisted Language Model Pretraining

**Nathan Truong** NATHANNGOCTRUONG@GMAIL.COM
*Independent, California, United States*

## Abstract

Standard causal language model pretraining uses a single-label cross-entropy objective that ignores the existence of multiple valid next-token continuations, resulting in sample inefficiency. Our initial analyses of language model outputs reveal that only a handful of tokens carry meaningful signal compared to the rest of the vocabulary. Predicated on this finding, we introduce a multi-label pretraining objective that modifies the loss to append a small set of context-conforming auxiliary tokens selected by a lightweight surrogate language model. Distinct from existing knowledge distillation methods, the surrogate is used only for token selection and suggestion rather than full distribution matching. We empirically show that adopting this multi-target stance achieves superior performance on benchmarks with notably less FLOPs utilization and tokens in our tested setting.

## 1. Introduction

It has been well established in recent years that scaling compute, model size, and training data can consistently improve model performance in next-token prediction and downstream tasks [3, 10, 12]. However, the act of scaling these subjects alone often incurs nontrivial costs, ultimately hindering faster training altogether. Incident to this, prevailing works have leveraged a teacher model to achieve enhanced model compression [8, 9], effectively reducing inference and training expenses.

Although knowledge distillation has been empirically proven to work in practice, inefficiencies regarding the teacher model's output distribution hinder further savings [18]. Conventional causal language modeling assumes that a singular target is the only valid continuation for a given sequence. While knowledge distillation forces a base model to match a distribution that is not typically "one-hot", it ignores the fact that most of the reference distribution is noise (Table 1). Multi-label learning [2] offers an alternative: assigning multiple plausible labels per example rather than a single target. Label smoothing [16, 21] achieves this in a nominal manner, but explicit applications of multi-label learning to LLM pretraining remain relatively untapped.

To quantify this, we swept over three language models of varying sizes and found that only a small subset of tokens within a model's vocabulary carries a meaningful probability mass. Motivated by this observation, we propose lightweight surrogate-assisted language model pretraining (LSAP): the usage of cheap reference models to enhance pretraining efficiency via suggesting plausible targets. That is, given an initial sequence and a target, we translate it to the surrogate's tokenizer using preconstructed mappings, pass the context and the translated version to the base and surrogate models respectively, and select tokens from the surrogate's output distribution that are assigned probabilities higher than a specified $\alpha$. Those tokens are then appended into the base model's loss objective as negative log likelihoods. We provide a diagram in Appendix A.

We empirically show that LSAP induces efficiency gains by pretraining OLMo2-1B from scratch on roughly 26B tokens from SlimPajama while leveraging Qwen3-0.6B as the reference LM. At

© N. Truong.

identical training FLOPs, our method improves the average zero-shot benchmark by $\sim$3.23% over the standard cross-entropy baseline while utilizing 14.4% fewer training tokens. Remarkably, challenging the standard knowledge-distillation assumption that teacher language models should be larger than the student, LSAP shows that the converse possesses tangible benefits.

## 2. Related Works

**Pretraining Data Optimization** Methods aimed at optimizing pretraining data have targeted distribution matching [3], quality filtering [6, 25], and perplexity-based pruning [1, 25]. However, scoring is typically storage-intensive and fixed thresholds can either admit harmful tokens or exclude beneficial ones.

**Knowledge Distillation (KD)** Hinton et al. [9] introduced training a smaller model to mimic a teacher's predictions, later adapted to causal language modeling by Kim and Rush [13]. Recent work has shown that teacher output distributions are sharply peaked, making tail probabilities contribute negligibly to the KL objective [5, 18]. Fixes include reweighting the KL divergence [5] or retaining only top-$k$ logits [18, 24]. While small LMs have been used to aid pretraining via KD [19], they still rely on distribution matching. By contrast, LSAP uses a lightweight surrogate solely for target selection, dynamically scaling auxiliary targets via a probability threshold rather than a fixed $k$.

**Multi-Label Learning** Standard supervised learning assumes unique class assignments [2], yet multi-label objectives allow multiple plausible labels per example [23]. In NLP, output token distributions exhibit lower entropy at larger scales [15], suggesting fewer tokens carry meaningful signal. To our knowledge, no prior work has introduced an auxiliary pretraining loss that is inherently multi-label through sampling from a lightweight surrogate's output distribution.

**Weak-to-Strong Generalization** Weak-to-Strong generalization provides a framework in which weak models are used to *supervise* stronger models by generating labels during fine-tuning [4]. The primary objective is to yield a fine-tuned model that is stronger than the one generating its labels. Although there has been work regarding WTSG's applications to pretraining, they typically involve data selection or data mixture optimization rather than providing alternative labels on the fly [11, 26]. By contrast, LSAP does not perform dataset filtering or curation, but provides surrogate label guidance in the pretraining regime. Moreover, LSAP does not directly optimize for superior performance over a teacher model and instead uses its suggested targets as a reference while still utilizing the underlying pretraining dataset.

## 3. Methodology

**Empirical Motivations** As shown in Table 1, when $\alpha$ is fixed to a value as low as 0.01, only roughly 8.83 tokens on average are assigned probabilities above that threshold. This concentration of probability mass suggests that the KL divergence in standard KD sums over thousands of tokens whose tail probabilities contribute negligibly. However, choosing an $\alpha$ too small could induce the selection of incorrect targets while an $\alpha$ too large could yield nominal gains. We compare three $\alpha$ values in Section 4.

**Causal Language Modeling** To begin, let $\mathcal{T}_b : \Sigma_b \to V_b$ denote the base model's tokenizer where $\Sigma_b$ and $V_b$ represents the set of token strings and the token ID's respectively. Provided an input

Table 1: **Average number of tokens above a probability threshold $\alpha$.** All models are evaluated on 5000 examples from each dataset. Each entry is averaged over seeds 21, 42, and 67. Here, WikiText (WT), OpenWebText (OWT), and FineWeb-Edu (FW-E) have been abbreviated for brevity and textual coherence.

| Model | $\alpha$ | WT | OWT | FW-E | ArXiv | C4 | IMDb | PG19 | AVG |
|---|---|---|---|---|---|---|---|---|---|
| Qwen3-0.6B | 0.01 | 9.26 | 9.77 | 9.37 | 9.54 | 9.91 | 10.37 | 8.94 | 9.59 |
| Qwen3-0.6B | 0.05 | 2.77 | 2.85 | 2.84 | 2.60 | 2.86 | 2.81 | 2.39 | 2.73 |
| Qwen3-0.6B | 0.2 | 0.87 | 0.86 | 0.91 | 0.83 | 0.84 | 0.78 | 0.81 | 0.84 |
| Llama-3.1-8B | 0.01 | 8.61 | 8.82 | 8.69 | 9.16 | 8.81 | 9.89 | 6.83 | 8.69 |
| Llama-3.1-8B | 0.05 | 2.77 | 2.85 | 2.84 | 2.74 | 2.79 | 2.94 | 2.29 | 3.14 |
| Llama-3.1-8B | 0.2 | 0.99 | 1.00 | 1.02 | 0.94 | 0.98 | 0.92 | 0.99 | 0.98 |
| Mistral-7B-v0.3 | 0.01 | 8.33 | 7.94 | 8.02 | 9.01 | 8.51 | 9.32 | 6.32 | 8.21 |
| Mistral-7B-v0.3 | 0.05 | 2.75 | 2.68 | 2.72 | 2.71 | 2.78 | 2.84 | 2.14 | 2.66 |
| Mistral-7B-v0.3 | 0.2 | 0.98 | 1.03 | 1.04 | 0.95 | 1.00 | 0.95 | 1.03 | 1.00 |

sequence $x = (x_1, x_2, \ldots, x_N)$ where $x_i \in V_b^*$ standard causal language modeling paradigms employ the following cross-entropy loss:

$$\mathcal{L}_{CLM}(\theta; x) = -\frac{1}{N}\sum_{i=1}^{N} \log P_\theta(x_i|x_{<i})$$

In this setting, $L_{CLM}(\theta; x)$ denotes the loss function parameterized by $\theta$ with an input sequence $x \in \mathcal{X}$ while $x_{<i}$ refers to $x_i$'s preceding tokens.

**Multi-Label Objective** Unlike conventional logit-based knowledge distillation [9], our proposal alters the loss function by appending additional loss objectives (i.e., pure negative log likelihoods rather than the KL-divergence) using a surrogate model $\mathcal{S}$. However, there may exist discrepancies in tokenization. For instance, the word "because" may be represented as a single token within the base model's vocabulary while split into "be" + "cause" for the surrogate's. To mitigate this issue, we only use tokens that exist in the intersection between both vocabularies. With $\mathcal{T}_s : \Sigma_s \to V_s$ being the surrogate model's tokenizer, let $\Sigma_b$ and $\Sigma_s$ denote the token string sets of $\mathcal{T}_b$ and $\mathcal{T}_s$ respectively. Then, $\Sigma_\cap := \Sigma_b \cap \Sigma_s$. We then establish translation mappings $\mathcal{F}_{b\to s} : \mathcal{T}_b(\Sigma_\cap) \to \mathcal{T}_s(V_s)$ for encoding inputs and $\mathcal{F}_{s\to b} : \mathcal{T}_s(\Sigma_\cap) \to \mathcal{T}_b(\Sigma_\cap)$ for decoding logits. Then, we can apply the following filters to the inputs and outputs to mitigate tokenizer discrepancies:

$$\mathcal{F}_{b\to s}(x_i) = \begin{cases} \mathcal{T}_{b\to s}(x_i) & \text{if } x_i \in \mathcal{T}_b(\Sigma_\cap) \\ -100 & \text{if } x_i \notin \mathcal{T}_b(\Sigma_\cap) \end{cases}, \quad \mathcal{F}_{s\to b}(v_i) = \begin{cases} \mathcal{T}_{s\to b}(v_i) & \text{if } v_i \in \mathcal{T}_s(\Sigma_\cap) \\ -\infty & \text{otherwise} \end{cases}$$

Here, $v_i \in V_s$ denotes surrogate token indices. In our experiments, -100 denotes the padding token id for the surrogate model's input mask, which forces tokens not existing within $\Sigma_\cap$ to be ignored by the attention mechanism. This imposes a constraint upon our proposal: the cardinality of $\Sigma_\cap$ should be similar to that of $\Sigma_b$ in order to retain input quality when passed to the surrogate.

For a provided target token $x_i$ and preceding tokens $x_{<i}$, we use the $\mathcal{S}$'s output logits to select tokens that fall above a probability threshold $a \in [0, 1]$. To this end, passing the translated sequence $\mathcal{F}_{b\to s}(x_{<i})$ into $\mathcal{S}$ yields an output distribution over $V_s$. We set the probability of the target $\mathcal{F}_{b\to s}(x_i)$

and tokens not in $\Sigma_\cap$ to $-\infty$ to avoid selection. For a given target and an array of surrogate target token ids $A$ formally shown as

$$A_i = \{v \in \mathcal{T}_s(\Sigma_\cap) : v \neq \mathcal{F}_{b\to s}(x_i),\ P_{\text{surrogate}}(v \mid \mathcal{F}_{b\to s}(x_{<i})) > \alpha\},$$

the auxiliary loss objective can be constructed in the following manner:

$$\mathcal{L}_{surrogate} = -\sum_{i=1}^{N}\sum_{v\in A} \log P_\theta(\mathcal{F}_{s\to b}(v)|x_{<i})$$

Here, $A_i$ denotes the set of alternative target tokens generated from the surrogate model for a given token position $i$ while $\mathcal{F}_{s\to b}(v_i)$ denotes a target selected by the surrogate translated to the base model's vocabulary. With this, we also acknowledge the fact that the validity of the surrogate's outputs are crucial to our method's success. If the reference model delegates high probabilities to tokens that are factually incorrect, the base model will leverage those indices nonetheless. To partially offset this issue, we multiply the auxiliary loss by a scaling factor following a cosine-annealing scheduler $\lambda$ to achieve the combined loss:

$$\mathcal{L}_{LSAP}(\theta; x) = \frac{1}{N}\mathcal{L}_{CLM} + \frac{1}{M}\lambda(t)\mathcal{L}_{surrogate}$$

For this specific formulation, $M$ is calculated as $\sum_{i=1}^{N} |A_i|$ and represents the total number of auxiliary targets selected for a sequence. The scheduler would prevent contaminating downstream training inputs with potentially false objectives. This allows the data distribution to dominate during later stages of training where accuracy is more important with respect to downstream performance [27].

## 4. Experiments

### 4.1. Setup

**Baseline Setting** For our baseline, we train OLMo2-1B [17] on ∼26 billion tokens from SlimPajama [20]. Note that the small model size was chosen primarily due to a lack of proper compute resources. With AdamW as the optimizer, we set the maximum sequence length to 1024 with a maximum learning rate of 4e-4 and a cosine decaying scheduler.

**LSAP Setup** Past work has found that using the instruct version of LMs as a teacher for knowledge distillation in a pretraining setting yields higher scores in benchmarks compared to their pretrained or "base" counterparts [14]. For this reason, we mainly use the post-trained variants of our reference models. In our experiments, we leverage Qwen3-0.6B [22] as our surrogate while pretraining OLMo2-1B from scratch. Additionally, the tokenizers used in both models contain highly similar tokens, further motivating its use. Based on our observations in Table 1, we extract 15 tokens with the highest probability from the surrogate's output distribution per position and filter the tokens according to our chosen $\alpha$ threshold of 0.05. Aside from this, we use the same training configuration as listed in **Baseline Setting**. Due to our relatively minimal dataset size, we did not find it necessary to use the cosine annealing scheduler for the surrogate loss term. However, we still included its results with our best $\alpha$.

Table 2: **Performance across benchmarks.** We compare the benchmark scores of LSAP to the baseline at matched TFLOPs (232 million). HellaSwag (HeSw), LAMBADA (LMB), ARC-Easy (ARC-E), ARC-Challenge (ARC-C), WinoGrande (WinoG), OpenBookQA (OBQA), and TruthfulQA (TrQA) are abbreviated for stylistic simplicity. The best result for each benchmark is bolded.

| Loss | $\alpha$ | HeSw | PIQA | LMB | ARC-E | ARC-C | WinoG | OBQA | TrQA |
|---|---|---|---|---|---|---|---|---|---|
| LSAP | 0.05-anneal | **47.40** | **69.40** | **32.40** | 46.60 | **26.00** | 56.40 | 32.20 | 38.69 |
| LSAP | 0.05 | **47.40** | 66.80 | 30.60 | **47.80** | 23.40 | **57.40** | **33.00** | 39.83 |
| LSAP | 0.01 | 45.40 | 66.80 | 27.20 | 47.00 | 23.20 | 55.20 | 29.40 | 40.30 |
| LSAP | 0.20 | 45.40 | 67.40 | 28.00 | 44.40 | 24.60 | 53.60 | 29.60 | **40.70** |
| Cross Entropy | 0 | 44.09 | 65.82 | 26.38 | 44.03 | 22.89 | 51.01 | 29.49 | 40.12 |

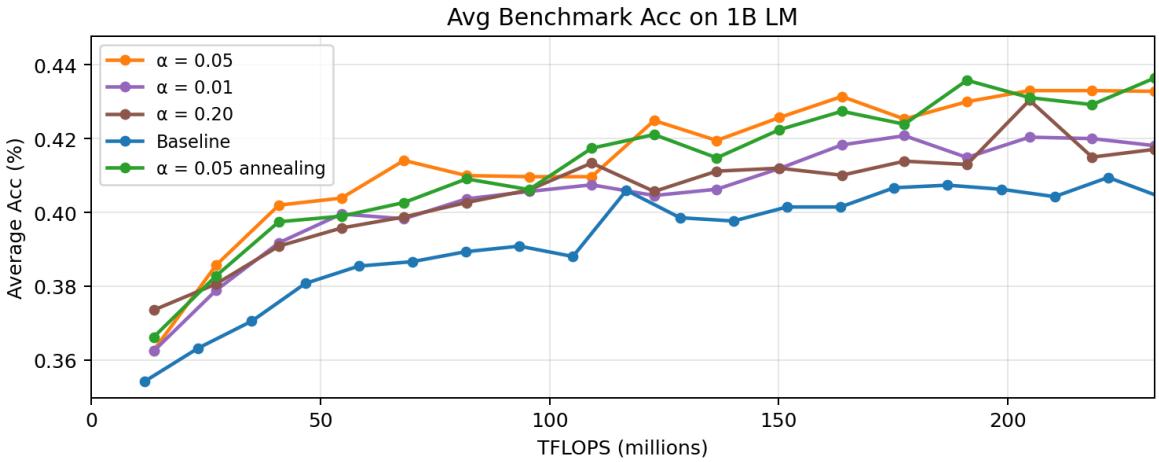


Figure 1: **Average benchmark scores.** We compare the mean zero-shot benchmark scores of LSAP against the baseline while varying $\alpha$.

**Evaluation Setup** To evaluate the effectiveness of our approach, we compare benchmark performance differences between a model trained with typical cross entropy and LSAP. To this end, we employ the lm-evaluation-harness [7].

### 4.2. Results

Here, we compare standard next-token pretraining against LSAP through matched FLOPs comparisons. All comparisons with LSAP includes the FLOPs overhead of the surrogate's inference.

Notably, we find that our method achieves higher benchmark performance at matched TFLOPs. As shown in Figure 1, LSAP with $\alpha$ = 0.05 and 0.05 without annealing achieves approximately 3.23% higher scores compared to the baseline at a matched 232 million training TFLOPs. Interestingly, our run with $\alpha$ = 0.05 generally outperformed runs with $\alpha = 0.01$ and 0.20, implying that a threshold of 0.01 may have been too permissive to the point of including false targets while 0.2 was overly restrictive and did not provide enough guidance.

Additionally, we note that applying the surrogate cosine annealing scheduler did not induce significant performance gains in our tested setting. We attribute this to our scope of training. Pretraining involves the task of broad knowledge acquisition while later stages (i.e., mid-training and post-training) require more factual suggestions that a small surrogate language model cannot provide consistently. We did not extend our experiments into those regions and thus leave it as a supplemental solution for future extensions.

LSAP generally outperforms our baseline in benchmark accuracy at matched FLOPs. Compared to the cross entropy loss, LSAP with $\alpha$ = 0.05 achieves +3.31% on HellaSwag, +3.58% on PIQA, +6.02% on LAMBADA, +2.57% on ARC-Easy, +3.11% on ARC-Challenge, +5.39% on WinoGrande, +2.71% on OpenBookQA, and -1.43% on TruthfulQA.

Given that LSAP achieves higher benchmark performance while using 14.4% fewer tokens at equivalent compute expenses, this suggests improved sample efficiency rather than gains driven purely by additional compute.

## 5. Limitations and Future Work

Due to a lack of proper compute resources, we were ultimately unable to apply LSAP to models of varying architecture and size. However, a natural extension of this work would be to analyze how we can make the surrogate prior to significant benchmark degradations. Additionally, we are still unsure how essential having a high tokenizer intersection percentage is to LSAP's efficacy. This is easily verifiable through a separate run wherein specific tokens are zeroed out to artificially decrease the intersection proportion.

Although it has not been empirically proven, it is intuitive to think that the benefits of LSAP should scale proportionally with the size of the base model while keeping the surrogate's size constant. In this scenario, the surrogate would provide the same target suggestions regardless of the base model's size. As a result, significantly scaling up the size of the base model would decrease the relative FLOPs overhead of the surrogate's inference.

## 6. Conclusion

We propose LSAP, a lightweight surrogate-assisted pretraining method that addresses the single-label inefficiency in conventional causal language modeling. By appending auxiliary tokens as negative log-likelihoods selected from a small surrogate rather than performing full distribution matching, we avoid the overhead of conventional knowledge distillation while benefiting from a multi-label objective. Our experiments on OLMo2-1B show that LSAP achieves superior benchmark performance using fewer tokens and matched FLOPs with a surrogate smaller than the student model.

## References

[1] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1GTARJhxtq.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006. ISBN 978-0-387-31073-2.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

[4] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ghNRg2mEgN.

[5] Sayantan Dasgupta, Trevor Cohn, and Timothy Baldwin. Don't ignore the tail: Decoupling top-k probabilities for efficient language model distillation, 2026. URL https://arxiv.org/abs/2602.20816.

[6] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/du22c.html.

[7] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

[8] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[10] 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.

[11] Kai Hua, Steven Wu, Ge Zhang, and Ke Shen. Attentioninfluence: Adopting attention head influence for weak-to-strong pretraining data selection, 2025. URL https://arxiv.org/abs/2505.07293.

[12] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

[13] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

[14] Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

[15] Marcus Ma, Georgios Chochlakis, Niyantha Maruthu Pandiyan, Jesse Thomason, and Shrikanth Narayanan. Large language models do multi-label classification differently. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Pro-*

*cessing*, pages 2472–2495, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.126. URL https://aclanthology.org/2025.emnlp-main.126/.

[16] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f1748d6b0fd9d439f71450117eba2725-Paper.pdf.

[17] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656.

[18] Mrigank Raman, Pranav Mani, Davis Liang, and Zachary Lipton. For distillation, tokens are not all you need. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023. URL https://openreview.net/forum?id=2fc5GOPYip.

[19] Ankit Singh Rawat, Veeranjaneyulu Sadhanala, Afshin Rostamizadeh, Ayan Chakrabarti, Wittawat Jitkrittum, Vladimir Feinberg, Seungyeon Kim, Hrayr Harutyunyan, Nikunj Saunshi, Zachary Nado, Rakesh Shivanna, Sashank J. Reddi, Aditya Krishna Menon, Rohan Anil, and Sanjiv Kumar. A little help goes a long way: Efficient LLM training by leveraging small LMs, 2026. URL https://openreview.net/forum?id=UrGsJphPnJ.

[20] et al. Soboleva, Daria. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023.

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. URL https://arxiv.org/abs/1512.00567.

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

[23] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3:1–13, 09 2009. doi: 10.4018/jdwm.2007070101.

[24] Qi Wang and Jinjia Zhou. Topkd: Top-scaled knowledge distillation. *ArXiv*, abs/2508.04539, 2025. URL https://api.semanticscholar.org/CorpusID:280536469.

[25] 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, 2019. URL https://arxiv.org/abs/1911.00359.

[26] 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 *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=lXuByUeHhd.

[27] Xuemiao Zhang, Xu Liangyu, Feiyu Duan, Yongwei Zhou, Sirui Wang, Rongxiang Weng, Jingang Wang, and Xunliang Cai. Preference curriculum: LLMs should always be pre-trained on their preferred data. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21181–21198, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1091. URL https://aclanthology.org/2025.findings-acl.1091/.

## Appendix A. LSAP Diagram

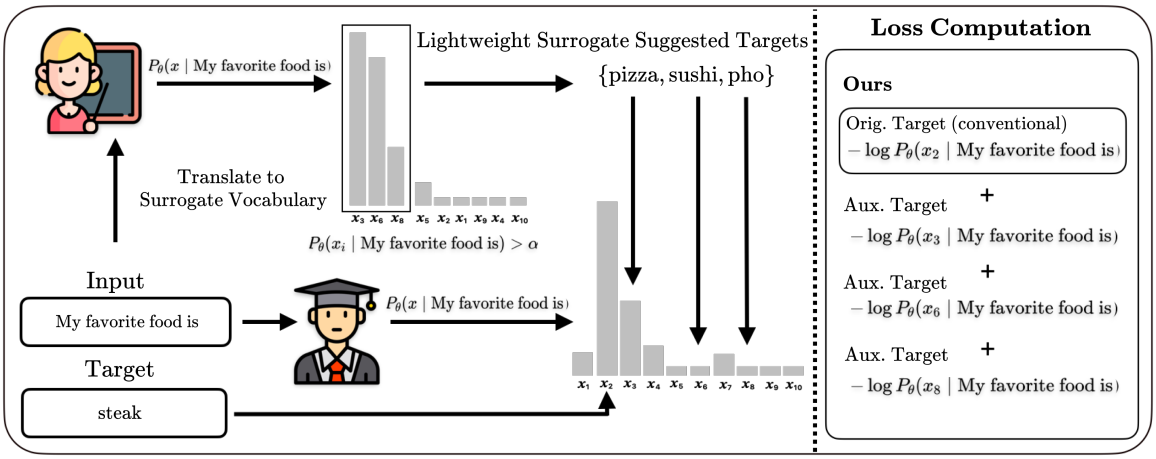


Figure 2: **Left:** A depiction of how additional target tokens are selected. **Right:** Our difference in loss function compared to typical cross entropy.