Leveraging Pretrained Language Models as Energy Functions for Glauber Dynamics Text Diffusion

Anonymous ACL submission

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 with or outperform prior diffusion models and GPT-2 style auto-regressive models on zero-shot common sense reasoning tasks as well asplanning and 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 LMs 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) building on their success in continuous domains like image and video generation. (Nichol and Dhariwal, 2021; Tae et al., 2022; Ho et al., 2022; Ramesh et al., 2024). Continuous space diffusion models first define a forward Markov process that gradually converges to an easy-to-sample-from distribution (e.g., Gaussian). They then to learn time reversal of this process, or denoising, that can transform the noise back into a clean data sample.

Directly applying diffusion models to discrete text is nontrivial. Prior work has explored various approaches to either approximate text in a continuous domain (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 (Varma et al., 2024). Among these models, masked diffusion LMs (MDLMs) have received significant recent attention showing fast inference while almost matching generative quality of AR models, as well as improved performance on downstream reasoning benchmark (Kim et al., 2025). However, Zheng et al. (2025) rigorously demonstrate that **MDLMs ultimately are highly unlikely to outperform AR models**. These limitations of MDLMs stem from a fundamentally flawed forward Markov process and noisy/stationary distribution; we expand on this in section 5. Furthermore, (Liu et al., 2025b) also show that certain problems are inherently non-parallelizable and highlight other key difficulties of MDLMs whose inherent strength is their parallel sampling at the cost of often poorer quality.

Instead, we propose a first-principles discrete diffusion approach based on Glauber dynamics, a well-studied Markov process in theoretical CS 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 of the underlying stochastic dynamics (measured

using a notion of curvature).

Glauber dynamics is a natural way to address the latter, typically being the dynamics of choice for sampling from many challenging discrete state space distributions. It 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. 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 AR 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 LM. 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 AR 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 a GPT-style models as energy functions in the appendix, 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 than all prior discrete diffusion LMs and are competitive with the generative and zero-shot perplexities of AR models. We 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 MDLMs and hence AR 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 is simulated via small Euler steps or scaled Poisson clocks(Campbell et al., 2022). Diffusion admits 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 $\overline{Q}_t$:

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

where the multiplicative factors $e^{\log p_t(y)-\log p_t(x)} = p_t(y)/p_t(x)$ 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 "denoising" version of score entropy loss applied in the context of discrete diffusion models, called **diffusion weighted denoising score entropy** loss $\mathcal{L}_{\text{DWDSE}}$ is defined as:

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

where $K(a) = a(\log a - 1)$ and $s_\theta(x,t)_y$ is the score function that is meant to be close to $p_{t|0}(y)/p_{t|0}(x)$ for all $x \in \Omega$ and all $y \sim x$, i.e. all neighbors $y \in \Omega$ of x.

This setup is general 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 focus our efforts on understanding what makes for good or bad Markov chains in discrete state spaces, especially for language modeling. As we discuss in section 5, prior work shows that it is unlikely that currently popular masked diffusion LMs with absorbing Markov chain can surpass autoregressive LMs. We design a better alternative.

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 of the underlying stochastic process is sufficiently good (Conforti et al., 2025) which typically manifests as the stochastic process having fast decay of entropy along the 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. Unlike diffusion models, which receive one or few gradient signals per step, AR models trained using teacher forcing loss to obtain L signals per step, facilitating more efficient learning of linguistic structure, which 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. The UL2 (Unified Language Learning) framework (Tay et al., 2022) consolidates multiple pre-training objectives 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). 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 Glauber Dynamics Discrete Diffusion

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 jth iteration in the ith 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 tth timestep, the same index is updated. 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.

While our focus will be on a setup using encoder-decoder UL2 models that we describe next which is more efficient and forms a more coherent picture, we note that one can in principle use GPT-style causal models as a way to devise an energy function for the forward Glauber dynamics as well which we describe in appendix D.

3.1 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. We convert a pretrained UL2 model θ into a diffusion transformer (Peebles and Xie, 2023) by adding time-embedding parameters γ, initialized to zero at $t = T$, but these time parameters 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.

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

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.

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.

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.

Also, like Lou et al. (2024b); Varma et al. (2024), our Diffusion Transformer architectures uses best practices like AdaLN-Zero and RoPE for the time and positional embeddings respectively.

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

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 train it on the mixture of denoisers objective to get the corresponding UL2 model we denote as UL2-large. Our custom UL2-medium uses a T5 architecture (16 encoder/decoder layers, $d_{model} = 1024$, $d_{ff} = 2048$) trained on the mixture-of-denoisers objective. 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 $N \times L$. Other baselines include GPT-2-M/L/XL, as well as a continous diffusion model for text (Gulrajani and Hashimoto, 2023) and MDLM (Sahoo et al., 2024) and SEDD Absorb (Lou et al., 2024b) as well as GGM (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 4. We achieve significantly better generative perplexity results than all prior diffusion based text generative models and furthermore match GPT-2-M and are very close to matching that of GPT-2-L. We also 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, 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.

We also report in Table 1 the zero-shot perplexities of our models and some baselines using some of the datasets from (Radford et al., 2019).

4.3 Additional Experimental Results

Model	LAMB.	WT2	WT103	1BW
GPT-2-M	**15.60**	22.76	26.37	55.72
SEDD-M *	42.77	31.04	29.98	61.19
Glauber-UL2-M (N=1) *	17.89	23.95	30.21	56.12
Glauber-UL2-M (N=3) *	17.14	**20.98**	**25.47**	**52.18**
GPT-2-L	10.87	**19.93**	22.05	44.58
Glauber-UL2-L (N=1)*	11.25	21.54	24.71	47.62
Glauber-UL2-L (N=3) *	**10.14**	20.35	**20.83**	**44.12**

Table 1: Zero-Shot Validation Perplexities (LAMB: LAMBADA, WT: WikiText). * marked model perplexities are upper bounds

In Table 2, we report the MAUVE score (Pillutla

et al., 2021) of our model and compare it against the baselines from (Lou et al., 2024b). Here, the generation happens conditioned on a fixed prefix. Both our Glauber dynamics based models (N=1,3) achieve better MAUVE scores than that of SEDD-medium and GPT-2-medium.

Method	Annealing	Mauve
GPT-2-M	Nucleus-0.95	0.955
	None	0.802
SSD-LM	Logit Thresh-0.95	0.919
	None	0.312
SEDD-M	None	0.957
Glauber-UL2-M (N=1)	None	0.959
Glauber-UL2-M (N=3)	None	0.966

Table 2: MAUVE scores (higher is better)

Next, Table 3 contains the performance of our model 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).These tasks involve prompting the model with multiple-choice questions (2–4 options) and evaluating accuracy based on the answer. Appendix B and C contain Sudoku/Zebra puzzle results and text generation examples, respectively.

Model	**HS**	**Wino**	**PIQA**	**SIQA**
GPT-2-M	38.3	50.7	67.4	37.7
SEDD-M	31.5	49.0	56.1	35.4
(Gong et al., 2025)-M	37.2	52.6	59.6	39.0
Glauber-M(N=1)	37.4	49.9	66.8	37.4
Glauber-M(N=3)	40.5	52.9	68.9	39.1

Table 3: Performance of models on Common Sense Reasoning Tasks (HS: HellaSwag, Wino: Winogrande)

5 Related Work

Continuous Diffusion for Text Generation Early continuous diffusion approaches for text focused on Gaussian diffusion in latent embedding spaces (Li et al., 2022; Gulrajani and Hashimoto, 2023) or the vocabulary simplex (Han et al., 2022, 2023; Tae et al., 2025). However, these generally underperform autoregressive models without extensive empirical tuning. 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 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, most prior discrete diffusion models use independent forward transition kernels. This independence means token indices updated during reverse sampling may not align with those from training, potentially undermining the benefits of the diffusion process's time-varying nature. 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. Similarly, (Gong et al., 2025) derive diffusion models from AR weights via attention-mask annealing. However, as they remain within the MDLM paradigm and use a comparable loss function, they likely inherit the fundamental limitations of MDLMs identified by (Zheng et al., 2025). Furthermore, their generative perplexity, when measured using GPT-2-large, results are worse than

Evaluation Model			GPT2-L (774M)	GPT2-XL (1.6B)	GPT-NEO (2.7B)
Evaluated Model	**Sampling Algorithm**	**Total Params**	**Gen. PPL (↓)**	**Gen. PPL (↓)**	**Gen. PPL (↓)**
Autoregressive Models					
GPT2-M (Radford et al., 2019)	top-p, $p = 0.8, L = 1024, T = 1024$	345M	12.4	13.0	14.5
GPT2-L (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)	$\tau = 0.9, L = 1024, T = 4096$	1.3B	19.7	19.7	17.9
SEDD-M(Lou et al., 2024b)	$L = 1024, T = 2048$	424M	27.3	28.0	25.2
MDLM (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-M (pre-SEDD CAUSAL-GEN)	$L = 1024$	368M	21.7	21.3	20.4
UL2-M (post-SEDD, T=3 $\times$ L CAUSAL-GEN)	$L = 1024$	419M	19.1	19.6	19.9
Glauber-UL2-M (post-SEDD, T=L, i.e., $N = 1$)	$L = 1024$	419M	17.1	17.5	16.6
Glauber-UL2-M (post-SEDD, T=3 $\times$ L, i.e., $N = 3$)	$L = 1024$	419M	13.2	13.7	14.9
UL2-L (pre-SEDD CAUSAL-GEN)	$L = 1024$	783M	–	14.2	14.9
UL2-L (post-SEDD, T=3 $\times$ L CAUSAL-GEN)	$L = 1024$	898M	–	11.4	11.5
Glauber-UL2-L (post-SEDD, T=L, i.e., $N = 1$)	$L = 1024$	898M	–	9.5	9.9
Glauber-UL2-L (post-SEDD, T=3 $\times$ L, i.e., $N = 3$)	$L = 1024$	898M	–	6.9	7.8

Table 4: 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-M).

PLAID (Gulrajani and Hashimoto, 2023) which our model outperforms and so we do not compare it for it's generative quality. 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. They 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. They frame training as $O(L)$ binary classification problems, outperforming SEDD at the 350M scale. Crucially, this binary framework restricts output complexity to $O(T|\Sigma|)$, avoiding the $O(T|\Sigma|^2)$ requirement of modeling full token-to-token transitions at every step. Despite a different setup, our UL2-based Glauber dynamics similarly scales as $(O(T|\Sigma|))$ rather than $O(T|\Sigma|^2)$.

6 Conclusion and Limitations

We presented a text diffusion model based on Glauber dynamics. By framing Glauber dynamics as a mask-infilling task and utilizing pretrained language models as energy functions, we developed a diffusion framework that, for the first time, matches the performance of GPT-2-medium and GPT-2-large in language modeling. While our approach is slower than prior discrete diffusion models and autoregressive (AR) baselines, this trade-off is justified by superior text quality, matching or outperforming GPT-2 models, unlike most prior MDLMs, and achieving state-of-the-art performance on complex search tasks like Sudoku/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 under 6 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 which in an iso-TFLOPs comparison corresponds to 24 H100s on a GPT-2-M model size as opposed to our training time on GPT-2-L 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 Muon (Liu et al., 2025a) which achieve significant improvements over AdamW for training AR LLMs. Furthermore, 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

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

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

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

Gregor Bachmann and Vaishnavh Nagarajan. 2024. The Pitfalls of Next-Token Prediction. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 2296–2318. PMLR.

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

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

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. 2022. 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*.

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, and 16 others. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53.

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

Hans Follmer. 1985. 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.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. 2024. Discrete flow matching. 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*.

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

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. 2024. Scaling Diffusion Language Models via Adaptation from Autoregressive Models. *Preprint*, arXiv:2410.17891.

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. 2025. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.

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

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

Xiaochuang Han, Sachin Kumar, Yulia Tsvetkov, and Marjan Ghazvininejad. 2023. SSD-2: Scaling and Inference-time Fusion of Diffusion Language Models. *Preprint*, arXiv:2305.14771.

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

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

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

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large Language Models Cannot Self-Correct Reasoning Yet. In *The Twelfth International Conference on Learning Representations*.

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

Holden Lee. 2024. Parallelising glauber dynamics. In *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.

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

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

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

Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. 2021. Limitations of autoregressive models and their alternatives. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5147–5173, Online. Association for Computational Linguistics.

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

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, and 9 others. 2025a. Muon is scalable for LLM training. *CoRR*, abs/2502.16982.

Yuxi Liu, Konpat Preechakul, Kananart Kuwaranancharoen, and Yutong Bai. 2025b. The serial scaling hypothesis. *Arxiv preprints 2507.12549*.

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

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

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

William Peebles and Saining Xie. 2023. 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.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Krithika Ramesh, Nupoor Gandhi, Pulkit Madaan, Lisa Bauer, Charith Peris, and Anjalie Field. 2024. Evaluating differentially private synthetic data generation in high-stakes domains. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15254–15269, Miami, Florida, USA. Association for Computational Linguistics.

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

Subham S. Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. 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*.

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

Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Panigrahy. 2024. Causal language modeling can elicit search and reasoning capabilities on logic puzzles. 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*.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. 2023. Parallel sampling of diffusion models. In *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*.

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

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

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

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

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, and 1 others. 2022. Ul2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

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

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

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

Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua M. Susskind, and Navdeep Jaitly. 2023. PLANNER: Generating Diversified Paragraph via Latent Language Diffusion Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. 2025. 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*.

A Training Algorithm Additional Details

In this section, we provide more details about our training algorithm in words. We reiterate that our noisy distribution is the causal generation from the DiT UL2 model with T = 1024 or 3072, the idea is that this corresponds to causal generation from the initial UL2 model but of course that distribution has changed somewhat since the weights are shared in the DiT UL2 model. For the training procedure, we will only need the mask infilling capabilities of this DiT UL2 model (across all times).

Now, at each epoch k, we freeze a copy of the model weights corresponding to that of the final timestep, T and denote that as T_k, this will essentially be our forward Markov chain used for noising samples. In actuality, within each epoch, every few iterations, we instead update the frozen copy of the model weights to be an exponential moving average of the frozen copy of the model weights and the current model weights in order to stabilize training. Now, in each iteration, we take a sample from our dataset x and sample uniformly a time between 0 and T. We then run t steps of the forward Markov chain coming from T_k. As the Markov chain is sequential, in order to reuse many intermediate steps to get more gradient signals, we actually consider the x_t not just for t but for 32 timesteps between 0 and t uniformly, i.e., $x_{i\times\lfloor t/32\rfloor}$. Now for each of these x_t's we will compute the SEDD loss.

Recall that the SEDD loss is the expectation over $x_0 \sim p_{\mathcal{D}}$ of

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

Now, for each term inside the integral, we understand how we can compute it to be used during backprop. The first term is $Q_t(x_t,y)s_\theta(x_t,t)_y$ which is essentially the probability of our model at t in the reverse process so this is essentially the DiT model outputs at time t so we can compute that directly. Now for the second term, to compute $\log s_\theta(x,t)_y$ we just write it as the first term $Q_t(x_t,y)s_\theta(x_t,t)_y$ (so computed as before) divided by the prediction according to T_k our frozen forward noising model. Also, for computing

$\frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)}$, we use the frozen model T_k by storing the probabilities as we ran the Markov chain from 0 to time t (by computing the probabilities upto time $t-1$ and then taking the tth step for getting to y versus getting to x_t). Since this is using the frozen model, there is no backpropogation on this.

B Sudoku and Zebra Puzzles Results

In this subsection, we report our results for solving Sudoku and Zebra/Einstein puzzles(Shah et al., 2024). In the Sudoku puzzle setup, we are given a 9×9 grid where each cell is to be occupied by a number in the range $\{1, 2, \ldots, 9\}$. The constraints are that the numbers along each row and column should be unique. In addition, the numbers within each 3×3 mini-grid should also be unique. Given a set of initially filled positions, the goal is to figure out the values that can occur in the unfilled cells. In standard Sudoku puzzles, there will always only exist a unique solution to the puzzle.

Our models are trained by taking a random but fixed permutation upfront for each round (including the initial auto-regressive generation) whereas the auto-regressive baseline is trained both with and without correct ordering information and the masked diffusion language model baseline results are taken directly from (Kim et al., 2025) and we refer to that paper for more details. We stress that picking a random ordering upfront rather than the correct ordering information handicaps our model relative to the AR baseline and we believe this to be reasonable given that we're allowed to do multiple passes to correct our model outputs. We do however do not just mask-unmask one cell at a time during the refinement process and instead at any given time mask the 6 next cells in the permutation and predict just one cell at a time. This is because we observe much slower convergence when we were only masking-unmasking one cell at a time likely because it gets stuck in local stuck configurations for the puzzles. We present our results in Table 5

Next, we present the Zebra or so-called Einstein riddles. These are a more verbal style of a puzzle where we need to fill in values in a grid again but this time the type of possible constraints is much richer. An example of such a riddle is as follows:

There are 3 people next to each other in a row. Everyone has a different name: Ali, Rose, Randy. Every one lives in a different colored house: gold, silver, indigo. Everyone likes a different drink: orange juice, beer, coffee. Match the people to the correct value for each of their characteristics using the clues.

1. The person who likes orange juice is immediately to the left of the person who likes coffee.
2. The person who likes beer is somewhere to the left of the person who lives in the indigo house.
3. The person at the 1st position is Rose.
4. Randy is not the person who likes orange juice.
5. Randy is the person who lives in the gold house.

We present the accuracy results in Table 6. Ordering for our models and baseline comparison setup is the same as in the Sudoku case however here we don't need to keep 6 cells masked but rather just one at a time works fine.

C Samples generated from our model

In this section we give some unconditional and some prefix-conditioned examples from our larger model.

C.1 Unconditional Samples

Example Generation 1

Urban gardening has emerged as a powerful movement with deep historical roots and a promising future, reshaping how people interact with their environments and food systems. The practice of cultivating plants in urban spaces is not a new phenomenon; it has existed in various forms for centuries, serving as a vital source of nutrition, community, and resilience. In ancient civilizations, city-dwellers often grew crops within or just outside city walls to ensure a steady food supply during times of siege or scarcity. The famous Hanging Gardens of Babylon, whether myth or reality, symbolize the enduring human desire to bring nature into urban settings. Throughout history, urban gardens have provided fresh produce, medicinal herbs, and a sense of connection to the natural world. During the Industrial Revolution, as cities expanded and populations grew, many people lost access to land and traditional farming, leading to increased food insecurity and health problems. In response, urban gardening initiatives began to take shape, particularly in Europe and North America. Community gardens, allotments, and school gardens became common, offering city residents a way to grow their own food, improve their diets, and build social bonds. In the early 20th century, urban gardening gained momentum as a response to economic hardship. The Victory Gardens of World War I and II encouraged citizens to grow food to support the war effort and reduce pressure on national

Model	Params	Accuracy
AR (w/o ordering)	42M	9.73%
AR (w/ ordering)	42M	87.18%
MDLM (vanilla)	6M	6.88%
MDLM (Top-K Prob.)	6M	18.51%
MDLM (Top-K Margin)	6M	89.49%
Glauber-UL2 (N=1)	7M	42.26%
Glauber-UL2 (N=3)	7M	91.82%

Table 5: Performance on Sudoku tasks

Model	Params	Accuracy
AR (w/o ordering)	42M	80.3%
AR (w/ ordering)	42M	91.2%
MDLM (vanilla)	19M	76.9%
MDLM (Top-K Prob.)	19M	98.5%
MDLM (Top-K Margin)	19M	98.3%
Glauber-UL2 (N=1)	21M	94.1%
Glauber-UL2 (N=3)	21M	98.7%

Table 6: Performance on Zebra riddles

food supplies. These gardens not only contributed to food security but also fostered a sense of patriotism and community spirit. In the decades that followed, urban gardening experienced periods of decline and resurgence, influenced by social, economic, and environmental factors. By the late 20th century, concerns about industrial agriculture, food safety, and environmental degradation led to a renewed interest in urban gardening. The rise of the environmental movement and increased awareness of climate change further fueled the popularity of urban agriculture. Today, urban gardening takes many forms, from rooftop gardens and balcony planters to community gardens and urban farms. These spaces provide fresh, locally grown food, reduce the environmental impact of food transportation, and help mitigate the urban heat island effect. Urban gardens also serve as important educational tools, teaching children and adults about the origins of food, the importance of biodiversity, and the benefits of sustainable living. In many cities, urban gardening has become a tool for social change, empowering marginalized communities to take control of their food supply and create green spaces in underserved neighborhoods. Community gardens often become hubs for cultural exchange, bringing together people from diverse backgrounds to share knowledge, traditions, and harvests. The future of urban gardening looks bright, as advances in technology and design make it easier and more efficient to grow food in cities. Innovations such as vertical farming, hydroponics, and aquaponics allow for year-round cultivation, even in small or unconventional spaces. Smart irrigation systems, automated lighting, and soil sensors help urban gardeners optimize plant growth and conserve resources. Urban gardening is also being integrated into urban planning and policy, with cities around the world implementing green infrastructure initiatives and supporting urban agriculture projects. These efforts not only enhance food security but also contribute to climate resilience, biodiversity conservation, and public health. As cities continue to grow and face challenges such as climate change, population growth, and resource scarcity, urban gardening will play an increasingly important role in creating sustainable, livable, and resilient urban environments. The movement is supported by a growing network of organizations, activists, and policymakers who recognize the multiple benefits of urban agriculture. Educational programs, workshops, and online resources make it easier than ever for people to get involved and start their own gardens. The rise of social media has also helped to spread ideas, share success stories, and connect urban gardeners across the globe. Looking ahead, urban gardening has the potential to transform cities into greener, healthier, and more equitable places. By reimagining unused spaces, fostering community engagement, and embracing innovation, urban gardening can help address some of the most pressing challenges of our time. Whether on a balcony, rooftop, or vacant lot, every garden contributes to a larger vision of sustainable urban living. The history and future of urban gardening remind us of the power of small actions to create meaningful change, and the importance of nurturing our connection to the earth, even in the heart of the city. As more people embrace urban gardening, the movement will continue to grow, inspiring new generations to cultivate not only plants but also hope, resilience, and a sense of belonging in an ever-changing world.

Example Generation 2

There's something truly special about rainy days, something that just makes you want to stay inside and watch the drops fall against the window. Rainy days are cozy, they're comforting, and they make you appreciate the little things in life, like a warm cup of tea or coffee, maybe a good book or a favorite movie. The sound of rain on the roof is soothing, almost hypnotic, and it can make you feel like you're wrapped in a blanket of calm, even if you're just sitting at your desk or lying on the couch. Rainy days slow things down, they make the world outside seem a little quieter, a little softer, and that's a feeling you don't get on sunny days. When it rains, people tend to stay indoors, and that can make the city feel different, more relaxed, less rushed. You might notice that everyone moves a little slower, talks a little softer, and seems to take their time. There's a sense of unity on rainy days, as if everyone is sharing the same experience, the same rhythm of the weather. Rainy days are also great for creativity, or at least that's what a lot of people say. Maybe it's because there are fewer distractions, or maybe it's because the rain itself inspires a certain mood, but rainy days seem to be perfect for writing, painting, or just thinking deeply about things. Sometimes, when it rains, you might find yourself daydreaming more, letting your mind wander to places you haven't thought about in a while. Rainy days can be nostalgic, too, bringing back memories of childhood, like splashing in puddles or watching cartoons on a lazy afternoon. For some people, rainy days are a time for reflection, a chance to pause and take stock of life, to think about where you are and where you want to go. Rainy days can also be a little sad, but in a good way, like the kind of sadness that feels cleansing, that helps you let go of things you've been holding onto. The rain washes away the dust and the dirt, and sometimes it feels like it washes away your worries, too. Of course, not everyone loves rainy days. Some people find them depressing, or boring, or just inconvenient, especially if they have to go outside and get wet. But for those who do enjoy them, rainy days are a gift, a chance to slow down and appreciate the world in a different way. Rainy days are also great for naps, because the sound of the rain is like nature's lullaby, and it's easy to drift off to sleep with that gentle patter in the background. Sometimes, when it rains, you might notice the smell of the earth, that fresh, earthy scent that comes after a good downpour. It's called petrichor, and it's one of those little details that make rainy days special. Rainy days are also a good excuse to stay in your pajamas all day, or to wear your favorite sweater, or to make soup or bake cookies. There's something about the combination of rain and comfort food that just feels right, like the universe is giving you permission to take it easy. Rainy days can also be romantic, especially if you're sharing them with someone you love. There's nothing quite like cuddling up with someone while the rain falls outside, or watching a movie together under a blanket, or just talking for hours without any distractions. Rainy days are a reminder that it's okay to slow down, to take a break, to let yourself relax and recharge. In a world that's always rushing, always demanding more, rainy days are a chance to pause, to breathe, to just be. Sometimes, when it rains, you might feel a little more introspective, a little more thoughtful, and that's not a bad thing. Rainy days can help you reconnect with yourself, with your thoughts and feelings, and with the people around you. They can also be a time for new beginnings, because the rain brings new life, new growth, and new possibilities. After the rain, the world looks different, brighter, fresher, as if it's been renewed. That's one of the things I love most about rainy days—the way they make everything feel new again. Rainy days are also a great time for writing letters, or sending messages to friends, or just reaching out to someone you haven't talked to in a while. There's something about the rain that makes you want to connect, to share your thoughts and feelings, to let someone know you're thinking of them. Rainy days can be a little lonely, too, but

Example Generation 3

sometimes I just forget things. Like, I'll walk into a room and forget why I went in there. It happens all the time, almost every day, and it's kind of funny but also a little annoying. I think everyone forgets things sometimes, like where they put their keys or what they were supposed to buy at the store. It's just part of being human, I guess. Forgetting things can be frustrating, especially when you really need to remember something important, but it's also kind of normal. I remember one time I forgot my friend's birthday, and I felt really bad, but they said it was okay, and we laughed about it. Forgetting things is just something that happens, and it's not always a big deal. Sometimes, when I forget something, I try to retrace my steps in my mind, like thinking about where I was or what I was doing before I forgot. That sometimes helps, but not always. Sometimes, I just have to accept that I forgot and move on. Forgetting things can be a little embarrassing, especially if it happens in front of other people, but most people understand because it happens to them too. I think forgetting things is just part of life, and it's not something to worry about too much. Of course, if you forget things all the time, like really important things, then maybe it's a good idea to talk to someone about it. But for the most part, forgetting things is just a normal part of being alive. Sometimes, I wonder why we forget things. Maybe it's because our brains are too full, or maybe it's because we're distracted, or maybe it's just because we're tired. I think being tired makes you forget things more easily, at least that's what I've noticed. When I don't get enough sleep, I forget things more often, like where I put my phone or what I was supposed to do that day. Forgetting things can be frustrating, but it's also kind of funny when you think about it. Like, how can you forget something you were just thinking about a minute ago? It's weird, but it happens. Sometimes, I forget things because I'm thinking about something else, or because I'm in a hurry, or because I'm just not paying attention. I think paying attention is really important if you want to remember things, but it's hard to pay attention all the time. Life is busy, and there's always something going on, so it's easy to forget things. Forgetting things is just part of being human, and it's not something to be ashamed of. I think most people forget things sometimes, even if they don't admit it. Sometimes, I forget things and then remember them later, like in the middle of the night or when I'm doing something completely unrelated. It's funny how that works. The other day, I forgot where I put my glasses, and I looked everywhere for them, and then I found them on my head. That was pretty funny, and I laughed about it. Forgetting things can be annoying, but it can also be kind of funny, especially when you realize how silly it is. Sometimes, I forget things because I'm stressed or worried about something else, and my mind is just somewhere else. I think stress makes you forget things more easily, at least that's what I've noticed. When I'm stressed, I forget things all the time, like appointments or deadlines or what I was supposed to do. It's frustrating, but it's also kind of normal. I think forgetting things is just part of being alive, and it's not something to worry about too much. Of course, if you forget things all the time, like really important things, then maybe it's a good idea to talk to someone about it. But for the most part, forgetting things is just a normal part of being human. Sometimes, I forget things and then remember them later, like in the middle of the night or when I'm doing something completely unrelated. It's funny how that works. The other day, I forgot where I put my keys, and I looked everywhere for them, and then I found them in my pocket. That was pretty funny, and I laughed about it. Forgetting things can be annoying, but it can also be kind of funny, especially when you realize how silly it is. Sometimes, I forget things because I'm distracted or because I'm thinking about something else, or because I'm just not paying attention. I think paying attention is really important if you want to remember things, but it's hard to pay

C.2 Prefix Conditioned Samples

Example Generation 1

The history of artificial intelligence is a fascinating journey that spans multiple decades, encompassing a series of breakthroughs, setbacks, and paradigm shifts. In the mid-twentieth century, the groundwork for AI was laid by visionary thinkers who dreamed of machines that could mimic human intelligence. Alan Turing, often regarded as the father of theoretical computer science, proposed the idea of a machine that could think and introduced the Turing test as a measure of intelligence. Early AI research was characterized by optimism and bold predictions. Scientists and mathematicians believed that within a few years, computers would be capable of complex reasoning, natural language understanding, and even creative thought. The initial wave of AI, known as symbolic AI, relied on rule-based systems and logic. Researchers painstakingly encoded human knowledge into computer programs, hoping to create systems that could solve problems, play games, and answer questions. The field experienced its first wave of enthusiasm in the 1950s and 1960s, with projects such as the General Problem Solver and the development of the first expert systems. However, as researchers soon discovered, human intelligence is far more complex than initially imagined. The limitations of symbolic AI became apparent as systems struggled to handle ambiguity, context, and real-world variability. The field entered a period known as the "AI winter," where funding and interest declined due to unmet expectations. Despite these setbacks, research continued behind the scenes, and new approaches began to emerge. The advent of machine learning in the 1980s and 1990s marked a significant turning point. Instead of relying on hand-coded rules, researchers began to develop algorithms that could learn from data. Early machine learning techniques, such as decision trees and neural networks, showed promise but were limited by the available computational power and data. The rise of the internet and the digital revolution in the late 1990s and early 2000s provided the necessary infrastructure and data for machine learning to flourish. With the explosion of online information, researchers had access to vast amounts of data that could be used to train more sophisticated models. The development of deep learning in the 2010s further accelerated progress in AI. Deep neural networks, inspired by the structure of the human brain, demonstrated remarkable capabilities in image and speech recognition, natural language processing, and even game playing. Breakthroughs such as AlphaGo, which defeated world champions in the ancient game of Go, captured the public imagination and demonstrated the potential of AI to surpass human performance in specific domains. Today, artificial intelligence is deeply integrated into many aspects of daily life, from virtual assistants and recommendation systems to autonomous vehicles and medical diagnostics. AI technologies are transforming industries, enabling new forms of creativity, and raising important ethical and societal questions. As AI continues to evolve, researchers are exploring new frontiers, such as explainable AI, which aims to make machine learning models more transparent and interpretable. The future of artificial intelligence holds both promise and challenges. On one hand, AI has the potential to solve some of the world's most pressing problems, from climate change to healthcare. On the other hand, concerns about job displacement, privacy, and the misuse of AI remain significant. The ongoing development of artificial intelligence will require collaboration across disciplines, thoughtful regulation, and a commitment to ethical principles. As we look ahead, it is clear that AI will play an increasingly central role in shaping the future of humanity, driving innovation and transforming the way we live, work, and interact with technology. The next decades will undoubtedly bring new breakthroughs, challenges, and opportunities, as artificial intelligence continues to push the boundaries of what is possible. The journey of AI is far from over, and the story of its evolution is still being written. Every day, researchers around the world are working to unlock new capabilities, address limitations, and ensure that artificial intelligence is developed and deployed in ways that benefit society as a whole. From its humble beginnings to its current state of rapid advancement, artificial intelligence represents one of the most exciting and transformative fields in science and technology. Whether AI will ever achieve true general intelligence or consciousness remains an open question, but the progress made so far is a testament to human ingenuity and the power of collaboration. As we continue to explore the possibilities of artificial intelligence, we must remain mindful of the ethical implications and strive to create a future where AI serves the greater good. The evolution of artificial intelligence is not just a story of technology, but also a story of human ambition, creativity, and the endless pursuit of knowledge.

Example Generation 2

Today the cold war ended. *Or at least, that's what they say. I'm not sure if it really ended, or if it just changed, or maybe it's still going on in some way. I think the Cold War was about the United States and the Soviet Union, and they were always arguing or threatening each other, but they never actually fought a big war. Or maybe they did, but not the kind of war with soldiers and tanks, more like a war of words and spies and maybe space. I remember something about the space race, where they tried to see who could get to the moon first. Or was it Mars? Maybe it was the moon. I think the United States got to the moon first, but I'm not sure. Anyway, today the Cold War ended, or at least that's what the news said. I don't know if it's true, because sometimes the news gets things wrong, or maybe they just want people to feel better. I think the Cold War was about nuclear weapons, too, and everyone was scared that someone would push the button and start a big explosion. I remember hearing about bomb shelters and people practicing hiding under their desks, which seems silly now, but maybe it made sense back then. I'm not sure. Today, the Cold War ended, but I don't know what that means for the world. Maybe things will be better now, or maybe they'll just be different. I think the Cold War lasted a long time, like maybe forty years, or maybe more. I'm not good with dates, so I might be wrong. I think it started after World War II, but I'm not sure. I remember something about the Berlin Wall, and how it fell, and people were happy, but I'm not sure if that was the end of the Cold War or just a part of it. Today, the Cold War ended, but I don't know if anyone will notice. I think most people are busy with other things, like work or school or watching TV. Maybe the Cold War ending is a big deal, but maybe it's not. I'm not sure. I think the Cold War was about politics and power, and maybe about who could control the world, or at least parts of it. I remember something about communism and capitalism, and how they were fighting over which one was better. I'm not sure which one is better, but I think most people like capitalism, at least where I live. I think the Cold War was also about movies and music, and how each side tried to show that their way of life was better. I remember something about jazz and rock and roll, and how the Soviet Union didn't like it, but I'm not sure if that's true. Today, the Cold War ended, but I don't know what will happen next. Maybe there will be more peace, or maybe there will be new problems. I think the world is always changing, and it's hard to keep up. I think the Cold War was a big part of history, and maybe it will be in the history books, but I'm not sure if kids will care. I think most people just want to live their lives and not worry about big things like wars or politics. Today, the Cold War ended, but I'm not sure if it really matters. I think it's good to have peace, but I don't know if that's what will happen. Maybe there will be new wars, or maybe things will stay the same. I'm not sure. I think the Cold War was about fear, and about people being scared of each other, and maybe that's still happening, even if the Cold War is over. I think fear is a big part of life, and maybe it will always be there, no matter what happens. Today, the Cold War ended, but I don't know if that will change anything. I think the world is complicated, and it's hard to understand everything. I think the Cold War was about more than just two countries, but I'm not sure what else it was about. Maybe it was about ideas, or about the future, or about who gets to decide what happens next. I'm not sure. Today, the Cold War ended, but I don't know if that's true, or if it's just something people are saying. I think the world is always changing, and it's hard to keep up. I think the Cold War was a big part of history, and maybe it will be in the history books, but I'm*

Example Generation 3

Jazz music is really something special. *Or at least that's what I think. I'm not sure where jazz started, but I think it was in America, maybe in New Orleans, or maybe somewhere else. I remember hearing something about jazz being a mix of different kinds of music, like blues and ragtime and maybe even classical, but I'm not sure. Jazz is kind of hard to explain, but it's all about improvising, or making things up as you go along. Sometimes jazz musicians just start playing and see where the music takes them, and that's really cool. I think jazz is about freedom, and about expressing yourself, and about not following all the rules. I'm not sure if that's true, but it seems like jazz is always changing, always moving, always trying new things. I remember hearing jazz on the radio when I was a kid, and it always made me feel happy, or maybe a little sad, depending on the song. I think jazz can make you feel all kinds of things, and that's part of what makes it special. Jazz is also about the instruments, like the saxophone and the trumpet and the piano, and sometimes the drums and the bass. I remember seeing a jazz band once, and the saxophone player was amazing, just playing all these crazy notes and making the music come alive. Sometimes jazz musicians play solos, where one person gets to shine, and everyone else supports them, and that's really nice. I think jazz is about community, and about sharing something special with other people. I'm not sure if that's true, but it seems like jazz brings people together, even if they don't know each other. I just know that jazz is always evolving, always trying new things, and that's part of what makes it exciting. I think jazz is about taking risks, and about not being afraid to make mistakes, because sometimes the best music comes from mistakes. I remember hearing a jazz musician say that, and it made sense to me. I think that's why jazz is so powerful, because it comes from the inside, from the soul, or at least that's what people say. I'm not sure if that's true, but it sounds nice. Jazz is also about rhythm, and about the beat, and about making people want to move. I think jazz can make you dance, or at least tap your foot, and that's always fun. I remember hearing jazz at a party once, and everyone was dancing and having a good time, and it was really cool. Jazz is also about the night, and about staying up late and listening to music, and about feeling alive. I think jazz is best at night, when everything is quiet and you can really listen. I'm not sure if that's true, but it seems like jazz is made for the night. Jazz is also about the past, and about remembering all the great musicians who played before, like Louis Armstrong and Miles Davis and Duke Ellington, but I'm not sure if I got all the names right. I think jazz is about tradition, but also about breaking the rules and trying new things. I'm not sure if that makes sense, but it seems like jazz is always changing, always moving, always trying to find new ways to express itself. I think jazz will always be around, because it's so flexible and so open to new ideas. I'm not sure if that's true, but I hope it is. Jazz is also about listening, and about paying attention, and about really hearing the music. I think jazz is best when you listen closely, and when you let yourself get lost in the sound. I remember hearing jazz on the radio, and sometimes I would just close my eyes and listen, and it felt like the music was taking me somewhere else. Jazz is also about mistakes, and about not being perfect, and about being okay with that. I think jazz is about taking chances, and about not being afraid to try something new, even if it doesn't work out. I remember hearing a jazz musician say that, and it made sense to me. Jazz is also about feeling, and about putting your heart into the music, and about letting go and just playing. I think that's why jazz is so powerful, because it comes from the soul. I'm not sure if that's true, but it sounds nice. Jazz is also about rhythm, and about the beat, and about making people want to move. I think jazz can make you dance, and that's always fun. I remember hearing*

D 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 (2.1). 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.

E Training Hyperparameters

We use the T5 tokenizer which has a vocabulary of 32100 and we train on sequences of length 1024. We use the AdamW optimizer (with $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 10^{-8}$ and 0.9999 EMA, no weight decay and a linear warmup schedule for the first 9000 steps to 3×10^{-4} and then decay to 10^{-6} using a cosine delay schedule over the remaining steps and we use a gradient accumulation of 4. We train our models using 32 H100s for 2 epochs which takes approx 6 days to finish training for our larger model (we highlight that based on the reported training time of the GGM paper (Varma et al., 2024), their training takes about 8 days to train a model one size smaller than ours using TPU compute which on a TFLOP basis is roughly equivalent to 24 H100s). We use 128 examples in each batch with 32 different time-steps bringing our batch size to 4096 effectively