Promises and Pitfalls of Generative Masked Language Modeling: Theoretical Framework and Practical Guidelines

Yuchen Li [1 2] Alexandre Kirchmeyer [* 1] Aashay Mehta [* 1] Yilong Qin [* 1]
Boris Dadachev [* 2] Kishore Papineni [* 2] Sanjiv Kumar [2] Andrej Risteski [1]

Abstract

Autoregressive language models are the currently dominant paradigm for text generation, but they have some fundamental limitations that cannot be remedied by scale—for example inherently sequential and unidirectional generation. While alternate classes of models have been explored, we have limited mathematical understanding of their fundamental power and limitations. In this paper we focus on *Generative Masked Language Models (GMLMs)*, a non-autoregressive paradigm in which we train a model to fit conditional probabilities of the data distribution via masking, which are subsequently used as inputs to a Markov Chain to draw samples from the model. These models empirically strike a promising speed-quality trade-off as each step can be typically parallelized by decoding the entire sequence in parallel. We develop a mathematical framework for analyzing and improving such models which sheds light on questions of sample complexity and inference speed and quality. Empirically, we adapt the T5 model for iteratively-refined parallel decoding, achieving 2-3x speedup in machine translation with minimal sacrifice in quality compared with autoregressive models. We run careful ablation experiments to give recommendations on key design choices, and make fine-grained observations on the common error modes in connection with our theory. Our mathematical analyses and empirical observations characterize both potentials and limitations of this approach, and can be applied to future works on improving understanding and performance of GMLMs.

1. Introduction

The current dominant approach to language modeling is *autoregressive* (AR): to generate a sequence of tokens, the language model starts by predicting the leftmost token, and then proceeds from left to right, each step predicting the next token based on everything on its left (Raffel et al., 2020; Brown et al., 2020; Touvron et al., 2023). AR models are not without limitations:

1. *Lack of parallelism:* To generate a sequence of N tokens, AR language models need N sequential decoding steps. Each step consists of a forward pass of the decoder component. When N is large, N sequential decoding steps incur high latency.

2. *Quality:* When predicting each token, the model cannot access its right hand side context, and has no natural way to revise earlier predictions on the left. This intuitive limitation was more formally explored in prior theoretical works (Li & Risteski, 2021; Lin et al., 2021; Bachmann & Nagarajan, 2024).

One promising alternative is based on *Generative Masked Language Models (GMLMs)*. They are trained to fit conditional probabilities for parts of the sequence (by applying a mask), conditioned on the rest. To produce samples, these conditionals are used as oracles for running Markov Chain, e.g. a Gibbs sampler (Wang & Cho, 2019; Goyal et al., 2022). Alternatively, we can think of these steps as an iterative refinement process, typically starting with pure noise (i.e. all tokens are masked or randomized). One can even fit conditional probabilities for noised versions of the input distribution, and use them as inputs to a denoiser to get certain types of discrete diffusion models (Austin et al., 2021). In GMLMs, typically one step of the Markov Chain is operationalized by a Transformer that generates the sequence in parallel (i.e. *parallel decoding* (Ghazvininejad et al., 2019; Gu & Kong, 2021; Savinov et al., 2022)). Hence, if the total number of steps is small, the latency is low. [1]

[*] Equal contribution [1] Carnegie Mellon University [2] Google Research. Correspondence to: Yuchen Li <yuchenl4@cs.cmu.edu>, and Boris Dadachev <dadachev@google.com>, and Andrej Risteski <aristesk@andrew.cmu.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] Our codes are released at `https://github.com/google-research/google-research/tree/master/padir`

However, none of these approaches robustly surpass autoregressive models in both speed and quality for a wider range of language generation tasks beyond machine translation. Thus, the following questions naturally arise:

(Q1) GMLMs are trained to learn conditional probabilities. When does it also imply learning the *joint* probability?

(Q2) What properties of the data distribution and training/inference algorithm govern the quality of the learned model and its generated samples?

(Q3) What are the best practices for training GMLMs, and can we use theory to elucidate the design space of losses, training and inference procedures?

Our contributions. Towards answering the questions above, we introduce a *theoretical framework* to characterize the potentials and limitations of GMLMs, for both training and inference. Precisely,

- The **asymptotic sample complexity** for estimating the parameters of a distribution via a broad class of masked-prediction losses can be related to the mixing time of a corresponding Markov Chain that can be used to sample from the distribution (Section 2.2). In particular, we prove that training with larger masks always improves statistical efficiency (Theorem 1).
- We show **finite-sample bounds** that relate how well the *conditional* distributions of the data distribution are learned, to how well the *joint* distribution is learned (Section 2.3) if we have some capacity control over the distribution class being learned (e.g. covering number bounds).
- **Transformers** are only able to represent decoding steps that factorize over the coordinates—preventing them from **efficiently sampling** even simple distributions with strong correlations between the coordinates (Section 2.4).

We accompany these theoretical findings with an extensive set of empirical investigations detailing important components and common error modes. Precisely:

- Our experiments (Section 3) suggest the **empirically critical components** include large masking ratio (c.f. theory in Section 2.2), custom vocabulary, distillation from AR models, and architecture improvements like positional attention. [2]
- GMLMs with parallel-decoding work well on **machine translation**: in fact, even *one single* forward pass can often produce reasonable translations. This aligns with our theoretical framework, as machine translation tasks typically involve lower-entropy and less multi-modal outputs, compared to other language generation tasks.
- Common **error modes** ("stuttering") suggest limitations for parallel-decoding GMLMs for modeling strong dependencies (c.f. theory in Section 2.4), which we empirically quantify (Section 3.4).

Jointly, our theoretical and empirical findings suggest synergistically designing better Markov Chains that mix fast in the presence of strong correlations in the target, and corresponding losses that inherit good statistical behavior.

2. Theoretical framework

We develop a mathematical framework for reasoning about the core ingredients for successfully training and using GMLMs: the *statistical complexity* to learn the model, and the *speed of inference*. We show that these two are surprisingly closely related: namely, we understand both the asymptotic and finite-sample statistical complexity through *functional inequalities* (e.g. Poincaré, approximate tensorization of entropy) corresponding to the Markov Chains we would use at inference time—which in turn characterize the mixing time of these chains. This picture closely mirrors an emerging picture in the continuous case for score-based (diffusion) models (Koehler et al., 2023; Qin & Risteski, 2023)—though with somewhat different proof techniques.

2.1. Setup and notation

The most classical way of fitting distributions from data is maximum likelihood: that is, finding the choice of parameters that maximize the likelihood of the training data. There are well-understood statistical reasons to do so: in the asymptotic sense (as the number of sample grows), maximum likelihood is the most sample-efficient way to estimate the distribution (Hájek, 1972). However, many families of distributions are such that optimizing maximum likelihood is computationally challenging. Thus, many alternate strategies and losses to fit the parameters have been developed.

For continuous distributions, a common choice of loss is the *score matching* loss, where instead of fitting the likelihood, we fit the gradient with respect to the input of the log-pdf, namely $\nabla_x \log p_\theta(x)$. In certain cases, this can provable computational benefits over maximum likelihood (Pabbaraju et al., 2023). For discrete distributions, we cannot take gradients with respect to the input: though a closely related strategy is available — trying to match the conditionals of subsets of variables. (This can be thought of as "flipping" the coordinates in the subsets, while keeping the remaining coordinates fixed.) Operationalizing this as a loss

[2] The benefit of distillation was verified in Kim & Rush (2016); Gu et al. (2018); Zhou et al. (2020); Gu & Kong (2021). Positional attention was tested in Gu et al. (2018); Kreutzer et al. (2020).

gives us the *pseudolikelihood* loss (Besag, 1975).

Variants of this strategy have been used in classical results for learning Ising models (Ravikumar et al., 2010; Vuffray et al., 2016). More recently, this strategy has been used in conjuction with neural models to both learn useful features in the guise of masked language modeling (MLM) (Devlin et al., 2019), which can be also used to produce a generative model (Wang & Cho, 2019; Goyal et al., 2022). The latter is done by using the learned conditionals inside a Gibbs sampler. However, when the conditionals are not *consistent*, i.e. there is not a joint distribution that satisfies these conditionals, Gibbs sampler may amplify errors. In general, mathematical understanding about sampling from masked language models is still lagging substantially behind.

Setup: Let Ω be a finite discrete set. Let $p_{\mathcal{X}}$ denote a distribution over a sequence of N variables $X = (X_1, X_2, \cdots, X_N) \in \Omega^N =: \mathcal{X}$. [3] Furthermore, for $K \subset [N]$, let X_K denote the subsequence $(X_i \,|\, i \in K)$, and X_{-K} denote the subsequence $(X_i \,|\, i \notin K)$.

We consider learning parameters θ parametrizing some distribution p_θ over $\mathcal{X}$, for $\theta \in \Theta$. The classical way of fitting θ is to maximize the likelihood of the training data:

Definition 1 (MLE, (Van der Vaart, 2000))**.** *Given i.i.d. samples $x_1, \ldots, x_n \sim p_{\theta^*}$, the maximum likelihood estimator is $\hat{\theta}_{\mathit{MLE}} = \arg\max_{\theta \in \Theta} \hat{\mathbb{E}}\left[\log p_\theta(X)\right]$, where $\hat{\mathbb{E}}$ denotes the expectation over the samples. As $n \to \infty$ and under suitable regularity conditions, we have $\sqrt{n}\left(\hat{\theta}_{\mathit{MLE}} - \theta^*\right) \to N\left(0, \Gamma_{MLE}\right)$, where $\Gamma_{MLE} := \mathcal{I}^{-1}$, $\mathcal{I} := \mathrm{Cov}_{X \sim p_{\theta^*}}\left(\nabla_\theta p_\theta(X)\right)_{|\theta = \theta^*}$ is the Fisher information matrix.*

A classical result due to Hájek-Le Cam (for modern exposition see (Van der Vaart, 2000)) is that maximum likelihood is the asymptotically most sample-efficient estimator among all "sufficiently regular" estimators (Section 8.5 in Van der Vaart (2000)) — so we will treat it as the "gold standard" against which we will compare other estimators. The class of estimators we will be focusing most is the a broad generalization of the *pseudo-likelihood estimator* (Besag, 1975).

Definition 2 (Weighted pseudolikelihood)**.** *Consider a partition of $[N]$, namely a collection of sets $\mathcal{K} := \{K_1, \ldots, K_{|\mathcal{K}|}\}$ such that $\bigcup_i K_i = [N]$, and a distribution $p_{\mathcal{K}} : \mathcal{K} \to \mathbb{R}^+$.*

Given n i.i.d samples of sequences and masks: $\{\left(X^{(i)}, K^{(i)}\right) | X^{(i)} \sim p_{\mathcal{X}}, K^{(i)} \sim p_{\mathcal{K}}\}_{i \in [n]}$, the weighted maximum pseudolikelihood estimator (MPLE) *is $\hat{\theta}_{PL} := \arg\min_\theta \sum_{i=1}^n -\log p_\theta(X^{(i)}_{K^{(i)}} | X^{(i)}_{-K^{(i)}})$. The population loss is* [4] $L_{PL}(\theta) := \mathbb{E}_{X \sim p_{\mathcal{X}}, K \sim p_{\mathcal{K}}}\left[-\log p_\theta(X_K | X_{-K})\right]$.

As a special case, if $\mathcal{K}$ contains all subsets of a certain size k and $p_{\mathcal{K}}$ is uniform over $\mathcal{K}$, we get the classical k-pseudolikelihood estimator:

Definition 3 (k-pseudolikelihood (Huang & Ogata, 2002))**.** *Same as Definition 2 except that $\mathcal{K} := \{K \subseteq [N] \mid |K| = k\}$, $p_{\mathcal{K}} = \mathit{Unif}(\mathcal{K})$.*

Remark 1. *The distribution of X and K in the above loss is independent. In Section 2.2.3 we will show that our results readily generalize to losses in which the distribution of the masks K can depend on the current X. We present the independent case first for ease of presentation.*

Informally, we predict the variables in positions $K \in \mathcal{K}$, conditioned on the remaining variables. The benefit is that parametrizing conditionals over smaller subsets K is often computationally cheaper. For instance, if $p_\theta(x)$ is an undirected graphical model, i.e. $p_\theta(x) \propto \exp(\sum_C \phi_{C,\theta}(x_C))$, where the sum is over all maximal cliques C of the graph describing the distribution, the conditional distribution of K only depends on its Markov blanket, which can be very small for sparse graphs and small sets K. Thus, computing the partition function corresponding to $p(x_K | x_{-K})$ takes time exponential in this Markov blanket. By contrast, computing the likelihood requires calculating the partition function of $p_\theta(x)$, which takes time exponential in the dimension of X. In fact, for Ising models, the corresponding loss is even *convex* [5]. A similar tradeoff exists for masked language models: fitting the conditionals for larger masks would likely require a larger model, thus would be computationally more expensive.

2.2. Asymptotic sample efficiency via functional inequalities

In this section, we will provide a framework for bounding the asymptotic sample complexity of learning the parameters θ of a discrete probability distribution by minimizing a loss in a broad family of "masked prediction" objectives. We will measure the quality of an estimator in terms of *parameter recovery*. To make this formal, we first recall that under mild technical conditions, the estimator will be asymptotically normal:

Lemma 1 (Asymptotic normality (Van der Vaart, 2000))**.** *Consider the weighted MPLE objective in Definition 2, and let $\theta^* \in \arg\min_\theta L_{PL}(\theta)$. Under mild regularity conditions (Lemma 11 in Appendix J), as $n \to \infty$, $\sqrt{n}(\hat{\theta}_{PL} - \theta^*) \xrightarrow{d} \mathcal{N}(0, (\nabla^2_\theta L_{PL}(\theta^*))^{-1} \mathrm{Cov}(\nabla_\theta l_{PL}(\theta^*)) (\nabla^2_\theta L_{PL}(\theta^*))^{-1})$*

[3] In language models, Ω is the set of tokens in the vocabulary.

[4] This is equivalent to minimizing the KL divergence of the ground-truth conditional distribution $p(X_K | X_{-K})$ from the predicted conditional distribution $p_\theta(X_K | X_{-K})$: $\mathbb{E}_{X \sim p_{\mathcal{X}}, K \sim p_{\mathcal{K}}}\left[D_{\mathrm{KL}}\left(p(\cdot | X_{-K}), p_\theta(\cdot | X_{-K})\right)\right]$

[5] This fact is well known, but for completeness included in Appendix K

If we know $\sqrt{n}(\hat{\theta}_{PL} - \theta^*) \xrightarrow{d} \mathcal{N}(0, \Gamma_{PL})$, we can extract bounds on the expected ℓ_2^2 distance between $\hat{\theta}_n$ and θ^*. Namely, from Markov's inequality, (see e.g., Remark 4 in Koehler et al. (2023)), for sufficiently large n, with probability at least 0.99 it holds that $\|\hat{\theta}_{PL} - \theta^*\|_2^2 \leq \frac{\mathrm{Tr}(\Gamma_{PL})}{n}$.

2.2.1. Masking more is (statistically) better

As a first application of our framework, we prove that increasing the number of variables k we predict in k-pseudolikelihood (Definition 3) strictly improves the statistical efficiency of the resulting estimator. Note, for larger k, we expect the computational cost to optimize the corresponding loss to be larger, and when $k = N$ we just get max likelihood. Thus, this naturally formalizes a *computational/statistical tradeoff* in choosing k.

Assumption 1 (Finite gradient and Hessian). *$\forall \theta \in \Theta, \forall x \in \mathcal{X}, K \subset [N]$, the norms of the gradient $\|\nabla_\theta \log p_\theta(x_K|x_{-K})\|_2$ and the Hessian $\|\nabla_\theta^2 \log p_\theta(x_K|x_{-K})\|_F$ exist and are finite .*

Assumption 2 (Realizability). *The data distribution $p_\mathcal{X}$ satisfies: $\exists \theta^* \in \Theta$ such that $p_{\theta^*} = p_\mathcal{X}$.*

Theorem 1 (Masking more is (statistically) better). *Let Assumption 1 and Assumption 2 be satisfied, and let Γ_{PL}^k denote the asymptotic variance of the k-MPLE estimator (Definition 3). Then, we have:*[6] $\Gamma_{PL}^{k+1} \preceq \Gamma_{PL}^k$

Remark 2. *By monotonicity of trace, Thm 1 implies $\mathrm{Tr}(\Gamma_{PL}^{k+1}) \leq \mathrm{Tr}(\Gamma_{PL}^k)$. By the remarks after Lemma 1, larger k implies a better asymptotic l_2 bound for learning θ since $\mathbb{E}_{X_{1:n}, K_{1:n}}\left[\|\hat{\theta}_{PL}^k - \theta\|_2^2\right] \to \frac{\mathrm{Tr}(\Gamma_{PL}^k)}{n}$.*

The main lemma needed for Theorem 1 is that the two matrices in the asymptotic covariance of MPLE, $\nabla_\theta^2 L_{PL}(\theta^*)$ and $\mathrm{Cov}_{X\sim p_\mathcal{X}, K\sim p_\mathcal{K}}(-\nabla_\theta \log p_\theta(X_K|X_{-K}))_{|\theta=\theta^*}$ are actually equal. For MLE (namely, when $k = N$) this is well-known and called the *information matrix equality*. Proofs of Lemma 2 and Theorem 1 are in Appendix A and Appendix B. We empirically verify Theorem 1 in Section 3.1.

Lemma 2 (Generalized information matrix equality). *Under Assumption 1 and Assumption 2, the weighted pseudolikelihood loss (Definition 2) verifies: $\nabla_\theta^2 L_{PL}(\theta^*) = \mathrm{Cov}_{X\sim p_\mathcal{X}, K\sim p_\mathcal{K}}(-\nabla_\theta \log p_\theta(X_K|X_{-K}))_{|\theta=\theta^*}$.*

2.2.2. Statistical efficiency bounds via mixing time bounds

We could in general conceive of masking strategies where certain subsets of variables get masked with different probabilities. For instance, in language, nearby words will tend to be more correlated; grammatical constraints will dictate the parts-of-speech that can occur in different positions. We would then like to have theoretical guidance on what choices of masking distributions are better. Remarkably, it turns out that we can relate the statistical efficiency — in the sense of $\mathbb{E}\|\hat{\theta} - \theta^*\|^2$ for the resulting estimator $\hat{\theta}$ — and the mixing time of an appropriately chosen Markov Chain. In fact, this is the Markov Chain that would be typically chosen at inference time. Towards making this formal, we will need several preliminary concepts and results for Markov chains. Recall, a Markov chain on a state space Ω is described by a (row-stochastic) transition matrix P. Moreover, we can assign a natural bilinear form called the Dirichlet form:

Definition 4 (Dirichlet form). *Let M be an ergodic, reversible Markov chain with transition matrix P on state space Ω. Let μ be its unique stationary distribution. $\forall f, g : \Omega \to \mathbb{R}$ the associated Dirichlet form is defined as $\mathcal{E}_P(f, g) := \langle f, (I - P)g\rangle_\mu = \frac{1}{2}\Sigma_{x,y\in\Omega}\mu(x)P(x,y)(f(x) - f(y))(g(x) - g(y))$*

Mixing time of the Markov chain can be bounded in the χ^2 sense by the gap between the 1st and 2nd eigenvalue of the Laplacian matrix $I - P$, expressed as Poincaré inequality:

Definition 5 (Poincaré inequality). *We say that a Markov chain satisfies a* Poincaré inequality *with constant C if for all $f : \Omega \to \mathbb{R}$, we have $\mathcal{E}_P(f, f) \geq \frac{1}{C}\mathrm{Var}_\mu(f)$.*

The Poincaré inequality implies exponential ergodicity of the Markov chain in χ^2-divergence, precisely $\chi^2(p_t, \mu) \leq e^{-2t/C}\chi^2(p_0, \mu)$, where μ is the stationary distribution of the chain and p_t is the distribution after running the Markov process for time t, starting at p_0. We will be particularly interested in several generalizations of Gibbs sampling:

Definition 6 (Weighted block dynamics). *Let $\mathcal{K} := \{K_1, \ldots, K_{|\mathcal{K}|}\}$ be a collection of sets (or blocks) such that $\bigcup_i K_i = [N]$. A block dynamics with blocks $\mathcal{K}$ is a Markov chain that picks a block K in each step according to some distribution $p_\mathcal{K} : \mathcal{K} \to \mathbb{R}^+$*[7] *and then updates the coordinates in K according to the conditional distribution $p_\mathcal{X}(X_K|X_{-K})$.*

The stationary distribution for the above Markov Chain is $p_\mathcal{X}$. Caputo & Parisi (2021) also derived the Dirichlet form (Definition 4) corresponding to this Markov chain:

$$\mathcal{E}(f, g) := \mathbb{E}_{K\sim p_\mathcal{K}}\mathbb{E}_{X_{-K}}\left[\mathrm{Cov}_{X_K|X_{-K}}(f, g)\right].$$

The crucial result we show is that the statistical efficiency of the weighted MPLE (Definition 2) as captured by the asymptotic variance can be related to the Poincaré constant of the corresponding weighted block dynamics (Definition 6). Proof of Theorem 2 is in Appendix C.4.

Theorem 2 (Asymptotic variance under a Poincaré Inequality). *Suppose the distribution p_{θ^*} satisfies a Poincaré inequality with constant C with respect to the weighted block*

[6] The notation $A \preceq B$ means $B - A$ is positive semidefinite.

[7] This is analogous to the training objective setting in Definition 2.

dynamics. Then under Assumption 1 and Assumption 2 the asymptotic variance of the weighted MPLE can be bounded as: $\Gamma_{PL} \preceq C\mathcal{I}^{-1}$ where $\mathcal{I}$ is the Fisher Information matrix (Definition 1).

2.2.3. Adaptive masking: masked positions depend on the sequence

The machinery we developed in Section 2.2.2 is in fact substantially more general — it applies to even "adaptive" masking losses in which the conditional distribution of the mask can depend on the current X (that is, for each X, there is a different conditional distribution $p_{\mathcal{K}}(K|X)$ which can be manually designed and is known to the model during training).

Definition 7 (Adaptively weighted pseudolikelihood)**.** *Given n i.i.d samples of sequences and masks: $\{(X^{(i)}, K^{(i)})\,|X^{(i)} \sim p_{\mathcal{X}}, K^{(i)} \sim p_{\mathcal{K}}(\cdot|X^{(i)})\}_{i\in[n]}$, the* weighted maximum pseudolikelihood estimator (MPLE) *is $\hat{\theta}_{PL} := \arg\min_\theta \sum_{i=1}^n -\log p_\theta(X_K|X_{-K}, K)$, where*

$$p_\theta(X_K|X_{-K}, K) := \frac{p_\theta(X)p_{\mathcal{K}}(K|X)}{\sum_{X'_K} p_\theta\left(X'_K, X_{-K}\right) p_{\mathcal{K}}(K|\left(X'_K, X_{-K}\right))} \tag{1}$$

The population loss is :

$$L_{PL}(\theta) := \mathbb{E}_{X\sim p_{\mathcal{X}}}\mathbb{E}_{K\sim p_{\mathcal{K}}(\cdot|X)}\left[-\log p_\theta(X_K|X_{-K}, K)\right].$$

Remark 3. *Note, the distribution $p_{\mathcal{K}}(\cdot|X)$ doesn't depend on θ, so $K^{(i)}$ can be generated readily by drawing samples from this distribution. Note also, the term $p_\theta(X_K|X_{-K}, K)$ is expressible in terms of the joint distribution $p_\theta(X)$ and $p_{\mathcal{K}}(\cdot|X)$ and the expression in (1) can be interpreted as a conditional distribution in the joint distribution $p_{\theta,\mathcal{K}}(X, K) := p_\theta(X)p_{\mathcal{K}}(K|X)$. Finally, note that conditioning on the set K is subtle, but important — see Lemma 3 in Appendix C.1.*

We can analogously generalize the sampling process to the following Markov chain in which K is sampled dependent on X:

Definition 8 (Adaptive weighted block dynamics)**.** *Let $\mathcal{K} := \{K_1, \dots, K_{|\mathcal{K}|}\}$ be a collection of sets (or blocks) such that $\bigcup_i K_i = [N]$. A block dynamics with blocks $\mathcal{K}$ is a Markov chain that picks a block K in each step according to some distribution[8] $p_{\mathcal{K}}(\cdot \mid X)$, and then updates the coordinates in K according to the conditional distribution $p_{\mathcal{X}}(X_K|X_{-K}, K)$.*[9]

If we understand the domain of this Markov chain to be $\{(X, K) \mid X \in \mathcal{X}, K \in \mathcal{K}\}$, its stationary distribution is

$$p_{\mathcal{X},\mathcal{K}}(X, K) := p_{\mathcal{X}}(X)p_{\mathcal{K}}(K \mid X).$$

The Dirichlet form can also be explicitly written down (note, f and g are functions of both X and K):

Proposition 1 (Dirichlet form for adaptive weighted block dynamics)**.** *The Dirichlet form corresponding to the weighted block dynamics (Definition 8) is:*

$$\mathcal{E}(f, g) = \mathbb{E}_{(X_{-K},K)\sim p_{\mathcal{X},\mathcal{K}}}\left[Cov_{X_K|(X_{-K},K)}(f, g)\right]$$

The proof of Proposition 1 is in Appendix C.3.

Analogous to Theorem 2, we again show that the statistical efficiency of the adaptively-weighted MPLE (Definition 7), captured by the asymptotic variance, can be related to the Poincaré constant of the corresponding adaptively-weighted Block dynamics (Definition 8):

Theorem 3 (Asymptotic variance of adaptively-weighted MPLE under a Poincaré Inequality, generalization of Theorem 2)**.** *Suppose the distribution p_{θ^*} satisfies a Poincaré inequality with constant C with respect to the adaptively-weighted block dynamics. Then under Assumption 1 and Assumption 2 where $p_\theta(x_K|x_{-K})$ is replaced by $p_\theta(x_K|x_{-K}, K)$, the asymptotic variance of the adaptively-weighted MPLE can be bounded as: $\Gamma_{PL} \preceq C\mathcal{I}^{-1}$ where $\mathcal{I}$ is the Fisher Information matrix (Definition 1).*

The proof of Theorem 3 is in Appendix C.4.

2.3. Finite sample bounds and distributional distance

The framework in Section 2.2 was asymptotic in nature, and used parameter closeness as a notion of "quality" of the estimator. In this section, we remove both requirements, at the cost of the bounds depending on a notion of "complexity" of the parametric class we are fitting. It turns out that we can prove very similar results, with the notion of "mixing" — as captured by the Poincaré constant — being replaced by a different constant called the "approximate tensorization constant". These results mirror results in Section 5.1 in (Koehler et al., 2023), who focus on 1-MPLE and use a differrent notion of "complexity" based on Rademacher complexity. We first introduce several preliminary concepts.

Definition 9 (Block approximate tensorization of entropy (Caputo & Parisi, 2021))**.** *Under fixed distribution $p_{\mathcal{K}}(\cdot \mid X)$ over binary masks $\mathcal{K}$ conditioned on X, we say the distribution $q_{\mathcal{X}}$ over $\mathcal{X}$ satisfies* block-generalized approximate tensorization of entropy *with constant $\bar{C}_{AT}(q_{\mathcal{X}})$ if for any distribution $r_{\mathcal{X}}$ over $\mathcal{X}$,*

$$D_{\mathrm{KL}}(r_{\mathcal{X}}, q_{\mathcal{X}}) \le \bar{C}_{AT}(q_{\mathcal{X}}) \cdot \mathbb{E}_{X\sim r_{\mathcal{X}}}[\mathbb{E}_{K\sim p_{\mathcal{K}}(\cdot|X)}[\\ D_{\mathrm{KL}}\left(r_{\mathcal{X}}(\cdot \mid X_{-K}, K), q_{\mathcal{X}}(\cdot \mid X_{-K}, K)\right)]]$$

This inequality is closely related to the mixing time of weighted-block dynamics (Definition 6). Namely, the inequality is weaker than the standard discrete version of the

[8] This is analogous to the training objective setting in Definition 7.

[9] Defined analogously as in Definition 1.

log-Sobolev inequality (Diaconis & Saloff-Coste, 1996) and stronger than the Modified Log-Sobolev Inequality (Bobkov & Tetali, 2006), which implies exponential ergodicity of the weighted block-dynamics in KL divergence[10], that is: $\mathrm{KL}(p_t, q) \leq e^{-2t/C_{AT}(q)} \mathrm{KL}(p_0, q)$.

To bound the distance between the population and empirical losses, as well as relate it to the distance between the estimated parameters and the ground truth, we first introduce a few useful pieces of notation.

Notation: For each sample $X^{(i)}, i \in [n]$, we assume we observe m masks $\{K_j^{(i)} \mid j \in [m]\}$ sampled iid from $p_{\mathcal{K}}(\cdot \mid X^{(i)})$. We denote the corresponding empirical loss by $\hat{L}_{PL}(\theta) := \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m - \log p_\theta(X^{(i)}_{K_j^{(i)}} | X^{(i)}_{-K_j^{(i)}}, K_j^{(i)})$. Furthermore, we will denote by $\tilde{p}_{\mathcal{X}}$ the uniform distribution over $\{X^{(i)}, i \in [n]\}$, and denote

$$\tilde{L}_{PL}(\theta) := \mathbb{E}_{X \sim \tilde{p}_{\mathcal{X}}, K \sim p_{\mathcal{K}}(\cdot | X)} \left[- \log p_\theta(X_K | X_{-K}, K) \right] \tag{2}$$

This is an intermediate quantity: it averages in the population sense over the masks, but it assumes a finite number of samples from $p_{\mathcal{X}}$. It will be a useful intermediate quantity for several concentration bounds.

We will also need a few mild assumptions on the distribution we are fitting. First, we assume that the learned conditional probabilities are uniformly lower-bounded by a constant:

Assumption 3 (Support margin)**.** *Exists constant* $\beta \in (0, 1)$ *s.t.* $\forall X \in \mathcal{X}, \forall K \subset [N]$, $\forall \theta \in \Theta$, *if* $p_{\mathcal{X}}(X_K | X_{-K}, K) > 0$, *then* $p_\theta(X_K | X_{-K}, K) \geq \beta$.

We also assume that the log-probabilities (and hence the losses $\hat{L}_{PL}$ and $\tilde{L}_{PL}$) are Lipschitz with respect to θ, and Θ has a finite covering bound. Namely:

Assumption 4 (Covering bound and Lipschitzness)**.** $\forall \epsilon > 0$, *there exists a finite partition* $\mathit{Par}_\epsilon(\Theta) = \{\Theta_1, \cdots, \Theta_{|\mathit{Par}(\Theta)|}\}$ *of* Θ, *such that* $\forall i, \forall \theta_1, \theta_2 \in \Theta_i$, *and* $\forall (X, K) \in \mathcal{X} \times \mathcal{K}$*:*

$$|\log p_{\theta_1}(X_K | X_{-K}, K) - \log p_{\theta_2}(X_K | X_{-K}, K)| \leq \frac{\epsilon}{2}.$$

Moreover, $C_\epsilon(\Theta)$ *denote the smallest possible cardinality among such partitions* $\mathit{Par}_\epsilon(\Theta)$.

With this setup, we can prove the following finite-sample bound on the closeness of the learned distribution, provided the weighted pseudolikelihood loss (Definition 2) is small:

Theorem 4 (Generalization bound for learning the joint distribution)**.** *Let* $\hat{\theta} := \arg\min_\theta \hat{L}_{PL}(\theta)$. *Under Assumption 3 and Assumption 4,* $\forall \epsilon > 0$, $\forall \delta \in (0, 1)$, *with probability at least* $1 - \delta$ *we have* $D_{\mathrm{TV}}\left(p_{\hat{\theta}}, p_{\mathcal{X}}\right) < \sqrt{\frac{1}{2} \bar{C}_{AT}(p_{\hat{\theta}}) \left(\hat{L}_{PL}(\hat{\theta}) + B \cdot \ln \frac{1}{\beta} + \epsilon \right)} + C$ *where* $B = \sqrt{\frac{2^{3N} |\Omega|^N C_\epsilon(\Theta)}{m \cdot \delta}} + \sqrt{\frac{\ln \frac{8 C_\epsilon(\Theta)}{\delta}}{2n}}$, *and* $C = \sqrt{\frac{|\Omega|^{3N}}{8 \delta n}}$.

Proof of Theorem 4 is in Appendix D. We can compare the statement to Theorem 3: (1) On the LHS, rather than parameter distance, we have total variation distance between the learned distribution and p. (2) On the RHS, rather than a Poincaré inequality, we have the $\bar{C}_{AT}(p_{\hat{\theta}})$ constant. (3) On the RHS, instead of the Fisher information matrix, we have quantities capturing the generalization error, through a notion of complexity of the class ($C_\epsilon(\Theta)$).

2.4. Inference-time limitations due to parallelism

In this section, we focus on limitations in the representational and computational efficiency that arise when using a parallel decoding approach to implement a step of the inference-time sampling algorithm. Precisely, at inference-time, using weighted block-dynamics with bigger blocks enables larger sets of coordinates to be re-randomized, facilitating a faster mixing time. A canonical example of this are k-Gibbs samplers:

Definition 10 (k-Gibbs sampler)**.** *The* k*-Gibbs sampler is a special case of the block dynamics (Definition 6) when* $\mathcal{K} := \{K \subseteq [N] \mid |K| = k\}$, *and* $p_{\mathcal{K}} = \mathit{Unif}(\mathcal{K})$.

$$X_K^{(t+1)} \sim p\left(\cdot \mid X_{-K}^{(t)}\right), \; X_j^{(t+1)} = X_j^{(t)} \, \forall j \notin K \tag{3}$$

Samplers with larger k are well-known to mix faster (e.g. (Lee, 2023) shows the Poincaré inequality improves by a factor of at least k). However, taking a step of this Markov Chain requires being able to re-randomize the k coordinates according to their conditional distribution — which is intuitively harder for larger k if we are trying to re-randomize the coordinates in parallel.

In Section 2.4.1, we show that the canonical GMLM-type parallel decoding language models can only implement Markov chains whose transitions are product distributions over the sequence positions. [11]

We also consider a natural Markov Chain whose transitions are product distribution, and show it can be substantially slower to reach the modes of the distribution compared to k-Gibbs for a large k, i.e. a Markov Chain with transitions that are far from a product distribution. Precisely, we consider:

Definition 11 (Independent parallel sampler)**.** *The* independent parallel sampler *performs* coordinate-wise *updates for*

[10] This in turns, also implies a Poincaré inequality and exponential ergodicity in χ^2 divergence.

[11] Remark 6 in Appendix I connects our results to technical details of model architectures in prior works.

all i in parallel [12]*, namely:*

$$\forall i \in [N],\ X_i^{(t+1)} \sim p\left(\cdot \mid X_{-\{i\}}^{(t)}\right) \tag{4}$$

In Section 2.4.2, we show that even if we do not care about mixing—just reaching the modes of the distribution—the independent parallel sampler can be much slower compared to k-Gibbs, for a large k.

2.4.1. Which Markov Chains are implementable via parallel decoding?

In this section, we characterize the power and restrictions of Transformers at inference time when they are restricted to decoding the tokens of the sequence in parallel. The inference algorithms for a model that has access to approximate conditional probabilities typically look like (potentially multiple) steps of block dynamics (Definition 6). We focus on understanding what kinds of transitions are implementable with a standard Transformer architecture.

Note that while there are well-known prior results about the expressive power of Transformers as sequence-to-sequence modelers (Yun et al., 2020), representing steps of a Markov Chain with parallel decoding is more subtle, due to the fact that a step of a Markov Chain requires randomness. First, we state a result characterizing the power of Transformers to approximate "deterministic" Markov Chains: that is, Markov Chains whose transition distributions are delta functions. Unsurprisingly, standard universal approximation results apply to this case. Namely:

Proposition 2 (informal)**.** *Transformers (with sufficient depth and width) can implement any number of transitions of any deterministic Markov Chain over sequences in Ω^N.*

On the other hand, Transformers using parallel decoding cannot implement general Markov chains over Ω^N. In fact, they can only implement Markov Chains for which the transition probabilities are product distributions:

Proposition 3 (informal)**.** *The class of Markov chains over sequences in Ω^N implementable by (sufficiently wide and deep) Transformers is those whose next-state transition probability distributions are product distributions over the positions, conditioned on the current state.*

Background information on the Transformer architecture, as well as proofs of Proposition 2 and Proposition 3 are relegated to Appendix I. Note that this does *not* mean one can only simulate Markov Chains whose *stationary* distribution is a product distribution. In fact, the standard 1-Gibbs sampler, by virtue of the fact that it only updates one coordinate at a time, encodes a product distribution for each transition. On the other hand, under fairly mild conditions on a joint p, the 1-Gibbs sampler corresponding to p is ergodic and has p as a stationary distribution. On the other hand, a step of a k-Gibbs sampler for $k > 1$ is in general not a product distribution, and will not be implementable by a Transformer with parallel decoding.

2.4.2. Markov Chains with dependent transitions can be (much) faster

In this section, we show that the k-Gibbs (Definition 10)—the prototypical example of a Markov Chain with dependent transitions—can reach modes of the distribution much faster than the independent parallel sampler (Definition 11)—the prototypical example of a Markov Chain with independent transitions. Intuitively, in cases where there is a strong dependence between subsets of variables, jointly updating them will bring us much faster to their modes.

The toy probabilistic family in which we will illustrate this phenomenon is *Ising models*. Specifically, we consider an undirected graphical model G that can be represented by the union of a clique C_G (in which $|C_G| \geq 2$, and the dependency among variables is strong) and a set of $N - |C_G|$ independent vertices. More formally, we consider: $p_G : \{\pm 1\}^N \to \mathbb{R}^+$,

$$p_G(\mathbf{x}) = \frac{1}{Z_G} \exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j\in C_G} J\mathbf{x}_i\mathbf{x}_j\right) \tag{5}$$

in which Z_G is the partition function, and $\boldsymbol{h}_i \in \mathbb{R}$ s.t. $\sum_{i\in C_G} \boldsymbol{h}_i > 0$ and $J > 0$ are scalar constants. This is a *ferromagnetic* Ising model (i.e. the pairwise interactions prefer the variables to have the same value), and when $J \gg \|\boldsymbol{h}\|_1$, the two "modes" of the distribution p_G are such that all variables have the same value:

$$\mathcal{R}_1 \coloneqq \{\boldsymbol{X} \in \{-1,1\}^N | X_i = 1\, \forall i \in C_G\} \tag{6}$$

$$\mathcal{R}_{-1} \coloneqq \{\boldsymbol{X} \in \{-1,1\}^N | X_i = -1\, \forall i \in C_G\} \tag{7}$$

The above distribution can be seen as a simple prototype of language tasks in which grammatical rules or semantic constraints create "clusters" of positions in which changing isolated words leads to very unlikely sentences. Next, we formalize the concentration around the "modes":

Assumption 5 (Strongly ferromagnetic Ising model)**.** *There exist constants $h_G > 0, J_0 > 0$ such that $h_G \coloneqq \sum_{i\in C_G} \boldsymbol{h}_i > \sum_{i\notin C_G} |\boldsymbol{h}_i|$, $J - \|\boldsymbol{h}\|_1 \geq J_0$.*

Informally, under Assumption 5, sequences in $\mathcal{R}_1$ are much more likely under the groundtruth distribution than those in $\mathcal{R}_{-1}$, which are further much more likely than all other sequences. The formal statement and proof are in Appendix E. As a result, we can think of sampling from $\mathcal{R}_1$ as analogous to sampling a high-quality sentence, and moreover, not

[12] The stationary distribution of this chain is unclear: in fact, it is not even clear the chain is ergodic.

reaching $\mathcal{R}_1$ implies the Markov chain sampling process has not mixed to the groundtruth distribution yet. In the analogy to language tasks, in tasks like machine translation, for each source sentence, sampling one high-quality target sentence is potentially good enough. In some other tasks like creative writing, producing well-calibrated samples might be desirable—so mixing would be needed.

We show that running k**-Gibbs sampler** requires a small number of steps to reach $\mathcal{R}_1$. This implies that if a model can efficiently approximate one step of k**-Gibbs sampler**, then it is fast to sample a high-probability sequence by iteratively applying the model. Proof is in Appendix F.

Proposition 4 (k-Gibbs sampler can reach the mode fast)**.** *On Ising model G in Equation (5) under Assumption 5, with any initial $\boldsymbol{X}^{(0)}$, $\forall \delta \in (0,1)$, with probability at least $1-\delta$, after $T := \left\lceil \log_{c_{\mathcal{R}_1}} \delta \right\rceil$ steps of k**-Gibbs sampler** (Definition 10) with $k \geq |C_G|$, we have $\{\boldsymbol{X}^{(t)} | t \in [T]\} \cap \mathcal{R}_1 \neq \emptyset$ in which the constant $c_{\mathcal{R}_1} \in (0,1)$, $c_{\mathcal{R}_1} := 1 - \frac{\binom{N-|C_G|}{k-|C_G|}}{\binom{N}{k}} \frac{e^{2(J_0+h_G)}}{e^{2(J_0+h_G)}+e^{2J_0}+2^{|C_G|}-2}$*

By contrast, we show that for nontrivial probability over the randomness in the initial sequence, running **independent parallel** requires a large number of steps to reach the largest mode $\mathcal{R}_1$ of the distribution. This implies that the sampling process may not reach a high-probability sequence in less than exponentially large number of iterations.

Proposition 5 (Independent parallel sampling stuck in bad samples)**.** *On Ising model G in Equation (5) under Assumption 5, if the initial $\boldsymbol{X}^{(0)}$ is such that $\sum_{i \in C_G} \boldsymbol{X}_i^{(0)} \leq -2$, $\forall \delta \in (0,1)$, with probability at least $1-\delta$, after $T := \left\lfloor \frac{\delta}{2} \exp\left(c_{stuck}\right) \right\rfloor$ steps of **independent parallel** (Definition 11), we have $\forall t \in [T], \sum_{i \in C_G} \boldsymbol{X}_i^{(t)} \leq -2$, in which $c_{stuck} := \frac{2\left(-1+\frac{1-\exp(-2J_0)}{\exp(-2J_0)+1}\frac{|C_G|}{2}\right)^2}{|C_G|}$*

The proof is in Appendix G. Combining Proposition 4 and Proposition 5 leads to a separation result between k**-Gibbs sampler** and **independent parallel**, in particular when the clique size in G is large and dependency is strong within the clique: with high probability, while the former reaches $\mathcal{R}_1$ in 1 step, the latter cannot do so in arbitrarily large number of steps. Proof is in Appendix H. We empirically verify our theory in Appendix L.7.

3. Experiments

3.1. Synthetic experiments on Ising model

To empirically validate Theorem 1 (Masking more is (statistically) better), we run controlled experiments with synthetic data generated by a ground-truth Ising model. We train an Ising model using k-pseudolikelihood, and measure the squared error of parameter estimation. The results verify that with the same training data size, larger k leads to lower error. We plot the results in Figure 1, with several related experiments, in Appendix L.6. [13]

3.2. Parallel Decoding by Iterative Refinement (PaDIR)

We consider an encoder-decoder architecture, in which the decoder is modified to be *non-autoregressive*: instead of iteratively predicting the next token, each of our decoder forward pass predicts an update to *all* target positions *in parallel*. The encoder extracts features from the source sequence, and based on these features, each decoder forward pass refines its current hypothesis of the target sequence. The initial decoder hypothesis is a purely random sequence, and more decoder forward passes correspond to more steps of refinement. Note that we are *not* the first in the literature to propose this language modeling paradigm. [14] Our focus in this paper is to provide theoretical and empirical analyses to characterize its potentials, limitations and document useful training practices. Details of inference and training frameworks are in Appendix L.1.

3.3. Evaluation

We train models on machine translation datasets, provide practical recommendations based on our empirical observations, and discuss their connections to our theory. Details of training recipe are provided in Appendix L.2.

Benchmarking PaDIR models and AR models reach similar BLEU (Papineni et al., 2002) and BLEURT (Sellam et al., 2020; Pu et al., 2021) scores. Quantitative experimental results and common baselines are shown in Table 1, Table 2, and Table 3 in Appendix L.4. We discuss several considerations for evaluation metrics in Appendix L.3. While bridging the gap between autoregressive and non-autoregressive model has so far focused on achieving parity in terms of BLEU scores, we believe this is insufficient. Since BLEU relies on n-gram overlaps between groundtruths and model predictions, it does not capture readability very well. Yet readability is paramount for most practical applications, and it is indisputably something that current autoregressive LMs excel at. To provide additional perspectives, we introduce a word-level stutter metric, computing how often consecutive words are repeated in the model output but not in the reference. For all datasets, we found that word-level stutter is 2 or more times more frequent for non-autoregressive models.

Speed The average target length in all datasets ranges between 28 and 33 tokens, including the `EOS` token. As such

[13] Related simulations were also reported in Huang & Ogata (2002).

[14] Representative prior works: Ghazvininejad et al. (2020); Savinov et al. (2022), inter alia. See Section 4.

a non-autogressive model using 4 decoding steps does 7 to 8 times fewer decoder passes. In practice we see an end-to-end speedup greater than >2x for the median and >5x for the 99th percentile latency on TPU v3 (with 4 decoding steps and batch size 1). The gap between expected and observed speedup is due to fixed costs (input tokenization, encoding, etc.) as well as a better optimization of AR decoding (e.g. through caching of intermediate results). For longer sequences, the constant number of decoding passes in GMLM is advantageous. For completeness, it is worth noting that the number of decoder passes necessary to achieve good quality (and thus model speed) is application dependent, with some tasks like non-autoregressive text in-painting remaining slower than their autoregressive counterparts, as shown in Savinov et al. (2022).

3.4. Connecting to theory: quantifying dependency via attention scores

Our theory suggests that stronger dependency between target positions leads to worse generalization guarantee and sampling efficiency. However, it is unclear how to measure such dependency for Transformer-based language models trained on natural language data. In this section, we empirically investigate: *how to predict what target positions have strong dependency which may be challenging for Transformers?* We test the following two hypotheses: (1) Strongly dependent target positions have larger **decoder self-attention** between each other. (2) Strongly dependent target positions have similar **cross-attention** distribution to source tokens.

For a pair of target positions, to measure how well their dependency is modeled in the generated output, we focus on adjacent repetitive tokens, a.k.a. *stutter*. Stuttering is a common error mode among parallel decoding models, and we use it as one reasonable proxy for measuring failures in modeling target-side dependency. We show:

- Hypothesis 1 is unlikely to hold: even on average, stuttering positions do not have larger **decoder self-attention** between each other, compared with non-stuttering adjacent positions. [15]
- By contrary, Hypothesis 2 is potentially promising: with various of distribution distance measures, stuttering positions in the generated output have more similar **cross-attention** distributions to source tokens, compared with non-stuttering adjacent positions.

Details are in Table 4 and Table 5 in Appendix L.5.

4. Related works

Our theory is inspired by recent progress in sampling: the connections between pseudolikelihood and approximate tensorization of entropy are discussed in Marton (2013; 2015); Caputo et al. (2015); Caputo & Parisi (2021); Koehler et al. (2023). Benefits of k-Gibbs sampler are discussed in Lee (2023). Our experiments follow the framework that trains generative masked language models and generates samples using parallel decoding by iterative refinement: (Lee et al., 2018; Ghazvininejad et al., 2019; 2020; Kasai et al., 2020; Savinov et al., 2022), which tend to be at least twice faster than autoregressive approaches with a small drop in quality for tasks like machine translation. The inference process, which converts complete noise to full samples, might resemble diffusion models (Hoogeboom et al., 2021; Austin et al., 2021; Li et al., 2022; Gong et al., 2023; Zheng et al., 2023; Lou et al., 2023), but a key conceptual difference is that diffusion models are trained to revert a small amount of noise at each step, whereas the family of models that we study in this work are more similar to *masked autoencoders*: the training objective encourages reconstructing the whole target sequence in each step of decoding. We discuss additional related works in Appendix M.

5. Conclusion

We introduce a new theoretical framework for understanding the power and limitations of generative masked language models (GMLM). In particular, our theory offers some guidance on the design spaces of learning and inference algorithms, through the perspectives of asymptotic sample complexity for parameter learning, finite-sample generalization bound for distribution learning, and the efficiency of Gibbs-like sampling algorithms. Empirically we adapt T5 to parallel decoding by iterative refinement (an non-autoregressive GMLM-based language generation strategy which showed strong speed-quality trade-off in the literature for tasks like machine translation). We recommend some rules of thumb for key design choices, and discuss the connection between the the empirical findings and our theory. For future works, we hope the theoretical framework and empirical observations can inspire new training objectives, inference algorithms, and neural network architectures better-suited for parallel decoding.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

[15] Since all stuttering positions are by definition adjacent, we think a fair comparison should only consider adjacent positions for non-stuttering position pairs.

References

Anari, N., Huang, Y., Liu, T., Vuong, T.-D., Xu, B., and Yu, K. Parallel discrete sampling via continuous walks. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pp. 103–116, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. doi: 10.1145/3564246.3585207. URL https://doi.org/10.1145/3564246.3585207.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

Bachmann, G. and Nagarajan, V. The pitfalls of next-token prediction, 2024.

Besag, J. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society Series D: The Statistician*, 24(3):179–195, 1975.

Bobkov, S. G. and Tetali, P. Modified logarithmic sobolev inequalities in discrete settings. *Journal of Theoretical Probability*, 19(2):289–336, 2006.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. s. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W14/W14-3302.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. Findings of the 2017 conference on machine translation (WMT17). In Bojar, O., Buck, C., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., and Kreutzer, J. (eds.), *Proceedings of the Second Conference on Machine Translation*, pp. 169–214, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4717. URL https://aclanthology.org/W17-4717.

Bojar, O. r., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pp. 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2301.

Bosc, T. and Vincent, P. Do sequence-to-sequence VAEs learn global features of sentences? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4296–4318, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.350. URL https://aclanthology.org/2020.emnlp-main.350.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://aclanthology.org/K16-1002.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J. D., Chen, D., and Dao, T. Medusa: Simple llm inference acceleration framework with multiple decoding heads, 2024.

Caputo, P. and Parisi, D. Block factorization of the relative entropy via spatial mixing. *Communications in Mathematical Physics*, 388(2):793–818, 2021.

Caputo, P., Menz, G., and Tetali, P. Approximate tensorization of entropy at high temperature. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 24, pp. 691–716, 2015.

Chan, W., Saharia, C., Hinton, G., Norouzi, M., and Jaitly, N. Imputer: Sequence modelling via imputation and dynamic programming. In *International Conference on Machine Learning*, pp. 1403–1413. PMLR, 2020.

Che, T., Li, Y., Zhang, R., Hjelm, R. D., Li, W., Song, Y., and Bengio, Y. Maximum-likelihood augmented discrete generative adversarial networks, 2017.

Deng, Y., Bakhtin, A., Ott, M., Szlam, A., and Ranzato, M. Residual energy-based models for text generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1l4SgHKDH.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Diaconis, P. and Saloff-Coste, L. Logarithmic sobolev inequalities for finite markov chains. *The Annals of Applied Probability*, 6(3):695–750, 1996.

Edelman, B. L., Goel, S., Kakade, S., and Zhang, C. Inductive biases and variable creation in self-attention mechanisms. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5793–5831. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/edelman22a.html.

Ghazvininejad, M., Levy, O., Liu, Y., and Zettlemoyer, L. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633.

Ghazvininejad, M., Levy, O., and Zettlemoyer, L. Semi-autoregressive training improves mask-predict decoding, 2020.

Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jQj-_rLVXsj.

Goyal, K., Dyer, C., and Berg-Kirkpatrick, T. Exposing the implicit energy networks behind masked language models via metropolis–hastings. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=6PvWo1kEvlT.

Gu, J. and Kong, X. Fully non-autoregressive neural machine translation: Tricks of the trade. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 120–133, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.11. URL https://aclanthology.org/2021.findings-acl.11.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1l8BtlCb.

Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., and Wang, J. Long text generation via adversarial training with leaked information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Guo, J., Xu, L., and Chen, E. Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 376–385, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.36. URL https://aclanthology.org/2020.acl-main.36.

Hájek, J. Local asymptotic minimax and admissibility in estimation. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 175–194, 1972.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=6nbpPqUCIi7.

Huang, F. and Ogata, Y. Generalized pseudo-likelihood estimates for markov random fields on lattice. *Annals of the Institute of Statistical Mathematics*, 54:1–18, 2002.

Jelassi, S., Sander, M. E., and Li, Y. Vision transformers provably learn spatial structure. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=eMW9AkXaREI.

Kasai, J., Cross, J., Ghazvininejad, M., and Gu, J. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Kasai, J., Pappas, N., Peng, H., Cross, J., and Smith, N. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KpfasTaLUpq.

Kim, Y. and Rush, A. M. Sequence-level knowledge distillation. In Su, J., Duh, K., and Carreras, X. (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139.

Koehler, F., Heckett, A., and Risteski, A. Statistical efficiency of score matching: The view from isoperimetry. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TD7AnQjNzR6.

Kong, X., Zhang, Z., and Hovy, E. Incorporating a local translation mechanism into non-autoregressive translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1067–1073, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.79. URL https://aclanthology.org/2020.emnlp-main.79.

Kreutzer, J., Foster, G., and Cherry, C. Inference strategies for machine translation with conditional masking. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5774–5782, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.465. URL https://aclanthology.org/2020.emnlp-main.465.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

Lee, H. Parallelising glauber dynamics, 2023.

Lee, J., Mansimov, E., and Cho, K. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1173–1182, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1149. URL https://aclanthology.org/D18-1149.

Lee, J., Shu, R., and Cho, K. Iterative refinement in the continuous space for non-autoregressive neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1006–1015, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.73. URL https://aclanthology.org/2020.emnlp-main.73.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. Diffusion-LM improves controllable text generation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=3s9IrEsjLyk.

Li, Y. and Risteski, A. The limitations of limited context for constituency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2675–2687, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.208. URL https://aclanthology.org/2021.acl-long.208.

Li, Y., Li, Y., and Risteski, A. How do transformers learn topic structure: Towards a mechanistic understanding. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19689–19729. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/li23p.html.

Lin, C.-C., Jaech, A., Li, X., Gormley, M. R., and Eisner, J. 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, pp. 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.

Lin, K., Li, D., He, X., Zhang, Z., and Sun, M.-T. Adversarial ranking for language generation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3158–3168, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Liu, B., Hsu, D., Ravikumar, P. K., and Risteski, A. Masked prediction: A parameter identifiability view. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Hbvlb4D1aFC.

Liu, B., Ash, J. T., Goel, S., Krishnamurthy, A., and Zhang, C. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=De4FYqjFueZ.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion language modeling by estimating the ratios of the data distribution, 2023.

Lu, H., Mao, Y., and Nayak, A. On the dynamics of training attention models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1OCTOShAmqB.

Ma, X., Zhou, C., Li, X., Neubig, G., and Hovy, E. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4282–4292, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1437. URL https://aclanthology.org/D19-1437.

Marton, K. An inequality for relative entropy and logarithmic sobolev inequalities in euclidean spaces. *Journal of Functional Analysis*, 264(1):34–61, 2013.

Marton, K. Logarithmic sobolev inequalities in discrete product spaces: a proof by a transportation cost distance. *arXiv preprint arXiv:1507.02803*, 2015.

Meng, Y., Krishnan, J., Wang, S., Wang, Q., Mao, Y., Fang, H., Ghazvininejad, M., Han, J., and Zettlemoyer, L. Representation deficiency in masked language modeling. *arXiv preprint arXiv:2302.02060*, 2023.

Pabbaraju, C., Rohatgi, D., Sevekari, A., Lee, H., Moitra, A., and Risteski, A. Provable benefits of score matching. *arXiv preprint arXiv:2306.01993*, 2023.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D. (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Post, M. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

Pu, A., Chung, H. W., Parikh, A. P., Gehrmann, S., and Sellam, T. Learning compact metrics for MT. *CoRR*, abs/2110.06341, 2021. URL https://arxiv.org/abs/2110.06341.

Qian, L., Zhou, H., Bao, Y., Wang, M., Qiu, L., Zhang, W., Yu, Y., and Li, L. Glancing transformer for non-autoregressive neural machine translation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1993–2003, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.155. URL https://aclanthology.org/2021.acl-long.155.

Qin, L., Welleck, S., Khashabi, D., and Choi, Y. COLD decoding: Energy-based constrained text generation with langevin dynamics. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TiZYrQ-mPup.

Qin, Y. and Risteski, A. Fit like you sample: Sample-efficient generalized score matching from fast mixing diffusions, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Ravikumar, P., Wainwright, M. J., and Lafferty, J. D. High-dimensional ising model selection using l1-regularized logistic regression. *Ann. Statist. 38(3): 1287-1319 (June 2010). DOI: 10.1214/09-AOS691*, 2010.

Reid, M., Hellendoorn, V. J., and Neubig, G. Diffuser: Diffusion via edit-based reconstruction. In *The Eleventh International Conference on Learning Representations*, 2022.

Ren, Y., Liu, J., Tan, X., Zhao, Z., Zhao, S., and Liu, T.-Y. A study of non-autoregressive model for sequence generation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 149–159, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.15. URL https://aclanthology.org/2020.acl-main.15.

Roberts, A., Chung, H. W., Levskaya, A., Mishra, G., Bradbury, J., Andor, D., Narang, S., Lester, B., Gaffney, C., Mohiuddin, A., Hawthorne, C., Lewkowycz, A., Salcianu, A., van Zee, M., Austin, J., Goodman, S., Soares, L. B., Hu, H., Tsvyashchenko, S., Chowdhery, A., Bastings, J., Bulian, J., Garcia, X., Ni, J., Chen, A., Kenealy, K., Clark, J. H., Lee, S., Garrette, D., Lee-Thorp, J., Raffel, C., Shazeer, N., Ritter, M., Bosma, M., Passos, A., Maitin-Shepard, J., Fiedel, N., Omernick, M., Saeta, B., Sepassi, R., Spiridonov, A., Newlan, J., and Gesmundo, A. Scaling up models and data with t5x and seqio. *arXiv preprint arXiv:2203.17189*, 2022. URL https://arxiv.org/abs/2203.17189.

Saharia, C., Chan, W., Saxena, S., and Norouzi, M. Non-autoregressive machine translation with latent alignments. *arXiv preprint arXiv:2004.07437*, 2020.

Savinov, N., Chung, J., Binkowski, M., Elsen, E., and van den Oord, A. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=T0GpzBQ1Fg6.

Schmidt, R. M., Pires, T., Peitz, S., and Lööf, J. Non-autoregressive neural machine translation: A call for clarity, 2022.

Sellam, T., Das, D., and Parikh, A. P. BLEURT: learning robust metrics for text generation. *CoRR*, abs/2004.04696, 2020. URL https://arxiv.org/abs/2004.04696.

Shazeer, N. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235, 2018. URL http://arxiv.org/abs/1804.04235.

Stern, M., Chan, W., Kiros, J., and Uszkoreit, J. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*, pp. 5976–5985. PMLR, 2019.

Toda, A. A. Operator reverse monotonicity of the inverse. *The American Mathematical Monthly*, 118(1): 82–83, 2011.

Torroba Hennigen, L. and Kim, Y. Deriving language models from masked language models. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1149–1159, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.99. URL https://aclanthology.org/2023.acl-short.99.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Van der Vaart, A. W. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vuffray, M., Misra, S., Lokhov, A., and Chertkov, M. Interaction screening: Efficient and sample-optimal learning of ising models. *Advances in neural information processing systems*, 29, 2016.

Wang, A. and Cho, K. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In Bosselut, A., Celikyilmaz, A., Ghazvininejad, M., Iyer, S., Khandelwal, U., Rashkin, H., and Wolf, T. (eds.), *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2304. URL https://aclanthology.org/W19-2304.

Wei, C., Chen, Y., and Ma, T. Statistically meaningful approximation: a case study on approximating turing machines with transformers, 2021. URL https://arxiv.org/abs/2107.13163.

Wen, K., Li, Y., Liu, B., and Risteski, A. Transformers are uninterpretable with myopic methods: a case study with bounded dyck grammars. In *Thirty-seventh*

Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=OitmaxSAUu.

Yao, S., Peng, B., Papadimitriou, C., and Narasimhan, K. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3770–3785, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.292. URL https://aclanthology.org/2021.acl-long.292.

Young, T. and You, Y. On the inconsistencies of conditionals learned by masked language models. *arXiv preprint arXiv:2301.00068*, 2022.

Yu, L., Zhang, W., Wang, J., and Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 2852–2858. AAAI Press, 2017.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S., and Kumar, S. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Zhao, H., Panigrahi, A., Ge, R., and Arora, S. Do transformers parse while predicting the masked word? In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 16513–16542, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.1029. URL https://aclanthology.org/2023.emnlp-main.1029.

Zheng, L., Yuan, J., Yu, L., and Kong, L. A reparameterized discrete diffusion model for text generation, 2023.

Zhou, C., Gu, J., and Neubig, G. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BygFVAEKDH.

Ziegler, Z. and Rush, A. Latent normalizing flows for discrete sequences. In *International Conference on Machine Learning*, pp. 7673–7682. PMLR, 2019.

Supplementary Material

A. Proof of Lemma 2: Generalized information matrix equality

For convenience, we restate the generalized information matrix equality we are going to show:

Lemma 2 (Generalized information matrix equality)**.** *Under Assumption 1 and Assumption 2, the weighted pseudolikelihood loss (Definition 2) verifies:* $\nabla^2_\theta L_{PL}(\theta^*) = \mathrm{Cov}_{X\sim p_{\mathcal{X}}, K\sim p_{\mathcal{K}}}(-\nabla_\theta \log p_\theta(X_K|X_{-K}))_{|\theta=\theta^*}$.

Proof. All the expectations in the proof will be taken with respect to $(X, K) \sim p_{\mathcal{X}} \times p_{\mathcal{K}}$. To decrease the notational load, we will not explicitly write $p_{\mathcal{X}} \times p_{\mathcal{K}}$. The proof proceeds by first exchanging the order of expectations and derivatives, and using that to show the appropriate terms in the expression for $\nabla^2_\theta L_{PL}(\theta^*)$ vanish.

Step 1: Changing the order of expectations and derivatives

We will show that the following two equalities hold:

$$\nabla_\theta \mathbb{E}_{(X,K)} \log p_\theta(x_K|x_{-K}) = \mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K}) \tag{A.8}$$

$$\nabla^2_\theta \mathbb{E}_{(X,K)} \log p_\theta(x_K|x_{-K}) = \mathbb{E}_{(X,K)} \nabla^2_\theta \log p_\theta(x_K|x_{-K}) \tag{A.9}$$

Since Ω, $[N]$, and $K \subset [N]$ are both discrete finite, the conditions for the Dominated Convergence Theorem holds under Assumption 1: namely, there exists a function $f : \Theta \times \Omega \times \mathcal{K} \mapsto \mathbb{R}$ such that $\forall \theta \in \Theta$, $\mathbb{E}_{(X,K)} [f(\theta, X, K)] < \infty$, $\|\nabla_\theta \log p_\theta(x_K|x_{-K})\|_2 \le f(\theta, X, K)$, and $\|\nabla^2_\theta \log p_\theta(x_K|x_{-K})\|_F \le f(\theta, X, K)$.

Denoting by $\mathbf{e}_i$ the i-th standard basis vector, we have:

$$\frac{\partial}{\partial \theta_j} \mathbb{E}_{(X,K)} [\log p_\theta(x_K|x_{-K})] = \lim_{h\to 0} \frac{1}{h} \left(\mathbb{E}_{(X,K)} \left[\log p_{\theta+\mathbf{e}_j h}(x_K|x_{-K})\right] - \mathbb{E}_{(X,K)} [\log p_\theta(x_K|x_{-K})] \right) \tag{A.10}$$

$$= \lim_{h\to 0} \mathbb{E}_{(X,K)} \left[\frac{\log p_{\theta+\mathbf{e}_j h}(x_K|x_{-K}) - \log p_\theta(x_K|x_{-K})}{h} \right] \tag{A.11}$$

By the Mean Value Theorem, there exists $\xi(h) \in (0, h)$ such that

$$\frac{\log p_{\theta+\mathbf{e}_j h}(x_K|x_{-K}) - \log p_\theta(x_K|x_{-K})}{h} = \frac{\partial}{\partial_{\theta_j}} \log p_{\theta+\mathbf{e}_j \xi(h)}(x_K|x_{-K})$$

So

$$\begin{aligned}
&\frac{\partial}{\partial \theta_j} \mathbb{E}_{(X,K)} [\log p_\theta(x_K|x_{-K})] \\
&= \lim_{h\to 0} \left(\mathbb{E}_{(X,K)} \left[\frac{\partial}{\partial_{\theta_j}} \log p_{\theta+\mathbf{e}_j \xi(h)}(x_K|x_{-K}) \right] \right) \\
&= \mathbb{E}_{(X,K)} \left[\lim_{h\to 0} \left(\frac{\partial}{\partial_{\theta_j}} \log p_{\theta+\mathbf{e}_j \xi(h)}(x_K|x_{-K}) \right) \right] \quad \text{(Dominated Convergence Thm and Assumption 1)} \\
&= \mathbb{E}_{(X,K)} \left[\frac{\partial}{\partial_{\theta_j}} \log p_\theta(x_K|x_{-K}) \right]
\end{aligned}$$

This implies that

$$\nabla_\theta \mathbb{E}_{(X,K)} \log p_\theta(x_K|x_{-K}) = \mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K})$$

which proves (A.10). The proof of (A.11) follows analogously.

Step 2: Rewrite $\nabla^2_\theta L_{PL}(\theta^*)$

$$\begin{aligned}
\nabla^2_\theta L_{PL}(\theta) &= -\nabla^2_\theta \mathbb{E}_{(X,K)} \log p_\theta(x_K|x_{-K})_{|\theta=\theta^*}\\
&\overset{\textcircled{1}}{=} -\mathbb{E}_{(X,K)}\nabla^2_\theta \log p_\theta(x_K|x_{-K})_{|\theta=\theta^*}\\
&\overset{\textcircled{2}}{=} \mathbb{E}_{(X,K)}\nabla_\theta \log p_\theta(x_K|x_{-K})\nabla_\theta \log p_\theta(x_K|x_{-K})^\top_{|\theta=\theta^*} - \frac{\nabla^2_\theta p_\theta(x_K|x_{-K})}{p_\theta(x_K|x_{-K})}_{|\theta=\theta^*}\\
&\overset{\textcircled{3}}{=} \mathbb{E}_{(X,K)}\nabla_\theta \log p_\theta(x_K|x_{-K})\nabla_\theta \log p_\theta(x_K|x_{-K})^\top_{|\theta=\theta^*}
\end{aligned} \tag{A.12}$$

where $\textcircled{1}$ follows by exchanging the order of expectation and Hessian ($S \in \mathcal{S}_k$ and $x \in \Omega$ are finite), and this is valid by **Step 1** above , $\textcircled{2}$ by an application of chain rule. The last equality $\textcircled{3}$ follows by a similar calculation as the proof of the classical information matrix equality:

$$\begin{aligned}
&\mathbb{E}_{(X,K)}\frac{\nabla^2_\theta p_\theta(x_K|x_{-K})}{p_\theta(x_K|x_{-K})}_{|\theta=\theta^*}\\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}}\mathbb{E}_{x_K|x_{-K}}\frac{\nabla^2_\theta p_\theta(x_K|x_{-K})}{p_\theta(x_K|x_{-K})}_{|\theta=\theta^*}\\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}}\int \frac{\nabla^2_\theta p_{\theta^*}(x_K|x_{-K})}{p_{\theta^*}(x_K|x_{-K})}\cdot p_{\mathcal{X}}(x_K|x_{-K})dx_K\\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}}\int \nabla^2_\theta p_{\theta^*}(x_K|x_{-K})dx_K \quad \text{, since } p_{\theta^*} = p_{\mathcal{X}} \text{ by Assumption 2)}\\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}}\nabla^2_\theta\int p_{\theta^*}(x_K|x_{-K})dx_K \quad \text{, by exchanging the order of expectation and Hessian}\\
&= 0
\end{aligned}$$

where the last equality follows since $\int p_{\theta^*}(x_K|x_{-K})dx_K = 1$ (so doesn't depend on θ). Similarly, we have:

$$\begin{aligned}
&\mathbb{E}_{(X,K)}\nabla_\theta \log p_\theta(x_K|x_{-K})_{|\theta=\theta^*}\\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}}\mathbb{E}_{x_K|x_{-K}}\frac{\nabla_\theta p_\theta(x_K|x_{-K})}{p_\theta(x_K|x_{-K})}_{|\theta=\theta^*}\\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}}\int \nabla_\theta p_\theta(x_K|x_{-K})dx_K|_{\theta=\theta^*}\\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}}\nabla_\theta\int p_\theta(x_K|x_{-K})dx_K|_{\theta=\theta^*}\\
&= 0
\end{aligned}$$

where the last equality follows since $\int p_\theta(x_K|x_{-K})dx_K = 1$ (so doesn't depend on θ). Plugging this into the definition of covariance, we have:

$$\begin{aligned}
&\mathrm{Cov}(\nabla_\theta - \log p_\theta(X_K|X_{-K}))_{|\theta=\theta^*}\\
&= \mathbb{E}_{(X,K)}\nabla_\theta \log p_\theta(x_K|x_{-K})\nabla_\theta \log p_\theta(x_K|x_{-K})^\top\\
&\quad - \mathbb{E}_{(X,K)}\nabla_\theta \log p_\theta(x_K|x_{-K})\cdot\mathbb{E}_{(X,K)}\nabla_\theta \log p_\theta(x_K|x_{-K})^\top_{|\theta=\theta^*}\\
&= \mathbb{E}_{(X,K)}\nabla_\theta \log p_\theta(x_K|x_{-K})\nabla_\theta \log p_\theta(x_K|x_{-K})^\top_{|\theta=\theta^*}
\end{aligned} \tag{A.13}$$

The proof of the lemma thus follows because the RHS of Equation (A.13) matches that of Equation (A.12).

□

B. Proof of Theorem 1: Masking more is (statistically) better

In this Section, we provide the proof for Theorem 1.

Proof of Theorem 1. All the expectations in the proof will be taken with respect to $(X, K) \sim p_{\mathcal{X}} \times p_{\mathcal{K}}$. To decrease the notational load, we will not explicitly write $p_{\mathcal{X}} \times p_{\mathcal{K}}$. By Lemma 2, we have:

$$\nabla^2_\theta L^k_{PL}(\theta^*) = \mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K}) \nabla_\theta \log p_\theta(x_K|x_{-K})^\top_{|\theta=\theta^*} \tag{B.14}$$

Let $\mathcal{S}_k$ denote the set $\{K \subset [N] \mid |K| = k\}$. For every $T \in \mathcal{S}_{k+1}$ and $a \in T$ we have:

$$\begin{aligned} \log p(x_T|x_{-T}) &= \log p(x_S, x_a|x_{-\{S\cup a\}}) \text{ where } S := T\backslash\{a\} \\ &= \log\left(p(x_a|x_{-\{S\cup a\}}) \cdot p(x_S|x_{-\{S\cup a\}}, x_a)\right) \\ &= \log p(x_a|x_{-\{S\cup a\}}) + \log p(x_S|x_{-S}) \end{aligned}$$

Using this identity, we can write:

$$\begin{aligned} \nabla^2_\theta L^{k+1}_{PL}(\theta^*) &= \mathbb{E}_{T\sim S_{k+1}} \mathbb{E}_{x_T, x_{-T}} \nabla_\theta \log p_\theta(x_T|x_{-T}) \nabla_\theta \log p_\theta(x_T|x_{-T})^\top_{|\theta=\theta^*} \\ &= \mathbb{E}_{S\sim S_k} \mathbb{E}_{a\notin S} \mathbb{E}_{x_S, x_a, x_{-\{S\cup a\}}} \nabla_\theta \log p_\theta(x_T|x_{-T}) \nabla_\theta \log p_\theta(x_T|x_{-T})^\top_{|\theta=\theta^*} \\ &= \mathbb{E}_{S\sim S_k} \mathbb{E}_{a\notin S} \mathbb{E}_{x_S, x_a, x_{-\{S\cup a\}}} \left(\nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}}) + \nabla_\theta \log p_\theta(x_S|x_{-S})\right) \\ &\quad \cdot \left(\nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}}) + \nabla \log p_\theta(x_S|x_{-S})\right)^\top_{|\theta=\theta^*} \end{aligned} \tag{B.15}$$

Let us denote:

$$\begin{aligned} A &:= \mathbb{E}_{S\sim S_k} \mathbb{E}_{a\notin S} \mathbb{E}_x \nabla_\theta \log p_\theta(x_S|x_{-S}) \cdot \nabla_\theta \log p_\theta(x_S|x_{-S})^\top_{|\theta=\theta^*} \\ B &:= \mathbb{E}_{S\sim S_k} \mathbb{E}_{a\notin S} \mathbb{E}_x \nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}}) \cdot \nabla_\theta \log p_\theta(x_S|x_{-S})^\top_{|\theta=\theta^*} \\ C &:= \mathbb{E}_{S\sim S_k} \mathbb{E}_{a\notin S} \mathbb{E}_x \nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}}) \cdot \nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}})^\top_{|\theta=\theta^*} \end{aligned}$$

By expanding the previous expression, we have

$$\nabla^2_\theta L^{k+1}_{PL}(\theta^*) = A + B + B^\top + C \tag{B.16}$$

Consider A first. Note that for a fixed $S \in S_k$, $\mathbb{E}_x \nabla_\theta \log p_\theta(x_S|x_{-S}) \cdot \nabla_\theta \log p_\theta(x_S|x_{-S})^\top$ is independent of $a \notin S$ and therefore:

$$\begin{aligned} A &= \mathbb{E}_{S\sim S_k} \mathbb{E}_x \nabla_\theta \log p_\theta(x_S|x_{-S}) \cdot \nabla_\theta \log p_\theta(x_S|x_{-S})^\top_{|\theta=\theta^*} \\ &= \nabla^2_\theta L^k_{PL}(\theta^*) \quad \text{(by Equation (B.14))} \end{aligned}$$

Proceeding to B, for a given $S \in S_k, x_{-S}$, we have

$$\begin{aligned} &\mathbb{E}_{x_S|x_{-S}} \left[\nabla_\theta \log p_\theta(x_S|x_{-S})^\top\right]_{|\theta=\theta^*} \\ &= \int \nabla_\theta \log p_\theta(x_S|x_{-S})^\top \cdot p(x_S|x_{-S}) dx_S|_{\theta=\theta^*} \\ &= \int \frac{\nabla_\theta p_\theta(x_S|x_{-S})}{p_{\theta^*}(x_S|x_{-S})}^\top \cdot p(x_S|x_{-S}) dx_S|_{\theta=\theta^*} \\ &= \int \nabla_\theta p_\theta(x_S|x_{-S})^\top dx_S|_{\theta=\theta^*} \\ &= \nabla_\theta \int p_\theta(x_S|x_{-S})^\top dx_S|_{\theta=\theta^*} \quad \text{(valid under Assumption 1, see Step 1 in the proof of Lemma 2)} \\ &= \nabla_\theta 1 = 0 \end{aligned} \tag{B.17}$$

Therefore:

$$\begin{aligned}
B &= \mathbb{E}_{S\sim S_k}\mathbb{E}_{a\not\in S}\mathbb{E}_{x_S,x_a,x_{-\{S\cup a\}}}\nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}})\cdot\nabla_\theta \log p_\theta(x_S|x_{-S})^\top|_{\theta=\theta^*}\\
&= \mathbb{E}_{S\sim S_k}\mathbb{E}_{a\not\in S}\mathbb{E}_{x_a,x_{-\{S\cup a\}}}\nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}})\cdot\mathbb{E}_{x_S}\left[\nabla_\theta \log p_\theta(x_S|x_{-S})^\top\right]|_{\theta=\theta^*}\\
&\quad\text{(valid under Assumption 1, see Step 1 in the proof of Lemma 2)}\\
&= \mathbb{E}_{S\sim S_k}\mathbb{E}_{a\not\in S}\mathbb{E}_{x_a,x_{-\{S\cup a\}}}\nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}})|_{\theta=\theta^*}\cdot 0 \quad \text{(by Equation (B.17))}\\
&= 0
\end{aligned}$$

Finally, each term $\nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}})\cdot\nabla_\theta \log p_\theta(x_a|x_{-\{S\cup a\}})^\top \succeq 0$ therefore $C \succeq 0$.

Plugging this back in (B.16), we have:

$$\nabla^2_\theta L^{k+1}_{PL}(\theta^*) = \nabla^2_\theta L^{k}_{PL}(\theta^*) + C \succeq \nabla^2_\theta L^{k}_{PL}(\theta^*)$$

Consequently, by monotonicity of the matrix inverse, we have

$$\Gamma^{k+1}_{PL} = \left(\nabla^2_\theta L^{k+1}_{PL}(\theta^*)\right)^{-1} \preceq \left(\nabla^2_\theta L^{k}_{PL}(\theta^*)\right)^{-1} = \Gamma^{k}_{PL}$$

as we need. □

C. Generalizations for adaptive masking

In this section, we provide proofs for several of the claims in Section 2.2.3.

C.1. Conditioning on K

First, we clarify a slightly subtle (and counterintuitive) point stressed in Remark 3: in general, $p_{\mathcal{X}}(x_K|x_{-K}, K) \neq p_{\mathcal{X}}(x_K|x_{-K})$.

Lemma 3. *Consider $\mathcal{X} = \{(0,0),(0,1),(1,0),(1,1)\}$. There exists a distribution $p_{\mathcal{X},\mathcal{K}}$ such that $p_{\mathcal{X}}(x_K|x_{-K}, K) \neq p_{\mathcal{X}}(x_K|x_{-K})$ for some $x \in \mathcal{X}$, $K \in \mathcal{K}$.*

Proof. To define $p_{\mathcal{X},\mathcal{K}}$, it suffices to define $p_{\mathcal{X}}$ and $p_{\mathcal{K}}(\cdot|x), \forall x \in \mathcal{X}$.

$$p_{\mathcal{X}}(X) = \begin{cases} (0,0), & \text{with probability } \frac{1}{2} \\ (0,1), & \text{with probability } \frac{1}{3} \\ (1,0), & \text{with probability } \frac{1}{6} \\ (0,0), & \text{with probability } 0 \end{cases}$$

and let

$$p_{\mathcal{K}}(K \mid X = (0,0)) = \begin{cases} \{0\}, & \text{with probability } \frac{1}{2} \\ \{1\}, & \text{with probability } \frac{1}{2} \end{cases}$$

$$p_{\mathcal{K}}(K \mid X = (0,1)) = \begin{cases} \{0\}, & \text{with probability } \frac{1}{3} \\ \{1\}, & \text{with probability } \frac{2}{3} \end{cases}$$

$$p_{\mathcal{K}}(K \mid X = (1,0)) = \begin{cases} \{0\}, & \text{with probability } \frac{1}{4} \\ \{1\}, & \text{with probability } \frac{3}{4} \end{cases}$$

By multiplying $p_{\mathcal{X}}(X)$ and $p_{\mathcal{K}}(K \mid X)$, we have

$$p((0,0),\{0\}) = \frac{1}{4}, \quad p((0,0),\{1\}) = \frac{1}{4}$$
$$p((0,1),\{0\}) = \frac{1}{9}, \quad p((0,1),\{1\}) = \frac{2}{9}$$
$$p((1,0),\{0\}) = \frac{1}{24}, \quad p((1,0),\{1\}) = \frac{1}{8}$$

Finally, we will see that $p_{\mathcal{X}}(x_1 = 0|x_0 = 0, \{0\}) \neq p_{\mathcal{X}}(x_1 = 0|x_0 = 0)$:

$$p_{\mathcal{X}}(x_1 = 0|x_0 = 0, \{0\}) = \frac{p((0,0),\{0\})}{p((0,0),\{0\}) + p((0,1),\{0\})} = \frac{9}{13}$$
$$p_{\mathcal{X}}(x_1 = 0|x_0 = 0) = \frac{p_{\mathcal{X}}((0,0))}{p_{\mathcal{X}}((0,0)) + p_{\mathcal{X}}((0,1))} = \frac{3}{5}$$

□

Instead, by correctly marginalizing, the following equality obtains:

Lemma 4. *For any distribution $p_{\mathcal{X},\mathcal{K}}$, we have:*

$$\forall x \in \mathcal{X},\ \forall K \in \mathcal{K},\ p_{\mathcal{X}}(x_K|x_{-K}) = \mathbb{E}_{K' \sim p_{\mathcal{K}}(\cdot|x_{-K})}\left[p_{\mathcal{X},\mathcal{K}}(x_K|x_{-K}, K')\right] \tag{C.18}$$

Proof. The proof proceeds by a sequence of straightforward rewrites:

$$\begin{aligned}
p_{\mathcal{X}}(x_K|x_{-K}) &= \frac{p_{\mathcal{X}}(x_K, x_{-K})}{p_{\mathcal{X}}(x_{-K})} \\
&= \sum_{K'} \frac{p_{\mathcal{X},\mathcal{K}}(x_K, x_{-K}, K')}{p_{\mathcal{X}}(x_{-K})} \\
&= \sum_{K'} \frac{p_{\mathcal{X},\mathcal{K}}(K', x_{-K})}{p_{\mathcal{X}}(x_{-K})} \cdot \frac{p_{\mathcal{X},\mathcal{K}}(x_K, x_{-K}, K')}{p_{\mathcal{X},\mathcal{K}}(x_{-K}, K')} \\
&= \sum_{K'} p_{\mathcal{K}}(K' \mid x_{-K}) p_{\mathcal{X},\mathcal{K}}(x_K|x_{-K}, K') \\
&= \mathbb{E}_{K' \sim p_{\mathcal{K}}(\cdot|x_{-s})} \left[p_{\mathcal{X},\mathcal{K}}(x_K|x_{-K}, K')\right]
\end{aligned}$$

□

C.2. Information matrix equality for adaptive masking

We prove a more general version of Lemma 2 when $p_{\mathcal{K}}$ is allowed to depend on X, which is needed for Theorem 3. Recall from Section 2.2.3 that the distribution $p_{\mathcal{X},\mathcal{K}}$ is defined such that:

$$p_{\mathcal{X},\mathcal{K}}(X, K) := p_{\mathcal{X}}(X) p_{\mathcal{K}}(K|X)$$

Lemma 5 (Generalized information matrix equality, adaptive masking)**.** *Under Assumption 1 and Assumption 2, the weighted pseudolikelihood loss (Definition 7) verifies:* $\nabla^2_\theta L_{PL}(\theta^*) = \mathrm{Cov}_{(X,K)\sim p_{\mathcal{X},\mathcal{K}}}(-\nabla_\theta \log p_\theta(X_K|X_{-K}, K))_{|\theta=\theta^*}$.

Proof. All the expectations in the proof will be taken with respect to $(X, K) \sim p_{\mathcal{X},\mathcal{K}}$. To decrease the notational load, we will not explicitly write $p_{\mathcal{X},\mathcal{K}}$. Same as Lemma 2, the proof proceeds by first exchanging the order of expectations and derivatives, and using that to show the appropriate terms in the expression for $\nabla^2_\theta L_{PL}(\theta^*)$ vanish.

In fact, it's readily seen that the proof of Step 1 in Lemma 2 (Appendix A) doesn't depend on $p_{\mathcal{X},\mathcal{K}}$ being a product distribution, and the same proof applies to our setting, namely we have:

$$\nabla_\theta \mathbb{E}_{(X,K)} \log p_\theta(x_K|x_{-K}, K) = \mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K}, K) \tag{C.19}$$

$$\nabla^2_\theta \mathbb{E}_{(X,K)} \log p_\theta(x_K|x_{-K}, K) = \mathbb{E}_{(X,K)} \nabla^2_\theta \log p_\theta(x_K|x_{-K}, K) \tag{C.20}$$

We can also rewrite the expression for $\nabla^2_\theta L_{PL}(\theta^*)$ almost the same way we did in Step 2 in Lemma 2:

$$\begin{aligned}
\nabla^2_\theta L_{PL}(\theta) &= -\nabla^2_\theta \mathbb{E}_{(X,K)} \log p_\theta(x_K|x_{-K}, K)_{|\theta=\theta^*} \\
&\overset{\textcircled{1}}{=} -\mathbb{E}_{(X,K)} \nabla^2_\theta \log p_\theta(x_K|x_{-K}, K)_{|\theta=\theta^*} \\
&\overset{\textcircled{2}}{=} \mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K}, K) \nabla_\theta \log p_\theta(x_K|x_{-K}, K)^\top_{|\theta=\theta^*} - {\frac{\nabla^2_\theta p_\theta(x_K|x_{-K}, K)}{p_\theta(x_K|x_{-K}, K)}}_{|\theta=\theta^*} \\
&\overset{\textcircled{3}}{=} \mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K}, K) \nabla_\theta \log p_\theta(x_K|x_{-K}, K)^\top_{|\theta=\theta^*}
\end{aligned} \tag{C.21}$$

where ① follows by exchanging the order of expectation and Hessian ($S \in \mathcal{S}_k$ and $x \in \Omega$ are finite), and this is valid by **Step 1** above , ② by an application of chain rule. The last equality ③ follows by a similar calculation as the proof of the classical information matrix equality (and again, analogously to the calculation in Lemma 2):

$$
\begin{aligned}
&\mathbb{E}_{(X,K)} \frac{\nabla^2_\theta p_\theta(x_K|x_{-K},K)}{p_\theta(x_K|x_{-K},K)}_{|\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \mathbb{E}_{x_K|x_{-K},K} \frac{\nabla^2_\theta p_\theta(x_K|x_{-K},K)}{p_\theta(x_K|x_{-K},K)}_{|\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \int \frac{\nabla^2_\theta p_{\theta^*}(x_K|x_{-K},K)}{p_{\theta^*}(x_K|x_{-K},K)} \cdot p_{\mathcal{X},\mathcal{K}}(x_K|x_{-K},K) dx_K \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \int \nabla^2_\theta p_{\theta^*}(x_K|x_{-K},K) dx_K \quad \text{, since } p_{\theta^*} = p_{\mathcal{X}} \text{ by Assumption 2 and Definition (1)} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \nabla^2_\theta \int p_{\theta^*}(x_K|x_{-K},K) dx_K \quad \text{, by exchanging the order of expectation and Hessian} \\
&= 0
\end{aligned}
$$

where the last equality follows since $\int p_{\theta^*}(x_K|x_{-K},K)dx_K = 1$ (so doesn't depend on θ). Similarly, we have:

$$
\begin{aligned}
&\mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K},K)_{|\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \mathbb{E}_{x_K|x_{-K},K} \frac{\nabla_\theta p_\theta(x_K|x_{-K},K)}{p_\theta(x_K|x_{-K},K)}_{|\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \int \nabla_\theta p_\theta(x_K|x_{-K},K) dx_K|_{\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \nabla_\theta \int p_\theta(x_K|x_{-K},K) dx_K|_{\theta=\theta^*} \\
&= 0
\end{aligned}
$$

where the last equality follows since $\int p_\theta(x_K|x_{-K},K)dx_K = 1$ (so doesn't depend on θ). Plugging this into the definition of covariance, we have:

$$
\begin{aligned}
&\mathrm{Cov}(\nabla_\theta l_{PL}(\theta^*)) \\
&= \mathrm{Cov}(\nabla_\theta - \log p_\theta(X_K|X_{-K},K))_{|\theta=\theta^*} \\
&= \mathbb{E}_{(X,K)} \nabla_\theta \log p_\theta(x_K|x_{-K},K) \nabla_\theta \log p_\theta(x_K|x_{-K},K)^\top_{|\theta=\theta^*}
\end{aligned} \tag{C.22}
$$

which finishes the proof of the Lemma. □

C.3. Proof of Proposition 1: Dirichlet form for adaptive block dynamics

Proposition 1 (Dirichlet form for adaptive weighted block dynamics)**.** *The Dirichlet form corresponding to the weighted block dynamics (Definition 8) is:*

$$
\mathcal{E}(f,g) = \mathbb{E}_{(X_{-K},K)\sim p_{\mathcal{X},\mathcal{K}}} \left[Cov_{X_K|(X_{-K},K)}(f,g) \right]
$$

Proof. Let us denote by $\Xi := \mathcal{X} \times \mathcal{K}$, and note that Ξ is the domain for both f and g. According definition of block dynamics in Definition 6, for each pair of states $(X,K_1),(Y,K_2) \in \Xi$, the transition matrix P is:

$$
P\left((X,K_1),(Y,K_2)\right) = \mathbb{1}_{K_1=K_2} \mathbb{1}_{X_{-K_1}=Y_{-K_1}} p(Y_{K_1} \mid X_{-K_1},K_1) \tag{C.23}
$$

The rest of the proof is straightforward calculation, expanding the expression in the definition of the Dirichlet form (Definition 4). Namely, we have:

$$
\begin{aligned}
&\mathcal{E}_P(f,g) \\
&= \frac{1}{2} \sum_{(X,K_1),(Y,K_2)\in\Xi} p(X,K_1)P\left((X,K_1),(Y,K_2)\right)(f(X,K_1)-f(Y,K_2))(g(X,K_1)-g(Y,K_2)) \\
&= \frac{1}{2} \sum_{(X,K_1),(Y,K_2)\in\Xi} p(X,K_1)\cdot(f(X,K_1)-f(Y,K_2))(g(X,K_1)-g(Y,K_2)) \\
&\quad \cdot \mathbb{1}_{K_1=K_2}\mathbb{1}_{X_{-K_1}=Y_{-K_1}} p(Y_{K_1} \mid X_{-K_1}, X_1) \\
&= \frac{1}{2} \sum_{X\in\mathcal{X}} p_{\mathcal{X}}(X) \sum_{K\in\mathcal{K}} p_{\mathcal{K}}(K\mid X) \sum_{Y\in\mathcal{X},Y_{-K}=X_{-K}} p(Y_K \mid X_{-K},K)\cdot(f(X,K)-f(Y,K))(g(X,K)-g(Y,K)) \\
&= \frac{1}{2}\mathbb{E}_X\mathbb{E}_{K|X}\mathbb{E}_{Y_K|X_{-K},K}(f(X,K)-f(Y,K))(g(X,K)-g(Y,K)) \\
&= \frac{1}{2}\cdot\left(2\cdot\mathbb{E}_K\mathbb{E}_{X_{-K}|K}\mathbb{E}_{X_K|X_{-K},K} f(X,K)g(X,K) - 2\cdot\mathbb{E}_K\mathbb{E}_{X_{-K}|K}\mathbb{E}_{X_K|X_{-K},K}\left[f(X,K)\right]\cdot\mathbb{E}_{X_K|X_{-K},K}\left[g(X,K)\right]\right) \\
&\quad \text{(we can merge terms because the roles of } X \text{ and } Y \text{ are symmetric)} \\
&= \mathbb{E}_{K\sim p}\big[\mathbb{E}_{x_{-K}\sim p(x_{-K}|K)}\big[\mathbb{E}_{y_K\sim p(x_K|x_{-K},K)}\left[f(y,K)g(y,K)\right] - \mathbb{E}_{y_K\sim p(x_K|x_{-K},K)}\left[f(y,K)\right] \\
&\quad \cdot \mathbb{E}_{y_K\sim p(x_K|x_{-K},K)}\left[g(y,K)\right]\big]\big] \\
&= \mathbb{E}_{(X_{-K},K)\sim p_{\mathcal{X},\mathcal{K}}}\left[\mathrm{Cov}_{X_K|(X_{-K},K)}(f,g)\right] \qquad \text{(C.24)}
\end{aligned}
$$

which completes the proof. □

C.4. Proof of Theorem 3: Asymptotic sample complexity for adaptively-weighted MPLE

Note that Theorem 3 generalizes Theorem 2. To reduce proof duplication, we only write the proof for the more general Theorem 3 here. Notational definition for $p_\theta(x_K|x_{-K},K)$ and other background info are in Section 2.2.3.

Theorem 3 (Asymptotic variance of adaptively-weighted MPLE under a Poincaré Inequality, generalization of Theorem 2)**.** *Suppose the distribution p_{θ^*} satisfies a Poincaré inequality with constant C with respect to the adaptively-weighted block dynamics. Then under Assumption 1 and Assumption 2 where $p_\theta(x_K|x_{-K})$ is replaced by $p_\theta(x_K|x_{-K},K)$, the asymptotic variance of the adaptively-weighted MPLE can be bounded as: $\Gamma_{PL} \preceq C\mathcal{I}^{-1}$ where $\mathcal{I}$ is the Fisher Information matrix (Definition 1).*

Proof. By Lemma 5 and Lemma 11, for n training samples as $n\to\infty$, we have:

$$
\sqrt{n}(\hat{\theta}_{PL}-\theta^*) \to \mathcal{N}\left(0, (\mathrm{Cov}_{(X,K)\sim p_{\mathcal{X},\mathcal{K}}}(-\nabla_\theta \log p_\theta(X_K|X_{-K},K))_{|\theta=\theta^*})^{-1}\right) \qquad \text{(C.25)}
$$

Now we relate the RHS of (C.25) to $\mathcal{I}$. Let d_Θ denote the dimensionality of θ, i.e. $\theta\in\mathbb{R}^{d_\Theta}$. Then, for any test vector $v\in\mathbb{R}^{d_\Theta}$ we have:

$$
\begin{aligned}
&v^\top\mathbb{E}_{(X,K)}\nabla_\theta\log p_\theta(x_K|x_{-S},K)\nabla_\theta\log p_\theta(x_K|x_{-S},K)^\top v_{|\theta=\theta^*} \\
&= \mathbb{E}_{(X,K)}(\nabla_\theta\log p_\theta(x_K|x_{-K},K)^\top v)^2_{|\theta=\theta^*} \\
&= \mathbb{E}_K\mathbb{E}_{x_{-K}|K}\mathrm{Var}_{x_K|x_{-K},K}(\nabla_\theta\log p_\theta(x_K|x_{-K},K)^\top v)_{|\theta=\theta^*} + (\mathbb{E}_{x_K|x_{-K},K}\nabla_\theta\log p_\theta(x_K|x_{-K},K)^\top v)^2_{|\theta=\theta^*} \qquad \text{(C.26)}
\end{aligned}
$$

Denote $f(X,K) := \nabla_\theta\log p_\theta(x_K|x_{-K},K)^\top v$ Consider the two parts in Equation (C.26) separately: the first term is simply $\mathbb{E}_K\mathbb{E}_{x_{-K}|K}\mathrm{Var}_{x_S|x_{-K,K}}(f(X,K))$, which, by Proposition 1, is equal to $\mathcal{E}_P(f,f)$. Moreover, by Poincaré inequality (Definition 5), this is $\geq \frac{1}{C}\mathrm{Var}_p(f)$.

The second term simplifies to

$$
\begin{aligned}
&\mathbb{E}_K \mathbb{E}_{x_{-K}|K} \left(\mathbb{E}_{x_K|x_{-K},K}\left[\nabla_\theta \log p_\theta(x_K|x_{-K},K)^\top v\right]\right)^2_{|\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \left(\mathbb{E}_{x_K|x_{-K},K}\left[\left(\frac{\nabla_\theta p_\theta(x_K|x_{-K},K)}{p_\theta(x_K|x_{-K},K)}\right)^\top v\right]\right)^2_{|\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \left[\int \left(\frac{\nabla_\theta p_\theta(x_K|x_{-K},K)}{p_\theta(x_K|x_{-K},K)}\right)^\top v \cdot p_{\mathcal{X},\mathcal{K}}(x_K|x_{-K},K)\, dx_K\right]^2_{|\theta=\theta^*} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \left[\int \left(\nabla_\theta p_\theta(x_K|x_{-K},K)\right)^\top v\, dx_K\right]^2_{|\theta=\theta^*} \quad \text{, since } p_{\theta^*} = p_{\mathcal{X}} \text{ by Assumption 2 and Definition (1)} \\
&= \mathbb{E}_K \mathbb{E}_{x_{-K}|K} \left[\nabla_\theta \left(\int p_\theta(x_K|x_{-K},K)\, dx_K\right)^\top v\right]^2_{|\theta=\theta^*} \quad \text{(since } p_\theta \text{ is differentiable wrt } \theta \text{ by Assumption 1)} \\
&= 0
\end{aligned}
$$

Therefore, we have $\mathrm{Cov}_{(X,K)\sim p_{\mathcal{X},\mathcal{K}}}(-\nabla_\theta \log p_\theta(X_K|X_{-K},K))_{|\theta=\theta^*} \succeq \frac{1}{C}\mathcal{I}$.

Plugging into Equation (C.25), and using the monotonicity of the matrix inverse (Toda, 2011), we obtain the upper bound on the asymptotic variance of our estimator we want:

$$\Gamma_{PL} \preceq C\mathcal{I}^{-1}$$

□

D. Proof of Theorem 4: Generalization bound for learning the joint distribution

We first state our overall structure of the proof of Theorem 4, and then state and prove the key lemmas mentioned therein.

Theorem 4 (Generalization bound for learning the joint distribution)**.** *Let* $\hat{\theta} := \arg\min_{\theta} \hat{L}_{PL}(\theta)$. *Under Assumption 3 and Assumption 4,* $\forall \epsilon > 0$, $\forall \delta \in (0,1)$, *with probability at least* $1-\delta$ *we have* $D_{\mathrm{TV}}\left(p_{\hat{\theta}}, p_{\mathcal{X}}\right) < \sqrt{\frac{1}{2}\bar{C}_{AT}(p_{\hat{\theta}})\left(\hat{L}_{PL}(\hat{\theta}) + B \cdot \ln\frac{1}{\beta} + \epsilon\right)} + C$ *where* $B = \sqrt{\frac{2^{3N}|\Omega|^N C_{\epsilon}(\Theta)}{m \cdot \delta}} + \sqrt{\frac{\ln\frac{8C_{\epsilon}(\Theta)}{\delta}}{2n}}$, *and* $C = \sqrt{\frac{|\Omega|^{3N}}{8\delta n}}$.

Proof. We first introduce a few pieces of notation. We will denote the data samples as $\mathcal{S}_{\mathcal{X}} := \{X^{(i)} | X^{(i)} \sim p(X)\}$, $|\mathcal{S}_{\mathcal{X}}| = n$, and for each $X^{(i)}$ we sample m masks $\mathcal{S}_{\mathcal{K}}{}^{(i)} := \{K_1^{(i)}, \ldots, K_m^{(i)}\}$ in which $K_j^{(i)}$ is sampled iid from $\mathcal{K}$ according to probabilities $p_{\mathcal{K}}(\cdot \mid X)$. [16] Theorem 4 follows by combining the following steps:

Step 1: relating closeness of the *conditional* distributions (i.e. the loss) to closeness of the *joint* distribution. The connection is established through the definition of the block-generalized approximate tensorization of entropy in Definition 9, by which we get[17]:

$$D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}, p_{\hat{\theta}}\right) \leq \bar{C}_{AT}(p_{\hat{\theta}}) \tilde{L}_{PL}(\hat{\theta})$$

The details are in Proposition 6. By Pinsker's inequality, this implies

$$D_{\mathrm{TV}}\left(\tilde{p}_{\mathcal{X}}, p_{\hat{\theta}}\right) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}, p_{\hat{\theta}}\right)} \leq \sqrt{\frac{1}{2}\bar{C}_{AT}(p_{\hat{\theta}}) \tilde{L}_{PL}(\hat{\theta})} \tag{D.27}$$

Step 2: generalization bound for learning the *conditional* distributions. We show that Assumption 3 and Assumption 4 imply a generalization guarantee for learning the *conditional* distributions from a finite sample of sequences and masked positions. We show that with probability at least $1 - \frac{\delta}{2}$, we have

$$\left| L_{PL}(\hat{\theta}) - \tilde{L}_{PL}(\hat{\theta}) \right| < \left(\sqrt{\frac{2^{3N} |\Omega|^N C_{\epsilon}(\Theta)}{m \cdot \delta}} + \sqrt{\frac{\ln \frac{8C_{\epsilon}(\Theta)}{\delta}}{2n}} \right) \cdot \ln \frac{1}{\beta} + \epsilon \tag{D.28}$$

Proof details of this step are in Corollary 2.

Step 3: *empirical* joint distribution converges to *population* joint distribution. With probability at least $1 - \frac{\delta}{2}$, we have:

$$D_{\mathrm{TV}}\left(\tilde{p}_{\mathcal{X}}, p_{\mathcal{X}}\right) < \sqrt{\frac{|\Omega|^{3N}}{8\delta n}} \tag{D.29}$$

The proof of this is standard and details are in Lemma 8.

Step 4: union bound and triangle inequality By union bound, with probability at least $1-\delta$, both Equation (D.28) and Equation (D.29) hold. Therefore, putting together the previous steps, we get:

$$
\begin{aligned}
& D_{\mathrm{TV}}\left(p_{\hat{\theta}}, p_{\mathcal{X}}\right) \leq D_{\mathrm{TV}}\left(\tilde{p}_{\mathcal{X}}, p_{\hat{\theta}}\right) + D_{\mathrm{TV}}\left(\tilde{p}_{\mathcal{X}}, p_{\mathcal{X}}\right) \quad \text{(by triangle inequality)} \\
\leq & \sqrt{\frac{1}{2}\bar{C}_{AT}(p_{\hat{\theta}}) \tilde{L}_{\mathrm{PL}}(\hat{\theta})} + \sqrt{\frac{|\Omega|^{3N}}{8\delta n}} \quad \text{(by Equation (D.27) and Equation (D.29))} \\
< & \sqrt{\frac{1}{2}\bar{C}_{AT}(p_{\hat{\theta}}) \left(\hat{L}_{PL}(\hat{\theta}) + \left(\sqrt{\frac{2^{3N} |\Omega|^N C_{\epsilon}(\Theta)}{m \cdot \delta}} + \sqrt{\frac{\ln \frac{8C_{\epsilon}(\Theta)}{\delta}}{2n}} \right) \cdot \ln \frac{1}{\beta} + \epsilon \right)} + \sqrt{\frac{|\Omega|^{3N}}{8\delta n}} \\
& \text{(by Equation (D.28))}
\end{aligned}
$$

This completes the proof of the Theorem. □

[16] Note that the $\{\cdot\}$ notation does not mean sets: duplicate entries are allowed in the training data $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{S}_{\mathcal{K}}{}^{(i)}$.

[17] Recall, $\tilde{L}_{PL}$ is defined in (2)

We proceed to Step 1 first. We show:

Proposition 6. *$D_{\mathrm{KL}}\left(p_{\mathcal{X}}, p_\theta\right) \leq \bar{C}_{AT}(p_\theta) L_{PL}(\theta)$ and $D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}, p_\theta\right) \leq \bar{C}_{AT}(p_\theta) \tilde{L}_{PL}(\theta)$*

Proof. By definition of block-generalized approximate tensorization of entropy in Definition 9

$$\begin{aligned} D_{\mathrm{KL}}(p_{\mathcal{X}}, p_\theta) &\leq \bar{C}_{AT}(p_\theta) \cdot \mathbb{E}_{X \sim p_{\mathcal{X}}}\left[\mathbb{E}_{K \sim p_{\mathcal{K}}(\cdot|X)}\left[D_{\mathrm{KL}}\left(p_{\mathcal{X}}(\cdot \mid X_{-K}, K), p_\theta(\cdot \mid X_{-K}, K)\right)\right]\right] \\ &= \bar{C}_{AT}(p_\theta) \cdot L_{\mathrm{PL}}(\theta) \end{aligned}$$

Likewise the latter holds when we replace p with $\tilde{p}$.

□

We will need the following simple observation in several concentration bounds we prove:

Proposition 7 (Bound on KL)**.** *Under Assumption 3,*

$$D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K}, K), p_\theta(\cdot|X_{-K}, K)\right) \in \left[0, \ln\frac{1}{\beta}\right]$$

Proof. By definition of D_{KL},

$$\begin{aligned} 0 \leq D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K}, K), p_\theta(\cdot|X_{-K}, K)\right) &= \sum_{X_K \in \Omega^{|K|}} \tilde{p}_{\mathcal{X}}(X_K|X_{-K}, K) \ln\frac{\tilde{p}_{\mathcal{X}}(X_K|X_{-K}, K)}{p_\theta(X_K|X_{-K}, K)} \\ &\leq \sum_{X_K \in \Omega^{|K|}} \tilde{p}_{\mathcal{X}}(X_K|X_{-K}, K) \ln\frac{1}{p_\theta(X_K|X_{-K}, K)} \\ &\leq \sum_{X_K \in \Omega^{|K|}} \tilde{p}_{\mathcal{X}}(X_K|X_{-K}, K) \ln\frac{1}{\beta} \quad \text{(by Assumption 3)} \\ &= \ln\frac{1}{\beta} \end{aligned}$$

□

we also recall a standard version of Hoeffding's inequality we'll use repeatedly:

Lemma 6 (Hoeffding's inequality)**.** *Let $Y_1, \cdots, Y_n$ be independent random variables such that $a \leq Y_i \leq b$ almost surely. Consider the sum of these random variables, $S_n = Y_1 + \cdots + Y_n$ whose expectation is $\mathbb{E}\left[S_n\right]$. Then, $\forall t > 0$, with probability at least $1 - 2e^{-\frac{2t^2}{n(b-a)^2}}$, we have $\left|S_n - \mathbb{E}\left[S_n\right]\right| < t$.*

Most of the generalization bounds we need for Step 2 (in particular, Corollary 2) will be derived from the following Lemma:

Lemma 7 (Point-wise generalization bound for learning conditional distributions)**.** *Fix a $\theta \in \Theta$ satisfying Assumption 3. $\forall \epsilon, t > 0$, with probability at least $1 - \frac{2^{N-2}|\Omega|^N}{\epsilon^2 m} - 2e^{-\frac{2t^2}{n \cdot \left(\ln\frac{1}{\beta}\right)^2}}$, we have*

$$\left|\hat{L}_{PL}(\theta) - \tilde{L}_{PL}(\theta)\right| < 2^N \epsilon \cdot \ln\frac{1}{\beta} + \frac{t}{n}$$

Proof. For notational convenience, let us denote by $\mathcal{S}_{\mathcal{X}}$ the training data points $\{X^{(i)}\}_{i \in [n]}$, and let us denote by $\mathcal{S}_{\mathcal{K}}(X)$ the set of masks corresponding to the training data point X.

Step 1: concentration over masked configurations

We first prove that $\hat{L}_{PL}(\theta)$ (Definition 2) concentrates to the expectation over masked positions K as m increases. [18]

[18] Note that the terms $D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K}, K), p_\theta(\cdot|X_{-K}, K)\right)$ are (generally) not independent for different K. Besides, the terms $\sum_{K \in \mathcal{S}_{\mathcal{K}}} D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K}, K), p_\theta(\cdot|X_{-K}, K)\right)$ are (generally) not independent for different $\mathcal{S}_{\mathcal{K}}$.

Denote

$$f(X) \coloneqq \mathbb{E}_{K\sim p_{\mathcal{K}}(\cdot|X)}\left[D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K},K), p_\theta(\cdot|X_{-K},K)\right)\right] \tag{D.30}$$

Then the expectation of $\hat{L}_{PL}(\theta)$ over the randomness of $\mathcal{S}_{\mathcal{K}}$ is:

$$\begin{aligned}\mathbb{E}_{\{\mathcal{S}_{\mathcal{K}}(j)\,|\,j\in[n]\}}\left[\hat{L}_{PL}(\theta)\right] &= \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}\frac{1}{m}\mathbb{E}_{\mathcal{S}_{\mathcal{K}}(j)}\left[\sum_{K\in\mathcal{S}_{\mathcal{K}}(X)} D_{\mathrm{KL}}\left(\tilde{p}(\cdot|X_{-K},K), p_\theta(\cdot|X_{-K},K)\right)\right]\\ &= \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}\mathbb{E}_{K\sim p_{\mathcal{K}}(\cdot|X)}\left[D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K},K), p_\theta(\cdot|X_{-K},K)\right)\right]\\ &= \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}} f(X) \qquad\qquad (\text{D.31})\end{aligned}$$

Moreover, for each K, the (observed) empirical probability $p_S(K \mid X)$ converges to the true probability $p_{\mathcal{K}}(K \mid X)$ as m increases, because the count, $p_S(K \mid X)\cdot m$, follows the binomial distribution Binomial$(m, p_{\mathcal{K}}(K \mid X))$. More specifically, by Chebyshev's inequality, $\forall\epsilon > 0$, and a fixed X we have:

$$\begin{aligned}\mathbb{P}\left\{|p_S(K \mid X) - p_{\mathcal{K}}(K \mid X)| \geq \epsilon\right\} &= \mathbb{P}\left\{|p_S(K \mid X)m - p_{\mathcal{K}}(K \mid X)m| \geq \epsilon m\right\}\\ &\leq \frac{\mathrm{Var}\left(p_S(K \mid X)m\right)}{\epsilon^2 m^2} \quad \text{(Chebyshev's inequality)}\\ &= \frac{m p_{\mathcal{K}}(K \mid X)(1-p_{\mathcal{K}}(K \mid X))}{\epsilon^2 m^2} \quad (\text{since } p_S(K)m \sim \text{Binomial}(m, p(K)))\\ &= \frac{p_{\mathcal{K}}(K \mid X)(1-p_{\mathcal{K}}(K \mid X))}{\epsilon^2 m}\\ &\leq \frac{1}{4\epsilon^2 m}\end{aligned}$$

Applying union bound over $K \in \{0,1\}^N, X \in \mathcal{X}$,

$$\mathbb{P}\left\{|p_S(K \mid X) - p_{\mathcal{K}}(K \mid X)| < \epsilon,\, \forall K \in \{0,1\}^N, \forall X \in \mathcal{S}_{\mathcal{X}}\right\} \geq 1 - \frac{2^N\left|\Omega\right|^N}{4\epsilon^2 m} = 1 - \frac{2^{N-2}\left|\Omega\right|^N}{\epsilon^2 m} \tag{D.32}$$

Plugging into Equation (D.30) and Equation (D.31), we get with probability at least $1 - \frac{2^{N-2}|\Omega|^N}{\epsilon^2 m}$,

$$\begin{aligned}&\left|\hat{L}_{PL}(\theta) - \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}} f(X)\right|\\ &= |\frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}\frac{1}{|\mathcal{S}_{\mathcal{K}}(X)|}\sum_{K\in\mathcal{S}_{\mathcal{K}}(X)} D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K},K), p_\theta(\cdot|X_{-K},K)\right)\\ &\quad - \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}\mathbb{E}_K\left[D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K},K), p_\theta(\cdot|X_{-K},K)\right)\right]|\\ &\leq \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}\sum_{K\in\{0,1\}^N}|p_S(K \mid X) - p_{\mathcal{K}}(K \mid X)|\cdot D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K},K), p_\theta(\cdot|X_{-K},K)\right) \quad \text{(triangle inequality)}\\ &< \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}\sum_{K\in\{0,1\}^N}\epsilon\cdot D_{\mathrm{KL}}\left(\tilde{p}_{\mathcal{X}}(\cdot|X_{-K},K), p_\theta(\cdot|X_{-K},K)\right) \quad \text{(by Equation (D.32))}\\ &\leq \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}\sum_{K\in\{0,1\}^N}\epsilon\cdot\ln\frac{1}{\beta} \quad \text{(by Proposition 7)}\\ &= \frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}} 2^N\epsilon\cdot\ln\frac{1}{\beta} \quad = 2^N\epsilon\cdot\ln\frac{1}{\beta} \qquad\qquad (\text{D.33})\end{aligned}$$

Step 2: concentration over sequences X in training data.

Recall $f(X)$ defined in Equation (D.30). We have:

$$\begin{aligned}\mathbb{E}\left[f(X)\right] &= \mathbb{E}_{X\sim\tilde{p}_{\mathcal{X}}}\left[\mathbb{E}_{K\sim p_{\mathcal{K}}(\cdot|X)}\left[D_{\mathrm{KL}}\left(\tilde{p}(\cdot|X_{-K},K),p_{\theta}(\cdot|X_{-K},K)\right)\right]\right]\\ &= \tilde{L}_{\mathrm{PL}}(\theta)\end{aligned}$$

Note that $f(X)\in[0,\ln\frac{1}{\beta}]$ by Proposition 7. Thus, applying Hoeffding's inequality (Lemma 6), $\forall t>0$, with probability at least $1-2e^{-\frac{2t^2}{n\cdot\left(\ln\frac{1}{\beta}\right)^2}}$, we have

$$\left|\frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}f(X)-\mathbb{E}\left[f(X)\right]\right|<\frac{t}{n} \tag{D.34}$$

Step 3: combining results: concentration over both masks K and sequences X.

By union bound, with probability at least

$$1-\frac{2^{N-2}\left|\Omega\right|^N}{\epsilon^2 m}-2e^{-\frac{2t^2}{n\cdot\left(\ln\frac{1}{\beta}\right)^2}}$$

both Equation (D.33) and Equation (D.34) hold, giving us:

$$\begin{aligned}&\left|\hat{L}_{PL}(\theta)-\tilde{L}_{\mathrm{PL}}(\theta)\right|\\ \leq&\left|\hat{L}_{PL}(\theta)-\frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}f(X)\right|+\left|\frac{1}{n}\sum_{X\in\mathcal{S}_{\mathcal{X}}}f(X)-\tilde{L}_{\mathrm{PL}}(\theta)\right| \quad \text{(triangle inequality)}\\ <&\,2^N\epsilon\cdot\ln\frac{1}{\beta}+\frac{t}{n}\end{aligned}$$

□

Remark 4. *The two terms in the bound given by Lemma 7, i.e. $2^N\epsilon\cdot\ln\frac{1}{\beta}$ and $\frac{t}{n}$, can be controlled by setting appropriate ϵ and t based on m and n, respectively. These two terms can reduce by increasing m and n, respectively, as we will show in the subsequent corollary. This is intuitive: we expect a smaller generalization gap when the model is trained on more mask configurations for each sequence, and when more sequences are included in the data. The first term grows with N — this is also intuitive: when the sequences are longer, it is natural to require observing more mask configurations.*

Corollary 1 (Point-wise generalization bound for learning conditional distributions, special case)**.** *Fix a $\theta\in\Theta$ satisfying Assumption 3. with probability at least $1-\delta$, we have*

$$\left|\hat{L}_{PL}(\theta)-\tilde{L}_{PL}(\theta)\right|<\left(\sqrt{\frac{2^{3N-1}\left|\Omega\right|^N}{m\cdot\delta}}+\sqrt{\frac{\ln\frac{4}{\delta}}{2n}}\right)\cdot\ln\frac{1}{\beta}$$

Proof. Apply Lemma 7 with ϵ and t satisfying

$$\begin{aligned}\epsilon&=\sqrt{\frac{2^{N-1}\left|\Omega\right|^N}{m\cdot\delta}}\\ t&=\sqrt{\frac{\ln\frac{4}{\delta}\cdot n}{2}}\ln\frac{1}{\beta}\end{aligned}$$

we have that with probability at least $1-\delta$, it holds:

$$\begin{aligned}&\left|\hat{L}_{PL}(\theta)-\tilde{L}_{\mathrm{PL}}(\theta)\right|\\&<2^N\epsilon\cdot\ln\frac{1}{\beta}+\frac{t}{n}\\&=\left(\sqrt{\frac{2^{3N-1}\left|\Omega\right|^N}{m\cdot\delta}}+\sqrt{\frac{\ln\frac{4}{\delta}}{2n}}\right)\cdot\ln\frac{1}{\beta}\end{aligned}$$

□

Corollary 2 (Uniform convergence generalization bound for learning conditional distributions)**.** *Under Assumption 3 and Assumption 4,* $\forall\delta\in(0,1)$*,* $\forall\epsilon>0$*, with probability at least* $1-\delta$*, we have*

$$\left|\hat{L}_{PL}(\theta)-\tilde{L}_{PL}(\theta)\right|<\left(\sqrt{\frac{2^{3N-1}\left|\Omega\right|^N C_\epsilon(\Theta)}{m\cdot\delta}}+\sqrt{\frac{\ln\frac{4C_\epsilon(\Theta)}{\delta}}{2n}}\right)\cdot\ln\frac{1}{\beta}+\epsilon$$

Proof. By Assumption 4, let $C_\epsilon(\Theta)$ denote the complexity of parameter space Θ, with the corresponding partition $\mathrm{Par}_\epsilon(\Theta)=\{\Theta_1,\cdots,\Theta_{C_\epsilon(\Theta)}\}$. As a corollary of Assumption 4, $\forall i,\forall\theta_1,\theta_2\in\Theta_i$, $\left|\tilde{L}_{PL}(\theta_1)-\tilde{L}_{PL}(\theta_2)\right|\leq\frac{\epsilon}{2}$, $\left|\hat{L}_{PL}(\theta_1)-\hat{L}_{PL}(\theta_2)\right|\leq\frac{\epsilon}{2}$.

Moreover, for each $i\in[C_\epsilon(\Theta)]$, arbitrarily select any point $\theta_i^*\in\Theta_i$ (as a "representative" of that region of the parameter space). Let the set of "representative points" be $\Theta^*=\{\theta_i^*\mid i\in[C_\epsilon(\Theta)]\}$. By Corollary 1, fixing any $\theta\in\Theta$ satisfying Assumption 3, then with probability at least $1-\frac{\delta}{C_\epsilon(\Theta)}$, we have

$$\left|\hat{L}_{PL}(\theta)-\tilde{L}_{\mathrm{PL}}(\theta)\right|<\left(\sqrt{\frac{2^{3N-1}\left|\Omega\right|^N C_\epsilon(\Theta)}{m\cdot\delta}}+\sqrt{\frac{\ln\frac{4C_\epsilon(\Theta)}{\delta}}{2n}}\right)\cdot\ln\frac{1}{\beta}$$

Applying union bound over $\theta_i^*\in\Theta^*$, since $|\Theta^*|=C_\epsilon(\Theta)$, with probability at least $1-\delta$,

$$\forall i\in[C_\epsilon(\Theta)],\quad\left|L_{\mathrm{PL}}(\theta_i^*)-\tilde{L}_{\mathrm{PL}}(\theta_i^*)\right|<\left(\sqrt{\frac{2^{3N-1}\left|\Omega\right|^N C_\epsilon(\Theta)}{m\cdot\delta}}+\sqrt{\frac{\ln\frac{4C_\epsilon(\Theta)}{\delta}}{2n}}\right)\cdot\ln\frac{1}{\beta}\tag{D.35}$$

Finally, by Assumption 4, $\forall\theta\in\Theta$, there exists $i\in[C_\epsilon(\Theta)]$ such that $\theta\in\Theta_i$ (i.e. θ falls into that partition), and

$$\begin{aligned}\left|\tilde{L}_{\mathrm{PL}}(\theta)-\tilde{L}_{\mathrm{PL}}(\theta_i^*)\right|&\leq\frac{\epsilon}{2}\\\left|\hat{L}_{PL}(\theta)-\hat{L}_{PL}(\theta_i^*)\right|&\leq\frac{\epsilon}{2}\end{aligned}\tag{D.36}$$

Combining Equation (D.35) and Equation (D.36) gives

$$\begin{aligned}&\left|\hat{L}_{PL}(\theta)-\tilde{L}_{\mathrm{PL}}(\theta)\right|\\&\leq\left|\hat{L}_{PL}(\theta)-\hat{L}_{PL}(\theta_i^*)\right|+\left|\hat{L}_{PL}(\theta_i^*)-\tilde{L}_{\mathrm{PL}}(\theta_i^*)\right|\\&\quad+\left|\tilde{L}_{\mathrm{PL}}(\theta_i^*)-\tilde{L}_{\mathrm{PL}}(\theta)\right|\quad\text{(by triangle inequality)}\\&<\frac{\epsilon}{2}+\left(\sqrt{\frac{2^{3N-1}\left|\Omega\right|^N C_\epsilon(\Theta)}{m\cdot\delta}}+\sqrt{\frac{\ln\frac{4C_\epsilon(\Theta)}{\delta}}{2n}}\right)\cdot\ln\frac{1}{\beta}+\frac{\epsilon}{2}\quad\text{(by Equation (D.35) and Equation (D.36))}\\&=\left(\sqrt{\frac{2^{3N-1}\left|\Omega\right|^N C_\epsilon(\Theta)}{m\cdot\delta}}+\sqrt{\frac{\ln\frac{4C_\epsilon(\Theta)}{\delta}}{2n}}\right)\cdot\ln\frac{1}{\beta}+\epsilon\end{aligned}$$

□

Finally, we complete Step 3 (proving Equation (D.29)):

Lemma 8 (Empirical PMF converges to population PMF)**.** *For any $\delta > 0$, with probability at least $1 - \delta$, we have:*

$$D_{\mathrm{TV}}\left(\tilde{p}_{\mathcal{X}}, p_{\mathcal{X}}\right) < \sqrt{\frac{|\Omega|^{3N}}{16\delta n}}$$

Proof. $\forall X \in \Omega^N$, the number of times that X appears in the training data $\mathcal{S}_{\mathcal{X}}$ follows the binomial distribution

$$\tilde{p}_{\mathcal{X}}(X)n \sim \text{Binomial}(n, p_{\mathcal{X}}(X))$$

with mean $np_{\mathcal{X}}(X)$ and variance $np_{\mathcal{X}}(X)(1 - p_{\mathcal{X}}(X))$. Hence, by Chebyshev's inequality, $\forall \epsilon > 0$

$$\begin{aligned}
\mathbb{P}\left\{|\tilde{p}_{\mathcal{X}}(X) - p_{\mathcal{X}}(X)| \geq \epsilon\right\} &= \mathbb{P}\left\{\tilde{p}_{\mathcal{X}}(X)n - p_{\mathcal{X}}(X)n \geq \epsilon n\right\} \\
&\leq \frac{\text{Var}\left(\tilde{p}_{\mathcal{X}}(X)n\right)}{\epsilon^2 n^2} \quad \text{(Chebyshev's inequality)} \\
&= \frac{np_{\mathcal{X}}(X)(1 - p_{\mathcal{X}}(X))}{\epsilon^2 n^2} \quad \text{(since } \tilde{p}_{\mathcal{X}}(X)n \sim \text{Binomial}(n, p_{\mathcal{X}}(X))\text{)} \\
&= \frac{p_{\mathcal{X}}(X)(1 - p_{\mathcal{X}}(X))}{\epsilon^2 n} \\
&\leq \frac{1}{4\epsilon^2 m}
\end{aligned}$$

Applying union bound over $X \in \Omega^N$,

$$\mathbb{P}\left\{|\tilde{p}_{\mathcal{X}}(X) - p_{\mathcal{X}}(X)| < \epsilon, \forall X \in \Omega^N\right\} \geq 1 - \frac{|\Omega|^N}{4\epsilon^2 n} \tag{D.37}$$

Hence, we get with probability at least $1 - \frac{|\Omega|^N}{4\epsilon^2 n}$,

$$D_{\mathrm{TV}}\left(\tilde{p}_{\mathcal{X}}, p_{\mathcal{X}}\right) = \frac{1}{2}\sum_{X \in \Omega^N} |\tilde{p}_{\mathcal{X}}(X) - p_{\mathcal{X}}(X)| < \frac{1}{2}\sum_{X \in \Omega^N} \epsilon = \frac{1}{2}|\Omega|^N \epsilon \tag{D.38}$$

Solving for $\delta = \frac{|\Omega|^N}{4\epsilon^2 n}$ gives $\epsilon = \sqrt{\frac{|\Omega|^N}{4\delta n}}$. Therefore, by Equation (D.37), with probability at least $1 - \delta$, we have

$$D_{\mathrm{TV}}\left(\tilde{p}_{\mathcal{X}}, p_{\mathcal{X}}\right) < \frac{1}{2}|\Omega|^N \epsilon = \sqrt{\frac{|\Omega|^{3N}}{16\delta n}}$$

□

E. Proof of Proposition 8: Modes of the strongly ferromagnetic Ising model

This section provides additional information for the discussion under Assumption 5 in Section 2.4.2.

Proposition 8 (Modes of the strongly ferromagnetic Ising model)**.** *On Ising model G in Equation (5) under Assumption 5, the high-probability regions $\mathcal{R}_1$ and $\mathcal{R}_{-1}$ defined in Equation (6) and Equation (7) satisfy*

1. $\forall \boldsymbol{x} \in \mathcal{R}_1$, $\forall \boldsymbol{y} \in \mathcal{R}_{-1}$, $\forall \boldsymbol{z} \in \{-1,1\}^N \backslash \mathcal{R}_1 \backslash \mathcal{R}_{-1}$, $p_G(\boldsymbol{x}) > p_G(\boldsymbol{y}) > e^{2J_0} p_G(\boldsymbol{z})$

2. *There exists a bijection* $f : \mathcal{R}_1 \mapsto \mathcal{R}_{-1}$ *such that* $\forall \boldsymbol{x} \in \mathcal{R}_1$, $p_G(\boldsymbol{x}) = e^{2h_G} p_G(f(\boldsymbol{x}))$

Proof. $\forall \boldsymbol{x} \in \mathcal{R}_1, \forall \boldsymbol{y} \in \mathcal{R}_{-1}$,

$$
\begin{aligned}
\frac{p_G(\boldsymbol{x})}{p_G(\boldsymbol{y})} &= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j \in C_G \subset [N]} J \mathbf{x}_i \mathbf{x}_j\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{y}_i + \sum_{i\neq j \in C_G \subset [N]} J \mathbf{y}_i \mathbf{y}_j\right)} \\
&= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{y}_i\right)} \quad (\text{since } \mathbf{x}_i\mathbf{x}_j = \mathbf{y}_i\mathbf{y}_j = 1 \, \forall \boldsymbol{x} \in \mathcal{R}_1, \forall \boldsymbol{y} \in \mathcal{R}_{-1}) \\
&= \frac{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{x}_i\right)}{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i \mathbf{y}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{y}_i\right)} \\
&= \frac{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{x}_i\right)}{\exp\left(-\sum_{i\in C_G} \boldsymbol{h}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{y}_i\right)} \quad (\text{since } \boldsymbol{x} \in \mathcal{R}_1, \boldsymbol{y} \in \mathcal{R}_{-1}) \\
&\geq \frac{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i - \sum_{i\notin C_G} |\boldsymbol{h}_i|\right)}{\exp\left(-\sum_{i\in C_G} \boldsymbol{h}_i + \sum_{i\notin C_G} |\boldsymbol{h}_i|\right)} \quad (\text{since } \mathbf{x}_i, \mathbf{y}_i \in \pm 1) \\
&= \exp\left(2\sum_{i\in C_G} \boldsymbol{h}_i - 2\sum_{i\notin C_G} |\boldsymbol{h}_i|\right) \\
&> \exp(0) = 1 \quad (\text{by Assumption 5})
\end{aligned}
$$

$\forall \boldsymbol{y} \in \mathcal{R}_{-1}, \forall \boldsymbol{z} \in \{-1,1\}^N \backslash \mathcal{R}_1 \backslash \mathcal{R}_{-1}$,

$$
\begin{aligned}
\frac{p_G(\boldsymbol{y})}{p_G(\boldsymbol{z})} &= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{y}_i + \sum_{i\neq j \in C_G \subset [N]} J \mathbf{y}_i \mathbf{y}_j\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{z}_i + \sum_{i\neq j \in C_G \subset [N]} J \mathbf{z}_i \mathbf{z}_j\right)} \\
&= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{y}_i + \sum_{i\neq j \in C_G \subset [N]} J\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{z}_i + \sum_{i\neq j \in C_G \subset [N]} J \mathbf{z}_i \mathbf{z}_j\right)} \quad (\text{since } \mathbf{y}_i\mathbf{y}_j = 1 \, \forall \boldsymbol{y} \in \mathcal{R}_{-1}) \\
&\geq \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{y}_i + \sum_{i\neq j \in C_G \subset [N]} J\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{z}_i + \sum_{i\neq j \in C_G \subset [N]} J - 2(|C_G|-1)J\right)} \\
&\quad (\text{since } \boldsymbol{z} \in \{-1,1\}^N \backslash \mathcal{R}_1 \backslash \mathcal{R}_{-1} \text{ and consider min edge number in bipartite graph}) \\
&= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{y}_i\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{z}_i - 2(|C_G|-1)J\right)} \geq \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{y}_i\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{z}_i - 2J\right)} \geq \frac{\exp\left(-\|\boldsymbol{h}\|_1\right)}{\exp\left(\|\boldsymbol{h}\|_1 - 2J\right)} \\
&= \exp\left(2(J - \|\boldsymbol{h}\|_1)\right) \geq \exp(2J_0) \quad (\text{by Assumption 5})
\end{aligned}
$$

For part 2, let $f : \mathcal{R}_1 \mapsto \mathcal{R}_{-1}$ be defined as

$$\forall \boldsymbol{x} \in \mathcal{R}_1, \quad f(\boldsymbol{x})_i = \begin{cases} -1, & \text{if } i \in C_G \\ \boldsymbol{x}_i, & \text{if } i \notin C_G \end{cases}$$

Let $\boldsymbol{w} \coloneqq f(\boldsymbol{x})$. Then,

$$\begin{aligned}
\frac{p_G(\boldsymbol{x})}{p_G(\boldsymbol{w})} &= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j \in C_G \subset [N]} J\mathbf{x}_i\mathbf{x}_j\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{w}_i + \sum_{i\neq j \in C_G \subset [N]} J\mathbf{w}_i\mathbf{w}_j\right)} \\
&= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i\right)}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{w}_i\right)} \quad (\text{since } \mathbf{x}_i\mathbf{x}_j = \mathbf{w}_i\mathbf{w}_j = 1\,\forall \boldsymbol{x} \in \mathcal{R}_1, \forall \boldsymbol{w} \in \mathcal{R}_{-1}) \\
&= \frac{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{x}_i\right)}{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i \mathbf{w}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{w}_i\right)} \\
&= \frac{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{x}_i\right)}{\exp\left(-\sum_{i\in C_G} \boldsymbol{h}_i + \sum_{i\notin C_G} \boldsymbol{h}_i \mathbf{w}_i\right)} \quad (\text{since } \boldsymbol{x} \in \mathcal{R}_1, \boldsymbol{w} \in \mathcal{R}_{-1}) \\
&= \frac{\exp\left(\sum_{i\in C_G} \boldsymbol{h}_i\right)}{\exp\left(-\sum_{i\in C_G} \boldsymbol{h}_i\right)} \quad (\text{since } \boldsymbol{w}_i = \boldsymbol{x}_i,\, \forall i \notin C_G) \\
&= \exp\left(2\sum_{i\in C_G} \boldsymbol{h}_i\right) \\
&= \exp\left(2h_G\right) \quad (\text{by Assumption } 5)
\end{aligned}$$

□

F. Proof of Proposition 4: k-Gibbs sampler can reach the mode fast

Proposition 4 (k-Gibbs sampler can reach the mode fast)**.** *On Ising model G in Equation (5) under Assumption 5, with any initial $\boldsymbol{X}^{(0)}$, $\forall \delta \in (0,1)$, with probability at least $1-\delta$, after $T := \left\lceil \log_{c_{\mathcal{R}_1}} \delta \right\rceil$ steps of k-**Gibbs sampler** (Definition 10) with $k \geq |C_G|$, we have $\{\boldsymbol{X}^{(t)} | t \in [T]\} \cap \mathcal{R}_1 \neq \emptyset$ in which the constant $c_{\mathcal{R}_1} \in (0,1)$, $c_{\mathcal{R}_1} := 1 - \frac{\binom{N-|C_G|}{k-|C_G|}}{\binom{N}{k}} \frac{e^{2(J_0+h_G)}}{e^{2(J_0+h_G)}+e^{2J_0}+2^{|C_G|}-2}$*

Proof. At any step, let K (with $|K|=k$) denote the set of coordinates to re-sample. We first consider the probability of $C_G \subset K$, which allows the whole C_G to be updated jointly:

$$\mathbb{P}\left\{C_G \subset K\right\} = \frac{\binom{N-|C_G|}{k-|C_G|}}{\binom{N}{k}} \tag{F.39}$$

$\forall t \in \mathbb{N}, \forall \boldsymbol{X}^{(t)} \in \{-1,1\}^N$, and $K \in [N]$ such that $|K|=k$ and $C_G \subset K$, consider $X_K^{(t+1)} \sim p_G(\cdot \mid X_{-K}^{(t)})$. There are three cases (a partition of all possibilities):

1. $\boldsymbol{X}^{(t+1)} \in \mathcal{R}_1$
2. $\boldsymbol{X}^{(t+1)} \in \mathcal{R}_{-1}$
3. $\boldsymbol{X}^{(t+1)} \in \{-1,1\}^N \backslash \mathcal{R}_1 \backslash \mathcal{R}_{-1}$

Then by Proposition 8 we show that Case 1 occurs with probability at least a (not arbitrarily small) constant,

$$\mathbb{P}\left\{\boldsymbol{X}^{(t+1)} \in \mathcal{R}_1\right\} = e^{2h_G}\mathbb{P}\left\{\boldsymbol{X}^{(t+1)} \in \mathcal{R}_{-1}\right\}$$

$$\frac{\mathbb{P}\left\{\boldsymbol{X}^{(t+1)} \in \mathcal{R}_{-1}\right\}}{\mathbb{P}\left\{\boldsymbol{X}^{(t+1)} \in \{-1,1\}^N \backslash \mathcal{R}_1 \backslash \mathcal{R}_{-1}\right\}} \geq e^{2J_0}\frac{|\mathcal{R}_{-1}|}{|\{-1,1\}^N \backslash \mathcal{R}_1 \backslash \mathcal{R}_{-1}|} = \frac{e^{2J_0}}{2^{|C_G|}-2}$$

Since the probabilities of the three cases sum up to 1,

$$\mathbb{P}\left\{\boldsymbol{X}^{(t+1)} \in \mathcal{R}_1\right\} \geq \frac{e^{2(J_0+h_G)}}{e^{2(J_0+h_G)}+e^{2J_0}+2^{|C_G|}-2}$$

Therefore, $\forall t \in \mathbb{N}, \forall \boldsymbol{X}^{(t)} \in \{-1,1\}^N$, combined with Equation (F.39),

$$\mathbb{P}\left\{\boldsymbol{X}^{(t+1)} \in \mathcal{R}_1\right\} \geq \mathbb{P}\left\{C_G \subset K, \boldsymbol{X}^{(t+1)} \in \mathcal{R}_1\right\} \geq \frac{\binom{N-|C_G|}{k-|C_G|}}{\binom{N}{k}} \frac{e^{2(J_0+h_G)}}{e^{2(J_0+h_G)}+e^{2J_0}+2^{|C_G|}-2} := 1 - c_{\mathcal{R}_1}$$

i.e. let constant $c_{\mathcal{R}_1}$ denote the above upper bound of $\mathbb{P}\left\{\boldsymbol{X}^{(t+1)} \notin \mathcal{R}_1\right\}$. Then

$$\mathbb{P}\left\{\{\boldsymbol{X}^{(t)} | t \in [T]\} \cap \mathcal{R}_1 = \emptyset\right\} \leq c_{\mathcal{R}_1}^T$$

Therefore, when $T \geq \log_{c_{\mathcal{R}_1}} \delta$,

$$\mathbb{P}\left\{\{\boldsymbol{X}^{(t)} | t \in [T]\} \cap \mathcal{R}_1 = \emptyset\right\} \leq \delta$$

□

G. Proof of Proposition 5 independent parallel sampling stuck in bad samples

Proposition 5 (Independent parallel sampling stuck in bad samples)**.** *On Ising model G in Equation (5) under Assumption 5, if the initial $\boldsymbol{X}^{(0)}$ is such that $\sum_{i\in C_G} \boldsymbol{X}_i^{(0)} \leq -2$, $\forall \delta \in (0,1)$, with probability at least $1-\delta$, after $T := \left\lfloor \frac{\delta}{2} \exp\left(c_{\textit{stuck}}\right) \right\rfloor$ steps of **independent parallel** (Definition 11), we have $\forall t \in [T], \sum_{i\in C_G} \boldsymbol{X}_i^{(t)} \leq -2$, in which $c_{\textit{stuck}} := \frac{2\left(-1+\frac{1-\exp(-2J_0)}{\exp(-2J_0)+1}\frac{|C_G|}{2}\right)^2}{|C_G|}$*

Proof. Suppose at step t, $\boldsymbol{X}^{(t)}$ is such that $\sum_{i\in C_G} \boldsymbol{X}_i^{(t)} \leq -2$ (satisfied at $t=0$), then

$$\forall j \in C_G, \sum_{i\in C_G, i\neq j} \boldsymbol{X}_i^{(t)} \leq -1 \tag{G.40}$$

Hence its next-step distribution $X_j^{(t+1)} \sim p(\cdot \mid X_{-\{j\}}^{(t)})$ satisfies

$$\begin{aligned}
\frac{\mathbb{P}\left\{X_j^{(t+1)}=1\right\}}{\mathbb{P}\left\{X_j^{(t+1)}=-1\right\}} &= \frac{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j \in C_G \subset [N]} J\mathbf{x}_i\mathbf{x}_j\right)|_{\mathbf{x}_j=1}}{\exp\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j \in C_G \subset [N]} J\mathbf{x}_i\mathbf{x}_j\right)|_{\mathbf{x}_j=-1}} \quad \text{(by definition in Equation (5))} \\
&= \frac{\exp\left(\boldsymbol{h}_j + \sum_{i\in C_G, i\neq j} J\mathbf{x}_i\right)}{\exp\left(-\boldsymbol{h}_j - \sum_{i\in C_G, i\neq j} J\mathbf{x}_i\right)} \quad \text{(canceling the same terms)} \\
&= \exp\left(2\boldsymbol{h}_j + 2J \sum_{i\in C_G, i\neq j} \mathbf{x}_i\right) \\
&\leq \exp\left(2\boldsymbol{h}_j - 2J\right) \quad \text{(by Equation (G.40))} \\
&\leq \exp\left(-2J_0\right) \quad \text{(by Assumption 5)}
\end{aligned}$$

Therefore

$$X_j^{(t+1)} = \begin{cases} 1, & \text{with prob } \leq \frac{\exp(-2J_0)}{\exp(-2J_0)+1} \\ -1, & \text{with prob } \geq \frac{1}{\exp(-2J_0)+1} \end{cases} \tag{G.41}$$

Denote

$$Y_j := \frac{X_j^{(t+1)}+1}{2} \tag{G.42}$$

Note that $\{Y_j \mid j \in [N]\}$ are independent Bernoulli random variables.

By Lemma 6, $\forall r > 0$, with probability at least $1 - 2e^{-\frac{2r^2}{|C_G|}}$,

$$\begin{aligned}
\frac{1}{|C_G|}\sum_{j\in C_G} Y_j &< \mathbb{E}_{j\in C_G}\left[Y_j\right] + \frac{r}{|C_G|} \quad \text{(by Hoeffding's inequality Lemma 6)} \\
&\leq \frac{\exp(-2J_0)}{\exp(-2J_0)+1} + \frac{r}{|C_G|} \quad \text{(by Equation (G.41) and definition of } Y_j \text{ in Equation (G.42))}
\end{aligned}$$

implying that with probability at least $1 - 2e^{-\frac{2r^2}{|C_G|}}$,

$$\frac{1}{|C_G|}\sum_{j\in C_G} X_j^{(t+1)} = 2\frac{1}{|C_G|}\sum_{j\in C_G} Y_j - 1 < 2\left(\frac{\exp(-2J_0)}{\exp(-2J_0)+1} + \frac{r}{|C_G|}\right) - 1$$

i.e.

$$\sum_{j\in C_G} X_j^{(t+1)} < \frac{\exp\left(-2J_0\right)-1}{\exp\left(-2J_0\right)+1}\left|C_G\right| + 2r$$

Setting RHS to -2 solves to

$$r = -1 + \frac{1-\exp\left(-2J_0\right)}{\exp\left(-2J_0\right)+1}\frac{\left|C_G\right|}{2}$$

Hence

$$\text{with probability at least } 1 - 2e^{-\frac{2\left(-1+\frac{1-\exp\left(-2J_0\right)}{\exp\left(-2J_0\right)+1}\frac{\left|C_G\right|}{2}\right)^2}{\left|C_G\right|}}, \quad \sum_{j\in C_G} X_j^{(t+1)} < -2 \tag{G.43}$$

By union bound, $\forall T \in \mathbb{N}_+$,

$$\text{with probability at least } 1 - 2Te^{-\frac{2\left(-1+\frac{1-\exp\left(-2J_0\right)}{\exp\left(-2J_0\right)+1}\frac{\left|C_G\right|}{2}\right)^2}{\left|C_G\right|}}, \quad \forall t \in [T], \quad \sum_{j\in C_G} X_j^{(t)} < -2 \tag{G.44}$$

Note that when $\sum_{j\in C_G} X_j^{(t)} < -2$, $\boldsymbol{X}^{(t)} \notin \mathcal{R}_1$.

Finally, aligning the probabilities: setting

$$2Te^{-\frac{2\left(-1+\frac{1-\exp\left(-2J_0\right)}{\exp\left(-2J_0\right)+1}\frac{\left|C_G\right|}{2}\right)^2}{\left|C_G\right|}} = \delta$$

solves to

$$T = \frac{\delta}{2}e^{\frac{2\left(-1+\frac{1-\exp\left(-2J_0\right)}{\exp\left(-2J_0\right)+1}\frac{\left|C_G\right|}{2}\right)^2}{\left|C_G\right|}}$$

□

H. Proof of Corollary 3: Separation between N-Gibbs sampler and independent parallel sampling

This section provides additional information for the discussion at the end of Section 2.4.2.

Assumption 6 (Strong interactions in Ising model)**.** *On Ising model G in Equation (5), for parameters $\delta \in (0,1)$ and $M \in \mathbb{N}_+$,*

$$\begin{aligned} |C_G| &\geq 8\left(1 + \ln \frac{4M}{\delta}\right) \\ h_G &\geq \frac{1}{2} \ln \frac{2(2-\delta)}{\delta} \\ J_0 &\geq \frac{1}{2} |C_G| \ln 2 \end{aligned}$$

Corollary 3 (Separation between N-Gibbs sampler and independent parallel sampling)**.** *On Ising model G in Equation (5) under Assumption 5, $\forall \delta \in (0,1)$, $\forall M \in \mathbb{N}_+$, If G additionally satisfies Assumption 6 and the initial $\boldsymbol{X}^{(0)}$ is such that $\sum_{i \in C_G} \boldsymbol{X}_i^{(0)} \leq -2$, then with probability at least $1-\delta$,*

1. *Running N-**Gibbs sampler**: $\boldsymbol{X}_{N.c.w.}^{(1)} \in \mathcal{R}_1$, and*
2. *Running **independent parallel**: $\{\boldsymbol{X}_{indep}^{(t)} | t \in [M]\} \cap \mathcal{R}_1 = \emptyset$*

Proof. Under the given conditions, with N-**Gibbs sampler**, by Proposition 4,

$$\text{with probability at least } 1 - \frac{\delta}{2}, \quad \{\boldsymbol{X}_{N.c.w.}^{(t)} | t \in [\left\lceil \log_{c_{\mathcal{R}_1}} \frac{\delta}{2} \right\rceil]\} \cap \mathcal{R}_1 \neq \emptyset \tag{H.45}$$

in which the constant

$$c_{\mathcal{R}_1} := 1 - \frac{\binom{N-|C_G|}{N-|C_G|}}{\binom{N}{N}} \frac{e^{2(J_0+h_G)}}{e^{2(J_0+h_G)} + e^{2J_0} + 2^{|C_G|} - 2} = 1 - \frac{e^{2(J_0+h_G)}}{e^{2(J_0+h_G)} + e^{2J_0} + 2^{|C_G|} - 2} \tag{H.46}$$

Applying Assumption 6 to bound parts of the RHS:

$$\frac{e^{2J_0}}{e^{2(J_0+h_G)}} = e^{-2h_G} \leq e^{-\ln \frac{2(2-\delta)}{\delta}} = \frac{\delta}{2(2-\delta)}$$

$$\frac{2^{|C_G|} - 2}{e^{2(J_0+h_G)}} \leq \frac{2^{|C_G|}}{e^{2(J_0+h_G)}} \leq \frac{2^{|C_G|}}{e^{|C_G| \ln 2 + \ln \frac{2(2-\delta)}{\delta}}} = \frac{2^{|C_G|}}{2^{|C_G|} \frac{2(2-\delta)}{\delta}} = \frac{1}{\frac{2(2-\delta)}{\delta}} = \frac{\delta}{2(2-\delta)}$$

Taking the sum:

$$\frac{e^{2J_0} + 2^{|C_G|} - 2}{e^{2(J_0+h_G)}} \leq \frac{\delta}{2-\delta}$$

Adding 1 to both sides:

$$\frac{e^{2(J_0+h_G)} + e^{2J_0} + 2^{|C_G|} - 2}{e^{2(J_0+h_G)}} \leq \frac{2}{2-\delta}$$

Taking the inverse:

$$\frac{e^{2(J_0+h_G)}}{e^{2(J_0+h_G)} + e^{2J_0} + 2^{|C_G|} - 2} \geq \frac{2-\delta}{2}$$

Plugging to Equation (H.46):

$$c_{\mathcal{R}_1} \leq 1 - \frac{2-\delta}{2} = \frac{\delta}{2}$$

Plugging into Equation (H.45):

$$\text{with probability at least } 1-\frac{\delta}{2}, \quad \{\boldsymbol{X}_{N.c.w.}^{(t)} | t \in [1]\} \cap \mathcal{R}_1 \neq \emptyset \tag{H.47}$$

On the other hand, with **independent parallel**, by Proposition 5,

$$\text{with probability at least } 1-\frac{\delta}{2}, \quad \{\boldsymbol{X}_{indep}^{(t)} | t \in [\left\lfloor \frac{\delta}{4} \exp\left(c_{\text{stuck}}\right) \right\rfloor]\} \cap \mathcal{R}_1 = \emptyset \tag{H.48}$$

in which the constant

$$c_{\text{stuck}} := \frac{2\left(-1+\frac{1-\exp\left(-2J_0\right)}{\exp\left(-2J_0\right)+1}\frac{|C_G|}{2}\right)^2}{|C_G|} \tag{H.49}$$

Applying Assumption 6 to bound parts of the RHS:

$$\frac{1-\exp\left(-2J_0\right)}{\exp\left(-2J_0\right)+1} \geq \frac{1}{2}$$

Plugging into Equation (H.49):

$$\begin{aligned}
c_{\text{stuck}} &\geq \frac{2\left(-1+\frac{1}{2}\frac{|C_G|}{2}\right)^2}{|C_G|} \\
&= \frac{2\left(1-\frac{|C_G|}{2}+\frac{|C_G|^2}{4}\right)}{|C_G|} \\
&\geq -1+\frac{|C_G|}{8} \\
&\geq -1+\left(1+\ln\frac{4M}{\delta}\right) \quad \text{(by Assumption 6)} \\
&= \ln\frac{4M}{\delta}
\end{aligned}$$

Plugging into Equation (H.48):

$$\text{with probability at least } 1-\frac{\delta}{2}, \quad \{\boldsymbol{X}_{indep}^{(t)} | t \in [\left\lfloor \frac{\delta}{4} \cdot \frac{4M}{\delta} \right\rfloor] = [M]\} \cap \mathcal{R}_1 = \emptyset \tag{H.50}$$

By union bound, with probability at least $1-\delta$, both Equation (H.47) and Equation (H.50) hold.

□

I. Background and proofs of Proposition 2 and Proposition 3: on the expressive power of Transformers for implementing sequence-to-sequence Markov chains in parallel

I.1. Technical setup and proofs

Background: Transformer network architecture. The transformer architecture (Vaswani et al., 2017) is a critical building block of many leading approaches to language modeling (Devlin et al., 2019; Brown et al., 2020). We refer the readers to these works for more details on the empirical promise that Transformer-based models have demonstrated. For theoretical understanding of Transformers, we refer the readers to prior works on their representational power (Yun et al., 2020; Yao et al., 2021; Liu et al., 2023; Zhao et al., 2023), statistical sample complexity (Wei et al., 2021; Edelman et al., 2022), optimization process (Lu et al., 2021; Jelassi et al., 2022; Li et al., 2023), and interpretability (Wen et al., 2023), and references cited therein.

Mathematical setup. In the following we adapt and use the mathematical notations for the Transformer network architecture in Yun et al. (2020) and Li et al. (2023).

For each position of an input sequence (N tokens), use a d-dimensional *positional embedding* to represent that *position*, and use a d-dimensional *token embedding* for the *content* at that position. Hence, for the input sequence, both the token embeddings $\boldsymbol{E}$ and the positional embeddings $\boldsymbol{P}$ are matrices in $\mathbb{R}^{d\times N}$. Following empirical convention, let the input to the Transformer be

$$\boldsymbol{X} \coloneqq \boldsymbol{E} + \boldsymbol{P}$$

A *Transformer block* $t^{h,m,r}$ (with h heads, head size m, and feed-forward hidden layer size r) is defined as

$$t^{h,m,r}(\boldsymbol{X}) \coloneqq \text{Attn}(\boldsymbol{X}) + \boldsymbol{W}_2 \cdot \text{ReLU}(\boldsymbol{W}_1 \cdot \text{Attn}(\boldsymbol{X}) + \boldsymbol{b}_1 \boldsymbol{1}_n^T) + \boldsymbol{b}_2 \boldsymbol{1}_n^T \tag{I.51}$$

where

$$\text{Attn}(\boldsymbol{X}) \coloneqq \boldsymbol{X} + \sum_{i=1}^{h} \boldsymbol{W}_O^i \boldsymbol{W}_V^i \boldsymbol{X} \cdot \sigma[(\boldsymbol{W}_K^i \boldsymbol{X})^T \boldsymbol{W}_Q^i \boldsymbol{X}] \tag{I.52}$$

where the weight parameters $\boldsymbol{W}_O^i \in \mathbb{R}^{d\times m}$, $\boldsymbol{W}_V^i, \boldsymbol{W}_K^i, \boldsymbol{W}_Q^i \in \mathbb{R}^{m\times d}$, $\boldsymbol{W}_2 \in \mathbb{R}^{d\times r}$, $\boldsymbol{W}_1 \in \mathbb{R}^{r\times d}$, $\boldsymbol{b}_2 \in \mathbb{R}^d$, $\boldsymbol{b}_1 \in \mathbb{R}^r$, and

$$\sigma : \mathbb{R}^{N_1 \times N_2} \mapsto (0,1)^{N_1 \times N_2}$$

is the column-wise softmax operation, such that

$$\sigma(A)_{ij} = \frac{\exp(A_{ij})}{\sum_{l=1}^{N} \exp(A_{lj})} \tag{I.53}$$

Finally, a Transformer is a composition of Transformer blocks:

$$\mathcal{T} \coloneqq \{g : \mathbb{R}^{d\times N} \to \mathbb{R}^{d\times N} \mid g \text{ is a composition of Transformer blocks } t^{h,m,r}\text{'s}\}. \tag{I.54}$$

and its output $\mathcal{T}(\boldsymbol{X}) \in \mathbb{R}^{d\times N}$ goes through a final affine transform and softmax (Equation (I.53)) to predict a distribution over tokens, for all positions

$$\mathcal{T}_{\text{pred}}(\boldsymbol{X}) \coloneqq \sigma\left(\boldsymbol{W}^{\text{pred}} \mathcal{T}(\boldsymbol{X}) + \boldsymbol{b}^{\text{pred}}\right) \in (0,1)^{|\Omega|\times N} \tag{I.55}$$

where $\boldsymbol{W}^{\text{pred}} \in \mathbb{R}^{|\Omega|\times d}$ and $\boldsymbol{b}^{\text{pred}} \in \mathbb{R}^{|\Omega|}$ are the prediction head weights and biases. Ω is the vocabulary of tokens.

For each position j, the predicted token τ_j is sampled from the predicted distribution $\mathcal{T}_{\text{pred}}(\boldsymbol{X})_{:,j}$ *independently* with other positions

$$\tau_j \sim \texttt{sample}(\mathcal{T}_{\text{pred}}(\boldsymbol{X})_{:,j}) \quad j \in [N] \tag{I.56}$$

where `sample` can be the standard sampling algorithm for multinomial distributions, or truncating the low-probability tail (Holtzman et al., 2020), or more conservatively, argmax sampling.

Yun et al. (2020) proved the following result on the expressivity of the Transformer network architecture:

Lemma 9 (Universal approximation by Transformers, informal (Yun et al., 2020))**.** *Let $1 \leq p < \infty$ and $\epsilon > 0$, then for any compact set $\mathcal{D} \subset \mathbb{R}^{d\times n}$, for any given function $f : \mathcal{D} \mapsto \mathbb{R}^{d\times n}$, there exists a Transformer network $g \in \mathcal{T}^{2,1,4}$ of $O(N\left(\frac{1}{\delta}\right)^{dN})$ layers such that*

$$\left(\int \|f(\boldsymbol{X}) - g(\boldsymbol{X})\|_p^p \, d\boldsymbol{X}\right)^{1/p} \leq \epsilon$$

in which δ is the smallest real number such that $\forall \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{d\times n}$, if $\|\boldsymbol{X} - \boldsymbol{Y}\|_\infty < \delta$, then $\left\|f(\boldsymbol{X}) - f(\boldsymbol{Y})\right\|_p < \epsilon$. Moreover, the bound on the size of the constructed Transformer is asymptotically tight.

Lemma 10 (Transformers can simulate parallel solution to automata, informal (Liu et al., 2023))**.** *Transformers can simulate the length-T output of all semiautomata with states Q, input alphabet Σ, and transition function $\delta : Q \times \Sigma \mapsto Q$. Moreover, the size of the simulating Transformer has depth $O(\log T)$, embedding dimension $O(|Q|)$, attention width $O(|Q|)$, and MLP width $O(|Q|^2)$.*

Remark 5. *Lemma 10 gives a more compact construction than a direct implication of more general universal approximation results Lemma 9 for Transformers.*

A direct corollary is Proposition 2:

Proposition 2 (informal)**.** *Transformers (with sufficient depth and width) can implement any number of transitions of any deterministic Markov Chain over sequences in Ω^N.*

Informal proof sketch. When each transition of a Markov chain is deterministic, i.e. if the next state distribution from any state is always a delta function, then the Markov chain reduces to a deterministic finite state automata, with states Ω^N, length N.

Applying Lemma 10, we get Transformers can simulate length-T output of this automata with depth $O(\log T)$, embedding dimension $O(|\Omega|^N)$, attention width $O(|\Omega|^N)$, and MLP width $O(|\Omega|^{2N})$. □

Proposition 3 (informal)**.** *The class of Markov chains over sequences in Ω^N implementable by (sufficiently wide and deep) Transformers is those whose next-state transition probability distributions are product distributions over the positions, conditioned on the current state.*

Informal proof sketch. The statement involves both a positive result and a negative result.

Positive: if the transition probability distribution is a product distribution conditioned on the current state, then the task of representing a Markov chain can be reduced to universally approximating a continuous function which maps all sequences to the correct logits $\boldsymbol{W}^{\text{pred}}\mathcal{T}(\boldsymbol{X}) + \boldsymbol{b}^{\text{pred}}$ in Equation (I.55), such that after softmax (Equation (I.53)) these logits produce the correct marginal distribution at each position. This is achievable by the construction in Lemma 9.

Negative: if the transition probability distribution is *not* a product distribution conditioned on the current state, then note that the sampling operations (Equation (I.56)) at positions j_1 and j_2 are *independent*, so Transformers cannot implement such Markov chains. □

I.2. Connection to prior works in GMLM

Among existing language generation approaches via iterative refinement, Wang & Cho (2019) uses 1-Gibbs sampler. The approaches in Ghazvininejad et al. (2019); Savinov et al. (2022) and our experiments do not closely fall into either of *independent parallel* (Equation (4)) or the k-Gibbs sampler (Equation (3)) in Section 2.4. See Remark 6 for technical details.

Moreover, these approaches train models to learn the parameterized conditional distributions, which empirically may not admit a consistent joint distribution (Young & You, 2022; Torroba Hennigen & Kim, 2023).

To formally reason about the iterative refinement process in GMLMs, in Section 2.4 we relax some of these limitations to focus on several underlying theoretical obstacles that these methods face.

Remark 6 (Technical details in theoretically formalizing GMLM architectures)**.** *By Proposition 3, the sampling process in Ghazvininejad et al. (2019); Savinov et al. (2022) and our experiments are different from N-Gibbs sampler. Moreover, the sampling process is also different from* ***independent parallel*** *(Gibbs sampler 11): note that* ***independent parallel*** *strictly* *freezes all* $X^{(t)}_{-\{i\}}$ *when sampling*

$$X_i^{(t+1)} \sim p(\cdot \mid X^{(t)}_{-\{i\}})$$

whereas in Savinov et al. (2022) *and our experiments, the model is trained to update all positions in parallel, which implies a different groundtruth next-iteration token distribution compared with* $p(\cdot \mid X^{(t)}_{-\{i\}})$. *In other words, although the updates are conditionally independent given the current state, the update probabilities are not trained to model* $p(\cdot \mid X^{(t)}_{-\{i\}})$.

Mechanistically, Savinov et al. (2022) *and our models in principle can take certain inter-position dependency into consideration (which* ***independent parallel*** *cannot): for example, in layer* L*, position* i *can attend to* [19] *other positions e.g.* j *in the layer-*$(L-1)$ *representations. This enables the layer-*L *computation at position* i *to be conditioned upon the intermediate representations at position* j*, which are* not independent *from the final prediction at position* j.

Ghazvininejad et al. (2019) *can be understood as predicting the subset of masked indices* K *in each update. The extent to which each update incorporates dependency between masked positions depends on implementation details: for example, whether attention masks are added to prevent any masked position from receiving attention.*

[19] via Transformer attention Equation (I.52)

J. Regularity conditions for asymptotic behavior of M-estimators

In the limite of infinite samples, M-estimators (in particular, maximum likelihood and the estimators in Definitions 2 and 7) converge in distribution to a normal distribution, under mild regularity conditions:

Lemma 11 ((Van der Vaart, 2000), Theorem 5.23; statement adapted from Qin & Risteski (2023))**.** *Consider a loss $L : \Theta \mapsto \mathbb{R}$, such that $L(\theta) = \mathbb{E}_p[\ell_\theta(x)]$ for $l_\theta : \mathcal{X} \mapsto \mathbb{R}$. Let Θ^* be the set of global minima of L, that is*

$$\Theta^* = \{\theta^* : L(\theta^*) = \min_{\theta \in \Theta} L(\theta)\}$$

Suppose the following conditions are met:

- *(Gradient bounds on l_θ) The map $\theta \mapsto l_\theta(x)$ is measurable and differentiable at every $\theta^* \in \Theta^*$ for p-almost every x. Furthermore, there exists a function $B(x)$, s.t. $\mathbb{E}\left[B(x)^2\right] < \infty$ and for every θ_1, θ_2 near θ^*, we have:*

$$|l_{\theta_1}(x) - l_{\theta_2}(x)| < B(x)\|\theta_1 - \theta_2\|_2$$

- *(Twice-differentiability of L) $L(\theta)$ is twice-differentiable at every $\theta^* \in \Theta^*$ with Hessian $\nabla^2_\theta L(\theta^*)$, and furthermore $\nabla^2_\theta L(\theta^*) \succ 0$.*
- *(Uniform law of large numbers) The loss L satisfies a uniform law of large numbers, that is*

$$\sup_{\theta \in \Theta} \left| \hat{\mathbb{E}}[l_\theta(x)] - L(\theta) \right| \xrightarrow{p} 0$$

- *(Realizability) The data distribution p satisfies: $\exists \theta^* \in \Theta$ such that $p_{\theta^*} = p$.*

Then, for every $\theta^* \in \Theta^*$, and every sufficiently small neighborhood S of θ^*, there exists a sufficiently large n, such that there is a unique minimizer $\hat{\theta}_n$ of $\hat{\mathbb{E}}[l_\theta(x)]$ in S. Furthermore, $\hat{\theta}_n$ satisfies:

$$\sqrt{n}(\hat{\theta}_n - \theta^*) \xrightarrow{d} \mathcal{N}\left(0, (\nabla^2_\theta L(\theta^*))^{-1} Cov(\nabla_\theta \ell(\theta^*; x))(\nabla^2_\theta L(\theta^*))^{-1}\right)$$

K. Convexity of pseudolikelihood for Ising models

Here, we expand on a comment in Section 2. We show that for a classic parameteric class of distributions (namely, Ising models — which appear also in Section 2.4) the k-MPLE loss is in fact *convex*. This is a known fact which has been used to design (provably) efficient algorithms for learning bounded-degree Ising models (Ravikumar et al., 2010; Vuffray et al., 2016), and is just included for completeness. Recall the definition of Ising models:

Ising models. For random variables $\boldsymbol{X} = \{X_i \in \{-1, 1\} : i \in [N]\}$, an Ising model with parameters $\boldsymbol{J} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{h} \in \mathbb{R}^N$ has joint distribution

$$p(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp\big(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j\in[N]} \boldsymbol{J}_{ij} \mathbf{x}_i \mathbf{x}_j\big), \tag{K.57}$$

in which Z is the partition function.

Proposition 9 (Fitting an Ising model over the conditional distributions is convex)**.** *When p_θ is an Ising model (Equation (K.57)), i.e. $\theta = (\boldsymbol{J}, \boldsymbol{h})$, the weighted pseudolikelihood objective (Definition 2) is convex.*

Proof. When p_θ is an Ising model (Equation (K.57)), we have:

$$\begin{aligned} -\ln p_\theta(\mathbf{x}_K|\mathbf{x}_{-K}) &= -\ln \frac{\exp\big(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j\in[N]} \boldsymbol{J}_{ij} \mathbf{x}_i \mathbf{x}_j\big)}{Z(\mathbf{x}_{-K})} \\ &= -\left(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j\in[N]} \boldsymbol{J}_{ij} \mathbf{x}_i \mathbf{x}_j\right) + \ln Z(\mathbf{x}_{-K}) \end{aligned}$$

in which the denominator

$$\begin{aligned} Z(\mathbf{x}_{-K}) = \sum_{X_K\in\{-1,1\}^{|K|}} \exp\Bigg(&\sum_{i\in K} \boldsymbol{h}_i X_K + \sum_{i\in[N]\setminus K} \boldsymbol{h}_i \mathbf{x}_i \\ &+ \sum_{i\neq j\in[K]} \boldsymbol{J}_{ij} X_i X_j + \sum_{i\in K, j\in[N]\setminus K} \boldsymbol{J}_{ij} X_i \mathbf{x}_j + \sum_{i\neq j\in[N]\setminus K} \boldsymbol{J}_{ij} \mathbf{x}_i \mathbf{x}_j\Bigg) \end{aligned}$$

Note that $-\Big(\sum_{i\in[N]} \boldsymbol{h}_i \mathbf{x}_i + \sum_{i\neq j\in[N]} \boldsymbol{J}_{ij} \mathbf{x}_i \mathbf{x}_j\Big)$ is linear in $(\mathbf{h}, \mathbf{J})$ and $\ln Z(\mathbf{x}_{-K})$ is convex in (h, J), so $-\ln p_\theta(X_K = \mathbf{x}_K | X_{-K} = \mathbf{x}_{-K})$ is convex in $(\mathbf{h}, \mathbf{J})$, which completes the last piece of the proof.

□

L. Additional experimental details

L.1. Training and inference approach for PaDIR

We provide a formal description of the training and inference strategy outlined in Section 3.2.

L.1.1. Inference

An input sequence X^{source} first goes through the encoder $f^{\text{enc}}_{\theta_e}$ (parameterized by θ_e) to produce the hidden representation h:

$$h = f^{\text{enc}}_{\theta_e}(X^{\text{source}})$$

A length predictor $f^{\text{len}}_{\theta_l}$ (parameterized by θ_l) takes h and predicts B_l most likely target lengths, where $B_l \in \mathbb{N}_+$ (beam size for length prediction) is an inference-time hyperparameter.

For each predicted length N, an initial hypothesis target sequence $X^{(0)} = X^{(0)}_1 \cdots X^{(0)}_N$ in which each $X^{(0)}_i$ can be a `[MASK]` token, or chosen uniformly randomly from the vocabulary of tokens.

For each decoder step $t \in 1 \cdots T$, the decoder $f^{\text{dec}}_{\theta_d}$ (parameterized by θ_d) takes two inputs: h and $X^{(t)}_{1\cdots N}$, and refines the hypothesis target sequence to $X^{(t+1)}_{1\cdots N}$, using one forward pass:

$$X^{(t+1)}_{1\cdots N} = f^{\text{dec}}_{\theta_d}(X^{(t)}_{1\cdots N}, h) \tag{L.58}$$

where $T \in \mathbb{N}_+$ (number of refinement steps) is an inference-time hyperparameter, and we can stop early if $X^{(t+1)} = X^{(t)}$.

L.1.2. Training

One-stage training Given source sequence X^{source} and target sequence X^{target} in the supervised training data $\mathcal{D}_{\text{train}}$, we use a preprocessing rule to create the initial hypothesis target sequence $X^{(0)}$. [20] The training objective is

$$L^{(1)} = \sum_{X^{\text{source}}, X^{\text{target}} \in \mathcal{D}_{\text{train}}} l(f^{\text{dec}}_{\theta_d}(X^{(0)}, f^{\text{enc}}_{\theta_e}(X^{\text{source}})) \tag{L.59}$$

where l is the cross-entropy loss applied to each position.

Multi-stage training One limitation of the one-stage training is that the inference situation is *out-of-distribution*: when decoder step $t > 1$, the model needs to refine its own predictions in step $t - 1$, which is not reflected in the training objective. Therefore, we use the multi-stage training objective (Ghazvininejad et al., 2020; Savinov et al., 2022): $L^{(S)} = \frac{1}{S}\sum_{s\in[S]} L^{(s)}$ where S is the number of training stages, and $L^{(s)} = \sum_{X^{\text{source}}, X^{\text{target}} \in \mathcal{D}_{\text{train}}} l(f^{\text{dec}}_{\theta_d}(X^{(s-1)}, f^{\text{enc}}_{\theta_e}(X^{\text{source}}))$

L.2. Details of training recipe

We provide more details to Section 3.3.

Model training We use Transformer encoder-decoder with size similar to Transformer-Base (Vaswani et al., 2017) and T5-Small-1.0 (Raffel et al., 2020): 6 encoder and decoder layers, 8 attention heads, 512 embedding dimensions and 2048 FFN hidden dim. We add a positional attention mechanism (Gu et al., 2018; Kreutzer et al., 2020) in each Transformer layer and use learnt positional embeddings. The total number of parameters is 67M. We initialize model parameters randomly and train using a batch size of 2048 for 500k iterations, with a 10% dropout rate, 15% unmasking rate [21] and 2 training stages. The optimizer is AdaFactor (Shazeer & Stern, 2018), with default T5X hyperparameters (Roberts et al., 2022). The learning rate peaks at 0.003 with a linear rampup for 10k steps followed by cosine decay, from and to a minimum value of $1e - 5$. Unlike most prior work, we do not use a remasking schedule; [22] we simply remask token-level stutter (i.e., consecutive

[20] Each position in $X^{(0)}$ may contain a `[MASK]` token, a random token, or the correct token in X^{source}, dependning on the preprocessing rule.

[21] This means, in Equation (L.59), 15% of the tokens in $X^{(0)}$ are the correct tokens in X^{target}, and the remaining 85% are random tokens in the vocabulary.

[22] We experimented with various remasking schedules but the results were not visibly affected.

repeated tokens) across iterations and drop repeated tokens after the final iteration. As commonly done, we distill our models by training on the output of an autoregressive model. For simplicity, we use the Google Cloud Translation API to generate this distillation data.

Datasets We evaluate our models on machine translation benchmarks commonly used in the non-autoregressive modeling literature. We conduct experiments on both directions of three WMT datasets: WMT14 DE↔EN ($4.5M$ examples) (Bojar et al., 2014), WMT16 RO↔EN ($610k$ examples) (Bojar et al., 2016) and WMT17 ZH↔EN ($20M$ examples) (Bojar et al., 2017). We load the data from the `tensorflow_datasets` library and do not apply any preprocessing other than sentence piece tokenization ((Kudo & Richardson, 2018)). Bilingual vocabularies of 32k tokens are created using the training sets of each language pair.

L.3. Discussion on modeling and metrics

This section provides additional information about Section 3.3.

Remark 7. *In principle, following a similar paradigm, a non-autoregressive* decoder-only *architecture is also possible. In this work we use encoder-decoder for two reasons: (1) Efficiency: in the iterative refinement process of the hypothesis target sequence, each forward pass only involves the decoder, but not the encoder. (2) Benchmarking: the encoder-decoder design is closer to a series of prior works, allowing for more informative comparison on benchmarks.*

We measure BLEU (Papineni et al., 2002) using the SacreBLEU implementation (Post, 2018) with language appropriate tokenizers [23]. For the same model, SacreBLEU on average reports a lower score than BLEU (e.g. see (Savinov et al., 2022)). Unfortunately, this does not allow a direct comparison with most of the existing literature. This is a deliberate choice since it has been shown that subtle differences in preprocessing can significantly impact metrics (Schmidt et al., 2022), making comparisons error prone, and SacreBLEU is the recommended metric in Post (2018). Furthermore, common preprocessing steps (lowercasing, separating punctuation, stripping diacritics, etc.) may artificially inflate scores while not being fully reversible, as such preventing real-world uses for such models.

For our experiments in Section 3.4, there are other error modes connected to the challenge of modeling target-side dependency, but they are more ambiguous for measuring and exactly locating. We do not aim to develop decoding algorithms tailored to just reducing stuttering rate. (After all, stuttering can be easily removed by rule-based postprocessing.) Instead, the above are general-purpose hypotheses which are potentially also predictive of other (more complex) failure modes related to target-side dependency.

L.4. Quantitative experimental results on machine translation task

This section provides quantitative evaluation results for Section 3.3.

Table 1. Test SacreBLEU scores on three WMT datasets. We report scores without any preprocessing. Our AR baselines are trained on the distilled dataset for a fair comparison. The 'Steps' column indicates the number of decoding iterations. The '# Hyp.' column denotes the number of hypotheses decoded in parallel (beam size for AR models and top_k predicted lengths for NAR models).

			WMT14		WMT16		WMT17	
Model	# Hyp.	Steps	DE→EN	EN→DE	RO→EN	EN→RO	ZH→EN	EN→ZH
AR Baselines	5	N	33.50	29.54	34.89	29.75	27.59	33.94
PaDIR	5	4	33.49	28.61	33.98	28.98	26.47	32.59
	5	10	33.63	28.58	33.99	28.97	26.54	32.68

[23] For public reproducibility: SacreBLEU signatures: BLEU+c.mixed+#.1+s.exp+tok.zh+v.1.3.0 for Chinese and BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.3.0 for other languages.

Table 2. Test BLEU scores on three WMT datasets for baselines. Note that they use different BLEU implementations and sometimes additional preprocessing than the results reported for our approach. We include results for our PaDIR under the T5X default BLEU score (SacreBLEU tok_intl). As we remarked in Appendix L.3, these different BLEU implementations may not be directly comparable.

			WMT14		WMT16		WMT17	
Model	# Hyp.	Steps	DE→EN	EN→DE	RO→EN	EN→RO	ZH→EN	EN→ZH
DisCo AR Baselines	5	N	31.71	28.60	34.46	34.16	24.65	35.01
CMLM	5	4	30.75	26.73	33.02	33.67	22.57	33.58
	5	10	31.24	27.39	33.67	33.33	23.76	34.24
DisCo Easy-First	5	3-6	31.31	27.34	33.25	33.22	23.83	34.63
SUNDAE Stochastic	16	4	32.10	27.94	-	-	-	-
	16	10	32.29	28.33	-	-	-	-
PaDIR	5	4	34.17	29.49	34.55	29.57	27.18	32.59
	5	10	34.33	29.48	34.57	29.56	27.25	32.60

Table 3. Test BLEURT scores on three WMT datasets for our models.

			WMT14		WMT16		WMT17	
Model	# Hyp.	Steps	DE→EN	EN→DE	RO→EN	EN→RO	ZH→EN	EN→ZH
AR Baselines	5	N	73.55	74.97	67.23	71.76	68.14	65.71
PaDIR	5	4	71.26	72.08	65.90	70.23	65.16	63.95
	5	10	71.82	73.28	66.09	70.49	66.19	64.30

L.5. Quantifying dependency via attention scores

We report quantitative results of the investigations introduced in Section 3.4.

Table 4. Stuttering positions have comparable average last-layer self-attentions compared with non-stuttering adjacent positions. For each pair of adjacent positions in the generated sequence: (1) the 'self-attention scores' include both directions ; (2) The column 'min' denotes only including the minimum among such score over all attention heads, and likewise for 'avg' and 'max'; (3) the entries are mean ± standard deviation; (4) $\mathbb{P}\{\text{top-}k \text{ overlap}\}$ denotes the chances that the self-attention distribution at one position includes the other position among its top-k "most attended to" positions.

	self-attention scores			$\mathbb{P}\{\text{top-}k \text{ overlap}\}$	
stutter	min	avg	max	$k = 1$	$k = 2$
yes	0.0004 ± 0.0007	0.032 ± 0.023	0.16 ± 0.11	0.20	0.39
no	0.0005 ± 0.0007	0.033 ± 0.025	0.17 ± 0.12	0.17	0.37

Table 5. Stuttering positions on average have more similar last-layer cross-attentions than non-stuttering adjacent positions. For each pair of adjacent positions in the generated sequence: (1) the 'total variation distance' and 'cosine distance' (both have range $[0, 1]$) are taken for the two corresponding cross-attention distributions; (2) The column 'min' denotes only including the minimum among such distance over all attention heads, and likewise for 'avg' and 'max'; (3) the entries are mean ± standard deviation; (4) $\mathbb{P}\{\text{top-}k \text{ overlap}\}$ denotes the chances that the two cross-attention distributions overlap in terms of their top-k "most attended to" source positions.

	total variation distance			cosine distance			$\mathbb{P}\{\text{top-}k \text{ overlap}\}$	
stutter	min	avg	max	min	avg	max	$k = 1$	$k = 2$
yes	0.06 ± 0.05	0.13 ± 0.09	0.23 ± 0.15	0.01 ± 0.01	0.10 ± 0.06	0.25 ± 0.11	0.57	0.89
no	0.11 ± 0.10	0.23 ± 0.14	0.35 ± 0.18	0.04 ± 0.08	0.20 ± 0.11	0.38 ± 0.12	0.40	0.81

L.6. Masking more is statistically better for learning synthetic Ising models

We show our observations in Section 3.1 and Figure 1 are robust to the shape of the groundtruth Ising model distribution: under a much more peaky groundtruth distribution (with 2 modes), it still holds that with the same training data size, larger k leads to lower error. We plot the results in Figure 2.

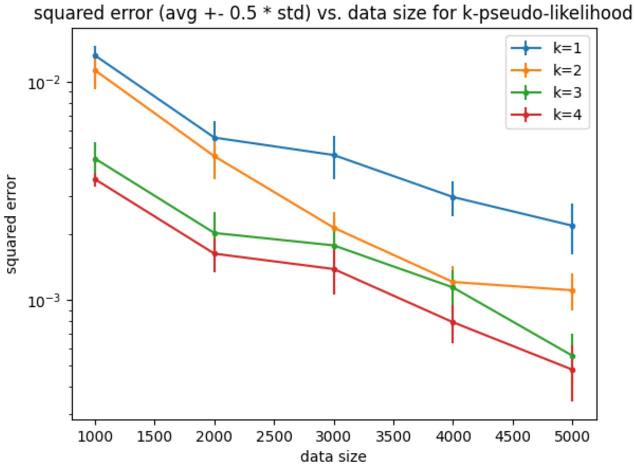

Figure 1. Average squared error in parameter estimation for fitting an Ising model on data generated by a groundtruth Ising model ($N = |C_G| = 4, J = 0.05, h_i = 0$ in Equation (5)) using the k-pseudolikelihood objective optimized by gradient descent. Error bars denote ± 0.5 * stdev for 10 repetitions of the experiment.

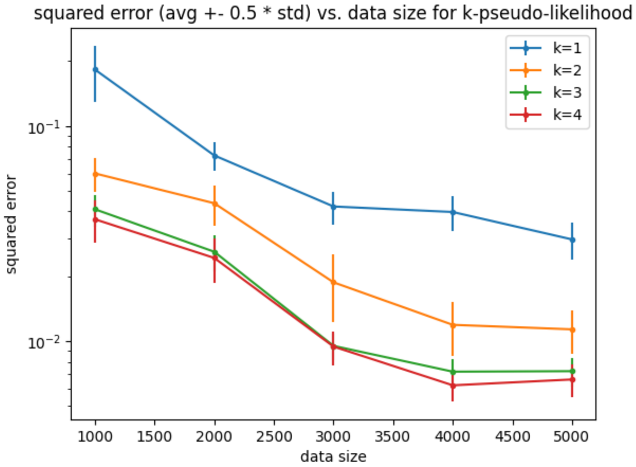

Figure 2. Average squared error in parameter estimation for fitting an Ising model on data generated by a groundtruth Ising model ($N = |C_G| = 4, J = 0.3, h_i = 0$ in Equation (5)) using the k-pseudolikelihood objective optimized by gradient descent. Error bars denote ± 0.5 * stdev for 10 repetitions of the experiment.

L.7. Markov Chains with dependent transitions can be (much) faster in sampling Ising models

To verify our theory in Section 2.4.2, we run controlled experiments benchmarking various sampling algorithms for Ising models: k-Gibbs sampler (Definition 10), and the independent parallel sampler (Definition 11).

The Ising model distribution that we sample from contains two modes, one larger and the other smaller, corresponding to $\mathcal{R}_1$ and $\mathcal{R}_{-1}$ defined in Equation (6) and Equation (7), respectively.

We show in Figure 3 that if we initialize the sample in the smaller mode $\mathcal{R}_{-1}$, running the k-Gibbs sampler (Definition 10) can often reach the larger mode $\mathcal{R}_1$ within a relatively small number of steps (though more peaky distributions i.e. those with larger J, are slower to sample). Moreover, larger k is faster than smaller k. By contrast, running the independent parallel sampler (Definition 11) cannot reach $\mathcal{R}_1$ within the compute budget we set. The results verify our theory in Section 2.4.2 that Markov Chains with dependent transitions can be (much) faster in sampling Ising models (compared with the independent parallel sampler).

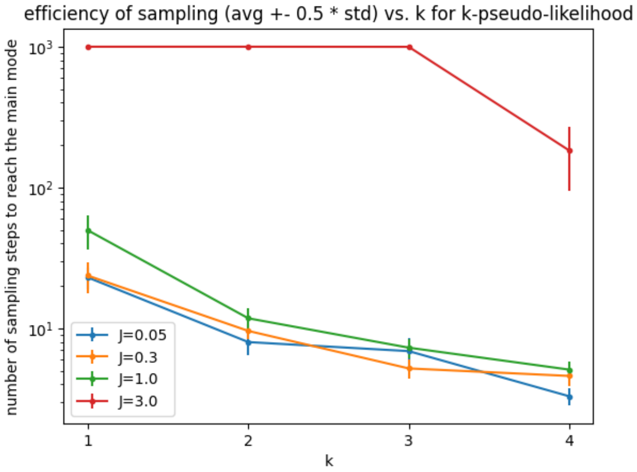

Figure 3. Number of steps for the k-Gibbs sampler (Definition 10) to reach the larger mode $\mathcal{R}_1$ (Equation (6)) of Ising models, starting from the smaller mode $\mathcal{R}_{-1}$ (Equation (7)). The parameters of our Ising models are: $N = 10, |C_G| = 4, h_i = 5.0$ in Equation (5). We vary the parameter J (a larger J corresponds to a more peaky distribution). Error bars denote ± 0.5 * stdev for 10 repetitions of the experiment. The compute budget is 1000 steps. Thus, a point with vertical coordinate 10^3 means that the sampler did not reach $\mathcal{R}_1$ within compute budget. The k-Gibbs sampler can often reach the $\mathcal{R}_1$ (larger k is faster). For context, the independent parallel sampler (not on the plot) can never reach $\mathcal{R}_1$ within the compute budget for any of the J's we tried.

M. Additional related works

We expand on the discussion in Section 4.

Non-autoregressive text generation Previous works applied various generative models to text, such as VAEs (Bowman et al., 2016; Bosc & Vincent, 2020), GANs (Che et al., 2017; Yu et al., 2017; Lin et al., 2017; Guo et al., 2018), and normalizing flows (Ziegler & Rush, 2019; Ma et al., 2019; Hoogeboom et al., 2021), but without a strong autoregressive component, the quality of generated text is often suboptimal. Later works achieve high-quality text generation through diffusion models (Hoogeboom et al., 2021; Austin et al., 2021; Li et al., 2022; Gong et al., 2023; Zheng et al., 2023) and energy-based models (Deng et al., 2020; Goyal et al., 2022; Qin et al., 2022), but their generation speeds tend to be much slower than autoregressive language models. Inference latency can be mitigated by approaches like Lee et al. (2020). Unlike the above paradigms that adapt continuous-domain generative models to text, our approach is closer to the following line of works that iteratively refine the generation process through parallel updates in the space of discrete token sequences, which tend to be at least twice faster than autoregressive approaches with a small drop in quality (Lee et al., 2018; Ghazvininejad et al., 2019; Stern et al., 2019; Guo et al., 2020; Ghazvininejad et al., 2020; Kasai et al., 2020; Savinov et al., 2022) (though autoregressive models also have the potential for speedup by using a shallower decoder for certain tasks (Kasai et al., 2021)). The generation quality of non-autoregressive models can be further improved by incorporating some autoregressive components (Kong et al., 2020; Reid et al., 2022) or input-output alignment (Chan et al., 2020; Saharia et al., 2020), or adaptive training curriculum (Qian et al., 2021). Insights such as the multimodality problem and components such as sequence-level knowledge distillation and input token fertility prediction were also proposed in (Gu et al., 2018). The benefit of distillation was verified in Kim & Rush (2016); Gu et al. (2018); Zhou et al. (2020); Gu & Kong (2021). Positional attention was tested in Gu et al. (2018); Kreutzer et al. (2020). Relevant to our experiments in Section 3.4, Ren et al. (2020) measure the target-side dependency as the proportion of attention paid to target tokens as opposed to the source tokens, in some modified attention architecture. Related to generation from MLMs, Wang & Cho (2019) use the learned conditionals inside a Gibbs sampler, but when the conditionals are not *consistent*, i.e. there is not a joint distribution that satisfies these conditionals, Gibbs sampler may amplify errors. In general, mathematical understanding about sampling from masked language models is still lagging substantially behind. Additionally, related to MLMs, Meng et al. (2023) analyzes some representational limitations, and Liu et al. (2022) analyzes subtleties from a parameter identifiability view. Related to parallel decoding, recent work (Cai et al., 2024) parallelizes the inference with multiple heads by finetuning autoregressive LLM backbones.

Theory about parallel sampling Koehler et al. (2023) proved a generalization bound for pseudolikelihood estimator via the classic ($k = 1$) approximate tensorization of entropy, in the “proper learning” setting. Our generalization bound (Theorem 4) uses the generalized notion of the approximate tensorization of entropy (Definition 9), also apply to “improper learning” settings, and the proof involves quite different techniques. The classic approximate tensorization of entropy are discussed in Marton (2013; 2015); Caputo et al. (2015), which was more recently generalized to the “α-weighted block” version (Definition 9) in Caputo & Parisi (2021). Lee (2023) proves that k-Gibbs sampler mixes at least k times faster than 1-Gibbs sampler. For future works, recent algorithmic advances in parallel sampling could potentially be incorporated into our framework to achieve finer-grained theoretical analysis or better empirical quality-efficiency trade-off (Anari et al., 2023).