LEVERAGING PRETRAINED LANGUAGE MODELS AS ENERGY FUNCTIONS FOR GLAUBER DYNAMICS TEXT DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a discrete diffusion-based language model using Glauber dynamics from statistical physics. Our main insight is that instead of trying to train a discrete state space diffusion model using Glauber dynamics with a uniform transition kernel as the forward process, one can setup an "energy function" based on pretrained causal/-masked language models. When viewed as the stationary distribution, this energy function allows us to significantly improve the quality of the generated text. Incorporating UL2 as the pretrained model into our diffusion pipeline, we outperform prior diffusion based LMs and perform competitively with autoregressive models of comparable model sizes. Furthermore, our models are competitive/outperform prior diffusion models and GPT-2 style auto-regressive models on zero-shot common sense reasoning tasks as well as some planning/search tasks like Sudoku and Zebra puzzles.

1 INTRODUCTION

The dominant paradigm for training language models has been autoregressive (AR), where given the preceding context, models are trained to generate the next token in the sequence. However, AR models face a number of challenges, especially in solving tasks involving global planning, complex structural constraints, and self-correction (Lin et al., 2021; Bachmann and Nagarajan, 2024; Huang et al., 2024). Diffusion language models are a promising alternative in addressing some of these limitations and unlocking new capabilities (Ye et al., 2024; Zhang et al., 2023; Tae et al., 2025) already having shown much success in continuous data domains such as audio, image, and video generation (Nichol and Dhariwal, 2021; Kong et al., 2020; Tae et al., 2022; Shen et al., 2023; Ho et al., 2022; Saharia et al., 2022; Ramesh et al., 2024). Continuous Space Diffusion models are trained by solving a class of regression tasks called score matching (Song et al., 2021). They first define a forward Markov process that gradually converges to the stationary distribution of the Markov process (and is typically viewed as adding noise to samples), transforming them into an easy-to-sample-from distribution (e.g. Gaussian distribution). The core idea is then to learn a time reversal of this process that can transform the noise back into a clean data sample. This process is often described as an iterative "denoising process".

Directly applying diffusion models to discrete text is nontrivial. Prior work has explored various approaches to either approximate text in a continuous domain with embedding or simplex spaces (Li et al., 2022; Han et al., 2022; Richemond et al., 2022) or modified the underlying diffusion mechanism to directly operate on discrete data (Lou et al., 2024a; Gong et al., 2024; Lou et al., 2024b). Existing discrete diffusion approaches, however, often suffer from instability, slow training, weak theoretical foundations, or inefficient sampling due to reliance on heuristic transition rules or approximations of score functions (Varma et al., 2024). Furthermore, while the so-called masked diffusion language models (MDLMs) have received significant recent attention with claims of faster inference while almost matching generative quality of auto-regressive models, as well as improved performance on downstream reasoning benchmarks, often with modifications to their training procedure Kim et al. (2025), **MDLMs ultimately are highly unlikely to outperform auto-regressive models on generative modelling** as rigorously demonstrated in work of (Zheng et al., 2025) (we expand on this in Section 5). This necessitates a first-principles stochastic process-focused approach of understanding the severe limitations of masked diffusion language models which stem from a fundamentally flawed

forward Markov process (and noisy/stationary distribution) and hence **devising forward Markov processes for language modeling which can match/outperform generative and downstream reasoning/planning capabilities of auto-regressive language models**.

In this work, we propose building a discrete diffusion LM based on Glauber dynamics, a popular Markov process used widely in theoretical computer science and statistical physics (Levin and Peres, 2017). Our main insights are that diffusion models are pathwise relative entropy minimizing (Follmer, 1985; Lehec, 2013) and, hence, their performance depends heavily on how far the "noisy distribution" is from the data distribution as well as the behavior (theoretically measured using some notion of curvature) of the underlying stochastic dynamics.

Glauber dynamics (Levin and Peres, 2017) is a natural way to address the latter (indeed it is sometimes referred to as Glauber-Langevin dynamics being a discrete analogue of Langevin dynamics) since it typically ends up being the dynamics of choice for sampling from many challenging discrete state space distributions. Glauber dynamics provides a well-defined and easy-to-implement Markov chain often with guaranteed convergence to a stationary distribution, offering a principled method for generative modeling. We demonstrate that using Glauber dynamics to generate text essentially resembles a (time-varying) masked language model. Furthermore, Glauber dynamics is typically expected to shine in the presence of an "energy function", i.e., sampling from $p(x) \propto e^{-f(x)}$ for some f operating on our discrete domain of interest. We propose to use pretrained LMs (such as autoregressive or masked LMs) as our energy function, treating them as our "noisy distribution" allowing us to leverage a significant amount of compute effort spent on training them and speed up the training process for our diffusion model. This choice also addresses the bottleneck of sample inefficiency of training diffusion LMs and poorer performance compared to autoregressive LMs as we would typically expect the sampling distribution of an autoregressive model to be closer to the data distribution as opposed to a uniform or unigram distribution as used in prior works (Lou et al., 2024b; Varma et al., 2024). While we sketch how to use GPT-style models as energy functions, we opt to use UL2 style models (Tay et al., 2022) as our energy functions, which are trained to span multiple tasks, including causal generation as well as mask infilling, as this provides a unifying framework under which our Glauber dynamics-based diffusion pipeline can be easily understood and implemented.

Starting with UL2 weights as initialization and adding additional variables for incorporating the temporal aspect of diffusion models, we train our model on the score-entropy loss function (Lou et al., 2024b; Benton et al., 2022) adapted to the forward Markov process defined by Glauber dynamics with an energy function. We obtain significantly better (unconditional) generative perplexities, as measured by a larger AR LM than all prior discrete diffusion based language models and are competitive with the generative and zero-shot perplexities of AR models. Our also report improved performance on MAUVE scores (Pillutla et al., 2021) as well as on some common-sense reasoning tasks like Winogrande (ai2, 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019) and HellaSwag (Zellers et al., 2019). Very recent work also shows that if masked diffusion language models are trained robustly account for token orderings, then they can outperform Autoregressive models of 7 times the parameter count on logic puzzles like Sudoku and Zebra puzzles (Kim et al., 2025). We show that without any changes to our training procedure to robustly account for token orderings, our UL2 Glauber dynamics can outperform masked diffusion language models and hence autoregressive models on the same Sudoku and Zebra puzzles tasks.

2 Preliminaries

2.1 Discrete Diffusion Models

Our goal is to model a probability distribution over sequences of length L and hence our state space is $\Omega = \Sigma^L$ where Σ is a finite vocabulary. A discrete diffusion process describes (time-varying) Markov chains Q_t with probability distributions $p_t \in \mathbb{R}^\Omega$ evolving according to a discrete heat equation:

$$\frac{dp_t}{dt} = Q_t p_t \quad p_0 \approx p_{\mathcal{D}}$$

where $Q_t \in \mathbb{R}^{\Omega \times \Omega}$ are the generators of the Markov process with non-negative off-diagonal entries and columns summing to zero and $p_{\mathcal{D}}$ is our data distribution. Furthermore, the process is assumed to be ergodic (and ideally fast-mixing) with a limiting distribution p_{base} as $t \to \infty$. The process can be simulated by either taking small Δt sized Euler steps and randomly sampling the resulting transitions

or by running Δt scaled Poisson clocks (meant to simulate the continous time discrete space Markov chain) corresponding to each transition (Campbell et al., 2022). While for any starting distribution, an ergodic Markov chain converges to the same limiting distribution, diffusion processes admit many different interpretations including one based on time-reversal of Markov processes (Anderson, 1982) where we again set up a (necessarily time-varying) Markov process $\overline{Q}_t$:

$$\frac{dp_{T-t}}{dt} = \overline{Q}_t p_{T-t} \quad \overline{Q}_t(y,x) = e^{\log p_t(y) - \log p_t(x)} Q_t(x,y) \quad \overline{Q}_t(x,x) = -\sum_{y \neq x} \overline{Q}_t(y,x)$$

where the multiplicative factors $e^{\log p_t(y) - \log p_t(x)} = \frac{p_t(y)}{P_t(x)}$ are probability ratios which are the analogues of the score function in continous space diffusion and are essentially the parameters to be learned. To train our model for reversing the process, we will use the score entropy loss framework of (Lou et al., 2024b), where the score entropy loss is defined as:

Definition 1. The **score entropy** $\mathcal{L}_{SE}$ for a distribution p, weights $w_{xy} \geq 0$ and a score network $s_\theta(x)_y$ is:

$$\mathcal{L}_{SE} = \mathbb{E}_{x \sim p}\left[\sum_{y \sim x} w_{xy}\left(s_\theta(x)_y - \frac{p(y)}{p(x)} \log s_\theta(x)_y + K\left(\frac{p(y)}{p(x)}\right)\right)\right]$$

where $K(a) = a(\log a - 1)$ is a normalizing constant function to ensure that $\mathcal{L}_{\text{SE}} \geq 0$ and $s_\theta(x)_y$ is our score network that's meant to be close to $p_(y)/p(x)$ for all $x \in \Omega$ and all $y \sim x$,i.e. all neighbors $y \in \Omega$ of x .

The tractable "denoising" version of this loss applied in the context of discrete diffusion models, called **diffusion weighted denoising score entropy** loss $\mathcal{L}_{\text{DWDSE}}$ is defined as:

Definition 2. (Diffusion Weighted Denoising Score Entropy)

$$\mathop{\mathbb{E}}_{x_0 \sim p_{\mathcal{D}}}\left[\int_0^T \mathop{\mathbb{E}}_{x_t \sim p_{t|0}(\cdot|x_0)} \sum_{y \sim x_t} Q_t(x_t, y)\left(s_\theta(x_t,t)_y - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)} \log s_\theta(x,t)_y + K\left(\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)}\right)\right) dt\right] \tag{1}$$

The advantage of this setup is that it is generic enough that once we fix the Markov chain and the corresponding stationary distribution, we can directly plug it into the loss for training our diffusion model. Hence, we can focus our efforts on understanding what makes for good/bad Markov chains in discrete state spaces, and especially focus our efforts when our task is that of language modelling. As we discuss in Section 5, it is unlikely that the dominant discrete text diffusion models, which are so-called masked diffusion models with the absorbing Markov chain, can be a promising alternative to autoregressive language models, and hence one needs to think through and design better Markov chains for the forward process to be used for diffusion models.

Instead of viewing diffusion models as a time reversal of a forward Markov process, we instead take the stochastic optimal control viewpoint of this process which is well-known to be the pathwise relative entropy minimizing stochastic process with end-point marginals corresponding to the data distribution and the stationary distribution of the corresponding Markov process (Follmer, 1985; Lehec, 2013). Given this insight, one desires that the stationary distribution is easy to sample from and it is efficient to make updates to a data point according to the (time-varying) Markov chains and the forward Markov chain. Furthermore, we desire that the data distribution is close in some sense to the stationary distribution (this ensures that one can converge in fewer steps to the data distribution starting from the stationary distribution). Another desirable property of the stochastic process, at least in the continuous diffusion case, is that some notion of curvature (usually Bakry-Emery) of the underlying stochastic process is sufficiently good (Conforti et al., 2025) which typically manifests as the stochastic process having fast decay of entropy or it's higher derivatives along the stochastic process trajectories.

To account for the ask of the stationary distribution to be close to the data distribution, we first observe that a natural distribution which certainly should be closer to the data distribution than uniform/unigram distributions are that of pretrained language models. There are actually significant advantages of trying to use pretrained language models for diffusion models if possible. Along with

the significant amounts of effort and compute spent already to train such models, another advantage of autoregressive models that is usually not present in diffusion models for text is that during training, autoregressive models trained using the so-called teacher forcing loss naturally get L gradient signals (one corresponding to predicting the next token for each prefix) at each step rather than diffusion models which typically get one or a few token gradient signals hence making them much slower or harder to train and consequently this might be one reason why AR models perform so much better at learning the structure of language compared to diffusion models.

To address the issue of natural Markov chains, which likely have good underlying entropic decay, a natural suggestion is that of Glauber dynamics, which has been used widely in sampling from challenging distributions coming from theoretical computer science, statistical physics, and Bayesian inference. We essentially notice that the way Glauber dynamics samples at each step corresponds in some sense to masked language models and furthermore, we can use pretrained language models as energy functions or equivalently, conditional samples which will directly fit in quite naturally into the design of the forward process for Glauber dynamics.

It is important to remark at this point that most prior discrete diffusion approaches like Lou et al. (2024b) use Markov chains which let them sample many tokens in each iteration and hence their sampling is very fast, typically much faster than auto-regressive models which necessarily take L model invocations to generate text. In our discrete diffusion model based on Glauber dynamics, generation will typically be slower than AR models (in our experiments it's $2L$ and $4L$ model invocations) which we acknowledge as an acceptable limitation especially because our focus is on generating much better quality text, which these prior masked diffusion language models struggle to do (Zheng et al., 2025) and furthermore, for tasks which require search or planning, Glauber dynamics inherently seems better suited than either AR models or masked diffusion language models as the concept of natural back-tracking to edit incorrect tokens is naturally baked into the process.

We remark that while our inference time is necessarily slower than auto-regressive and masked diffusion language models, there are ways to improve our inference time that might also lead to a reduction in our training time. In particular, there is work on parallelizing Glauber dynamics (Lee, 2024) that can be leveraged here leading to improved sampling complexity, similar to work of (Shih et al., 2023) for continous diffusion models for image generation.

2.2 UL2 Models

UL2 models form the backbone of our proposed Glauber diffusion models. Proposed to overcome the limitations inherent in single-objective pretraining, UL2 (Unified Language Learning) model (Tay et al., 2022) aims to consolidate the strengths of various pre-training strategies into a single, versatile model. Based on an encoder-decoder transformer architecture, UL2 is pretrained using a varied set of denoising tasks, where the model learns to reconstruct original text from corrupted versions. These tasks include R-denoising (regular span corruption, for general knowledge acquisition), S-denoising (sequential PrefixLM or sequence-to-sequence, for causal generation capabilities), and X-denoising (extreme span corruption, for recovering large missing portions of text). This diverse training regimen allows the model to internalize different inductive biases and skills necessary for a wide array of NLP applications. We remark that while one could use AR models as energy functions in Glauber dynamics and we sketch one such way to do so, the flexibility of UL2 models to do both causal generation and mask infilling in a unified framework is instrumental to an efficient implementation of the diffusion transformer backbone in our Glauber dynamics based diffusion model.

3 Discrete Diffusion using Glauber Dynamics

In this section, we describe the training and inference procedure for our discrete diffusion model based on Glauber dynamics. We first describe the Glauber dynamics Markov chain to sample from a distribution on sequences of length L with each index taking values in a discrete set Σ, where we are interested in sampling from a probability distribution $p : \Sigma^L \to [0, 1]$.

The sampling is broken up into n different rounds where in each round, we apriori pick a permutation of $\{1, \ldots, L\}$ denoted by π_i for $i = 1, \ldots, n$ and sequentially update all L tokens by updating just the $k = \pi_i(j)^{th}$ index in the j^{th} iteration in the i^{th} round according to the distribution $p(x_k = \cdot\,|x_{\setminus k})$, i.e., we sample x_k from the stationary distribution conditioned on all but x_k (denoted as $x_{\setminus k}$ indices

being fixed. If $p(x) = e^{-f(x)}$ for some energy function f, the conditional distribution, with a Metropolis filter, can be written as $p(x_k = \sigma \,|x_{\setminus k}) = \min\left\{1, e^{-f(x_{\setminus k}, \sigma)+f(x)}\right\}$, possibly with some self-loops to ensure lazy chains. Even if the stationary distribution is not specified upfront and we just have the conditonal distributions, one can show that under some mild conditions, there is a stationary distribution and the Glauber dynamics, which just needs these conditonal distributions to run the sampler converges to this distribution. We remark that Glauber dynamics is often described as picking a random index to update, in each iteration, conditioned on the remaining indices being unchanged.However, in our setup, we will fix the permutations for each round upfront and for each data point, in the t^{th} timestep, the same index is updated. Furthermore, as the time reversed Markov chain at each time-step has the same edge structure (upto reversal of edges) as the forward Markov chain at that time-step, in the reverse process during sampling, we will update the same index.

We first briefly describe a procedure to use GPT-style causal models as a way to devise an energy function for the forward Glauber dynamics which can then be reversed. However, our focus will be on a second kind of setup using encoder-decoder UL2 models that we describe later which is more efficient and forms a more coherent picture.

3.1 GPT-style Energy Functions

Given a GPT-style causal generative model with parameters θ, we consider the generative perplexity of this model, i.e., $PPL_\theta(x) = \frac{1}{L}\sum_{i=1}^{L-1} \log p_\theta(x_{i+1}|x_{1:i})$ as our energy function. In any iteration then, our forward process essentially has to take some index, say i and update x_i to some other token y with probability $p(x_i = y_i \,|x_{\setminus i}) = \min\left\{1, e^{-PPL_\theta(x_{\setminus i}, y_i)+PPL_\theta(x)}\right\}$ for $y_i \in \Sigma$ (possibly with some self-loops for lazy chains). This ensures that the stationary distribution of the Markov chain is the distribution of the GPT-style causal generative model itself. Given this setup, we can then reverse our Markov process by learning (time-varying) jump probabilities using this transition kernel and stationary distribution and feeding it into a denoising score entropy loss function ([1]). However, we will have to keep a set of parameters for this GPT-style model for our energy function and need additional parameters for the score network required in the reverse process. From a statistical viewpoint, it would then only be fair to compare such models with say GPT-style models with parameter count equaling the parameter count of the GPT-model used for energy PLUS the one used for the denoising score network. Furthermore, we would have to train the denoising score network from scratch which can be costly.

As we will show in our second approach based on UL2 style models, we can devise a strategy where essentially the number of model parameters are roughly unchanged and the training can also be setup to mostly look like fine-tuning a model on a score entropy loss function. It is also important at this point to remark that to execute each forward transition in the GPT-style Energy Function Glauber Dynamics, we would need to compute $|\Sigma|$ many perplexity evalulations, one for each y_i which can be quite costly but note that all these perplexity calculations can be evaluated in parallel. However, the computational time overhead can still be significant. It is still worth considering this approach to compare against other methods provided one has enough computational resources and we leave this comparison for future work.

3.2 UL2 based Glauber Dynamics

We now describe our UL2 based forward process for Glauber dynamics. As described above, UL2 is a language model which is trained on both causal generation tasks as well as mask infilling tasks while sharing the same set of parameters. The first insight is that at each step of Glauber dynamics, we are essentially asking the model to take a specific index and conditioned on all other indices, asking it to give a probability distribution for what to fill this token with. So it is rather easy to see that this is essentially a mask infilling task! Furthermore, to understand the stationary distribution, it would essentially correspond to running our masked language model iteratively over each token multiple times. However this can be rather slow but given that in UL2, the masked language model shares the same weights for the causal generation task, it's a reasonable idea to just consider a causal generation from the UL2 model as an (approximate) sample from the stationary distribution which is what we will do during inference from the model.

Before describing training, we first describe our model architecture, especially in light of wanting to reuse pretrained model weights. However, the default UL2 model doesn't really have any time-varying aspect baked into it which is required for diffusion models. Hence, what we do is convert a UL2 model into a diffusion transformer model (Peebles and Xie, 2023) by taking a pretrained UL2 model θ and then adding additional time embedding parameters γ which are initialized such that time = T corresponds to these time embeddings outputting values of zero initially but this will of course change as we train the diffusion transformer.

During training, we will consider the UL2 model conditioned at time $T = N \times L$ and create a copy of these parameters because they are meant to serve as the Markov transition kernel as well as corresponding to the noisy distribution and not to be backpropagated on in our diffusion score entropy loss. This Markov transition kernel is then used to noise our data points up to time t picked randomly and then fed into the Diffusion Score Entropy loss. The important thing to note is that our UL2 model already outputs a probability distribution that predicts filling a masked token with whereas in the score entropy loss function we are trying to learn the multiplicative changes in the transition probabilities $s_\theta(x \to (y = (x_{\setminus i}, y_i), t)$ corresponding to $p_t(y)/p_t(x)$ but notice that given the Markov transition kernel of the forward process, which is just the frozen copy of the UL2 model with time fixed to be T, we can just get the ratios by dividing this from the probabilities of the UL2 model at time t. After one finishes a single epoch of training (we actually do this multiple times within an epoch to expedite training), one then discards the frozen copy of the UL2 model at time T and considers the new updated model for creating the frozen copy.

Our entire training algorithm is described in Algorithm 2 with some specific details about the forward process as well as how the SEDD loss can be computing just by the mask infilling probability distribution at time T as well as at t deferred to Appendix A. It is however important to note that because the model weights are shared for the causal generation and mask infilling task as well as across time, the frozen copy of weights that we're using to noise our data points is also changing. Hence the distribution for causal generation as well as mask infilling at the time endpoint of $T = N \times L$ is going to change over the course of the training.

Algorithm 1 UL2 Based Glauber Dynamics Diffusion Transformer Training

procedure TRAINING($\mathcal{D}$ dataset, N number of diffusion rounds)
  DiT UL2 model (θ, γ) initialized with pretrained weights for θ
  for Epoch $k \leftarrow 1$ to K **do**
    Deep Copy $(\theta_k, \gamma_k)(\cdot, t = T) \to T_k$
    for Each iteration **do**
      Sample $x \in \mathcal{D}$ and $t \in [0, T]$
      From x, get x_t by running T_k as the forward process for t iters (Details in Appendix A)
      Compute SEDD Loss using (θ_k, γ_k) and T_k on x and x_t (Details in Appendix A)
      Update (θ_k, γ_k)
    end for
  end for
  return (θ_K, γ_K)
end procedure

Next, we describe the unconditional inference algorithm. Specifically, we take our trained UL2 based diffusion transformer model and given the N permutations that were used during training, we first generate our L tokens by invoking the CAUSAL-GEN mode from our model at time $T = N \times L$. Following that, for N rounds in reverse, we update the tokens in reverse order of the corresponding permutation by repeatedly calling our UL2 diffusion transformer in MASK-INFILL model, at different values of time corresponding to which token is to be updated at that time step.

Algorithm 2 UL2 Glauber Dynamics Inference

procedure INFERENCE
DiT UL2 model (θ, γ), N number of diffusion rounds, permutations $\pi_1, \ldots, \pi_N$
$x \leftarrow$ [CAUSAL-GEN] using (θ, γ) invoked at fixed time $t = T$
for $n \leftarrow N$ to 1 **do**
for $i \leftarrow L$ to 1 **do**
$j \leftarrow \pi_n(i)$

Update x_j via [MASK-INFILL] using (θ, γ) at time $t = n \times i$ on $(x_{1:j-1}$ [MASK] $x_{j+1:L})$
end for
end for
return x
end procedure

We remark that the above algorithm describes unconditional generation. If a prefix is conditioned to be some string PFX, then we just freeze those indices and do causal generation on the tokens following the prefix and then during the Glauber dynamics Mask-Infilling steps, skip the token indices corresponding to the prefix.

We also remark that, like Lou et al. (2024b); Varma et al. (2024), our Diffusion Transformer model architecture uses best practices like AdaLN-Zero for the time embeddings and RoPE positional embeddings. In many ways our entire pipeline can basically be understood as first following the UL2 training pipeline from Tay et al. (2022) and then adding time variables to the trained model followed by continued training with the diffusion score entropy loss.

4 EXPERIMENTS

4.1 SETUP

Datasets and Models To enable a fair comparison with Varma et al. (2024) without retraining their setup, we train our models on OpenWebText (Gokaslan and Cohen, 2019) and consider model architectures of similar size as GPT-2 Medium (350M) and GPT-2 Large (745M). Our UL2 models are trained following the procedure described in (Tay et al., 2022) which uses Flan-T5 architecture and instruction following procedure (Chung et al., 2024) and then trained on the mixture-of-denoisers objective, following which we add approximately 15% more parameters for the time variables. We would ideally have wanted to use a pre-trained UL2 model however the released model checkpoint was only of 20 billion parameters and we could not find a checkpoint for a smaller model size. Instead, for our larger model, we simply take the FLAN-T5-LARGE model from HuggingFace and then train it on the mixture of denoisers objective to get the corresponding UL2 model we denote as UL2-large. For our medium sized model however, the released model checkpoint labelled as FLAN-T5-BASE has about 247 million parameters which is about 100 million parameters less than GPT-2-medium. Hence, we instead define our own T5 model setup with 16 encoder and 16 decoder layers (instead of 12 in FLAN-T5-BASE), d_{model} being 1024, d_{ff}, the dimension of the feed forward network in the transformer layers being 2048 and train the model on the mixture of denoisers objective to get the corresponding UL2 model we denote as UL2-medium.

We will use these model checkpoints as a baseline by doing causal generation before we attach any time variables and train on a score entropy loss referred to as UL2 pre-SEDD CAUSAL-GEN. After we attach the time variables and train these UL2 models according to the score entropy loss in Glauber dynamics, since the parameters are shared for mask infilling and causal generation, the probability distribution for causal generation corresponding to the end-point time of $N \times L$ (the noisy distribution) would have changed compared to pre-SEDD (Here, N is the number of rounds in Glauber dynamics, i.e., the number of times each token is touched after doing a causal generation). Hence, we also add causal generation from this model as an additional baseline denoted as UL2 post-SEDD CAUSAL-GEN at time $N \times L$ for $N = 3$. Finally, we have our Glauber dynamics based UL2 models trained on the score entropy loss and we report results for the two model sizes for $N = 1$

and $N = 3$ denoted as Glauber-UL2 post-SEDD and the total number of time-steps/end-point time, which is $N \times L$.

Our other baselines include GPT-2-medium/large/xl, as well as a continous diffusion model for text (Gulrajani and Hashimoto, 2023) as well as the masked diffusion language models MDLM (Sahoo et al., 2024) and SEDD Absorb (Lou et al., 2024b) as well as the prior Glauber dynamics based generative model (Varma et al., 2024).

Hyperparameters We consider $L = 1024$, the number of tokens in the generation and N, the number of Glauber dynamics rounds to be 1 and 3 bringing our inference steps (number of model invocations) to be 2048 and 4096 (coming from the 1024 causal generation steps followed by $1024 \times N$ mask infilling steps). Again to maintain a fair comparison to GGM, we evaluate the perplexity of our 1024 samples using large GPT models (GPT2 Large, XL, Neo). We describe our training hyperparameters, optimizer, compute setup and other details in Appendix D.

4.2 Perplexity Results

We summarize our main results for generative perplexities in Table 1. We achieve significantly better generative perplexity results than all prior diffusion based text generative models and furthermore match GPT-2-medium and are very close to matching that of GPT-2-large. We furthermore compare how causal generation from UL2 prior to fine-tuning on the SEDD loss as well as the UL2 model at time T after fine-tuning the GGM on the SEDD loss and see that both those models lag in perplexity compared to the corresponding GPT-2 models illustrating the importance of running the reverse Glauber dynamics procedure to obtain our improved results. It is however interesting to see that the model perplexities for causal generation from UL2 models post SEDD improve relative to pre SEDD suggesting that training on score entropy via Glauber dynamics does improve the langugage understanding and generative capabilities of the UL2 model even if used as a causal generative model. Furthermore, as can be seen, while doing one round of reverse Glauber improves the performance slightly, it improves significantly after 3 rounds of Glauber dynamics and we believe it is possible for performance to improve and surpass GPT-2 with more rounds of Glauber dynamics.

Furthermore, we also report zero-shot perplexities of our models and some baselines using some of the datasets from the GPT-2 paper as is standard (Radford et al., 2019). These results are provided in Table 2.

We also report MAUVE scores (Pillutla et al., 2021), performance on Common Sense Reasoning tasks like Winogrande (ai2, 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019) and HellaSwag (Zellers et al., 2019) as well as performance of solving Sudoku and Zebra/Einstein puzzles for our models and some of the baselines in Appendix B and present examples of generated unconditional and prefix-conditioned text from our Glauber dynamics based generative model in Appendix C.

5 Related Work

Continuous Diffusion for Text Generation Since text is discrete, diffusion models designed for images, audio, or video cannot be directly applied to them. Early attempts to resolve this issue thus focussed on continuous approximations. One of line of work (Li et al., 2022; Gulrajani and Hashimoto, 2023) first embeds the text into some continuous latent embedding space and runs continuous (Gaussian) diffusion in that space after which discrete text is recovered. Another line of work focuses on continuous diffusion in the probability simplex of vocabulary (Han et al., 2022; 2023; Tae et al., 2025). Most work in this direction underperforms autoregressive models. The ones that do match or outperform them rely on heavy annealing and empirical alterations (Han et al., 2022).

Discrete Space Markov Chain Based Text Diffusion This line of work considers the space to be fixed length sequences in the discrete space corresponding to the text vocabulary (or 256 different pixel values in the context of discrete diffusion for images) and designs forward Markov chains directly in this space (Sohl-Dickstein et al., 2015). In the context of text diffusion, most papers consider discrete time, discrete space Markov chains with transitions corresponding to each token index being treated independently and for each index, each token is uniformly transitioning to any other token (Ho et al., 2020; Nichol and Dhariwal, 2021; Austin et al., 2021) . The reverse Markov

Table 1: Generative Perplexities evaluated over 1024 unconditional generations. We do not evaluate a larger model (e.g., GPT2-XL) with a smaller model (e.g., GPT2-small).

			Evaluation Model	**GPT2-large (774M)**	**GPT2-xl (1.6B)**	**GPT-neo (2.7B)**
Evaluated Model	**Sampling Algorithm**	**Total Params**	**Gen. PPL (↓)**	**Gen. PPL (↓)**	**Gen. PPL (↓)**
Autoregressive Models					
GPT2-medium (Radford et al., 2019)	top-p, $p = 0.8$ $L = 1024, T = 1024$	345M	12.4	13.0	14.5
GPT2-large (Radford et al., 2019)	top-p, $p = 0.8$ $L = 1024, T = 1024$	774M	–	6.5	7.4
GPT2-xl (Radford et al., 2019)	top-p, $p = 0.8$ $L = 1024, T = 1024$	1.6B	–	–	6.8
Prior Diffusion Models					
Plaid (Gulrajani and Hashimoto, 2023)	$\tau = 0.9$ as per (Gulrajani and Hashimoto, 2023) $L = 1024, T = 4096$	1.3B	19.7	19.7	17.9
SEDD-medium (Lou et al., 2024b)	default as per (Lou et al., 2024b) $L = 1024, T = 2048$	424M	27.3	28.0	25.2
MDLM (Sahoo et al., 2024)	default as per (Sahoo et al., 2024) $L = 1024, T = 1000$	170M	44.2	45.4	40.9
GGM (Varma et al., 2024)	top-p, $p = 0.8$ $L = 1024, T = 4096$	387M	19.5	19.9	18.0
UL2 and our UL2 DiTs					
UL2-medium (pre-SEDD CAUSAL-GEN)	$L = 1024$	368M	21.7	21.3	20.4
UL2-medium (post-SEDD, T=3 $\times$ L CAUSAL-GEN)	$L = 1024$	419M	19.1	19.6	19.9
Glauber-UL2-medium (post-SEDD, T=L, i.e., $N = 1$)	$L = 1024$	419M	17.1	17.5	16.6
Glauber-UL2-medium (post-SEDD, T=3 $\times$ L, i.e., $N = 3$)	$L = 1024$	419M	13.2	13.7	14.9
UL2-large (pre-SEDD CAUSAL-GEN)	$L = 1024$	783M	–	14.2	14.9
UL2-large (post-SEDD, T=3 $\times$ L CAUSAL-GEN)	$L = 1024$	898M	–	11.4	11.5
Glauber-UL2-large (post-SEDD, T=L, i.e., $N = 1$)	$L = 1024$	898M	–	9.5	9.9
Glauber-UL2-large (post-SEDD, T=3 $\times$ L, i.e., $N = 3$)	$L = 1024$	898M	–	6.9	7.8

Table 2: Zero-Shot Validation Perplexities of baseline models compared to our models

Model	**LAMBADA**	**WikiText2**	**WikiText103**	**1BW**
GPT-2-medium	**15.60**	22.76	26.37	55.72
SEDD-medium	$\leq$ 42.77	$\leq$ 31.04	$\leq$ 29.98	$\leq$ 61.19
Glauber-UL2-medium (N=1)	$\leq$ 17.89	$\leq$ 23.95	$\leq$ 30.21	$\leq$ 56.12
Glauber-UL2-medium (N=3)	$\leq$ 17.14	$\leq$ **20.98**	$\leq$ **25.47**	$\leq$ **52.18**
GPT-2-large	10.87	**19.93**	22.05	**44.58**
Glauber-UL2-large (N=1)	$\leq$ 11.25	$\leq$ 21.54	$\leq$ 24.71	$\leq$ 47.62
Glauber-UL2-large (N=3)	$\leq$ **10.14**	$\leq$ 20.35	$\leq$ **20.83**	$\leq$ 45.04

process is then a series of time-varying Markov chains. However, their text diffusion results still underperform significantly compared to autoregressive models.

Some recent work has also attempted to devise a framework for score matching in the discrete space akin to continuous diffusion (Song et al., 2021) with the denoising score entropy framework (Lou et al., 2024b; Benton et al., 2022) emerging as a promising approach especially in the context of text diffusion models with a 150M parameter model matching the generative perplexities of a similar parameter size GPT-2 model. Also, MDLM (Sahoo et al., 2024) simplifies the D3PM setup with a simpler training objective designed for the specific masked diffusion model Markov chain and obtain seemingly better results. However, almost all prior work on discrete diffusion for text has focused on using forward transition kernels which treat each token index independently. The problem with treating every token independently in the forward process is that when sampling from the reverse process, the token indices touched at different time steps do not correspond to the same token indices during training making it unclear how the time varying nature of the diffusion process helps here in generating good quality samples. Indeed (Zheng et al., 2025) formally show that for the masked diffusion model, the optimum model of the loss function in this context is actually equivalent to a (time-invariant) masked language model thus making it hard to justify masked diffusion models as a compelling alternative to autoregressive language models. Furthermore, they show that at lower floating point accuracy, these masked diffusion language models can do a kind of temperature hacking

which makes their perplexities look better than they really are and these values degrade significantly once these models are evaluated at 64 bit floating point precision. We do remark that recent work of Gong et al. (2025) also motivated by trying to take AR models and devising diffusion language models from them. They however still follow the masked diffusion language model paradigm and are more focused on a continual pre-training from an AR model, via attention-mask annealing, to train it on a loss function which is effectively the same as the MDLM loss function and hence should suffer from the same problems as outlined by Zheng et al. (2025). Furthermore, their generative perplexity, when measured using GPT-2-large, results are worse than PLAID Gulrajani and Hashimoto (2023) (Figure 3 in their paper) which our model outperforms and hence we do not compare it for it's generative quality. We do however keep that as a baseline for our Common Sense Reasoning tasks in the appendix.

An exception to this line of MDLM (and modifications) however, which is also relevant to our work, is concurrent work of (Varma et al., 2024) where they also propose a discrete diffusion model based on Glauber dynamics. Their setup considers the noisy distribution as the unigram distribution of the data corpus and their training and inference pipeline is modeled by considering a token at each timestep and considering whether that token is a prediction due to noise or an actual signal. With this setup, they then setup a training objective which essentially constitutes T many binary classification problems where $T = O(L)$ is the number of diffusion steps and obtain better generative perplexity results than SEDD in the 350M parameter model size, however in this parameter size, they still lag behind GPT-2 models of a similar size. We emphasize that one of the reasons that they consider this binary classification framework is to ensure that they only have to learn $T|\Sigma|$ many outputs as opposed to $T|\Sigma|^2$ that might be required if at each time step, we needed to understand transitions from each token to every other token in the vocabulary. Our Glauber dynamics setup, while rather different than theirs, besides a similar underlying Markov chain graph structure, when executed using our UL2 style setup will also just need to learn $T|\Sigma|$ outputs rather than scale quadratically in $|\Sigma|$.

6 Conclusion and Limitations

In this work, we presented a Glauber Dynamics based diffusion generative model for text. Our innovations come from observing that Glauber dynamics can be written as a mask infilling model as well as the fact that designing an Glauber dynamics using pretrained language model as an energy function allows us to create significantly better diffusion models and for the first time match GPT-2-medium and GPT-2-large autoregressive models in language modelling tasks.

While our diffusion models performance is better than prior discrete diffusion models which have extremely fast sampling speed whereas our models are necessarily much slower than even auto-regressive generative models of similar size, we believe this limitation is reasonable given that we nearly match the performance of GPT-style autoregressive models and also outperform them (as well as MDLMs) on planning/search tasks like the Sudoku and Zebra puzzles.

Another limitation of our work is the slow training time and need for significant computation resources. As we outline in an appendix, we needed 32 H100 GPUs running for a little over 7 days to train our larger model which is a significant time (however the GGM paper (Varma et al., 2024) report taking 8 days on TPU compute that roughly corresponds to 24 H100s on a iso-TFLOPs comparison on a GPT-2-medium model size as opposed to our training time on GPT-2-large model size). While we do believe that this can be improved using many engineering optimizations that we did not pursue as well as the possibility of using kronecker factorization based second order optimizers like Shampoo(Morwani et al., 2025), SOAP (Vyas et al., 2025) and Muon (Liu et al., 2025) which have showed significant improvements over AdamW for training AR LLMs as well as diffusion models, we also mention that it might be possible to use flow matching versions of our energy based Glauber dynamics model which may be easier to train and perform better as their training seems to be more stable in the case of continous space diffusion as well as for masked text language models (Lipman et al., 2023; Gat et al., 2024; Holderrieth et al., 2024) and we leave this investigation for future work.

References

Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. Limitations of autoregressive models and their alternatives. In Kristina Toutanova, Anna Rumshisky, Luke

Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5147–5173, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.405. URL https://aclanthology.org/2021.naacl-main.405/.

Gregor Bachmann and Vaishnavh Nagarajan. The Pitfalls of Next-Token Prediction. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 2296–2318. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bachmann24a.html.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. Diffusion of Thought: Chain-of-Thought Reasoning in Diffusion Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=G0v0TxX01N.

Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua M. Susskind, and Navdeep Jaitly. PLANNER: Generating Diversified Paragraph via Latent Language Diffusion Model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=SLwy8UVS8Y.

Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model. *arXiv preprint arXiv:2502.13917*, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *ICLR*, 2020.

Jaesung Tae, Hyeongju Kim, and Taesu Kim. EdiTTS: Score-based Editing for Controllable Text-to-Speech. In *Interspeech*, 2022.

Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. NaturalSpeech 2: Latent Diffusion Models are Natural and Zero-Shot Speech and Singing Synthesizers. *ArXiv*, abs/2304.09116, 2023.

Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video Diffusion Models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *NeurIPS*, 2022.

Krithika Ramesh, Nupoor Gandhi, Pulkit Madaan, Lisa Bauer, Charith Peris, and Anjalie Field. Evaluating differentially private synthetic data generation in high-stakes domains. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15254–15269, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.894. URL https://aclanthology.org/2024.findings-emnlp.894/.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-LM Improves Controllable Text Generation. In *NeurIPS*, 2022.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive Simplex-based Diffusion Language Model for Text Generation and Modular Control. *arXiv preprint arXiv:2210.17432*, 2022.

Pierre H Richemond, Sander Dieleman, and Arnaud Doucet. Categorical SDEs with Simplex Diffusion. *arXiv preprint arXiv:2210.14784*, 2022.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2024a.

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, 2024. URL https://arxiv.org/abs/2410.17891.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution, 2024b. URL https://arxiv.org/abs/2310.16834.

Harshit Varma, Dheeraj Nagaraj, and Karthikeyan Shanmugam. Glauber generative model: Discrete diffusion models via binary classification. *arXiv preprint arXiv:2405.17035*, 2024.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Hans Follmer. An entropy approach to the time reversal of diffusion processes. In *Stochastic differential systems (Marseille-Luminy, 1984), volume 69 of Lecture Notes in Control and Inform. Sci*, page 156–163. Springer, Berlin, 1985.

Joseph Lehec. Representation formula for the entropy and functional inequalities. In *Inst. Henri Poincare Probab. Stat.*, volume 49(3), page 885–899, 2013.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, et al. Ul2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022.

Joe Benton, Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. From denoising diffusions to denoising markov models. In *Arxiv eprints*, volume 2211.03595, 2022.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *NeurIPS*, 2021.

Winogrande: An adversarial winograd schema challenge at scale. 2019.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems 35 Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

B.D.O. Anderson. Reverse-time diffusion equation models. In *Stochastic Process. Appl.*, volume 12(3), 1982.

Giovanni Conforti, Alain Durmus, and Marta Gentiloni Silveri. Kl convergence guarantees for score diffusion models under minimal data assumptions. *SIAM Journal on Mathematics of Data Science*, pages 86–109, 2025.

Holden Lee. Parallelising glauber dynamics. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX-/RANDOM 2024, August 28-30, 2024, London School of Economics, London, UK*, volume 317 of *LIPIcs*, pages 49:1–49:24, 2024.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://skylion007.github.io/. *OpenWebTextCorpus*, 2019.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53, 2024.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-Based Diffusion Language Models. *arXiv preprint arXiv:2305.18619*, 2023.

Subham S. 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 *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Xiaochuang Han, Sachin Kumar, Yulia Tsvetkov, and Marjan Ghazvininejad. SSD-2: Scaling and Inference-time Fusion of Diffusion Language Models, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured Denoising Diffusion Models in Discrete State-Spaces. In *NeurIPS*, 2021.

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, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Depen Morwani, Itai Shapira, Nikhil Vyas, Eran Malach, Sham M. Kakade, and Lucas Janson. A new perspective on shampoo's preconditioner. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: improving and stabilizing shampoo using adam for language modeling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=IDxZhXrpNf.

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. *CoRR*, abs/2502.16982, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi S. Jaakkola, Brian Karrer, Ricky T. Q. Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. *Arxiv preprints*, abs/2410.20587, 2024.

Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/67b31ca159553d5593e62d7b998d63ea-Abstract-Conference.html.