DUAL LANGUAGE MODELS: BALANCING TRAINING EFFICIENCY AND OVERFITTING RESILIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper combines autoregressive and masked-diffusion training objectives without any architectural modifications, resulting in flexible models that outperform the standard single-objective models in both settings. Autoregressive language modeling has been a popular approach, partly because of its training efficiency; however, this comes at the cost of susceptibility to overfitting. On the other hand, masked-diffusion language models are less efficient to train while being more resilient to overfitting. In this work, we demonstrate that dual-objective training achieves the best of both worlds. To derive the optimal ratio of the masked-diffusion and autoregressive objectives, we train and evaluate 50 language models under varying levels of data repetition. We show that it is optimal to combine both objectives under all evaluated settings and that the optimal ratio is similar whether targeting autoregressive or masked-diffusion downstream performance.

1 INTRODUCTION

The dominant paradigm for training large language models has been autoregressive next-token prediction (Brown et al., 2020). This approach is remarkably efficient in training, allowing models to quickly absorb vast amounts of text. However, this efficiency comes with a significant drawback: a tendency to overfit, especially when training data is limited or repeated (Muennighoff et al., 2023). This issue is becoming increasingly critical as the community reaches the so-called "data wall" – the imminent exhaustion of high-quality text data required to train ever-larger models according to established scaling laws (Villalobos et al., 2024).

An alternative approach, masked-diffusion language modeling, offers a compelling solution to the overfitting problem. These models are inherently more robust to data repetition and can learn powerful bidirectional representations (Prabhudesai et al., 2025; Ni, 2025). Yet, this robustness comes at the cost of lower training efficiency; masked-diffusion models are known to be 16 times less sample-efficient than their autoregressive counterparts (Nie et al., 2025a), requiring significantly more computation to reach comparable performance levels. This presents a fundamental trade-off: the fast convergence of autoregressive models versus the training stability of masked-diffusion models.

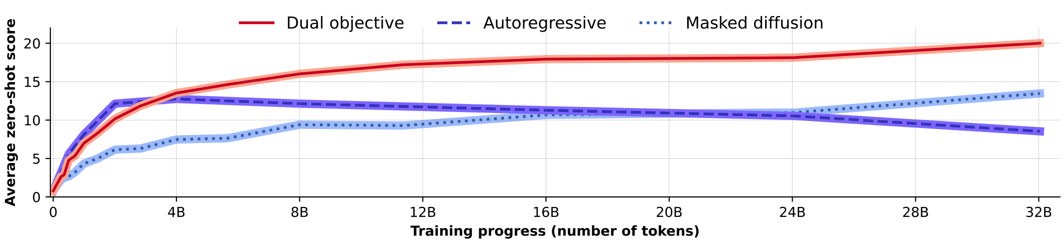

Figure 1: **The dynamics of zero-shot performance throughout training.** The three models are trained in a rather extreme setting – 128 repetitions of the same training corpus – which highlights the different behaviors caused by the three training objectives. The autoregressive objective (dashed line) converges the fastest but also very quickly overfits; the masked-diffusion objective (dotted line) converges slowly but without being negatively affected by the high amount of repetitions. Combining both objectives together (full line) results in fast convergence as well as to robustness to overfitting.

In this work, we show that it is possible to achieve the best of both worlds by simultaneously training a single language model on both autoregressive and masked-diffusion objectives. The core idea is to use the training efficiency of the autoregressive objective for rapid initial learning while using the masked-diffusion objective to regularize the model and prevent it from overfitting. The effectiveness of this dual-objective approach is illustrated in Figure 1. In the extreme data-constrained setting with 128 data repetitions, the purely autoregressive model learns quickly but then catastrophically overfits. The masked-diffusion model is immune to overfitting but converges very slowly. Our proposed dual-objective model combines the strengths of both and successfully leverages the given compute and data.

Building on this observation, we conduct a large-scale systematic study to find the optimal balance between these two objectives under varying degrees of data constraint. Our primary contributions are:

- We propose a dual-objective training method that combines autoregressive and masked-diffusion losses, enabling a single model to excel at both unidirectional and bidirectional tasks.
- Through an extensive empirical study, we systematically map the relationship between data repetition, the ratio of training objectives, and final downstream performance.
- We demonstrate that a dual-objective approach is superior to single-objective training in all evaluated settings, for both autoregressive and masked-diffusion evaluation.
- We derive two practical recommendations for setting the optimal objective ratio when training in both regular and data-constrained regimes, providing a concrete guideline for future training of large language models.
- We show that the dual language models can generalize to prefix language models at inference time, which further increases their downstream performance.

2 Background

As the name suggests, *language models* are statistical models $p_{\boldsymbol{\theta}}(\cdot)$ of the true language distribution of some training corpus $\mathcal{D}$. The training corpus consists of sequences $\boldsymbol{x} = (x_1, x_2 \dots x_N) \in \mathcal{D}$ of subword tokens. The language models are trained by finding such parameters $\boldsymbol{\theta}$ that maximize the likelihood estimation (MLE; Fisher, 1922; 1925):

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \underset{\boldsymbol{x} \sim \mathcal{D}}{\mathbb{E}} \Big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \Big]. \tag{1}$$

In this paper, we combine two popular approaches for computing $p_{\boldsymbol{\theta}}(\cdot)$, *autoregressive language models* and *masked-diffusion language models*.

2.1 Autoregressive language modeling

Language models have a long tradition and since their inception in the seminal paper by Shannon (1951), they have been factored into a chain of next-token prediction terms $p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}_{<i})$:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=1}^{|\boldsymbol{x}|} \log p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}_{<i}). \tag{2}$$

Computation of the next-token likelihoods can be efficiently parallelized when modeled by transformer networks (Vaswani et al., 2017), and thanks to their scalability, it has been the most popular paradigm behind the recent era of large language models (Brown et al., 2020).

2.2 Masked-diffusion language modeling

Masked-diffusion language models have recently become a popular alternative to autoregressive models (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2025; Ou et al., 2025; Nie et al., 2025b). Computing $p_{\boldsymbol{\theta}}(\cdot)$ with masked-diffusion is slightly more complicated than with autoregression, but the resulting language model learns to handle full *bidirectional* context, which can lead to increased performance on downstream tasks (Berglund et al., 2024; Samuel, 2025).

First, following Austin et al. (2021), we define the forward (and backward) diffusion process that gradually turns a sequence of tokens $\boldsymbol{x}$ into special mask tokens (and vice-versa). The diffusion process $\{\boldsymbol{x}^t\}$ depends on the time variable $t \in [0, 1]$ so that $\boldsymbol{x}^{(0)} = \boldsymbol{x}$ and $\boldsymbol{x}^{(1)}$ is a fully masked sequence. The intermediate values are defined by the probability distribution q:

$$q_{t|0}(\boldsymbol{x}^t \mid \boldsymbol{x}) \stackrel{\text{def}}{=} \prod_{i=1}^{|\boldsymbol{x}|} q_{t|0}(x_i^t \mid x_i); \text{ where } q_{t|0}(x_i^t \mid x_i) \stackrel{\text{def}}{=} \begin{cases} 1-t, & x_i^t = x_i, \\ t, & x_i^t = \texttt{mask}. \end{cases} \tag{3}$$

We can see that each token can either remain unchanged or turn into a mask token with probability t. The forward process is fully reversible and we can thus accordingly define the backward process, which gradually unmasks a sequence (Austin et al., 2021). Using the results from Ou et al. (2025), the probability distribution $q_{0|t}(x_i| \boldsymbol{x}^t)$ governing the backward process can be modeled with a time-independent transformer language model with parameters $\boldsymbol{\theta}$ as $p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}^t)$. This model can be fitted to the training data by maximizing the lower bound on the log-likelihood estimate (Ou et al., 2025):

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \int_0^1 \mathop{\mathbb{E}}_{\boldsymbol{x}^t \sim q_{t|0}(\cdot|\boldsymbol{x})} \left[\frac{1}{t} \sum_{\{i | x_i^t = \texttt{mask}\}} \log p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}^t) \right] \mathrm{d}t. \tag{4}$$

The integral can be equivalently written as the expectation over $t \sim \mathcal{U}(0, 1)$, thus, it can be directly used as a training objective when estimated by Monte-Carlo sampling (Metropolis & Ulam, 1949). Such a Monte-Carlo estimate can also be used at inference-time for likelihood-based evaluation, similarly to Equation (2). Note that the resulting objective is very similar to the one used to train masked language models such as BERT (Devlin et al., 2019).

3 Dual language modeling

The method of combining autoregressive and masked (diffusion) objectives is mostly based on the earlier GPT-BERT approach by Charpentier & Samuel (2024). They showcased promising results for very small language models trained within the limitations of the BabyLM Challenge (Hu et al., 2024). We extend their approach to masked-diffusion language models and to orders of magnitude larger computation scale.

Dual objective and next-token prediction Our goal is to align the two factorizations of the MLE objective in Equations (2) and (4) so that they can be parameterized by a single transformer model. For this reason, we use a slightly modified version of masked language modeling called *masked next-token prediction* (MNTP; Lv et al., 2024). With this approach, the model always uses the hidden state at position i to predict the next token at position $i + 1$ (we prove that this parameterization is as expressive in Appendix I). In this way, both modes of operation are unified as they both, perform next-token prediction; as illustrated in Figure 2. MNTP has also been used in recent work for adapting a masked diffusion model from an autoregressive checkpoint (Gong et al., 2025; Ye et al., 2025).

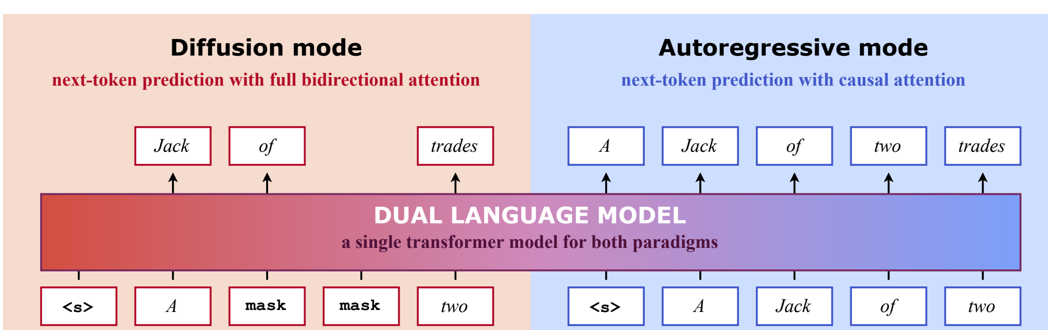

Figure 2: **Two modes of operation inside a single model.** We use the same transformer architecture with the same parameters to do both diffusion and autoregression language modeling, the only difference between the two modes is the input sequence and the attention mask.

Standard transformer architecture The main benefits of using masked next-token prediction are that we can use exactly the same transformer architecture as standard autoregressive models, and we can optimize its parameters with both objectives at the same time. The only difference between the two modes of operation are the inputs – they are either (partially) masked inputs with empty (fully bidirectional) attention masks, or full unchanged inputs with causal (unidirectional) attention masks.

Loss weighting It is crucial to correctly weight the masked-diffusion objective by $1/t$, as in Equation (4), to maintain the lower bound. Thus, on average, the masked diffusion objective is weighted by $\mathbb{E}_{t \sim \mathcal{U}(0,1)}[1/t] = 2$. To address this imbalance in regards to the autoregressive objective, we double the weight of the autoregressive loss.

GPU-wise objective separation In practice, naively mixing both objectives within a single batch could result in reduced throughput. For this reason, we assign each GPU device to a single objective so that the computation graph remains simple and static, and can be efficiently compiled. To be specific, we distribute the training of each model across 256 devices, which allows for choosing between 256 ratios of diffusion and autoregressive training. For example, if we wanted each global batch to contain as many diffusion samples as autoregressive samples, we would refer to this setting as the $1:1$ *AR-D ratio* (autoregressive-diffusion). Standard autoregressive model would be trained with $1:0$ AR-D ratio, and a model heavily skewed towards the masked-diffusion objective would be trained with $1:255$ AR-D, for example.

4 Evaluation

While it is a common practice to only consider the value of loss on a held-out set when evaluating language models (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023), it is important to measure the actual downstream performance to accurately assess the effect of different training configurations. This is especially crucial when training with two incompatible training losses.

Tasks We evaluate our models on nine standard language modeling tasks in a zero-shot fashion. All tasks consist of a context (which can be empty) and multiple different completions where one is correct and the others are incorrect. We evaluate the sum of the log-likelihood of each completion and assign the completion with the maximum sum as the prediction of the model. Table 1 lists the tasks:

Table 1: **The list of evaluation tasks.** The ARC† datasets contain some examples with 3 or 5 completions rather than 4. All tasks are evaluated zero-shot.

Task	# Examples	# Completions	Split	Reference
ARC-Easy (ARC-E)	2 376	$4^{\dagger}$	test	Clark et al. (2018)
ARC-Challenge (ARC-C)	1 172	$4^{\dagger}$	test	Clark et al. (2018)
BLiMP	67 000	2	—	Warstadt et al. (2020)
Commonsense QA (CSQA)	1 221	5	val	Talmor et al. (2019)
HellaSwag (HSwag)	10 042	4	val	Zellers et al. (2019)
MMLU	14 042	4	test	Hendrycks et al. (2021)
OpenBook QA (OBQA)	500	4	test	Mihaylov et al. (2018)
Physical Interaction QA (PIQA)	1 838	2	val	Bisk et al. (2020)
Social IQa (SIQA)	1 954	3	val	Sap et al. (2019)

Evaluation setup We follow the guidelines of the OLMES paper (Gu et al., 2025) for the normalization of our log-likelihood estimations as well as the prompt format, with two changes: 1) we only evaluate in a zero-shot fashion to simplify the setup, 2) we only consider their “cloze” formulation of each task, which is more suitable for smaller models. For the BLiMP task, which is not considered in the OLMES evaluation suite, we do not apply any length normalization and take the raw log-likelihood score. Since the BLiMP and MMLU tasks contain multiple sub-tasks (67 for BLiMP, and 57 for MMLU), we report their macro-average as the final score. More information on how each task is normalized can be found in Appendix B.

Normalized score averaging To ensure a fair aggregation of the different task scores, we first normalize the scores such that the random baseline of task is at 0 and the maximum is at 1; similarly to the Open LLM Leaderboard (Fourrier et al., 2024). To achieve this we apply the following formula to our scores: $\text{score}(x, t) = {}^{(x-r_t)}/_{(m_t-r_t)}$, where x is the raw score, r_t is the random baseline and m_t is the optimal score for task t. We then take the simple average of the normalized scores across all tasks as the final performance of our model.

4.1 AUTOREGRESSIVE (UNIDIRECTIONAL) EVALUATION

To evaluate the autoregressive capabilities of our models, we use Equation (2) to estimate the log-likelihood of each completion. Specifically, given a completion ($\boldsymbol{w}$) and context ($\boldsymbol{c}$), we calculate the conditional log-likelihood as $\log p_{\boldsymbol{\theta}}(\boldsymbol{w} \mid \boldsymbol{c}) = \sum_i \log p_{\boldsymbol{\theta}}(w_i \mid \boldsymbol{c}, \boldsymbol{w}_{<i})$.

4.2 MASKED-DIFFUSION (BIDIRECTIONAL) EVALUATION

One possibility to evaluate the masked-diffusion capabilities of our models is to also leverage the training objective in Equation (4) and estimate the conditional log-likelihood of each completion by Monte-Carlo sampling. We describe this approach in more detail in Appendix C. While it provides accurate downstream scores, it is computationally expensive and less accurate than using simpler pseudo log-likelihood (PLL; Wang & Cho, 2019; Salazar et al., 2020; Samuel, 2025) estimation.

PLL allows us to do bidirectional evaluation more than ten times faster while being more accurate than Monte-Carlo sampling (Appendix F). Therefore, we use PLL for evaluating the bidirectional capability of our models. We fully describe this method in Appendix D. As visualized in Figure 3 on the left, we specifically use the semi-autoregressive variation of PLL proposed by Samuel (2025).

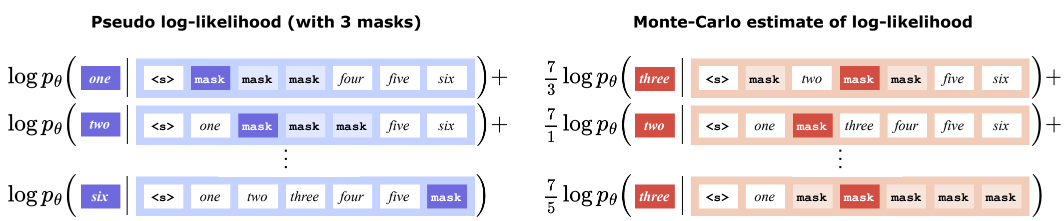

Figure 3: **Visual representations of bidirectional evaluation methods.** Pseudo log-likelihood estimation (on the left) reaches accurate likelihood scores substantially faster than the (theoretically grounded) Monte-Carlo estimation (on the right).

5 EXPERIMENTS

5.1 PRETRAINING SETUP

We train 470-million-parameter language models (with 360M non-embedding weights) on 32 billion tokens. This token budget is more than $4\times$ past the Chinchilla compute-optimal point (Hoffmann et al., 2022); we specifically decided to conduct the experiments in this regime as it reflects how modern language models are trained in practice. This compute budget is also large enough to induce non-trivial zero-shot downstream performance, enabling us to measure clear differences between different configurations.

Model architecture The language models have 24 layers with hidden size of 1 024, their self-attention operations are divided into 16 parallel heads, the feed-forward modules have intermediate size of 3 554, and the vocabulary is set to 51 200 tokens. As for the architecture itself, we follow the usual modifications of the original transformer recipe (Vaswani et al., 2017) – pre-normalization (Nguyen & Salazar, 2019) with RMSNorm (Zhang & Sennrich, 2019), rotational positional embedding (Su et al., 2024) and Swish-gated linear units (Ramachandran et al., 2018; Shazeer, 2020).

Optimization The parameters are optimized by the Muon optimizer for faster convergence (Jordan et al., 2024), specifically its variation proposed by Liu et al. (2025). The learning rate is set to 0.007 and decayed according to the warmup-stable-decay (WSD; Hägele et al., 2024) schedule (without warmup steps and 2 048 steps of linear decay). In total, each model is trained for 8 192 steps with 4M tokens in each global batch and with a sequence length of 2 048 tokens. The optimization is regularized by weight decay (with strength of 10^{-1}) and by an auxiliary z-loss term (with strength of 10^{-4}; Chowdhery et al., 2022).

Training corpus and tokenizer Even though we limit the training data to 32B tokens, we deliberately choose a text corpus that is not excessively filtered and that is representative of large-scale web crawls used in practice. We randomly sample English documents with 32B tokens in total from the HPLT v2 corpus (Burchell et al., 2025), which combines extracted webpages from the Internet Archive and CommonCrawl. We also use a smaller disjoint subset to monitor the validation loss. To prevent a potential bias from using an external tokenizer, we train a standard byte-level BPE tokenizer (Gage, 1994) with 51 200 subwords directly on the full training data.

5.2 Finding the optimal autoregressive-diffusion ratio

We trained and evaluated 50 language models in total, the results are plotted in Figure 4. In order to deal with the noisy nature of this data and to better understand the relation between the amount of data repetitions and the optimal autoregressive-diffusion ratio, we use simple statistical models.

Interpolation with Gaussian process We use Gaussian process regression (GPR; Williams & Rasmussen, 1995) with a composite kernel structure to model the relationship between data repetitions, AR-D ratios and downstream performance. The kernel consists of a constant kernel multiplied by an anisotropic Matérn kernel ($\nu = 1.5$; Stein, 1999) combined additively with a white noise kernel to account for observation noise. The input features are standardized to zero mean and unit variance, and the output features are normalized. The kernel parameters are optimized by L-BFGS-B (Liu & Nocedal, 1989) using `SciPy` (Virtanen et al., 2020). The resulting interpolations in Figure 4 show regular structure while closely fitting the data with R^2 over 0.99 in all cases.

The optimal autoregressive-diffusion ratios The fitted Gaussian process is a probabilistic model of the downstream performance given the amount data repetition and the AR-D ratio. Thus, we can transform this to the probability that a particular AR-D ratio is optimal for the given data repetition. More concretely, we can estimate the density of this distribution by sampling from the posterior of the GPR model. The result of this is visualized in the bottom part of Figure 4.

5.3 Results and discussion

The structure of Figure 4 becomes clearer once we identify which training settings result in overfitting during training.[1] The density of optimal ratios highlights that there are two regions to consider: *1) Regular-data region* where a language model trained solely on the autoregressive objective does not overfit – this roughly corresponds to 16 repetitions of training data and less, as also shown by Muennighoff et al. (2023). *2) Data-constrained region* – roughly corresponding to 32 data repetitions and more – where overfitting is an important consideration.

In the first case, it is clearly beneficial to put more weight to the autoregressive training than to masked-diffusion. Yet, training only autoregressively does not lead to any improvement in any experiments within the regular-data region. Even when evaluated purely autoregressively, the differences between $256 : 0$ and $15 : 1$ ratios are negligible. Switching to bidirectional evaluation, the single-objective $256 : 0$ ratio performs poorly while all models trained with ratios between $255 : 1$ and $15 : 1$ perform similarly – notably, they all substantially outperform models trained only with masked-diffusion. We hypothesize that the reason for these strong results (and basically 'free-lunch' masked-diffusion capability) is that the prevalence of the autoregressive objective leads to fast convergence and the small amount of masked-diffusion balances its slower convergence by inducing useful modeling priors. This leads us to formulating the first practical recommendation:

Remark 1 (Language modeling under regular data settings)**.** When training a language model in a regular data setting (16 repetitions or less), train with a small amount of masked-diffusion objective (roughly every 64th sequence) to get a strong bidirectional model without losing any autoregressive performance.

In the second data-constrained case, the relation between data repetition, AR-D ratio, and final performance seems more complicated. We risk overfitting by putting too much weight to autoregression

[1] Here, *overfitted training runs* are those runs, in which the held-out loss starts diverging while the training loss keeps converging (Appendix J). Such runs are highlighted in Figure 4 by $\times$ marks.

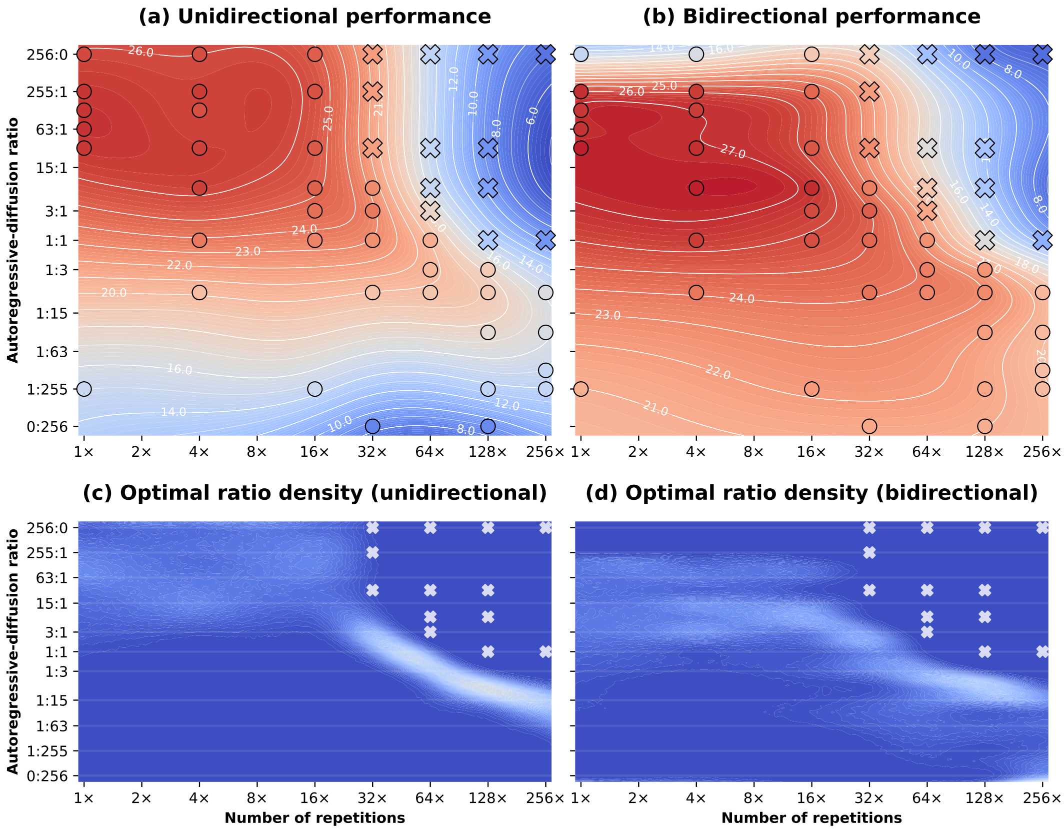

Figure 4: **Interpolated unidirectional and bidirectional results.** *The (a) and (b) figures on top* show the relation between repetitions (x-axis) and autoregressive-diffusion ratios (y-axis); the contours follow the Gaussion process model that interpolates the average performance of language models trained according to the specified settings. The respective results are plotted either as crosses when the model overfitted during training, or as circles. *The (c) and (d) figures below* visualize the estimated probability that a particular ratio (y-axis) is optimal for a given number of repetitions (x-axis).

and underfitting by focusing too much on masked-diffusion; as evident from Figure 4, the interval of optimal ratios is fairly narrow. On the other hand, the optimal ratios are surprisingly similar for the unidirectional and bidirectional performance. We can notice that the region of optimal ratios is right beneath the region of ratios that lead to overfitting, but the question is how to identify such an AR-D ratio. It is possible to have an alternative interpretation of the ratios and count the number of data repetitions that each objective is individually trained on – then we can see that more than 32 autoregressive repetitions lead to overfitting while less than 8 autoregressive repetitions lead to underfitting. Thus, based on the empirical results, our recommendation for this scenario is:

Remark 2 (Data-constrained language modeling)**.** When training a language model in a data-constrained setting (more than 32 repetitions), choose an autoregressive-diffusion ratio that exposes the autoregressive objective to roughly 16 repetitions of the training data.

Generalization to larger language models An obvious question is whether the recommendations hold even at much bigger scale for larger language models. Reliably answering this question would require expensive experimentation, but we believe that the conclusions hold for two reasons. Firstly, according to our results, the optimal AR-D ratios are clearly correlated with overfitting of autoregressive language models. Since the overfitting behavior does not depend on model size according of previous work (Muennighoff et al., 2023; Prabhudesai et al., 2025), we believe that the optimal AR-D ratios should also not change. Secondly, the relative burden of representing two modes of operation within the learned parameters decreases with model size, so we believe that the benefit of the dual training objective should actually increase with model size.

Table 2: **The normalized autoregressive performance of selected models.** We show the results on all nine evaluated tasks for three repetition values; each repetition group contains the results of the best-performing autoregressive-diffusion ratio and of the autoregressive-only model. The scores for each task are normalized so that 0% corresponds to random baseline and 100% is the perfect score. The best result for each dataset size is boldfaced.

Model configuration	ARC-C	ARC-E	BLiMP	CSQA	HSwag	MMLU	OBQA	PIQA	SIQA	Average
1 REPETITION										
Dual (63 : 1)	5.7	28.6	**63.7**	**35.1**	31.1	**4.9**	**17.6**	**40.9**	14.3	**26.9**
Autoregressive (1 : 0)	**5.9**	**30.3**	61.3	33.5	**31.7**	3.8	13.6	39.4	**15.2**	26.1
32 REPETITIONS										
Dual (3 : 1)	3.3	**28.0**	**57.9**	**31.1**	**26.4**	3.6	**14.4**	**36.1**	**14.6**	**23.9**
Autoregressive (1 : 0)	**5.0**	24.9	53.3	28.5	25.4	**3.8**	9.9	33.3	14.2	22.0
128 REPETITIONS										
Dual (1 : 7)	**1.7**	**23.6**	**56.1**	**24.8**	**14.2**	**1.6**	**8.5**	**28.1**	**13.3**	**19.1**
Autoregressive (1 : 0)	-1.0	12.3	33.2	6.8	8.1	1.1	-0.5	15.8	8.9	9.4

Detailed results To put the abstract average scores into another perspective, we look at the individual (normalized) scores per task in Table 2. The results show that the improvement in performance from using a dual objective is observed on a majority of tasks. This is especially true the more repetitions there are. The detailed scores also highlight how effectively the dual objective learns from limited data, reaching nontrivial performance even when exposed to just 256M tokens of training data (under 128 repetitions). We observe similar trends for masked-diffusion evaluation except that as the number of repetitions decreases, the performance gap increases rather than decreases. Detailed performance for the masked-diffusion evaluation can be found in Appendix H.

5.4 Generalization to prefix language modeling

Prefix language modeling (Dong et al., 2019; Raffel et al., 2020; Wang et al., 2022) is a promising alternative to the two training objectives investigated in this work. It processes the conditioning part (prefix, $\boldsymbol{c}$ in notation from Section 4.1) of a text fully bidirectionally while the completion part ($\boldsymbol{w}$ in Section 4.1) is processed autoregressively. Given that our models are trained with both unidirectional and bidirectional attention, we test whether the exposure to both can induce generalization to prefix language modeling without any further training. We repeat the earlier autoregressive evaluation with prefix attention masks and plot the results in Figure 5.

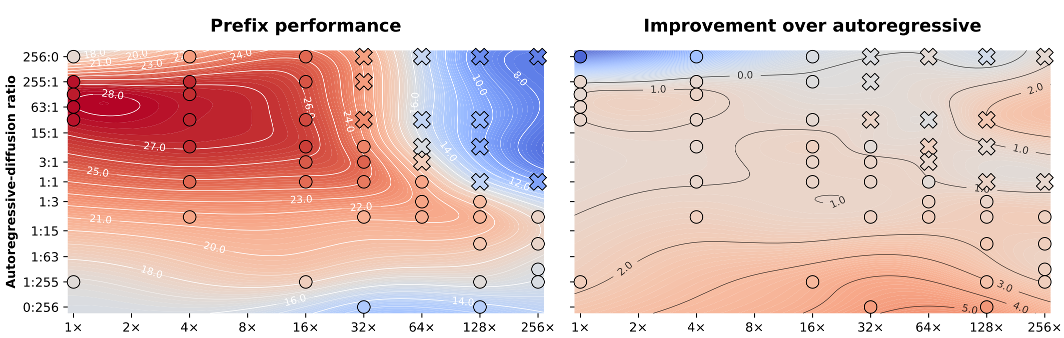

Figure 5: **Interpolated prefix results.** The figures show the relation between data repetitions (x-axis), autoregressive-diffusion ratios (y-axis), and downstream performance (color-coded). The individual results are interpolated by a GPR model. The right figure demonstrates the relative improvement of prefix-masked evaluation compared to fully unidirectional evaluation (blue color denotes decreased performance and red color denotes a performance increase).

The right side of Figure 5 shows the overall improvement of the prefix evaluation over the autoregressive one. Notably, we can see that it is reliably over one percentage point better across most configurations that combine both training objectives. This finding leads to our third recommendation:

Remark 3 (Induced prefix language modeling)**.** The autoregressive performance of dual language models can be reliably improved at inference time by processing the conditional part of a prompt fully bidirectionally.

6 Related work

Combining autoregressive and masked (diffusion) language modeling This paper builds upon the GPT-BERT training objective by Charpentier & Samuel (2024), validating its effectivity in a more practical setting. However, there is a long history of papers that tried to combine bidirectional masked language modeling with unidirectional autoregressive modeling: T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) were the first to train with autoregressive fill-in-the-blank training objectives by relying on encoder-decoder transformer architectures. Later, Du et al. (2022) proposed GLM, which uses the same objective as T5 while using a simpler decoder-only architecture with complicated scheme of positional encodings. CM3 by Aghajanyan et al. (2022) further simplifies training by not requiring any non-standard architectural modifications like the previous work. As they also add autoregressive language-modeling objective, their work is close to our approach – a model trained with CM3 can be used as any other autoregressive model at inference time, similarly to us. However, our objective also generalizes masked-diffusion language modeling and allows for fine-grained balance of the two objectives throughout training. More recently AntLM by Yu et al. (2024) proposed to switch from one objective to the other in a curriculum fashion, starting with a short autoregressive training, followed by a long masked language training and finishing on another short autoregressive training. While this does show promise, the transition between one objective to the other leads to forgetting of the previous objective whereas our objective continuously learns both objectives. Other notable works include prefix language models (Dong et al., 2019; Raffel et al., 2020; Wang et al., 2022) and UL2 (Tay et al., 2023).

Scaling of autoregressive and masked-diffusion models Concurrent works by Prabhudesai et al. (2025) and Ni (2025) have demonstrated that masked-diffusion models outperform autoregressive models in data-constrained training regimes. Our results confirm their findings but we show that using either of these training objectives is never optimal – combining them together is always better, not only in data-constrained settings.

Bidirectional masking of user and system prompts A recent paper by Katz et al. (2025) shows that using a bidirectional mask on user and system prompts improves performance on a wide variety of task, in line with Remark 3. However, for models to be able to use such masks, the authors first need to train adapters. Our work shows that by training both autoregressive and masked-diffusion at the same time, we are able to induce the prefix mask without any additional training.

Data-constrained scaling laws Muennighoff et al. (2023) studies the scaling laws of autoregressive models in data-constrained settings with a similar motivation to this paper. They show that autoregressive models cannot meaningfully learn from more than 16 data repetitions, we demonstrate that this value is an order of magnitude larger when training with the dual objective.

7 Conclusion

In this work, we addressed the fundamental trade-off between the training efficiency of autoregressive models and the overfitting resilience of masked-diffusion models. We have empirically demonstrated that a dual-objective training strategy successfully achieves the best of both worlds, resulting in models that converge rapidly without any performance degradation in data-constrained settings. We established that combining objectives is universally beneficial and derived practical guidelines for selecting the optimal training ratio based on the degree of data repetition. We showed that prefix language modeling is induced and that it performs better than autoregressive on downstream tasks. Our findings suggest that this unified approach provides a more robust and compute-efficient path forward for training the next generation of language models, especially as the field contends with the limits of available high-quality data.

Reproducibility Statement

To ensure reproducibility of our work we provided the guidelines on how to train language models on both objectives at the time in Section 3. For our model parameters and hyperparameters we specified those in Section 5.1. We describe how we perform the evaluations, the number of mask tokens used for PLL, the prompt formats, and log-likelihood normalizations in Section 4, Appendix B, and Appendix D. We openly release our custom training and evaluation code at `https://github.com/censored-for-review`. The training code is based on the common and freely distributed deep-learning framework `PyTorch` (Paszke et al., 2019).

References

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. **CM3: A causal masked multimodal model of the internet**, 2022.

Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. **Block diffusion: Interpolating between autoregressive and diffusion language models**. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. **Structured denoising diffusion models in discrete state-spaces**. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. **The reversal curse: LLMs trained on "a is b" fail to learn "b is a"**. In *The Twelfth International Conference on Learning Representations*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. **PIQA: Reasoning about physical commonsense in natural language**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020. doi: 10.1609/aaai.v34i05.6239.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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**. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Laurie Burchell, Ona de Gibert, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Pinzhen Chen, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajič, Jindřich Helcl, Erik Henriksson, Mateusz Klimaszewski, Ville Komulainen, Andrey Kutuzov, Joona Kytöniemi, Veronika Laippala, Petter Mæhlum, Bhavitvya Malik, Farrokh Mehryary, Vladislav Mikhailov, Nikita Moghe, Amanda Myntti, Dayyán O'Brien, Stephan Oepen, Proyag Pal, Jousia Piha, Sampo Pyysalo, Gema Ramírez-Sánchez, David Samuel, Pavel Stepachev, Jörg Tiedemann, Dušan Variš, Tereza Vojtěchová, and Jaume Zaragoza-Bernabeu. **An expanded massive multilingual dataset for high-performance language technologies (HPLT)**. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17452–17485, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.854.

Lucas Georges Gabriel Charpentier and David Samuel. **GPT or BERT: why not both?** In Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox (eds.), *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pp. 262–283, Miami, FL, USA, November 2024. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. **PaLM: Scaling language modeling with pathways**, 2022.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. **Think you have solved question answering? try ARC, the AI2 reasoning challenge**, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. **Unified language model pre-training for natural language understanding and generation**. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. **GLM: General language model pretraining with autoregressive blank infilling**. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 320–335, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.26.

R. A. Fisher. **On the mathematical foundations of theoretical statistics**. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368, 1922. doi: 10.1098/rsta.1922.0009.

R. A. Fisher. **Theory of statistical estimation**. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, 1925. doi: 10.1017/S0305004100009580.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. **Open LLM leaderboard v2**, 2024.

Philip Gage. **A new algorithm for data compression**. *C Users J.*, 12(2):23–38, February 1994. ISSN 0898-9788.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. **Scaling diffusion language models via adaptation from autoregressive models**. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. **OLMES: A standard for language model evaluations**. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5005–5033, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.282.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben allal, Leandro Von Werra, and Martin Jaggi. **Scaling laws and compute-optimal training beyond fixed training durations**. In *Workshop on Efficient Systems for Foundation Models II @ ICML2024*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. **Measuring massive multitask language understanding**. In *International Conference on Learning Representations*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. **Training compute-optimal large language models**. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. **Surface form competition: Why the highest probability answer isn't always right**. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.564.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox (eds.). **The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning**, Miami, FL, USA, November 2024. Association for Computational Linguistics.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. **Muon: An optimizer for hidden layers in neural networks**, 2024.

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

Shahar Katz, Liran Ringel, Yaniv Romano, and Lior Wolf. **Segment-based attention masking for GPTs**. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19308–19322, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.947.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. **Tracr: compiled transformers as a laboratory for interpretability**. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Dong C. Liu and Jorge Nocedal. **On the limited memory BFGS method for large scale optimization**. *Math. Program.*, 45(1–3):503–528, August 1989. ISSN 0025-5610.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. **Muon is scalable for LLM training**, 2025.

Aaron Lou, Chenlin Meng, and Stefano Ermon. **Discrete diffusion modeling by estimating the ratios of the data distribution**. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Ang Lv, Kaiyi Zhang, Shufang Xie, Quan Tu, Yuhan Chen, Ji-Rong Wen, and Rui Yan. **An analysis and mitigation of the reversal curse**. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13603–13615, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.754.

Nicholas Metropolis and S. Ulam. **The Monte Carlo method**. *Journal of the American Statistical Association*, 44(247):335–341, 1949. doi: 10.1080/01621459.1949.10483310.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. **Can a suit of armor conduct electricity? A new dataset for open book question answering**. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. **Scaling data-constrained language models**. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 50358–50376. Curran Associates, Inc., 2023.

Toan Q. Nguyen and Julian Salazar. **Transformers without tears: Improving the normalization of self-attention**. In Jan Niehues, Rolando Cattoni, Sebastian Stüker, Matteo Negri, Marco Turchi, Thanh-Le Ha, Elizabeth Salesky, Ramon Sanabria, Loic Barrault, Lucia Specia, and Marcello Federico (eds.), *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong, November 2-3 2019. Association for Computational Linguistics.

Jinjie Ni. **Diffusion language models are super data learners**, 2025. Notion Blog.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. **Scaling up masked diffusion models on text**. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. **Large language diffusion models**. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025b.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. **Your absorbing discrete diffusion secretly models the conditional distributions of clean data**. In *The Thirteenth International Conference on Learning Representations*, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. **PyTorch: an imperative style, high-performance deep learning library**. Curran Associates Inc., Red Hook, NY, USA, 2019.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. **Diffusion beats autoregressive in data-constrained settings**, 2025.

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**. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. **Searching for activation functions**, 2018.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. **Simple and effective masked diffusion language models**. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. **Masked language model scoring**. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2699–2712, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.240.

David Samuel. **BERTs are generative in-context learners**. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. **Social IQa: Commonsense reasoning about social interactions**. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454.

Claude E. Shannon. **Prediction and entropy of printed English**. *Bell System Technical Journal*, 30 (1):50–64, January 1951. doi: 10.1002/j.1538-7305.1951.tb01366.x.

Noam Shazeer. **GLU variants improve transformer**, 2020.

Michael L. Stein. **Interpolation of spatial data**. Springer Series in Statistics. Springer-Verlag, New York, 1999. ISBN 0-387-98629-4. doi: 10.1007/978-1-4612-1494-6. Some theory for Kriging.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. **RoFormer: Enhanced transformer with rotary position embedding**. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. **CommonsenseQA: A question answering challenge targeting commonsense knowledge**. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. **UL2: Unifying language learning paradigms**. In *The Eleventh International Conference on Learning Representations*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. **Attention is all you need**. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. **Position: will we run out of data? limits of llm scaling based on human-generated data**. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. **SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python**. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Alex Wang and Kyunghyun Cho. **BERT has a mouth, and it must speak: BERT as a Markov random field language model**. In Antoine Bosselut, Asli Celikyilmaz, Marjan Ghazvininejad, Srinivasan Iyer, Urvashi Khandelwal, Hannah Rashkin, and Thomas Wolf (eds.), *Proceedings* *of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2304.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. **What language model architecture and pretraining objective works best for zero-shot generalization?** In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22964–22984. PMLR, 17–23 Jul 2022.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. **BLiMP: The benchmark of linguistic minimal pairs for English**. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321.

Gail Weiss, Yoav Goldberg, and Eran Yahav. **Thinking like transformers**. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11080–11090. PMLR, 18–24 Jul 2021.

Christopher Williams and Carl Rasmussen. **Gaussian processes for regression**. In D. Touretzky, M.C. Mozer, and M. Hasselmo (eds.), *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Yelong Shen, Jian Jiao, Juntao Li, Zhongyu Wei, Jian Guo, Nan Duan, and Weizhu Chen. **AR-Diffusion: auto-regressive diffusion model for text generation**. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Shuchen Xue, Tianyu Xie, Tianyang Hu, Zijin Feng, Jiacheng Sun, Kenji Kawaguchi, Zhenguo Li, and Zhi-Ming Ma. **Any-order GPT as masked diffusion model: Decoupling formulation and architecture**. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. **Dream 7B: Diffusion large language models**, 2025.

Xinru Yu, Bin Guo, Shiwei Luo, Jie Wang, Tao Ji, and Yuanbin Wu. **AntLM: Bridging causal and masked language models**. In Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox (eds.), *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pp. 324–331, Miami, FL, USA, November 2024. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. **HellaSwag: Can a machine really finish your sentence?** In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472.

Biao Zhang and Rico Sennrich. **Root mean square layer normalization**. Curran Associates Inc., Red Hook, NY, USA, 2019.

A THE USE OF LARGE LANGUAGE MODELS

Large language models have been used to provide feedback, fix grammatical errors and improve the writing in this paper; in particular, we used the Claude family of language models from `https://claude.ai`. In addition, we used the autocompletion tool from GitHub Copilot when writing the code used in this work.

B LOG-LIKELIHOOD NORMALIZATION

For the BLiMP task, which is not considered in the OLMES evaluation suite, we do not apply any normalization and take the raw log-likelihood. We also stick to the no-context form of this task, where the whole sentence is considered the completion. We apply character length normalization to ARC-Easy, HellaSwag, MMLU, PIQA, and SIQA. Finally, we apply point-wise mutual information normalization (Holtzman et al., 2021), where the log-likelihood of the context-informed completion is divided by the log-likelihood of the uncontrained context completion, this can be seen in Equation (5), to ARC-Challenge Commonsense QA, and OpenBook QA.

$$\mathrm{PMI}(\boldsymbol{w}) = \sum_{i=1}^{|\boldsymbol{w}|} \log\left(\frac{p_{\boldsymbol{\theta}}\left(w_i \mid \boldsymbol{c} \oplus \boldsymbol{w}_{<i}\right)}{p_{\boldsymbol{\theta}}\left(w_i \mid \boldsymbol{u} \oplus \boldsymbol{w}_{<i}\right)}\right), \tag{5}$$

where $\boldsymbol{w}$ is the completion, $\boldsymbol{c}$ is the context, and $\boldsymbol{u}$ is the unconstrained context (in our case, this would be "Answer:")

C MONTE CARLO ESTIMATION OF LOG-LIKELIHOOD

To evaluate the masked-diffusion capabilities of our models, we use Equation (4) with the same modification as for the autoregressive evaluation as well as an adaptation of Monte-Carlo sampling to estimate the log-likelihood of each completion. Instead of taking the expectation over $t \sim \mathcal{U}(0, 1)$, we take the expectation between N equally spaced points between 0 and 1. This reduces the variance of the estimation and allows for a faster convergence. However, accurate estimation still requires $N \geq 256$, which is unbearably slow – especially when compared to simple autoregressive calculation of log-likelihood that requires only a single forward pass.

D PSEUDO LOG-LIKELIHOOD ESTIMATION

The base PLL equation can be described by a slight modification of Equation (2):

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{w}) = \sum_{i=1}^{|w|} \log p_{\boldsymbol{\theta}}\big(w_i \mid \boldsymbol{c} \oplus w_0 \oplus \cdots \oplus w_{i-1} \\ \oplus [\mathrm{MASK}] \\ \oplus w_{i+1} \oplus \cdots \oplus w_{|\boldsymbol{w}|}\big)\end{aligned} \tag{6}$$

This means that instead of doing a single forward pass, we need to do $|\boldsymbol{w}|$ forward passes to estimate the PLL. However, using a single mask token could lead underestimating the log-likelihood of words split into multiple tokens. Therefore we can further modify Equation (6) to have a variable (but constant) number of mask token after the token we are trying to estimate:

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{w}) = \sum_{i=1}^{|\boldsymbol{w}|} \log p_{\boldsymbol{\theta}}\big(w_i \mid \boldsymbol{c} \oplus w_0 \oplus \cdots \oplus w_{i-1} \\ \oplus [\mathrm{MASK}] \oplus \cdots \oplus [\mathrm{MASK}] \\ \oplus\ w_{i+n} \oplus \cdots \oplus w_{|\boldsymbol{w}|}\big),\end{aligned}$$

where n represents the number of $[\mathrm{MASK}]$ tokens. In our case we take a combination of two different number of mask tokens (1 and 6), by taking the best score of the two for each task. The two values were chosen experimentally, more details on the results of each number of mask tokens can be found in Appendix E.

E EFFECTS OF NUMBER OF MASK TOKENS ON THE PLL

We first look at whether using a single number of mask tokens can lead to a good estimation of the PLL in general. For this, we evaluate five different models from 1 to 6 mask tokens and report the results in Tables 3 to 7.

Table 3: **PLL performance depending on the number of mask tokens.** We show the PLL performance on the 9 tasks of the model trained with an equal ratio of masked-diffusion and AR and 32 repetitions with different number of masks. Best results per task are boldfaced.

Task	**Number of masks**					
	1	**2**	**3**	**4**	**5**	**6**
ARC Easy	18.7	24.1	25.5	**26.3**	26.0	**26.3**
ARC Challenge	**4.7**	3.3	3.8	2.7	1.9	2.6
BLiMP	**65.2**	63.9	62.5	60.3	60.5	60.3
Commonsense QA	29.4	32.8	33.9	**34.1**	**34.1**	**34.1**
HellaSwag	**29.8**	27.0	26.7	27.1	26.7	26.4
MMLU	2.0	**3.5**	3.1	2.9	3.3	3.3
OpenBook QA	9.1	7.7	8.5	**9.3**	7.2	6.9
PIQA	33.1	34.3	35.1	35.4	35.6	**36.8**
SIQA	11.4	13.3	13.7	13.5	**14.4**	**14.4**
Average	22.6	23.3	**23.6**	23.5	23.3	23.4

Table 4: **PLL performance depending on the number of mask tokens.** We show the PLL performance on the 9 tasks of the model trained with a 1 masked-diffusion to 7 autoregressive ratio and 32 repetitions with different number of masks. Best results per task are boldfaced.

Task	**Number of masks**					
	1	**2**	**3**	**4**	**5**	**6**
ARC Easy	18.2	25.6	27.1	28.2	26.9	**27.5**
ARC Challenge	1.9	3.2	2.4	2.6	3.6	**4.7**
BLiMP	**61.2**	60.0	58.3	56.9	57.0	57.3
Commonsense QA	24.2	29.1	29.0	**29.4**	**29.4**	**29.4**
HellaSwag	25.2	25.7	26.6	27.0	**26.8**	**26.8**
MMLU	1.9	3.4	4.0	3.9	4.0	**4.2**
OpenBook QA	9.9	10.1	**12.3**	10.9	10.1	9.6
PIQA	31.0	34.7	**36.1**	36.0	35.0	35.9
SIQA	11.7	11.8	14.2	13.7	14.1	**14.3**
Average	20.6	22.6	**23.3**	23.2	23.0	**23.3**

We can see two clear trends from the results. The first is that the BLiMP and HellaSwag tasks are better evaluated with a single mask token, rather than multiple. This could be due to the simpler language found in these datasets. The second trend is that ARC-Easy, Commonsense QA, PIQA, and SIQA tend to do better with multi-token masking, this could be due to the more complex answers using more infrequent words, that have a higher likelihood of being split into subwords. We therefore decide that using a combination of a single token mask for some tasks and a multiple tokens for others is the best solution. To find the optimal combination, we test all possible combinations. The results can be seen in Table 8.

Based on Table 8, we decide to evaluate PLL for all models with both a single mask token and six mask tokens. Then we take the max performance between the two for each task.

Table 5: **PLL performance depending on the number of mask tokens.** We show the PLL performance on the 9 tasks of the model trained with a 7 masked-diffusion to 1 autoregressive ratio and 32 repetitions with different number of masks. Best results per task are boldfaced.

Task	**Number of masks**					
	1	**2**	**3**	**4**	**5**	**6**
ARC Easy	16.3	20.8	23.9	24.0	**24.9**	**24.9**
ARC Challenge	**5.7**	3.9	3.5	1.8	3.3	2.2
BLiMP	**69.5**	67.6	64.0	60.7	60.1	60.1
Commonsense QA	25.4	29.7	30.6	31.1	31.1	**31.2**
HellaSwag	**25.5**	22.8	21.0	21.2	20.5	19.8
MMLU	0.5	2.2	2.2	2.0	**2.5**	2.4
OpenBook QA	13.1	12.0	**15.2**	14.4	13.1	13.9
PIQA	29.6	30.3	30.8	30.1	**31.2**	31.0
SIQA	12.2	15.0	**15.2**	13.6	13.8	13.9
Average	22.0	22.7	**22.9**	22.1	22.3	22.2

Table 6: **PLL performance depending on the number of mask tokens.** We show the PLL performance on the 9 tasks of the model trained with an equal ratio of masked-diffusion and AR and 16 repetitions with different number of masks. Best results per task are boldfaced.

Task	**Number of masks**					
	1	**2**	**3**	**4**	**5**	**6**
ARC Easy	16.8	23.7	25.8	25.8	**26.1**	**26.1**
ARC Challenge	**7.2**	4.4	4.4	4.8	3.2	4.5
BLiMP	**65.3**	64.8	63.1	60.7	60.6	60.4
Commonsense QA	29.7	33.8	35.1	35.1	**35.2**	**35.2**
HellaSwag	**30.5**	27.9	27.8	27.9	27.2	26.8
MMLU	1.3	2.4	**2.9**	2.5	2.7	2.5
OpenBook QA	12.3	12.0	**13.1**	11.2	11.7	11.7
PIQA	33.8	34.6	36.0	34.7	36.3	**37.0**
SIQA	14.3	13.9	15.9	15.3	15.9	**16.1**
Average	23.5	24.2	**24.9**	24.2	24.3	24.5

Table 7: **PLL performance depending on the number of mask tokens.** We show the PLL performance on the 9 tasks of the model trained with an equal ratio of masked-diffusion and AR and 64 repetitions with different number of masks. Best results per task are boldfaced.

Task	**Number of masks**					
	1	**2**	**3**	**4**	**5**	**6**
ARC Easy	16.6	21.8	23.5	23.4	23.1	23.1
ARC Challenge	1.8	3.9	3.9	3.2	**4.0**	3.5
BLiMP	**63.1**	61.2	59.6	57.5	56.9	56.9
Commonsense QA	24.6	27.6	28.5	**28.7**	**28.7**	**28.7**
HellaSwag	**26.8**	25.2	24.2	24.7	24.3	24.1
MMLU	1.2	3.1	3.0	3.2	**3.4**	3.2
OpenBook QA	8.3	8.5	**11.7**	10.1	8.3	8.0
PIQA	31.0	31.7	32.1	33.7	**34.3**	34.1
SIQA	**14.3**	12.3	**14.3**	13.1	13.3	13.5
Average	20.8	21.7	**22.3**	22.0	21.8	21.7

Table 8: PLL performance for combinations of one mask token and multi-mask token. Best results per model are bolfaced.

Repetitions - Causal Ratio	Mask combination				
	1-2	1-3	1-4	1-5	1-6
32 - 50%	24.1	24.5	24.6	24.7	**24.8**
32 - 87.5%	22.8	23.6	23.7	23.5	**23.8**
32 - 12.5%	23.5	**24.3**	24.0	24.1	24.2
16 - 50%	24.9	25.7	25.4	25.7	**25.8**
64 - 50%	22.3	**23.0**	22.9	22.9	22.8

Table 9: **Normalized PLL versus Masked-Diffusion evaluation.** The scores for each task are normalized so that 0% corresponds to the random baseline and 100% is the perfect score. The best result for each task is in boldfaced. We evaluate a model trained with equal AR and masked-diffusion ratio and 32 repetitions.

Task	PLL	Masked-Diffusion
ARC-Easy	26.3	**27.1**
BLiMP	**65.2**	56.5
Commonsense QA	**34.1**	32.7
HellaSwag	**29.8**	21.3
PIQA	**36.8**	32.0

F PLL VERSUS MASKED-DIFFUSION

Table 9 shows that the performance of the masked-diffusion model is in general lower than that of the combined (1 and 6 mask) PLL. In addition, the two PLL evaluations took about 2 hours to complete while the masked-diffusion evaluation takes 12 hours to complete on a MI250X AMD GPU.

G PREFIX VERSUS AUTOREGRESSIVE ON OPTIMAL MODELS.

Table 10: **Normalized autoregressive and prefix performance of selected models.** The scores for each task are normalized so that 0% corresponds to the random baseline and 100% is the perfect score. The best result for each dataset size is in boldfaced. The results for BLiMP are the same, since there is no context and the prefix evaluation defaults to the autoregressive one. The AR ratio for the models are 12.5% for the 128 repetitions, 75% for the 32 repetitions, and 98.4% for the single repetition.

Model	ARC-C	ARC-E	BLiMP	CSQA	HSwag	MMLU	OBQA	PIQA	SIQA	Average
1 REPETITION										
Autoregressive	5.7	28.6	**63.7**	35.1	31.1	4.9	**17.6**	40.9	14.3	26.9
Prefix	**6.5**	**31.0**	**63.7**	**40.0**	**31.2**	**4.5**	16.5	**42.1**	**15.2**	**27.9**
32 REPETITIONS										
Autoregressive	3.3	28.0	**57.9**	31.1	26.4	3.6	14.4	36.1	14.64	23.9
Prefix	**6.3**	**28.9**	**57.9**	**33.1**	**27.1**	**4.3**	**15.2**	**36.7**	**15.4**	**25.0**
128 REPETITIONS										
Autoregressive	**1.7**	23.6	**56.1**	24.8	14.2	1.6	8.5	28.1	13.3	19.1
Prefix	1.3	**24.1**	**56.1**	**28.5**	12.4	**2.3**	**10.9**	**30.9**	**15.2**	**20.5**

Table 10 shows that evaluting with the prefix mask almost always outperforms using the causal mask when the models are optimally trained. This is true in both the regular and constrained data settings.

H Detailed results of diffusion-masked evaluation

Table 11: **The normalized PLL performance of selected models.** We show the results on all nine evaluated tasks for three repetition values; each repetition group contains the results of the best-performing autoregressive-diffusion ratio and of the autoregressive-only model. The scores for each task are normalized so that 0% corresponds to random baseline and 100% is the perfect score. The best result for each dataset size is boldfaced.

Model configuration	**ARC-C**	**ARC-E**	**BLiMP**	**CSQA**	**HSwag**	**MMLU**	**OBQA**	**PIQA**	**SIQA**	**Average**
32 REPETITIONS										
Dual (3 : 1)	**6.0**	**28.3**	62.7	**33.4**	**27.8**	**4.3**	**12.3**	**37.4**	**15.4**	**25.3**
Masked-Diffusion (0 : 1)	-0.1	22.3	**64.8**	29.0	24.1	1.6	9.1	27.2	14.4	21.4
128 REPETITIONS										
Dual (1 : 7)	2.8	**23.3**	**63.5**	**30.5**	**25.0**	2.1	**12.8**	**31.8**	**15.2**	**23.0**
Masked-Diffusion (0 : 1)	**3.3**	19.2	63.3	29.2	22.1	**2.6**	9.3	28.3	12.0	21.0

Table 11 shows similar trends to those found in Table 2. The notable exception being for BLiMP where the performances are similar between both models. Unlike the autoregressive models, the performance of the purely masked-diffusion models are similar to each other. This is partially due to the model not overfitting, but also to it not being sample efficient. On the other hand we see that for the Dual Models, the performance significantly increases as we increase the training data set size.

I Proof of left-shift closure

This section proves that when we parameterize masked-diffusion language models as bidirectional transformers with shifted output, we do not lose any expressivity compared to standard non-shifted bidirectional models. We prove it constructively by defining a shift operation in the RASP language (which can then be compiled into an equivalent transformer model).

Definition 1 (RASP programs)**.** The Restricted Access Sequence Processing language (RASP; Weiss et al., 2021) is a sequence processing language that uses two types of variables: *sequence operators* and *selectors*; and two types of operators: *element-wise* and *select-aggregate* operators. Valid *programs* in RASP are operations on sequence operators formed by a finite composition of element-wise and select-aggregate operators.

- *Sequence operators* represent sequences of values (akin to hidden states in transformer models). `tokens` and `indices` are two pre-defined sequence operators; the first directly returns a sequence of the input tokens ($\texttt{tokens}(\texttt{"hello"}) = [\texttt{h}, \texttt{e}, \texttt{l}, \texttt{l}, \texttt{o}]$), and the second returns the positional indices ($\texttt{indices}(\texttt{"hello"}) = [0, 1, 2, 3, 4]$).
- *Selectors* are binary matrices (akin to attention matrices in transformers).
- *Element-wise operators* are arbitrary element-wise transformations on sequence operators (akin to feed-forward layers in transformers). For example $(\texttt{indices} + 2)(\texttt{"hello"}) = [2, 3, 4, 5, 6]$.
- *Select-aggregate operators* consist of two sequentially applied operators `select` and `aggregate` (corresponding to the attention operation).
- $\texttt{select}(\boldsymbol{x}, \boldsymbol{y}, p)$ is an operator defined on two sequence operators $\boldsymbol{x}$ and $\boldsymbol{y}$, and an element-wise boolean operator p defined on two sequence operators; the result is a selector matrix $\boldsymbol{M}$, where $M_{ij} = p(x_i, y_j)$. For example, $\texttt{select}([0, 1, 2], [1, 2, 3], <)$ results in a upper-triangular 3×3 binary matrix (selector).
- $\texttt{aggregate}(\boldsymbol{M}, \boldsymbol{x}; c)$ is an operator defined on a selector $\boldsymbol{M}$, a sequence operator $\boldsymbol{x}$ and a default value c (usually set to 0 and omitted for convenience). It produces a sequence operator $\boldsymbol{y}$

such that:

$$y_i = \begin{cases} \frac{1}{|\{j:\, M_{ij}=1\}|} \sum_{j:\, M_{ij}=1} x_j, & \text{if } |\{j : M_{ij} = 1\}| > 0, \\ c, & \text{otherwise.} \end{cases}$$

Fact 1 (RASP-transformer reduction)**.** *For every valid program written in RASP, there exists an equivalent fully-bidirectional transformer model that computes the same per-position operation; see Weiss et al. (2021); Lindner et al. (2023).*

Definition 2 (Σ-realizable functions)**.** We consider programs defined on an input alphabet Σ with a special token <s> $\in \Sigma$. A valid input sequence $\boldsymbol{x} = (x_1, x_2 \dots x_n) \in \mathcal{X}$ is every sequence where $x_1 =$ <s> and all $x_i \in \Sigma$. The output space $\mathcal{Y}$ is made of sequences $\boldsymbol{y} = (y_1, y_2 \dots y_n) \in \mathcal{Y}$, where every element is a probability distribution over the alphabet Σ: that is all $y_i \in [0,1]^{|\Sigma|}$ and $\sum_j (y_i)_j = 1$.

A function $f : \mathcal{X} \to \mathcal{Y}$ is Σ*-realizable* if there exists a transformer whose output on every input $\boldsymbol{x} \in \mathcal{X}$ equals $f(\boldsymbol{x})$ position-wise. Let $\mathcal{R}_\Sigma$ be the class of all Σ-realizable functions.

Theorem 1 (Left-shift closure)**.** *$\mathcal{R}_\Sigma$ is closed under unit left-shifts: for every $f \in \mathcal{R}_\Sigma$, there exists $g \in \mathcal{R}_\Sigma$ such that for all $\boldsymbol{x} \in \mathcal{X}$ and $i \in [1, n-1]$: $g(\boldsymbol{x})_i = f(\boldsymbol{x})_{i+1}$ (note that $f(\boldsymbol{x})_1$ and $g(\boldsymbol{x})_n$ are not constrained).*

Proof. The proof constructs a suitable function $g \in \mathcal{R}_\Sigma$ for any $f \in \mathcal{R}_\Sigma$. The new function g will mirror function f and then shift its output so that $g(\boldsymbol{x})_i = f(\boldsymbol{x})_{i+1}$, the shift will be constructed in RASP so that g is Σ-realizable.

Let $f \in \mathcal{R}_\Sigma$ be any Σ-realizable function and set T_f as a fully-bidirectional transformer that realizes f, so $T_f(\boldsymbol{x})_i = f(\boldsymbol{x})_i$ for all valid inputs $\boldsymbol{x} \in \mathcal{X}$ and all positions $i \in [1, n]$.

First, we define a RASP selector $\boldsymbol{S} = \mathtt{select}(\mathtt{indices} + 1,\ \mathtt{indices},\ =)$, whose entries therefore satisfy $S_{ij} = 1$ iff $j = i + 1$ (each row i selects exactly the next position $i + 1$, and the last row selects none).

Then, for any sequence operator $\boldsymbol{z}$ (possibly vector-valued), we define a RASP program $\mathtt{shift}(\boldsymbol{z}) = \mathtt{aggregate}(\boldsymbol{S},\ \boldsymbol{z};\ c)$, where c is arbitrary and can be simply set to z_n. By construction of $\boldsymbol{S}$ and the definition of $\mathtt{aggregate}$, we have $\mathtt{shift}(\boldsymbol{z})_n = c = z_n$ and for every $i \in [1, n-1]$:

$$\mathtt{shift}(\boldsymbol{z})_i = \frac{1}{|\{j : S_{ij} = 1\}|} \sum_{j:\, S_{ij}=1} z_j = z_{i+1}. \tag{7}$$

Using Fact 1, there exists a transformer $T_{\mathtt{shift}}$ that computes the RASP program $\mathtt{shift}$. Therefore, we can construct a transformer T_g as $T_{\mathtt{shift}} \circ T_f$. This corresponds to the function g we are looking for – T_g operates in the same input and output space as T_f, so $g \in \mathcal{R}_\Sigma$; furthermore, this function satisfies for all $\boldsymbol{x} \in \mathcal{X}$ and $i \in [1, n-1]$: $g(\boldsymbol{x})_i = \mathtt{shift}(f(\boldsymbol{x}))_i = f(\boldsymbol{x})_{i+1}$. $\square$

Corollary 1.1. Theorem 1 implies that when we parameterize a masked-diffusion model with a shifted transformer, it as expressive as the standard non-shifted parameterization. More specifically, masked diffusion is defined in Equation (4), and $p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}^t)$ is typically implemented as a fully-bidirectional transformer model that outputs this probability at the ith position. When we set Σ as our subword vocabulary, we get that the space of all possible transformer realizations of $p_{\boldsymbol{\theta}}(x_i \mid \boldsymbol{x}^t)$ are the Σ-realizable functions $\mathcal{R}_\Sigma$ (Definition 2). Theorem 1 shows that if we instead expect the output at the $(i-1)$th position, we do not lose any expressivity. Thus, transformer-based dual language models are a generalization of standard masked-diffusion language models. Note that the left-shift closure in Theorem 1 works up to the first token – which is guaranteed to be the special <s> token in Definition 2 as well as in the actual implementation.

J VALIDATION LOSS CURVES

While we focused on actual downstream performance in the main experiments, we also show the validation loss below to demonstrate the training dynamics.

The validation curves in Figure 6 focus on an extremely data-constrained scenario with 128 data repetitions. There, it is crucial to avoid overfitting, which can be achieved by increasing the proportion of masked diffusion during training. Note that the noise of some of the curves is only due to our implementation of measuring the validation loss – the sample size can be too small when the proportion of the respective training objective is low.

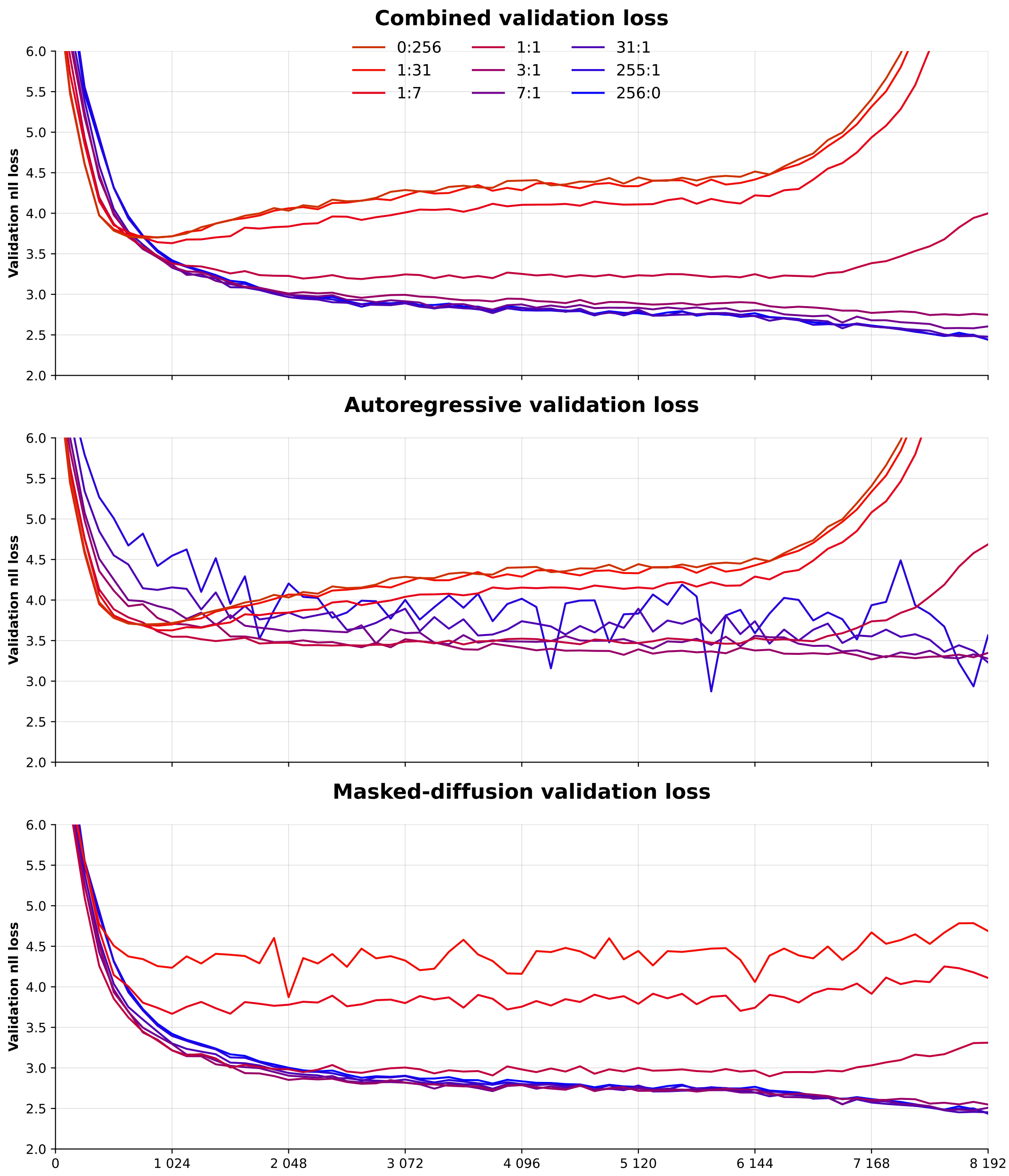

Figure 6: **Validation loss curves for 128 repetitions.** These plots clearly demonstrates how training runs with high autoregressive ratio (in red) overfit. High masked-diffusion ratios are in blue

Contrary to the previous figure, Figure 7 shows validation curves for 4 data repetitions. Here, overfitting is not an issue, instead it is crucial to improve the learning speed by increasing the proportion of autoregressive language modeling.

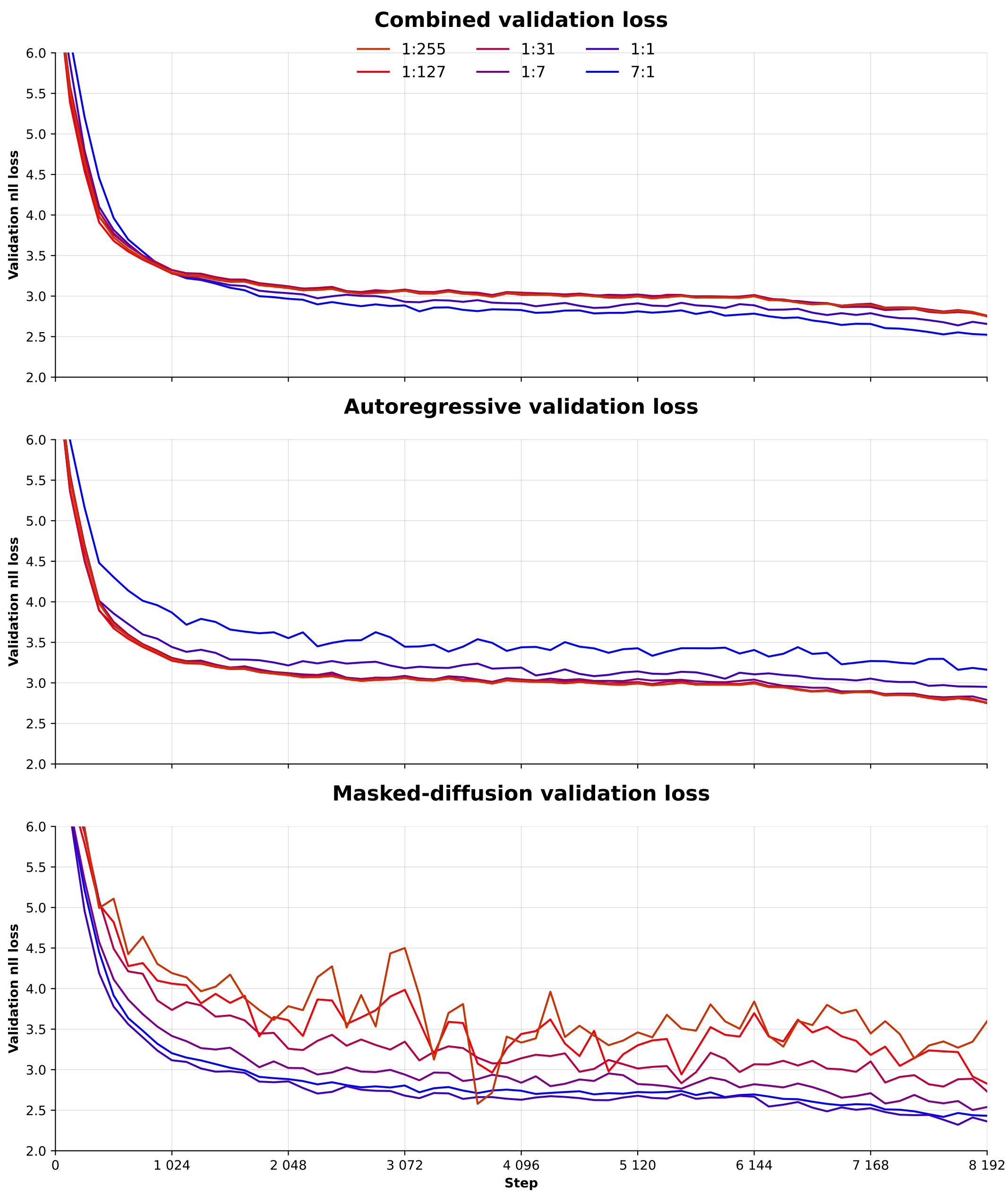

Figure 7: **Validation loss curves for 4 repetitions.** All losses monotonically decrease because overfitting is not a concern in this setting. High autoregressive ratios are plotted in red and high masked-diffusion ratios are shown in blue.

K Extended related work

Autoregressive diffusion Our work shares motivation with the autoregressive-diffusion models proposed by Wu et al. (2023). The diffusion process in that work is biased towards left-to-right denoising, which improved the decoding efficiency of the diffusion language models at that time. Similarly, Arriola et al. (2025) speeds-up decoding of masked-diffusion models by autoregressively generating chunks of tokens where each chunk is decoded by a diffusion process. In both cases, the resulting models are still diffusion models – albeit faster; these approaches do not generalize over autoregressive and masked-diffusion language modeling as our method.

Fair MD-AR comparison The recent work by Xue et al. (2025) modifies masked-diffusion language models by parameterizing them with causally-masked transformers, which makes the diffusion models more comparable to standard autoregressive models – decoupling their architectural differences from differences in training objectives. Their conclusion is that masked diffusion alone is a suboptimal objective for language, which is also confirmed by our experiments (Figure 4). However, we found that by simply combining both objectives, we can get the benefits of diffusion without losing any performance.

Approaching the data wall Large language models are known to reliably follow the empirical *scaling laws* that describe how their performance should improve with increased compute, model size, and training data. Kaplan et al. (2020) first demonstrated these relationships, showing how the training loss decreases as a power law with respect to these three parameters. These laws were later polished by Hoffmann et al. (2022), who showed that compute-optimal training requires scaling data and model size together. Related to our work, the scaling laws reveal a fundamental problem: achieving each incremental gain in performance requires exponentially more training data. Thus, data-constrained language modeling is quickly becoming a relevant field of study even for high-resource languages such as English.