SONAR-LLM: Autoregressive Transformer that Thinks in Sentence Embeddings and Speaks in Tokens

Anonymous ACL submission

Abstract

The recently proposed *Large Concept Model* (LCM) (Barrault et al., 2024) generates text by predicting a sequence of sentence-level embeddings and training with either mean–squared error or diffusion objectives. We present **SONAR-LLM**, a decoder-only transformer that *thinks* in the same continuous SONAR (Duquenne et al., 2023) embedding space yet is *supervised* through token-level cross-entropy propagated via the frozen SONAR decoder. This hybrid objective retains the semantic abstraction of LCM while eliminating its diffusion sampler and restoring a likelihood-based training signal. Across model sizes from 100 M to 900 M parameters, SONAR-LLM attains competitive generation quality. We report scaling trends, ablations, and benchmark results and, release the complete training code and all pretrained checkpoints to foster reproducibility and future research.

1 Introduction

Most autoregressive language models learn token-by-token: they minimise cross-entropy over a discrete vocabulary and emit one token per forward step (Brown et al., 2020; Raffel et al., 2020). This fine-grained decoding is simple to train and evaluate but becomes a throughput bottleneck for long sequences. Meta's recently introduced *Large Concept Model* (LCM) (Barrault et al., 2024) addresses the latency issue by predicting a much shorter trajectory of sentence-level embeddings trained with diffusion or MSE objective. Yet removing token-level likelihoods makes optimization less stable.

We present **SONAR-LLM**, an autoregressive decoder-only transformer that keeps LCM's "think in sentence embeddings" idea while leveraging the advantages of cross-entropy learning. The model predicts SONAR sentence embeddings but propagates loss through the frozen SONAR decoder down to individual tokens, coupling continuous reasoning with discrete supervision. This yields a single-shot sentence generator that is diffusion-free, likelihood-consistent, and fast at inference time.

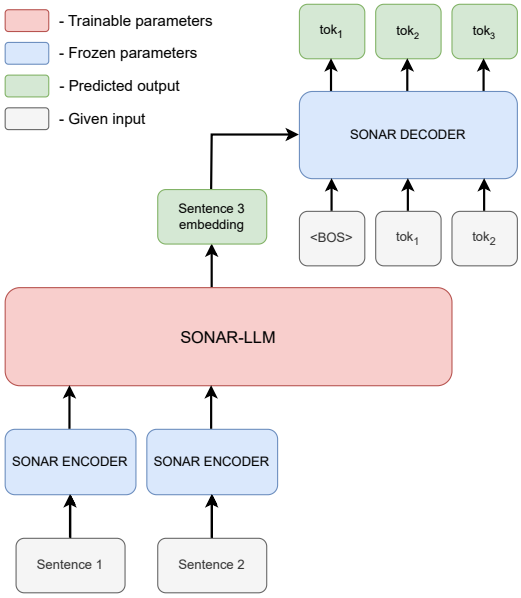

Figure 1: Architecture of **SONAR-LLM**. The model autoregressively predicts the next sentence embedding given a prefix of embeddings and decodes it via the frozen SONAR decoder.

Our contributions are:

1. **Token-Aware Embedding Objective.** We introduce a training objective that back-propagates token-level cross-entropy through a frozen SONAR decoder, aligning continuous predictions with discrete targets.

2. **Scaling Laws Analysis.** We provide a detailed scaling law fit for validation losses across model sizes, quantifying the scaling exponents for LLM, LCMs, and SONAR-LLM architectures.

3. **Inference Efficiency Analysis.** We present a

Example 1

Prefix: Once upon a time, in a small yard, there was a small daisy. The daisy had a name.

Generated: Her name was Daisy. Daisy lived in a garden full of flowers. She loved to play with her friends. One day, Daisy saw a big butterfly. The butterfly was very pretty. Daisy wanted to be friends with the butterfly. She said, "Hi, butterfly! Do you want to play with me?" The butterfly said, "Yes, I want to play with you!" Daisy and the butterfly played all day. They had so much fun. At the end of the day, Daisy was happy to go home. She knew she would have a new friend to play with.

Example 2

Prefix: Once upon a time, in a small house, there lived a kind and honest girl named Lily. She loved to bake cakes for her friends and family.

Generated: One day, Lily decided to make a big cake for her friend. Lily went to the store to mix cake, flour, and sugar. She mixed and mixed all the ingredients until it was time to bake. Then, she put the cake in the oven and waited for it to bake. When the cake was done, Lily took it out of the house and shared it with her friends. They all loved the cake and said, "Thank you, Lily!" Lily felt happy and proud that she could make her friends happy.

Figure 2: Examples of texts generated by SONAR-LLM 900 M

theoretical analysis of inference FLOPs, showing that SONAR-LLM achieves superior computational efficiency on long sequences compared to standard LLMs.

4. **Reproducible Open-Source Release.** All training code, evaluation scripts, and model checkpoints are publicly released to facilitate follow-up research.[1]

2 Related Works

Token-level autoregressive models. Large language models are trained by next-token prediction with cross-entropy over a discrete vocabulary (Brown et al., 2020), inheriting the Transformer architecture (Vaswani et al., 2017). Recent research has explored alternatives to self-attention for faster long-sequence processing; for example, MAMBA replaces attention with selective state-space updates and achieves linear-time generation while matching Transformer quality (Gu et al., 2023).

Latent-variable text generators. Continuous and discrete VAEs generate sentences from latent codes (Bowman et al., 2016). Vector-Quantised VAE (VQ-VAE) models compress sentences into a short sequence of discrete indices and decode them with an autoregressive prior (van den Oord et al., 2017). The **SONAR** encoder–decoder extends this idea to a language-agnostic, multimodal sentence embedding space covering 200 languages (Duquenne et al., 2023). Meta's *Large Concept Model* (LCM) builds an autoregressive prior over SONAR embeddings and investigates MSE, quantisation and diffusion losses in that space (Barrault et al., 2024). Our **SONAR-LLM** also operates in SONAR space but reinstates token-level cross-entropy by back-propagating through the frozen decoder.

Diffusion and discrete denoising models for text. Diffusion-LM denoises continuous word-embedding sequences to enable controllable generation without left-to-right constraints (Li et al., 2022). Discrete Denoising Diffusion Probabilistic Models (D3PMs) corrupt token sequences and learn to reverse the process in discrete space (Austin et al., 2021). Recent work improves training with a score-entropy objective, narrowing the perplexity gap to autoregressive baselines (Lou et al., 2024).

Flow and ODE-based generators. Flow Matching trains continuous normalising flows without expensive simulation and subsumes diffusion as a special case (Lipman et al., 2023). Applying flow matching to text, FLOWSEQ generates high-quality sentences in a handful of ODE steps, greatly accelerating sampling (Hu et al., 2024).

In summary, research has progressed from token-wise decoding to latent concept prediction (LCM), diffusion and flow-based models. SONAR-LLM bridges these by learning an autoregressive prior *in* sentence embedding space while retaining likelihood-based supervision.

3 SONAR-LLM

Suggested **SONAR-LLM** is an autoregressive decoder–only Transformer that operates directly in the SONAR sentence-embedding space while being supervised with token-level cross-entropy. The

[1] Available at https://anonymous.4open.science/r/SONAR-LLM-775D/

overall architecture of our approach is illustrated in Figure 1.

3.1 Pre-processing and Sentence Segmentation

We segment text into small units using the *Punkt* unsupervised sentence tokenizer implemented in NLTK (Kiss and Strunk, 2006). Each sentence s_t is encoded with the frozen multilingual SONAR encoder (Duquenne et al., 2023), yielding a fixed-length vector $\mathbf{e}_t \in \mathbb{R}^d$ (d=1024 in all experiments). Given a prefix of sentence embeddings $(\mathbf{e}_1, \ldots, \mathbf{e}_t)$, the model predicts the embedding $\hat{\mathbf{e}}_{t+1}$ of the next sentence. This predicted vector is then decoded using the frozen SONAR decoder, and the generated sentence is compared to the true next sentence s_{t+1}, which serves as the training target.

3.2 Model Architecture

SONAR-LLM is a decoder-only Transformer with the same layer pattern as Llama 3 (Llama Team, AI @ Meta, 2024) but an *embedding vocab* of size one: the model predicts a continuous vector rather than a discrete token at each step. Formally, given prefix $\mathbf{e}_{<t} = (\mathbf{e}_1, \ldots, \mathbf{e}_{t-1})$, the network outputs $\hat{\mathbf{e}}_t = f_\theta(\mathbf{e}_{<t}) \in \mathbb{R}^d$. We train variants from 100 M to 900 M parameters by scaling width and depth; all use rotary position encodings and RMS-norm.

3.3 Cross-Entropy Through the Frozen Decoder

To avoid MSE or diffusion objectives yet keep likelihood-based training, we *decode* $\hat{\mathbf{e}}_t$ back to token logits with the frozen SONAR decoder $\mathcal{D}$:

$$\mathbf{z}_t = \mathcal{D}(\hat{\mathbf{e}}_t) \in \mathbb{R}^{|\mathcal{V}|}.$$

We minimise standard cross-entropy between $\mathbf{z}_t$ and the ground-truth token sequence of sentence s_t:

$$\begin{aligned}\mathcal{L} &= -\sum_{t=1}^{T} \log p_\theta(s_t \mid \mathbf{e}_{<t}) \\ &= -\sum_{t=1}^{T}\sum_{i=1}^{|s_t|} \log\left(\mathrm{softmax}(\mathbf{z}_t)_{s_{t,i}}\right) \qquad (1)\end{aligned}$$

Back-propagation flows through $\mathcal{D}$ keeping SONAR frozen and reducing memory overhead. Teacher–forcing supplies the ground-truth embedding $\mathbf{e}_t$ at the next time step.

3.4 End of sequence

We append a special literal sentence `"End of sequence."` to every document and encode it once with the SONAR encoder to obtain $\mathbf{e}_{\text{eot}}$. At inference, generation halts when the cosine similarity between the latest predicted embedding and $\mathbf{e}_{\text{eot}}$ exceeds τ_{stop}=0.98, or when $T_{\max} = 32$ sentences are produced.

4 Results

We trained large language models (LLMs) of four different scales (100 M, 300 M, 600 M, and 900 M parameters) for four epochs each, using the Llama 3 architecture on the TinyStories dataset (Eldan and Li, 2023). Each run was conducted on a server equipped with up to 8 NVIDIA A100 GPUs (80GB). When reporting model sizes for LLMs, we included the embedding matrices in the parameter list, as these were fully trained. We also trained SONAR-LLM, MSE-based LCM, and diffusion-based LCM. For SONAR-LLM and MSE-based LCM models, we used the same architecture configurations as their LLM counterparts, but excluded the embedding and decoder parameters from training. As a result, these models contain fewer trainable parameters: 34 M, 170 M, 450 M, and 700 M, respectively, having the same depth and width. For consistency, we refer to model sizes (100 M – 900 M) based on the full LLM configurations, even when the number of trainable parameters is smaller. For the diffusion-based LCM, we employed the two-tower architecture from the original paper. Both LCM versions were trained using the official implementation provided by the authors (Barrault et al., 2024).

All models were trained using a cosine learning rate scheduler. We experimented with two learning rates: 5×10^{-4} and 1×10^{-3}. Based on validation loss performance, we found 1×10^{-3} to be optimal for SONAR-LLM, while the other models (LLM, MSE-based LCM, and diffusion-based LCM) performed better with a learning rate of 5×10^{-4}.

Examples of generated texts can be found in Figure 2

4.1 Scaling laws

The empirical scaling properties of the evaluated architectures, illustrated in Figure 3, offer insights into their efficiency in leveraging increased model parameters and training compute. This analysis

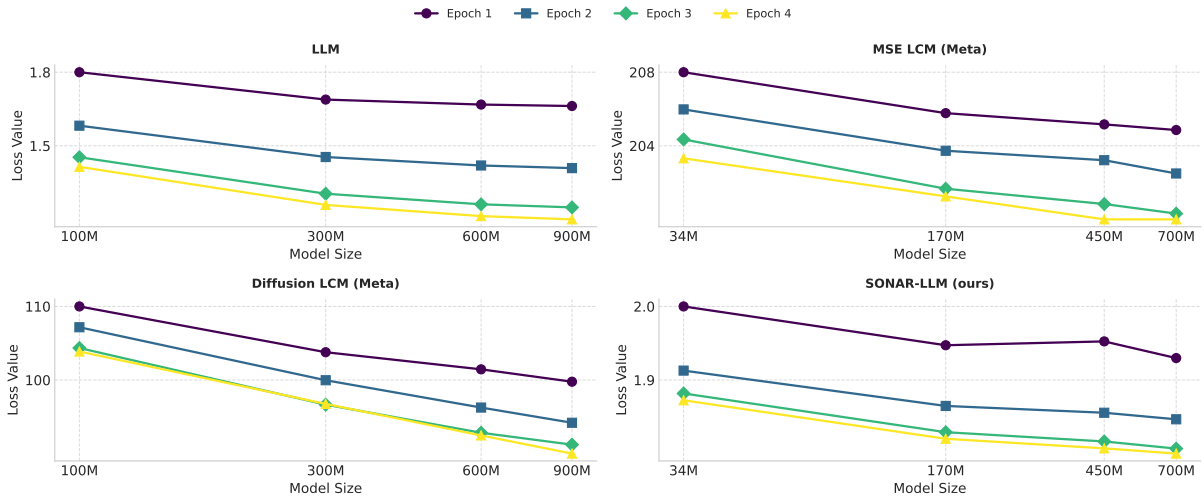

Figure 3: Scaling laws: validation loss dynamics vs. number of trainable parameters.

focuses on the implications of these observed validation loss dynamics for each model type.

We fitted the classical scaling law

$$L(N) = aN^{-\alpha} + b$$

to the validation losses of all models at epoch 4. The results (Table 1) confirm that SONAR-LLM achieves a strong scaling exponent ($\alpha \approx 0.074$), matching or surpassing other embedding-based models. This demonstrates that SONAR-LLM can efficiently leverage increased model capacity, benefiting from both semantic abstraction and effective scaling behaviour.

Table 1: Fitted scaling law parameters $L(N) = aN^{-\alpha} + b$ for each model at epoch 4.

Model	a	α	b	R^2
LLM	1.07	0.170	1.09	0.997
MSE LCM (Meta)	6.40	0.071	199.6	0.994
Diffusion LCM (Meta)	14.9	0.072	89.1	0.997
SONAR-LLM (ours)	0.24	0.074	1.74	0.994

4.2 Automatic Evaluation with GPT-4o

We evaluated the performance of all four model types on a dataset consisting of 512 generated stories, assessing grammatical correctness, creativity, coherence, and plot consistency, following the methodology proposed by (Eldan and Li, 2023). To initiate story generation, we used the first two sentences from validation set stories as prompts. During evaluation, GPT-4o was shown the full story—including the prompt and the generated continuation—but was explicitly instructed to assess only the continuation starting from the third sentence. All models were evaluated after four epochs of training. For the LLM, we experimented with both greedy decoding and beam sampling with four beams.

As illustrated in Figure 4, the classic token-level LLM clearly demonstrates the best performance. Among the concept-based models, our proposed SONAR-LLM achieves the highest story generation quality, significantly outperforming both the diffusion-based and MSE-based LCM variants.

4.3 NLG Metrics

To assess how effectively models capture the distribution of the original data, we evaluated standard NLG metrics, including BLEU, ROUGE-L, and METEOR. Specifically, we selected 512 stories from the validation set and used the first two sentences from each story as a context (short prefix) to generate the third sentence. We then measured similarity between the generated sentence and the corresponding reference sentence from the validation set using the aforementioned metrics. Additionally, we performed the same evaluation using half of each story in terms of sentence count as a context (long prefix), to investigate model performance under varying context lengths. Results are provided in Figure 6.

The NLG evaluation demonstrates that SONAR-LLM achieves results closely matching—and frequently slightly surpassing—those of a standard autoregressive LLM across all metrics. In contrast, original concept-based methods, such as diffusion-based and MSE-based LCMs, consistently show

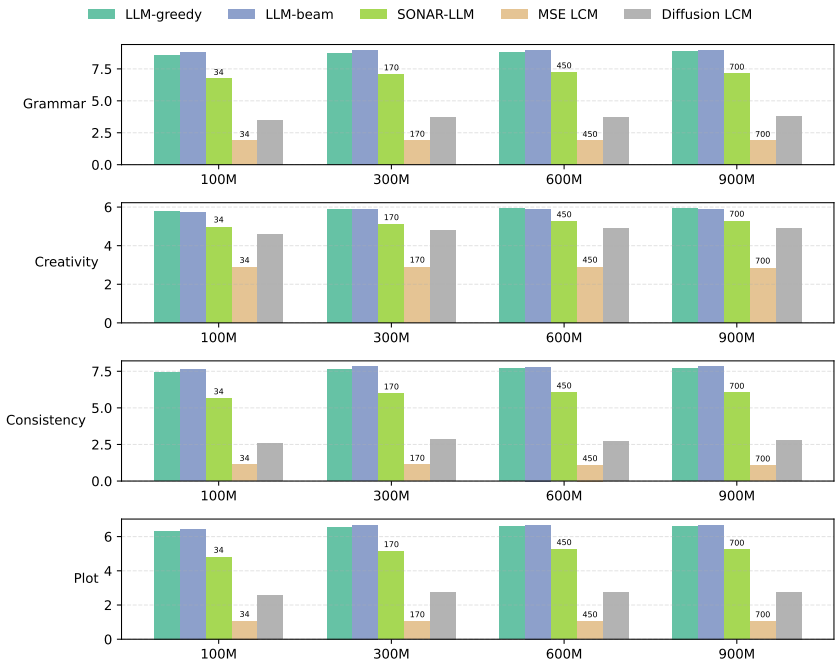

Figure 4: GPT-4o-based evaluation scores (grammar, creativity, consistency, plot) by model and size. Trainable parameter counts are shown above bars for SONAR-LLM and MSE LCM.

lower-quality generations, lagging notably behind both SONAR-LLM and standard LLMs, regardless of prompt length or model size.

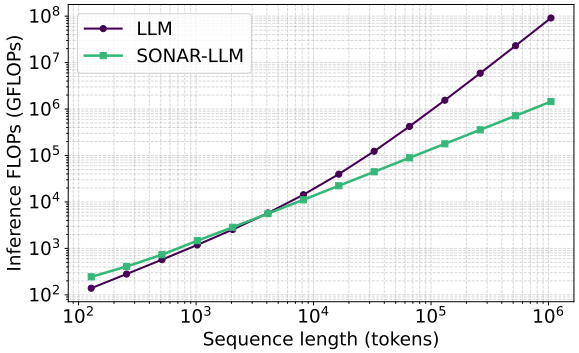

Figure 5: Theoretical inference FLOPs for autoregressive LLM and SONAR-LLM as a function of sequence length (log–log scale).

4.4 Inference Efficiency

We compared the theoretical inference complexity in FLOPs of SONAR-LLM and a standard LLM depending on the input sequence length. The comparison was performed for models with identical architectures configured at 600 M parameters. In the case of SONAR-LLM, we assumed an average sentence length of 60 tokens and, in addition to the complexity of the main SONAR-LLM model, we also included the FLOPs of the SONAR encoder and decoder. The inference setup of SONAR-LLM follows the same structural principles as the MSE-based LCM proposed by Barrault et al. (2024), suggesting that both models exhibit similar inference efficiency due to similar design.

The results presented in Figure 5 indicate that, for shorter sequences, standard token-level LLMs maintain a computational advantage due to their optimized token-wise autoregressive decoding. However, as the input length increases, this advantage diminishes: starting from approximately 4096 tokens, SONAR-LLM surpasses the standard LLM in inference efficiency. This is attributable to SONAR-LLM's design, which processes entire sentences as atomic units, thereby reducing the number of required decoding steps relative to token-based models. While the theoretical computational complexity remains quadratic for both approaches, the effective cost for SONAR-LLM grows much more slowly with sequence length because it operates on a compressed sequence of sentence embeddings. In practice, this yields an almost linear growth in FLOPs up to 1 million tokens, as the quadratic term is scaled by the inverse square of the average sentence length.

5 Conclusion

We presented **SONAR-LLM**, a decoder-only Transformer that predicts sentence embeddings

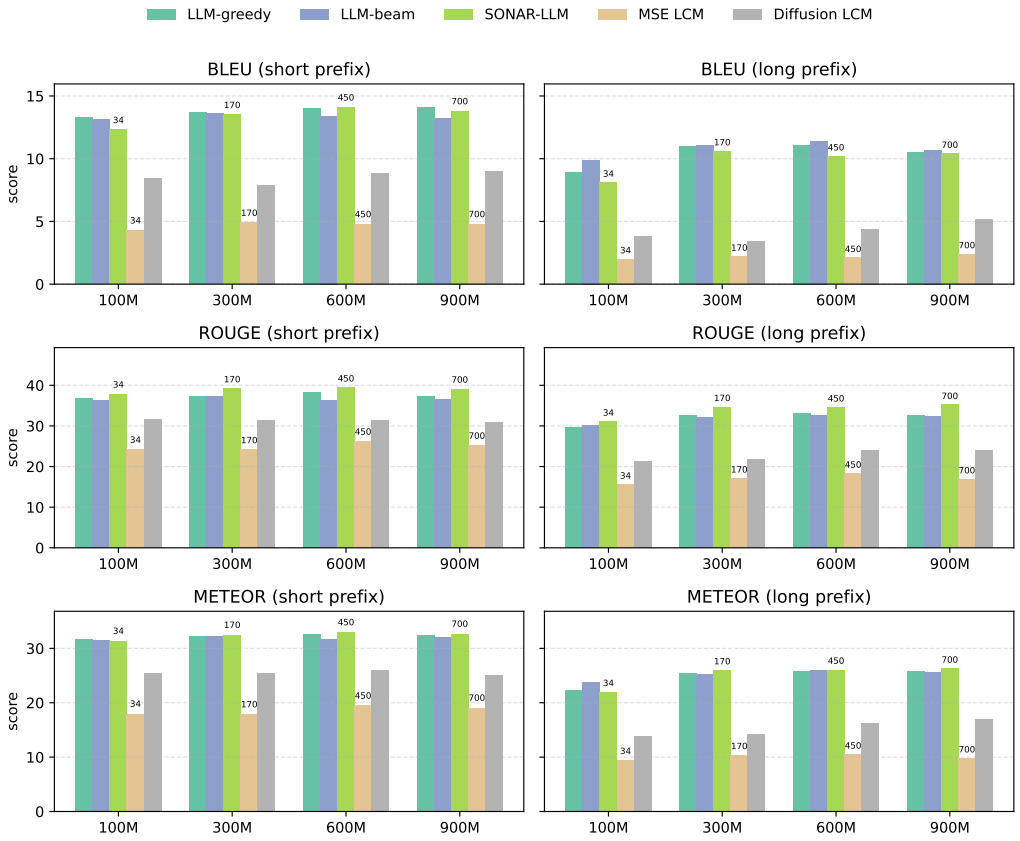

Figure 6: NLG scores by model and size; trainable parameter counts are shown above bars for SONAR-LLM and MSE LCM.

and is supervised via token-level cross-entropy propagated through a frozen SONAR decoder. This approach retains the semantic abstraction of concept-based models like LCM while restoring a likelihood-based training signal.

As a proof of concept, we trained SONAR-LLM on the TINYSTORIES dataset. It showed faster loss reduction across training epochs than both MSE-based and diffusion-based LCMs, and demonstrated favorable scaling behaviour as model size increased. In GPT-4o evaluations, SONAR-LLM outperformed both LCM variants in grammar, coherence, creativity, and plot consistency. On standard NLG metrics, SONAR-LLM demonstrated strong performance, consistently matching or slightly surpassing the standard token-level LLM. It also outperformed both the MSE-based and diffusion-based LCMs across all prefix lengths, establishing it as a competitive and reliable alternative for sentence-level generation tasks.

Our theoretical FLOPs analysis further demonstrates that SONAR-LLM achieves superior inference efficiency for long contexts: beyond 4096 tokens, its total computational cost grows **almost linearly** with sequence length up to 1 million tokens. Importantly, this effect results from operating on sentence-level segments, but the underlying complexity is still quadratic. This property enables SONAR-LLM to serve as a practical and scalable architecture for long-context generation.

We plan to extend our research to more diverse and open-ended datasets, as well as explore scaling to larger model sizes to further assess the generalization and expressiveness of SONAR-LLM.

6 Limitations

While our study reveals clear trends among the evaluated model architectures, several limitations remain.

First, all experiments were conducted on the synthetic TINYSTORIES dataset, which contains short and structurally simple narratives. The extent to which our findings generalize to longer-form, more diverse, or real-world text remains uncertain.

Second, our evaluation of generation quality combines standard automatic metrics (BLEU, ROUGE-L, METEOR) with GPT-4o-based assessments of grammar, coherence, creativity, and plot

consistency. While the latter offers a stronger proxy for human judgment, it is still limited by the behavior and biases of the underlying model. A more complete evaluation would benefit from direct human annotation or broader qualitative analysis.

Third, due to computational constraints, we limited training to four epochs and model sizes up to 900 M parameters, with minimal hyperparameter tuning. Larger-scale training or more extensive exploration might change the observed scaling trends.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *arXiv preprint arXiv:2107.03006*, arXiv:2107.03006.

Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, David Dale, Hady Elsahar, Kevin Heffernan, João Maria Janeiro, Tuan Tran, Christophe Ropers, Eduardo Sánchez, Robin San Roman, Alexandre Mourachko, Safiyyah Saleem, and Holger Schwenk. 2024. Large concept models: Language modeling in a sentence representation space. *arXiv preprint arXiv:2412.08821*, arXiv:2412.08821.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th Conference on Computational Natural Language Learning*, pages 10–21. Association for Computational Linguistics.

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, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. Sonar: Sentence-level multimodal and language-agnostic representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4969–4983. Association for Computational Linguistics.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, arXiv:2305.07759.

Albert Gu, Tri Dao, David Dohan, Rishi Bommasani, and Percy Liang. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, arXiv:2312.00752.

Vincent Hu, Di Wu, Yuki Asano, Pascal Mettes, Basura Fernando, Björn Ommer, and Cees Snoek. 2024. Flow matching for conditional text generation in a few sampling steps. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 380–392, St. Julian's, Malta. Association for Computational Linguistics.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, arXiv:2210.02747.

Llama Team, AI @ Meta. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, arXiv:2407.21783.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32819–32848. PMLR.

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

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 6306–6315.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.