JOINT DISCRETE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion models generate tokens in any order, making conditionally independent predictions within each denoising step. Sampling multiple tokens independently in parallel can however severely degrade generation quality, motivating tractable models of joint token distributions. Here we introduce *joint discrete diffusion*, which uses a small learned conditional random field (CRF) head on a frozen diffusion backbone. The head defines a joint distribution over the sequence at each denoising step, allowing exact normalization, marginalization, and joint sampling. Two novel contributions are introduced to make this idea work in practice: A low-rank parameterization of the CRF that allows efficient use of the conditioning sequence and a one-step reveal objective. This reveal the remaining tokens objective is motivated by the fact that the CRF ELBO objective in the low reveal rate regime assigns a very low probability to revealing adjacent tokens simultaneously. Supervision of token dependence is therefore by construction substantially weakened. We tested our approach across large diffusion language models iLLaDA, Dream, SDLLM-MDLM, and SDLLM-Duo in prompt continuation tasks. Comparing the joint against the native backbone model, we observe over a range of conditions that the native is outperformed by a joint with half the number of generation steps. Our joint token modeling achieves a measured 1.8–1.9× speedup at equal or better scores under both text evaluators.

Code: anonymous.4open.science/r/joint-diffusion-C978

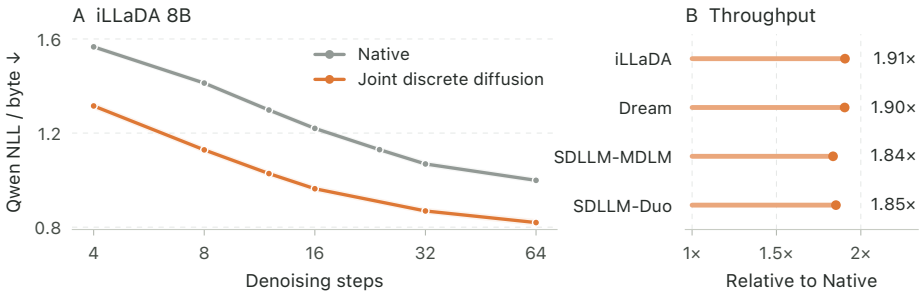


Figure 1: **Fewer denoising steps with better generation scores.** We use Qwen2.5-7B NLL (negative log likelihood) to score 256-token continuations from 32-token prefixes, using 1,024 held-out source documents for every backbone. **A:** Native (frozen backbone marginals) versus Joint discrete diffusion NLL on iLLaDA across measured budgets; shading connects per-arm pointwise 95% intervals. **B:** Measured throughput relative to double-step Native: 16 versus 32 steps for iLLaDA, Dream and MDLM; 64 versus 128 steps for Duo, at batch size eight. Figure 2 shows all quality–runtime curves; Table 1A gives these comparisons.

1 INTRODUCTION

Diffusion language models generate text by repeatedly predicting clean tokens from a masked or corrupted sequence (Sahoo et al., 2024; Ye et al., 2025). Each model evaluation can update many positions, making it possible to generate text with fewer evaluations than there are tokens. Standard decoders are trained to predict *the marginal token distribution* at each position. Sampling the marginals can produce incompatible choices even when the individual token marginals are correct. For example, "the cat meows" and "the dog barks" are both plausible completions, but independently

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

sampling the last tokens also permits "the cat barks." Sampling with a lower rate of revealed tokens trades speed for accuracy.

A more principled approach is to model the joint distribution of the missing tokens. Its usefulness depends on how the joint is parameterized and on what the training objective supervises.

We make the joint modeling practical with a small contextual conditional random field (CRF) (Lafferty et al., 2001) attached to a frozen diffusion backbone. The backbone supplies token predictions and global conditioning context, while the CRF models dependence between neighboring tokens. Low-rank pair factors permit exact normalization, marginalization and joint sampling over the full vocabulary. Only the added head is trained. The pair factors combine shared vocabulary features with interaction strengths learned from the conditioning context. This makes the learned dependence structure very versatile despite being very parameter-light.

In masked diffusion, a reverse step reveals a random subset of the missing tokens and scores only those, so the loss evaluates a marginal of the model's joint distribution rather than the joint itself. For two tokens with correct singleton marginals, an error in their dependence affects the loss only when both are revealed in the same step. If each token is revealed independently with probability r, an error in such a dependence contributes with weight r^2, whereas an error in a single-token prediction contributes with weight r; dependence errors are thus attenuated by a factor r. Smaller reveal probabilities therefore make the objective less sensitive to this dependence. A model with exact single-token distributions and incorrect dependence can incur little additional loss despite a substantial error in its joint prediction.

This analysis provides a rationale for *joint reconstruction*: scoring the complete conditional distribution of the missing clean tokens. The objective directly penalizes errors in their joint prediction, including dependencies that partial reveals can obscure. At a fixed corrupted input, the KL divergence of the complete prediction bounds that of every subset marginal (Appendix A.1). Learning this distribution therefore provides a common training target for different generation budgets: the model learns a joint completion, while the decoder determines how many tokens to update at each step.

In matched training comparisons on iLLaDA, joint reconstruction improves generated-text scores over continued finite-step ELBO training, despite no consistent improvement in the validation ELBO. Comparisons with independent sampling from the same CRF's exact marginals show that coordinated choices contribute beyond improvements in individual token predictions. Across iLLaDA, MDLM, Duo and Dream, a single trained head per backbone improves the measured quality–runtime frontier across multiple generation budgets, with decoding schedules held fixed (Figure 2). At selected operating points, the heads outperform their native backbones under both automatic text evaluators while using half as many denoising steps, yielding 1.8–$1.9\times$ measured throughput (Figure 1).

Related works. Existing joint diffusion methods highlight difficulties in exploiting dependence modeling for efficient generation. Discrete Copula Diffusion sampler uses an auxiliary autoregressive model whose computation can dominate runtime at small denoising budgets (Liu et al., 2025). CoDD combines a probabilistic circuit with a window-based decoding scheduler (Li et al., 2026); in our results using Dream at 64 steps, its scheduler alone recovers the entire observed accuracy gain (Appendix C.3). Tensor-train diffusion reports improvements that vary by backbone: in its MDLM/OpenWebText comparison, none of the comparable reported operating points matches the baseline's generative perplexity at twice the denoising budget (Kim & Sung, 2026, Table 1). Joint reconstruction and frozen-backbone adaptation are shared ingredients of CoDD and tensor-train diffusion. Our focus is dependence supervision, a compact chain that constructs a tractable joint, and precise attributions to learn more about the setting.

2 Joint discrete diffusion

We first review discrete diffusion (Hoogeboom et al., 2021; Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024), then introduce the joint CRF head, its parameterization and training objective. Our derivation follows a different route than Sahoo et al. (2024); Shi et al. (2024) while still ending up with an equivalent result for the standard case of independent tokens (see Appendix B.3). We work with text x of length L over a vocabulary $\mathcal{V}$, so $x \in \mathcal{V}^L$, and with partially revealed latent states $z \in (\mathcal{V} \cup \{\text{mask}\})^L$. We write $U(z)$ for *the set of masked positions* of z.

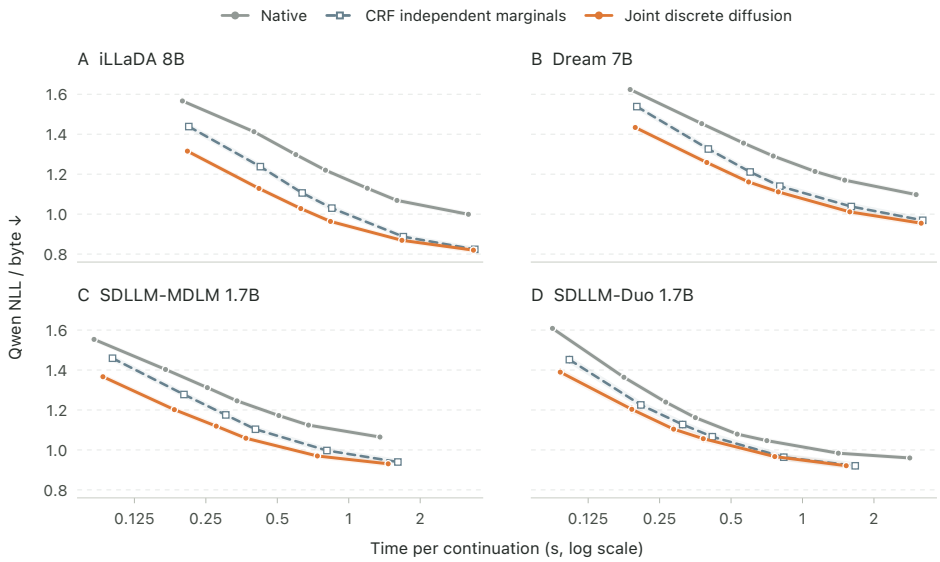


Figure 2: **Joint discrete diffusion improves the quality–runtime frontier.** We train one CRF head per backbone and use it for every budget. Curves compare joint draws (orange), independent draws from the same head's exact marginals (blue), and Native (gray), with decoding schedules fixed. Runtime measures throughput at batch size eight. All panels use the same 1,024 held-out source documents, 32-token prefixes and 256-token continuations. Error bars are pointwise 95% bootstrap intervals; shading connects their endpoints. Appendix D reports the GPT-2 results.

2.1 Generation as a sequence of reveals

The model generates by revealing tokens according to a reveal law. Let z_0 be the initial state, fully or partially masked. At step k the model draws a set of positions $R_k \subseteq U(z_{k-1})$ from the reveal law $\nu(\cdot \mid z_{k-1})$, then draws their clean values from the generative model $p_\theta(x^{R_k} \mid z_{k-1})$. The next state z_k is z_{k-1} with those positions filled in. After K steps every initially masked position has been filled.

To obtain the likelihood, we marginalize over the reveal order:

$$p_\theta(x) = \mathbb{E}_\nu\Big[\prod_{k=1}^{K} p_\theta\big(x^{R_k} \mid z_{k-1}\big)\Big] \tag{1}$$

We can turn this into a bound using Jensen's inequality

$$\log p_\theta(x) \geq \mathbb{E}_\nu\Big[\sum_{k=1}^{K} \log p_\theta\big(x^{R_k} \mid z_{k-1}\big)\Big]\,. \tag{2}$$

Each step contributes the log score of one subset of the clean tokens, so a step supervises only those marginals of the model's joint at z_{k-1} that its reveal set exposes.

The reveal sets form a sequence, whose law we write autoregressively,

$$\nu\big(R_{1:K}\big) = \prod_{k=1}^{K} \nu\big(R_k \mid z_{k-1}\big). \tag{3}$$

We assume throughout that ν reads only which positions are still masked, not the values placed at the revealed ones. The mask pattern of z_{k-1} is then determined by z_0 and $R_{1:k-1}$, so the law of the reveal sequence does not depend on x: one distribution over orders serves every text, and the expectation in Equation (2) is over orders alone.

We can for example make ν memoryless: at each step every masked position is revealed independently with probability r,

$$\nu\big(R \mid z\big) = r^{|R|}(1-r)^{|U(z)|-|R|}, \qquad R \subseteq U(z). \tag{4}$$

A position then waits a geometric number of steps before it is revealed, and after k steps it is still masked with probability $(1-r)^k$, independently of the others. Writing $\alpha_k = 1-(1-r)^k$ for the revealed fraction, the state z_k is available in closed form: mask each position independently with probability $1-\alpha_k$ and fill the rest from x. Training therefore needs no roll-out — draw a step, a state at that step, and a reveal set. Equation (2) is the evidence lower bound with the reveal order as latent variable and the reveal law itself as the variational distribution. No separate variational family is introduced.

Each term of Equation (2) depends on the reveal sequence only through the pair (z_{k-1}, R_k), so the expectation over orders reduces to the marginal law of that pair. Under Equation (4) both factors are available in closed form: z_{k-1} masks each position independently with probability $1-\alpha_{k-1}$, and R_k is an independent $\mathrm{Bernoulli}(r)$ draw on the masked positions. Writing $\mathcal{M}_\alpha(x)$ for the first and $\mathcal{B}_r(z)$ for the second,

$$\log p_\theta(x) \geq \sum_{k=1}^{K} \mathbb{E}_{z\sim\mathcal{M}_{\alpha_{k-1}}(x),R\sim\mathcal{B}_r(z)}\big[\log p_\theta\big(x^R \mid z\big)\big]. \tag{5}$$

No trajectory is simulated: an unbiased estimate of the whole sum draws k uniformly, a state at level α_{k-1}, a reveal set on its masked positions, and multiplies by K.

More generally the reveal probability may vary across steps. Let α_t be the fraction of positions revealed at time t, increasing from $\alpha_0 = 0$ to $\alpha_1 = 1$, evaluated on the grid $0 = t_0 < \cdots < t_K = 1$. The step from t_{k-1} to t_k reveals each masked position with probability

$$r_k = \frac{\alpha_{t_k} - \alpha_{t_{k-1}}}{1-\alpha_{t_{k-1}}}, \qquad \text{equivalently} \qquad 1-\alpha_{t_k} = \prod_{j=1}^{k}\big(1-r_j\big), \tag{6}$$

since a position is still masked exactly when it has failed every reveal so far. Constant $r_k = r$ gives the geometric schedule above, $\alpha_{t_k} = 1-(1-r)^k$. Everything preceding holds with r replaced by r_k: positions are still revealed independently, so the state marginal remains a product and Equation (5) is unchanged apart from the per-step rate. The backbone is conditioned on α, which is the only role the schedule plays here.

2.2 Joint distribution over the revealed tokens

We will derive two expressions for the subset generative probability $p_\theta\big(x^R \mid z\big)$ that we can plug into the evidence lower bound expression (5) or the one-step reveal objective introduced below.

The factorized model. The standard diffusion model uses independent generation:

$$p_\theta\big(x^R \mid z\big) = \prod_{i\in R} u_i(x_i; z), \tag{7}$$

where $u_i(x_i; z)$ will be a softmax probability given by the generative backbone. Every subset score is immediate, which is what makes a diffusion backbone an any-order autoregressive model. It is also the entire limitation: the tokens a step reveals are conditionally independent given z.

The chain. We keep u_i as a factor and add factors between each pair of neighbors,

$$p_\theta(x \mid z) = \frac{1}{Z}\prod_{i=1}^{L} u_i(x_i; z)\prod_{i=1}^{L-1} K_i(x_i, x_{i+1}; z). \tag{8}$$

The normalizer is one member of a family. For $R \subseteq U(z)$ let

$$Z(R) = \sum_{x^{U(z)\setminus R}}\prod_{i=1}^{L} u_i(x_i; z)\prod_{i=1}^{L-1} K_i(x_i, x_{i+1}; z), \tag{9}$$

the sum over the still-missing tokens with the positions of R held at x^R, so that $Z = Z(\emptyset)$. Scoring a subset is marginalization, and is again a ratio of two such sums,

$$p_\theta\big(x^R \mid z\big) = \frac{Z(R)}{Z}, \tag{10}$$

two sum–product passes, for any $R \subseteq U(z)$. A clamped position separates the chain: tokens on either side of it are conditionally independent given its value. The head therefore couples positions only within a run of consecutive masked tokens, and those runs shorten as generation proceeds.

Uniform corruption instead of masking. Duo replaces tokens rather than masking them, so no position is known to be clean: $U(z)$ is the whole sequence and nothing is clamped. The observed z_i is evidence about the clean token rather than a copy of it, and the unary is the backbone's clean-token potential π_i reweighted by the likelihood of that evidence under the known channel,

$$u_i(c; z) \propto \pi_i(c; z)\big[(1-\alpha) + \alpha|\mathcal{V}|\,\mathbf{1}\{c = z_i\}\big]. \tag{11}$$

The evidence at position i enters here and nowhere else. The head itself is unchanged: pair factors, partition sums and subset ratios never refer to masking.

2.3 Low-rank contextual parameterization of transitions

The chain defines a conditional random field (CRF) (Lafferty et al., 2001) that permits exact normalization, marginalization and sampling. A naive parameterization scales like $L|\mathcal{V}|^2$, prohibitive at language-model vocabularies. We therefore use a low-rank parameterization that draws on context through the backbone's last hidden layer $h(z)$. We write the transition kernel as one plus a low-rank product of nonnegative contextual features of dimension $\rho = 16$,

$$K_i(c, d; z) = 1 + \rho\,\eta_i(z) \sum_{a=1}^{\rho} A_a(c) B_a(d), \tag{12}$$

$$A(c) = \mathrm{softmax}\big(W_L^\top \bar{e}_c\big) \tag{13}$$

$$B(c) = \mathrm{softmax}\big(W_R^\top \bar{e}_c\big), \tag{14}$$

where $\bar{e}_c$ is the unit-normalized row of the backbone's frozen output embedding matrix for token c. It depends on the token alone, so A and B are computed once; the only contextual quantity in the pair factor is the scalar gate $\eta_i(z)$. The softmax runs over the ρ feature components, not over tokens. The projections W_L, W_R describe which vocabulary items are compatible; a scalar gate sets the strength of that compatibility at each edge,

$$\eta_i(z) = \exp\big\{s + 2\tanh\big[(w^\top h_i(z) + b)/2\big]\big\}. \tag{15}$$

When the gate is zero we revert to independent tokens. At initialization, $s = -3$, so $\eta_i = e^{-3} \approx 0.05$. The model is therefore initialized close to the independent model. The constant term in Equation (12) keeps every pair weight positive and contributes one further rank-one component. The resulting $\rho + 1 = 17$ components permit exact sum–product normalization, marginalization and joint sampling without materializing a vocabulary-pair matrix (Appendix B).

2.4 The one-step reveal objective

Under $\mathcal{B}_r$ a given missing position lies in R with probability r, and a given pair lies in R with probability r^2. A step's score $\log p_\theta(x^R \mid z)$ is a marginal, so it constrains the dependence between two positions only when both are revealed: in Equation (5) singleton errors enter with weight r and dependence errors with weight r^2. When there are many steps, r is small and the bound is dominated by the singleton predictions. Appendix A makes this precise, and Figure 6 shows its effect on optimization. This can partially be alleviated by changing the reveal set distribution to preferentially reveal pairs. However, we observed empirically that changing the objective was a much better choice.

Replacing R by all of $U(z)$ removes the attenuation: every missing pair is then scored at every step. Since $p_\theta(x^U \mid z) \le p_\theta(x^R \mid z)$ for $R \subseteq U$, this only lowers each term, so

$$\log p_\theta(x) \ge \sum_{k=1}^{K} \mathbb{E}_{z \sim \mathcal{M}_{\alpha_{k-1}}(x)}\big[\log p_\theta\big(x^{U(z)} \mid z\big)\big] \tag{16}$$

is again a lower bound — a looser one, which scores the complete reconstruction at every state the process visits instead of only the tokens that step reveals.

Equation (16) could serve as the objective directly, but K and α appear in it, so a head fitted to one grid would be fitted to one budget. Under the geometric schedule the steps visit masked counts $m_k = L(1-\alpha_{k-1})$ with $\mathrm{d}k \propto \mathrm{d}m/m$, so the sum is $1/|\log(1-r)|$ times a grid-free average that weights each count by $1/m$. The factor is positive and independent of θ, so the two have the same minimizer and we train on the grid-free form,

$$\mathcal{L}_{\mathrm{JR}} = \mathbb{E}_x \left[\frac{1}{L} \sum_{m=1}^{L} \mathbb{E}_U \left[-\frac{1}{m} \log p_\theta \left(x^U \mid z \right) \right] \right], \tag{17}$$

where the inner expectation is over uniform subsets U of size m and z is x with those positions masked. No step count or stage index enters: one fitted head serves multiple generation budgets, and the decoder alone chooses how many tokens to update per step.

For Duo we instead sample the clean retention $\alpha \in [0.001, 1)$ uniformly and score the complete clean sequence per token.

3 Experiments

The central question for our work is whether a small head can compensate for taking fewer denoising steps by coordinating token choices within each step. We directly examine each of our contributions: how far we shift the speed–quality tradeoff, the benefit of jointly reconstructing all missing tokens during training, and the gains from coordinating token choices. We use four frozen backbones: iLLaDA Base (8B), Dream Base (7B), SDLLM-MDLM Base (1.7B) and SDLLM-Duo Base (1.7B) (Nie et al., 2026; Ye et al., 2025; NVIDIA, 2026b;a). The first three use absorbing masking while Duo uses uniform corruption.

Evaluation. 32-token prefixes at temperature 1.0 unless stated otherwise. All four backbones use the same 1,024 held-out source documents, tokenized by each backbone's own tokenizer. *Native* is the factorized backbone with the same reverse process and reveal schedule as the CRF. We score each continuation with GPT-2-large NLL per token and Qwen2.5-7B NLL per UTF-8 byte (Radford et al., 2019; Qwen et al., 2025); lower scores indicate better agreement with these external text models. Runtime is warmed generation time on one LUMI MI250X card, divided by the eight continuations in a batch, giving a throughput measure. Confidence intervals are pointwise 95%; score intervals

Table 1: **Faster generation and gains beyond factorized adaptation.** GPT-2 NLL is per token and Qwen NLL per byte (lower is better); speedup is the throughput ratio and time is seconds per continuation. Panel B uses one fresh seed per trained head, separate from A, with the same examples, corruption draws and 24,576 updates. The CRF and factorized adapter each have 71,810 parameters. Tensor-train uses a rank-4 head and optimizer, with a different parameter count due to its parameterization. CRF independent marginals and Joint discrete diffusion use the same fitted CRF.

A Better scores with half the steps *Four backbones*

	Steps		GPT-2 NLL ↓		Qwen NLL ↓		
Backbone	Native	Joint	Native	Joint	Native	Joint	Speedup
iLLaDA	32	16	5.088	4.634	1.069	0.963	**1.91**×
Dream	32	16	5.396	5.165	1.170	1.111	**1.90**×
SDLLM-MDLM	32	16	5.297	4.983	1.124	1.059	**1.84**×
SDLLM-Duo	128	64	4.516	4.253	0.960	0.921	**1.85**×

B Matched training exposure *SDLLM-MDLM*

	8 steps			16 steps			32 steps		
Method	GPT-2	Qwen	Time (s)	GPT-2	Qwen	Time (s)	GPT-2	Qwen	Time (s)
Native	6.536	1.407	0.168	5.839	1.252	0.337	5.354	1.139	0.673
Factorized adapter	6.516	1.378	0.187	5.806	1.223	0.374	5.318	1.111	0.748
Tensor-train (rank 4)	6.202	1.284	0.580	5.596	1.152	0.749	5.122	1.057	1.090
CRF independent marginals	6.041	1.315	0.201	5.253	1.142	0.403	4.759	1.034	0.806
Joint discrete diffusion	**5.727**	**1.238**	0.184	**5.072**	**1.099**	0.367	**4.663**	**1.009**	0.735

resample paired documents while timing intervals resample batches. Appendix D reports training histories, full results and evaluation protocols.

3.1 Joint prediction improves the quality–runtime frontier

Joint prediction adds computation to each denoising step, but can reduce the number of steps needed to obtain good text. Figure 2 shows that this tradeoff is favorable on all four backbones. At the operating points in Table 1A, joint discrete diffusion obtains better scores under both evaluators with half as many backbone calls as Native. The extra cost per step is approximately 5–9%, so the reduction in calls translates into 1.84–$1.91\times$ higher measured throughput. Example generations appear in Appendix Table 5.

These gains are obtained with a single fitted head per backbone, reused across generation budgets. In a separate comparison on SDLLM-MDLM, joint discrete diffusion has significantly lower NLL under both evaluators and lower runtime than tensor-train joint modeling at all six budgets from 4 to 64 calls (Appendix D.2).

Quality beyond lower token entropy. Lower evaluator NLL can often accompany a less varied token distribution. We therefore test whether the advantage persists when we increase the joint model's token entropy. This control uses iLLaDA at 32 steps, with 256 development prefixes and 128-token continuations. Raising the backbone-logit temperature to $T = 1.05$ gives joint discrete diffusion higher corpus unigram entropy than Native at $T = 1$, while retaining lower NLL under both evaluators (Figure 3). Both paired intervals exclude zero. The score improvement thus cannot be explained by lower token entropy alone. The protocol and additional text diagnostics are given in Appendices D.7 and D.8.

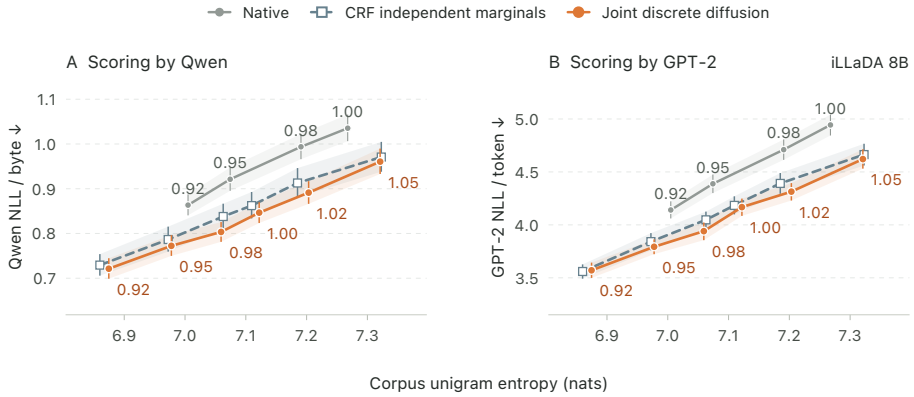


Figure 3: **Quality beyond lower token entropy.** iLLaDA (8B), 32 denoising steps, 256 development prefixes and 128-token outputs. Backbone-logit temperatures are labeled; both CRF decoders share the orange grid. Bars and shading show pointwise 95% document-bootstrap intervals. Protocol and repetition: Appendix D.7.

3.2 The learned dependence is mainly local

Although the head connects neighboring positions, a chain can also represent dependence between more distant tokens. We examine how far this dependence extends in the fitted iLLaDA head using 256 held-out documents with initially masked 128-token continuations. Mutual information under the joint is 0.894 bits between adjacent tokens, but already falls to approximately 0.006 bits at a separation of three tokens (Figure 4A). In other words, tokens three positions apart therefore share less than 1% as much information as adjacent tokens. The straight line in the log scale plot clearly suggests that average dependence decays exponentially with token separation.

To see whether this dependence is useful for prediction, we reveal one correct token and measure the change in a held-out target token's log probability. The joint can incorporate this token by conditioning its existing distribution, without another backbone evaluation. We observe its predictive gain is concentrated near the revealed token; rerunning the backbone gives larger gains over longer

distances (Figure 4B). The head therefore coordinates nearby choices between backbone evaluations, while the backbone remains important for propagating evidence over longer distances. Appendix D.10 gives the measurement protocol and the block-size diagnostic.

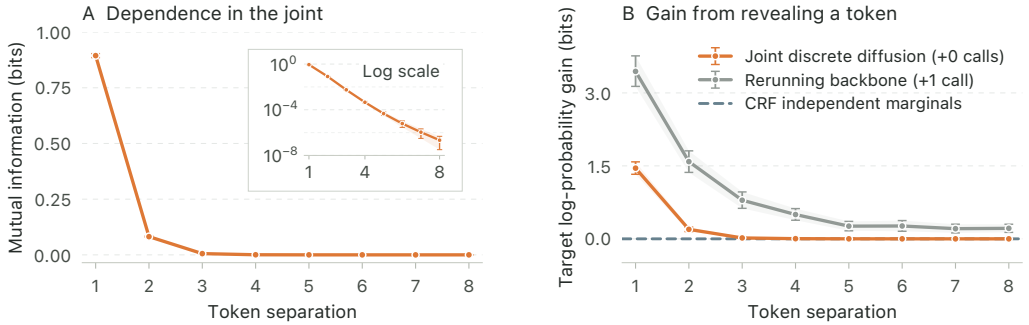


Figure 4: **Local dependence and predictive benefit between backbone evaluations. A:** Full-vocabulary mutual information under the frozen joint; the inset shows the same measurements on a logarithmic axis. **B:** Gain in the correct target token's log probability after revealing one true token, relative to each method's initial prediction. Native uses one additional backbone call; the joint conditions its cached distribution. Results use one iLLaDA-8B-Base head and 256 held-out documents with initially masked continuations. Bars and shading show pointwise 95% document-bootstrap intervals. Appendix D.10 gives the protocol and block-size diagnostic.

3.3 Direct supervision of the joint improves generation

To isolate the training objective, we continue two ELBO-trained iLLaDA heads under either finite-step ELBO or joint reconstruction. Each pair receives 12,288 additional updates from identical weights, examples and corruption draws, holding architecture and training history fixed.

Joint reconstruction improves four-step generation under both evaluators for both starting heads (Figure 5A). On the 256-prefix, 128-token development task, Qwen NLL falls by about 0.09 nats/byte in both comparisons. Both reconstruction-trained heads also outperform eight-step Native under both evaluators at approximately half its generation time. Yet their validation NELBOs show no consistently robust improvement over continued ELBO training (Figure 9B).

The reveal size also affects which model receives a better likelihood score. We next evaluate reconstruction likelihood directly on held-out text, comparing the joint iLLaDA head from Figure 2 with Native on 256 documents (Figure 5C). We keep the fitted models and corrupted inputs fixed while varying the reveal probability. To compare different reveal sizes, we divide subset NLL by the expected number of revealed tokens. Native scores better at $r = 1/8$, but the joint scores much better when all missing tokens are revealed: 2.190 versus 2.403 nats per missing token. Small-reveal scores can therefore favor a model that is worse at predicting the missing tokens jointly, consistent with our supervision analysis in Section 2.4. Appendix D.9 gives the full likelihood results.

3.4 Coordinated sampling contributes beyond better marginals

Even with independent samples, finetuning a model can potentially improve generation by revising individual token probabilities. Better generation scores alone therefore do not establish that our coordinated sampling is useful. To isolate its contribution, we compare joint draws with *CRF independent marginals*: independent draws from our fitted head's exact single-token distributions, calculated by simply marginalizing over each token in the joint distribution. This control preserves token probabilities but completely removes their dependence.

Joint sampling improves generation scores at the tested 4–32-step budgets for iLLaDA, MDLM and Dream (Figure 5B). For example, at four steps it lowers iLLaDA's Qwen NLL by 0.123 nats/byte. Improved marginals however account for much of the gain over Native. From the curves, the gain from token coordination clearly depends on both backbone and generation budget.

In a separate experiment, we test whether similar gains can be obtained by simply training a factorized adapter with the same parameter and training budget. We train a fresh CRF and adapter on SDLLM-

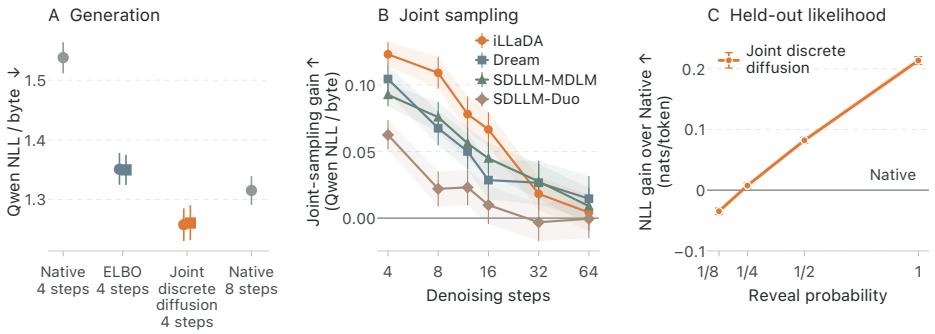


Figure 5: **Benefits of joint reconstruction and coordinated sampling.** **A:** Four-step iLLaDA generation after 12,288 ELBO or reconstruction updates from identical paired weights, using 256 development prefixes and 128-token outputs; marker shapes distinguish two lineages. **B:** On the main 1,024-document cohort, Qwen NLL of independent-marginal samples minus joint samples; positive favors joint sampling. **C:** Held-out data likelihood of the joint iLLaDA head on 256 documents: Native minus joint subset NLL at fixed corrupted inputs, normalized by the expected number of revealed tokens; positive favors the joint. Bars show pointwise 95% document-bootstrap intervals for individual means (A) and paired differences (B–C); shading connects intervals.

MDLM, each with 71,810 parameters and 24,576 updates. In this comparison, joint sampling outperforms both the adapter and CRF independent marginals under both evaluator models at 8, 16 and 32 steps (Table 1B) with paired intervals excluding zero. A comparably sized factorized head hence does not reproduce the gains from joint prediction in this experiment and only improves slightly over the frozen model (Appendix D.3). Joint discrete diffusion also outperforms the rank-4 tensor-train baseline under both evaluators at all three budgets, with lower generation runtime.

4 Discussion

Our results suggest that a small joint head can effectively reduce the number of backbone evaluations needed for diffusion language generation (Figure 2; Table 1A). The head improves individual token predictions and coordinates tokens sampled in the same step. Both contribute to the observed gains, although their relative importance varies across backbones and generation budgets (Figure 5B). The independent-marginal control makes this distinction explicit: improving the predictions of individual tokens does not account for the full benefit of the joint model.

Our measurements of token locality suggest a division of work between the backbone and the joint head (Figure 4). In our joint iLLaDA head, dependence is concentrated among nearby tokens, and conditioning on a newly revealed token mainly improves the nearby predictions. In contrast, a new backbone evaluation helps propagate information over longer distances. These observations suggest that inexpensive local coordination can practically, and efficiently, complement the predictions supplied by the backbone.

The training objective helps determine how strongly this dependence is supervised. When only small subsets are revealed, a model can predict individual tokens accurately while receiving relatively little penalty for errors in their dependence. Joint reconstruction instead directly scores the complete missing sequence. Our controlled training comparisons and the flip in model ranking with reveal size support the practical importance of this (Figure 5A and C). We further show that generation can improve without a consistent improvement in validation NELBO (Figure 9B), suggesting that validation NELBO alone can be incomplete when evaluating generative performance.

The scope of these conclusions remains limited by the evaluation. External language-model scores measure agreement with other models, and the entropy control addresses only one possible explanation for their improvement. Human evaluation and downstream tasks would help establish whether the gains translate into more useful text. The mechanistic measurements also concern particular fitted heads, and document-level confidence intervals do not capture variation across training runs. Repeating these comparisons across training seeds and longer continuations would clarify how reliably local coordination reduces the need for backbone evaluations.

REPRODUCIBILITY STATEMENT

Our anonymous code repository provides the CRF implementation, training, generation and scoring code, trained CRF heads for all four backbones, and fixed evaluation inputs. It includes scripts for reproducing the paper's figures and tables from retained measurements, dependency specifications, data preparation instructions, and CPU tests of inference and sampling. Backbone and evaluator weights are downloaded separately at specified revisions. Appendix A states the theoretical assumptions and proofs; Appendix B details exact inference and training objectives; and Appendix D documents data cohorts, training configurations, checkpoint selection, decoding, hardware, and uncertainty estimation.

AI USE STATEMENT

Generative AI tools assisted with code development, model training and evaluation on HPC clusters, analyzing experimental results, and preparing the manuscript and supporting materials. The authors reviewed AI-assisted work and take full responsibility for the final manuscript; this includes theory, methods, results and conclusions.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968. doi: 10.1109/TIT.1968.1054142.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *Advances in neural information processing systems*, 34(4), 2021.

Byoungkwon Kim and Minhyuk Sung. Tensor-train joint modeling for few-step masked diffusion. *arXiv preprint arXiv:2607.03788*, 2026. URL https://arxiv.org/abs/2607.03788.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289. Morgan Kaufmann, 2001. URL https://www.cs.columbia.edu/~jebara/6772/papers/crf.pdf.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the factorization barrier in diffusion language models. *arXiv preprint arXiv:2603.00045*, 2026. URL https://arxiv.org/abs/2603.00045.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FXw0okNcOb.

Thomas A. Louis. Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):226–233, 1982. doi: 10.1111/j.2517-6161.1982.tb01203.x. URL https://doi.org/10.1111/j.2517-6161.1982.tb01203.x.

Shen Nie, Qiyang Min, Shaoxuan Xu, Zihao Huang, Yuxuan Song, Yong Shan, Yankai Lin, Wayne Xin Zhao, Chongxuan Li, and Ji-Rong Wen. Improved large language diffusion models. *arXiv preprint arXiv:2606.25331*, 2026. URL https://arxiv.org/abs/2606.25331.

NVIDIA. SDLLM-Duo-1.7B-Base. Model release, 2026a. URL https://huggingface.co/nvidia/SDLLM-Duo-1.7B-Base.

NVIDIA. SDLLM-MDLM-1.7B-Base. Model release, 2026b. URL https://huggingface.co/nvidia/SDLLM-MDLM-1.7B-Base.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a365e37c18fb91af547a2f0012a89e98-Abstract-Conference.html.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115v2. Version 2, January 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T. Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52584–52619. PMLR, 2025. URL https://proceedings.mlr.press/v267/sahoo25a.html.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models. *arXiv preprint arXiv:2602.15014*, 2026. URL https://arxiv.org/abs/2602.15014.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025. URL https://arxiv.org/abs/2508.15487.

A SENSITIVITY OF THE SUBSET SCORE TO DEPENDENCE

Fix a corrupted context z with missing positions $U = \{1, \dots, m\}$, and let p and p' be positive distributions over the clean values x^U. We work on finite sample spaces and use natural logarithms. For $R \subseteq U$ write p_R for the marginal on x^R.

A step of Equation (5) scores $\log p_\theta(x^R \mid z)$ with $R \sim \mathcal{B}_r(z)$, so the discrepancy it can see between two models is

$$D_r(p\|p') = \mathbb{E}_{R\sim\mathcal{B}_r(z)}\big[\mathrm{KL}(p_R\|p'_R)\big]. \tag{18}$$

The reveal law does not depend on the clean values, so it cancels from the log ratio and leaves the average subset divergence; this is what makes Equation (18) the right object. A policy that inspected sampled token values before choosing positions would require a different analysis.

A.1 QUADRATIC ATTENUATION ON A CHAIN

Theorem A.1. *If p and p' are positive Markov distributions on the same ordered path,*

$$r^2\,\mathrm{KL}(p\|p') \;\le\; D_r(p\|p') \;\le\; \mathrm{KL}(p\|p'). \tag{19}$$

The factor r^2 is sharp, already for a positive binary pair.

Proof. Write $p(x) = p_1(x_1)\prod_{i<m} p_{i+1|i}(x_{i+1} \mid x_i)$ and likewise for p', and let $p_{j|i}$ denote the multistep conditional for $i < j$. For a nonempty $R = \{j_1 < \cdots < j_k\}$, summing out the intervening positions gives $p_R(x^R) = p_{j_1}(x_{j_1})\prod_{\ell<k} p_{j_{\ell+1}|j_\ell}$, and the same for p'_R, so

$$\mathrm{KL}(p_R\|p'_R) = \mathrm{KL}(p_{j_1}\|p'_{j_1}) + \sum_{\ell=1}^{k-1} \mathbb{E}_{p_{j_\ell}}\mathrm{KL}\big(p_{j_{\ell+1}|j_\ell}\|p'_{j_{\ell+1}|j_\ell}\big), \tag{20}$$

every term nonnegative. Put $A = \mathrm{KL}(p_1\|p'_1)$ and $B_i = \mathbb{E}_{p_i}\mathrm{KL}(p_{i+1|i}\|p'_{i+1|i})$. The term A appears whenever position 1 is revealed, with probability r. The term B_i appears whenever i and $i+1$ are both revealed — no position can lie between them, so they are necessarily consecutive in R — with probability r^2. Discarding the remaining nonnegative terms,

$$D_r(p\|p') \;\ge\; rA + r^2\sum_{i<m} B_i \;=\; r^2\mathrm{KL}(p\|p') + (r - r^2)A, \tag{21}$$

which gives the lower bound. For the upper bound, marginalization cannot increase divergence, so $\mathrm{KL}(p_R\|p'_R) \le \mathrm{KL}(p\|p')$ for every R; average over R. Sharpness follows from the construction below. □

Beyond chains. For an arbitrary positive p, the same argument applied to its chain projection p^{ch} — the chain matching p's singleton and adjacent-pair marginals (Chow & Liu, 1968) — gives $D_r(p\|p') \ge r^2\mathrm{KL}(p^{\mathrm{ch}}\|p')$, the residual $\mathrm{KL}(p\|p^{\mathrm{ch}})$ lying outside the chain family.

A.2 SHARPNESS INSIDE THE RANK-ρ FAMILY

For binary $x, y \in \{-1, +1\}$ let

$$q_\gamma(x, y) = \tfrac{1}{4}\,(1 + \gamma xy), \qquad 0 < \gamma < 1. \tag{22}$$

Its singleton marginals are uniform for every γ, so the empty and singleton divergences between $q_{\gamma_\star}$ and q_γ vanish and both positions are revealed with probability r^2:

$$D_r\big(q_{\gamma_\star}\|q_\gamma\big) = r^2\,\mathrm{KL}\big(q_{\gamma_\star}\|q_\gamma\big), \tag{23}$$

attaining the lower bound of Theorem A.1.

This lies inside the family of Equations (12)–(14) with finite parameters. Take $\rho = 16$ and signs $s_a \in \{-1, +1\}$, eight of each. Choose $0 < \delta < 1$ and set $h = \operatorname{arctanh}\delta$, with logits $s_a hx$ and $s_a hy$; since $x, y = \pm 1$ the softmax features are

$$A_a(x) = \frac{1 + s_a\delta x}{\rho}, \qquad B_a(y) = \frac{1 + s_a\delta y}{\rho}, \tag{24}$$

strictly positive and summing to one over a. Because $\sum_a s_a = 0$ and $s_a^2 = 1$,

$$\rho \sum_{a=1}^{\rho} A_a(x) B_a(y) = 1 + \delta^2 xy, \tag{25}$$

so with uniform unaries the implemented potential is

$$K(x, y) = 1 + \eta\, \rho \sum_a A_a(x) B_a(y) = (1 + \eta)\Big(1 + \tfrac{\eta\delta^2}{1+\eta}\, xy\Big), \tag{26}$$

which normalizes to q_γ with $\gamma = \eta\delta^2/(1 + \eta)$. Any $0 < \gamma < \delta^2$ is reached by the finite choice $\eta = \gamma/(\delta^2 - \gamma)$. A scalar embedding $\bar{e}_x = x$ realizes the logits through linear projections, with zero contextual-gate weights and log strength $\log \eta$. The r^2 rate is therefore attained by a model the head can represent, not only by a degenerate one.

A.3 Fisher information

The same rate appears without reference to any second distribution. Since XY has mean γ under q_γ, the full-pair score is $\partial_\gamma \log q_\gamma(X, Y) = XY/(1+\gamma XY)$, with mean square $\mathcal{I}_{\text{pair},1}(\gamma) = 1/(1-\gamma^2)$. A partial observation has a γ-independent law unless both positions are revealed, so

$$\mathcal{I}_{\text{pair},r}(\gamma) = \frac{r^2}{1-\gamma^2}, \qquad \text{against} \qquad \mathcal{I}_{\text{single},r}(\mu) = \frac{r}{1-\mu^2} \tag{27}$$

for a binary singleton $s_\mu(u) = (1 + \mu u)/2$. This is the information the subset score carries about a dependence parameter, which is what makes its gradient both small and infrequently non-zero; multiplying the loss by a constant rescales curvature without changing the observation (Louis, 1982).

A.4 Controlled synthetic experiment

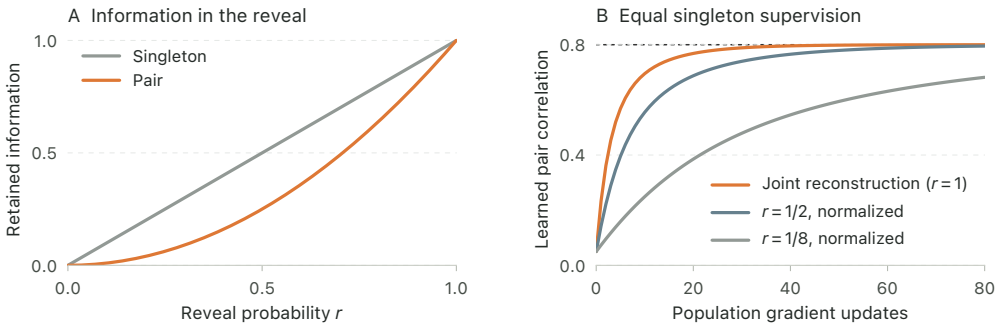


Figure 6: **Small reveals weaken supervision of dependence.** Left: information retained by independently revealing a biased singleton or a correlated binary pair, normalized by the corresponding clean observation. Right: learning the pair correlation by exact population gradient descent, with a common parameterization, initialization and learning rate. Dividing the partial-reveal objective by r equalizes singleton supervision but leaves pair supervision scaled by r. The one-step reveal objective corresponds to $r = 1$. All objectives have the same population optimum. These deterministic binary calculations isolate the effect of objective weighting; language-model comparisons appear in Section 3.

Figure 6 is generated by the accompanying `code/mechanism_experiment.py`; exact probabilities, optimization trajectories and numerical checks are saved in `data/mechanism.json`. The experiment uses exact population probabilities and gradients.

We use two binary tasks: a biased singleton with target mean $\mu_\star = 0.6$, and the correlated pair of Appendix A.2 with target $\gamma_\star = 0.8$. Let their natural coordinates be $a = \operatorname{arctanh} \mu$ and

$b = \operatorname{arctanh}\gamma$. Up to constants independent of the modeled parameters, the complete-observation cross-entropies are

$$\ell_{\text{single}}(a) = \log(2\cosh a) - \mu_\star a, \qquad \ell_{\text{pair}}(b) = \log(2\cosh b) - \gamma_\star b.$$

Under partial reveal the parameter-dependent cross-entropies are $r\,\ell_{\text{single}}$ and $r^2\ell_{\text{pair}}$. We divide both by r, giving equal singleton weighting at every reveal probability and pair weighting proportional to r. With step size $\lambda = 0.25$ the exact population-gradient updates are

$$a_{n+1} = a_n - \lambda\big(\tanh a_n - \mu_\star\big),$$
$$b_{n+1} = b_n - \lambda\, r\big(\tanh b_n - \gamma_\star\big).$$

We initialize $\mu_0 = \gamma_0 = 0.05$, run 80 updates, and compare $r \in \{1, \frac{1}{2}, \frac{1}{8}\}$; $r = 1$ is the one-step reveal objective. Every finite iterate has $0 < \gamma < 0.8$, so with $\delta = 0.95$ it admits the finite rank-ρ realization of Appendix A.2. The optimizer uses the same natural coordinate in every condition, isolating the effect of relative objective weighting after singleton normalization, and all conditions share the same population optimum.

The script also enumerates the reveal experiment directly to verify the r^2 identity of Equation (23), verifies the rank-ρ factor realization, and checks the chain-projection bound on 100 random positive length-five binary distributions p paired with positive Markov p' at six reveal probabilities.

B Exact inference and training objectives

The CRF's low-rank pair factors turn full-vocabulary inference into a sequence of small matrix contractions. We first derive this representation, then show how the backbone and corruption channel determine the unary evidence.

Throughout, $Z[u]$ denotes the partition sum of Equation (8) with unaries u, so that $Z = Z[u]$ and the clamped sums of the main text are $Z(R) = Z[u \odot \mathbf{1}_{x^R}]$.

B.1 Transfer-matrix representation

Using the vocabulary features A, B of Equation (14), define extended features $\tilde{A}_0(c) = \tilde{B}_0(c) = 1$ and, for $a > 0$, $\tilde{A}_a(c) = \rho\, A_a(c)$, $\tilde{B}_a(c) = B_a(c)$. Let $\delta_i(0) = 1$ and $\delta_i(a) = \eta_i$ for $a > 0$. Each pair factor becomes

$$K_i(c, c') = \sum_{a=0}^{\rho} \tilde{A}_a(c)\delta_i(a)\tilde{B}_a(c'). \tag{28}$$

For a chain of length L, fix edge-boundary states $a_0 = a_L = 0$ and form

$$M_i(p, q) = \sum_c u_i(c; z)\tilde{B}_p(c)\tilde{A}_q(c)\delta_i(q). \tag{29}$$

Summing over the internal edge states gives $Z[u] = e_0^\top M_1 \cdots M_L e_0$. The final state is fixed to zero, so the last strength does not enter the partition sum. Define left and right messages

$$\lambda_i^\top = e_0^\top M_1 \cdots M_{i-1}, \qquad \mu_i = M_{i+1} \cdots M_L e_0.$$

The singleton probability is

$$p_\theta(x_i = c \mid z) = \frac{u_i(c; z)}{Z[u]}\Big[\sum_p \lambda_i(p)\tilde{B}_p(c)\Big]\Big[\sum_q \tilde{A}_q(c)\,\delta_i(q)\,\mu_i(q)\Big]. \tag{30}$$

Messages are normalized during computation to control their numerical range. An exact joint draw is obtained by sampling the edge-state chain backwards, then sampling each token conditional on its two adjacent states.

For local evidence $e_i(c) \geq 0$,

$$\mathbb{E}_{p_\theta} \prod_i e_i(x_i) = Z[u \odot e]/Z[u]. \tag{31}$$

Clamping a queried subset with $e_i(c) = \mathbf{1}\{c = x_i\}$ and $e_i \equiv 1$ elsewhere recovers $p_\theta(x^R \mid z) = Z(R)/Z$ of Equation (10). Substituting $e_i(c) = C_i(y_i \mid c, z_i)$ computes the reverse transition as $Z[u \odot C_y]/Z[u]$, with $(u \odot C_y)_i(c) = u_i(c; z)\, C_i(y_i \mid c, z_i)$. Each query holds the factors and backbone state fixed; a new backbone evaluation constructs a new conditional joint.

B.2 Backbone interfaces and reverse channels

For absorbing-mask backbones, unknown-position unaries are clean-token probabilities with mask excluded from support. Visible continuation tokens are clamped to their observed value; the prefix conditions the backbone but is not an additional CRF chain. Dream uses its specified one-token output shift. All output IDs other than mask remain in its clean support, including padded IDs; the experimental appendix records decoding treatment.

For Duo, write its plug-in likelihood potentials as $f_i(c)$. With vocabulary size V and clean-retention parameter α_t,

$$u_i(c; z) \propto f_i(c)\, q_t(z_i \mid c), \qquad q_t(z \mid c) = \alpha_t \mathbf{1}\{z = c\} + (1 - \alpha_t)/V, \tag{32}$$

the evidence being applied exactly once. For $0 \le \alpha_t \le \alpha_s \le 1$ with $\alpha_t < 1$, $\alpha_s > 0$, and $\kappa = \alpha_t/\alpha_s$, the uniform posterior channel is

$$C_i(y \mid c, z) = \frac{\big[\kappa \mathbf{1}\{z = y\} + (1 - \kappa)/V\big]\big[\alpha_s \mathbf{1}\{y = c\} + (1 - \alpha_s)/V\big]}{\alpha_t \mathbf{1}\{z = c\} + (1 - \alpha_t)/V}. \tag{33}$$

With neutral factors this interface recovers the declared factorized uniform reverse law.

For a masked position, the channel at reveal probability r is

$$C_i(y \mid c, \mathrm{mask}) = (1 - r)\mathbf{1}\{y = \mathrm{mask}\} + r\mathbf{1}\{y = c\}. \tag{34}$$

If the currently unknown set is U and the newly revealed subset is R, the reverse probability is

$$p_\theta(y \mid z) = r^{|R|}(1 - r)^{|U \setminus R|}\, p_\theta\big(y^R \mid z\big), \tag{35}$$

the reveal factor being exactly $\nu(R \mid z)$ of Equation (4). This factorization requires R to be selected independently of the sampled clean token values given the current context; candidate-dependent confidence scheduling needs separate accounting.

B.3 Finite-grid variational bound

This is the classical diffusion derivation of the same bound. Equation (5) obtains it generatively, without a forward law or terminal prior; the two agree when the reveal law is shared, and the terminal term is the constant the generative route absorbs.

Let $z_0 = x$ and fix a forward law $q(z_{1:K} \mid x)$ and terminal prior p_K. For the normalized reverse chain,

$$\begin{aligned} -\log p_\theta(x) &\le \mathcal{F}_K(\theta; x) \\ &= \mathbb{E}_q \left[\log q(z_{1:K} \mid x) - \log p_K(z_K) - \sum_{k=1}^{K} \log p_\theta(z_{k-1} \mid z_k) \right]. \end{aligned} \tag{36}$$

Under the uniform-reveal-time grid used in the ELBO comparison the terminal state is all mask and the schedule factors cancel, leaving

$$\mathcal{L}_{\mathrm{ELBO},K} = \frac{1}{L} \sum_{j=0}^{K-1} \mathbb{E}\left[-\log p_\theta\big(x^{R_j} \mid z_j\big) \right]. \tag{37}$$

Figure 7 reports held-out ELBO gains and the effect of cutting a fitted head into shorter chains.

Given a uniform stage J and two conditionally independent revealed subsets $R^{(1)}, R^{(2)}$, the retained estimator of the per-token trainable loss is

$$\widehat{\mathcal{L}}_{\mathrm{ELBO},K} = -\frac{K}{2L} \sum_{b=1}^{2} \log p_\theta\big(x^{R^{(b)}} \mid z_J\big). \tag{38}$$

Under a factorized p_θ the conditional expectation of $-\log p_\theta(x^R \mid z)$ is $r\big[-\log p_\theta(x^U \mid z)\big]$, giving the mixed-step reconstruction weight $Kr/L = K/[L(K - J)]$. The identity fails in general for dependent p_θ, which is the content of Section 2.4.

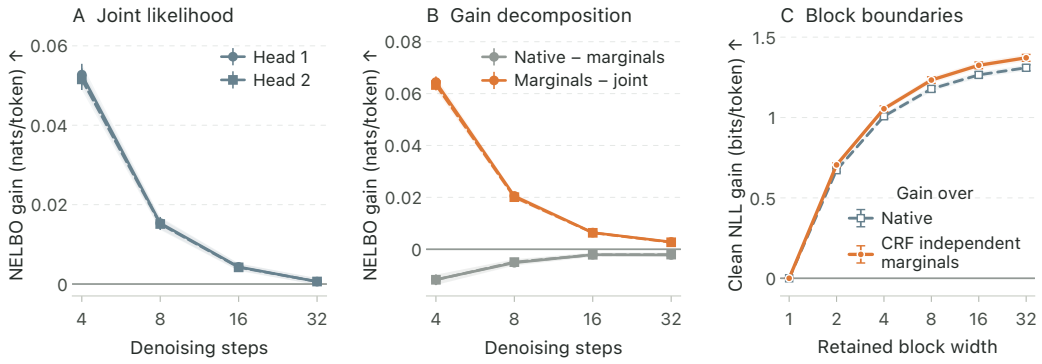


Figure 7: **Held-out likelihood gains and the effect of chain boundaries. A–B:** Two ELBO-trained iLLaDA heads evaluated on 1,024 reserved documents. Panel A shows Native minus joint finite-grid negative ELBO; panel B decomposes this gain into Native minus CRF independent marginals and marginals minus joint. Positive values indicate a lower finite-grid negative ELBO. Circles/solid and squares/dashed identify the two head seeds. Intervals are family-adjusted 95% over the 24 prespecified contrasts. **C:** For a separate iLLaDA head trained by the one-step reveal objective, clean reconstruction NLL improvement over Native and over the cut joint's CRF independent marginals is measured in bits per missing token. All widths use the same fitted head and 256 held-out documents; no head is retrained. Intervals are pointwise 95% from 10,000 paired document-bootstrap resamples. In all panels, bars show uncertainty and light shading connects their endpoints. Boundary cuts remove adjacent edges as well as restricting inference range.

Uniform corruption has additional fixed forward and prior terms. At terminal retention $\epsilon = 0.001$ with a uniform prior, its per-token terminal KL is

$$\pi \log(V\pi) + (V-1)\omega \log(V\omega), \qquad \pi = \epsilon + (1-\epsilon)/V, \quad \omega = (1-\epsilon)/V.$$

Such constants cancel in paired objectives under a common process but must be included in absolute likelihood-bound claims.

B.4 Step-free and mixed-step strengths

The step-free head uses one learned log strength s in Equation (15). The mixed-step head replaces it by $s_{K,j}$ for $K \in \{4, 8, 16, 32\}$ and $j \in \{0, \dots, K-1\}$, giving 60 strengths. The two iLLaDA variants therefore have 135,170 and 135,229 learned parameters. Mixed-step reconstruction samples a budget and one of its stages uniformly, masks each token with probability $(K-j)/K$, and weights the complete missing-clean joint NLL by $K/[L(K-j)]$. Equation (17) instead samples the mask count directly and requires no grid input. Training histories and initialization differences are recorded in Appendix D.

B.5 Computational cost and model restrictions

For hidden and embedding dimension d, rank ρ, and n_s strength parameters, the head has $2d\rho + d + 1 + n_s$ learned parameters; frozen embeddings and caches are separate resources. Writing $H = \rho + 1$, the retained dense moment construction costs $O(LVH^2)$, chain messages cost $O(LH^2)$, and materializing all token marginals costs $O(LVH)$. The inference cache of vocabulary outer-product features uses $O(VH^2)$ memory and is rebuilt when the learned factors change. Static cache preparation is excluded from warmed-generation throughput measurements.

Inference exactly normalizes and samples the clean law over the full output support. Visible interior tokens separate their adjacent chains at a fixed neural state. Global contextual information still enters the unaries and gates through the backbone. The native product law is recovered as $\eta \to 0$, a limit of the positive finite-parameter model.

C Related work and structural comparisons

C.1 Diffusion and clean conditional prediction

Masked diffusion models predict clean tokens from partially observed text (Sahoo et al., 2024). RADD identifies time-independent clean conditionals for absorbing diffusion (Ou et al., 2025), while Duo develops uniform-state diffusion and consistency distillation (Sahoo et al., 2025). SDLLM studies scaling across these diffusion families (Sahoo et al., 2026); Dream and iLLaDA provide larger pretrained backbones (Ye et al., 2025; Nie et al., 2026). These results establish clean conditional prediction as a useful diffusion parameterization. Our focus is the supervision of dependence when that prediction is a joint distribution and several tokens are updated together.

C.2 Tractable joint heads

Discrete Copula Diffusion combines a diffusion model with a separate dependence model (Liu et al., 2025). CoDD uses a probabilistic circuit together with contextual token evidence and a dynamic window policy (Li et al., 2026). Tensor-train diffusion parameterizes the clean joint through nonnegative contextual cores (Kim & Sung, 2026). Our construction uses classical CRF inference (Lafferty et al., 2001) with a restricted observed-token chain, whose relationship to tensor trains is given below.

Shared ingredients and different restrictions. CoDD and tensor-train diffusion already learn tractable clean conditional joints (Li et al., 2026; Kim & Sung, 2026). Both can adapt a frozen backbone; the released OpenWebText tensor-train scripts freeze the Transformer and vocabulary projection. Clean-joint training and frozen-backbone adaptation are therefore shared ingredients. Our contribution combines an analysis of the dependence exposed by a denoising objective with matched training comparisons and a compact contextual head. Comparing joint draws with independent draws from that head's exact marginals then tests whether coordinated choices contribute beyond changes in individual token predictions.

The released CoDD circuit prior is static; its token evidence depends on the current context. Our edge strengths also depend on the current backbone state, whereas general tensor-train cores can express richer contextual interactions. The tensor-train implementation further offers an accelerated path that predicts vocabulary-summed cores with an auxiliary head. Exact contraction of the original token-specific cores and this predicted-core path have different computational guarantees. Our exactness claims refer to inference in the specified block law, with its stated representational restrictions.

Our chain is a constrained tensor train. Fix the context and a block of $n \geq 2$ tokens, and set $f_i(c) = (1, \rho\eta_i A(c)^\top)^\top$ and $g(c) = (1, B(c)^\top)^\top$. Then $K_i(c,d) = f_i(c)^\top g(d)$. The nonnegative cores

$$\begin{aligned} G_1(c) &= u_1(c) f_1(c)^\top / Z, \\ G_i(c) &= u_i(c) g(c) f_i(c)^\top \quad (1 < i < n), \\ G_n(c) &= u_n(c) g(c) \end{aligned} \tag{39}$$

satisfy $Q(x) = G_1(x_1) \cdots G_n(x_n)$ exactly. Thus the TT bond is at most $\rho+1 = 17$, and is one across independent block boundaries. Each interior core slice has matrix rank one. More fundamentally, the observed tokens form a Markov chain at fixed context; a general hidden-state tensor train need not have this property. Comparing TT rank four with our edge rank sixteen is consequently not a comparison of equivalent model capacities.

Scope of empirical comparisons. Our principal frontiers use Base checkpoints and declared factorized baselines. The CoDD math checkpoints use Instruct backbones and task-specific priors, while the tensor-train text checkpoint uses the original MDLM/OpenWebText setting rather than SDLLM-MDLM. Their published aggregate scores are not directly comparable with our FineWeb evaluations. We therefore adapt the released tensor-train head to our SDLLM-MDLM setting (Appendix D.2). This matches evaluation and total training presentations, with the different training histories disclosed below. CoDD changes both the joint head and its window-based reveal policy. Our matched-decoder tensor-train comparison isolates a different question: how these trained implementations perform with common evaluation settings.

C.3 SCHEDULER ATTRIBUTION IN CODD

We replay CoDD components (Li et al., 2026) using frozen Dream-v0-Instruct-7B and the released Dream-GSM circuit on 128 reused GSM8K development questions (training indices 384–511). Four generation seeds give 512 paired trajectories per arm, each with 64 backbone calls and 512 response positions. We use 32-position windows, backbone temperature 0.2, top-$p = 0.95$, circuit temperature 0.1, and activation threshold 0.3. The corrected replay uses normalized categorical sampling and the release's coverage-one forcing branch.

Native achieves 35.16% accuracy; the circuit-free window/forcing policy and CoDD both achieve 53.13%, with identical paired correctness. Independent and joint circuit interventions also achieve 53.13%. The scheduler alone therefore recovers the entire observed 17.97-point gain in this configuration. A paired bootstrap over questions, retaining the four seeds together, gives a pointwise 95% interval of $[10.16, 25.78]$ points. This result motivates holding the decoding policy fixed when attributing gains to a joint head; it concerns the released CoDD system and does not compare its training against ours.

D EXPERIMENTAL DETAILS AND COMPLETE RESULTS

D.1 BACKBONES AND DATA COHORTS

All CRF experiments use frozen pretrained backbones and the retained positive low-rank CRF with a scalar gate derived from the backbone hidden state. No backbone weights are updated. Native and CRF methods within a backbone have identical token support, temperature, sequence lengths, and reverse schedule. The shared context is the complete partially corrupted sequence; contiguous CRF blocks restrict the additional dependence modeled by the head, not the backbone's receptive field.

Training and evaluation text comes from a cached FineWeb shard. The iLLaDA study reserves 32,768 training documents, 128 validation documents, 256 generation-development documents, 1,024 density-test documents, and 2,048 generation-confirmation documents. The iLLaDA and Dream heads were developed using the 256 generation-development documents and 128-token continuations; those numerical records remain in the source bundle. The main frontiers instead evaluate their fixed selected heads on the separately reserved 1,024-document curve cohort used for MDLM and Duo, with 256-token outputs. Source-document and prefix hashes exclude every iLLaDA/Dream fitting, validation and development reservation. Heads, schedules and primary comparisons were fixed before this confirmation; no head was retrained or selected using its scores. Later decoder controls reuse subsets of this cohort and are not additional independent confirmation populations.

MDLM and Duo use 262,144 established training documents and the first 128 documents of their established validation set. Their 1,024-document curve cohort was reserved separately and is disjoint from training and validation by document and canonical-prefix hashes. These exclusions prevent exact document/prefix overlap; they do not establish semantic deduplication or absence from backbone pretraining. Each backbone's own tokenizer defines the 32-token prefix and target length. We compare methods within a backbone, not absolute evaluator scores or latency across differently tokenized tasks.

Table 2: Training configuration of the general reconstruction experiments. "Additional" updates exclude the disclosed inherited training history.

Backbone	Rank	Block	Train length	Updates	Initialization
iLLaDA-8B-Base	16	32	128	24,576	Additional; reconstruction parent
Dream-v0-Base-7B	16	32	128	49,152	Fresh head
SDLLM-MDLM-1.7B	16	16	256	24,576	Additional; ELBO parent
SDLLM-Duo-1.7B	16	16	256	24,576	Additional; ELBO parent

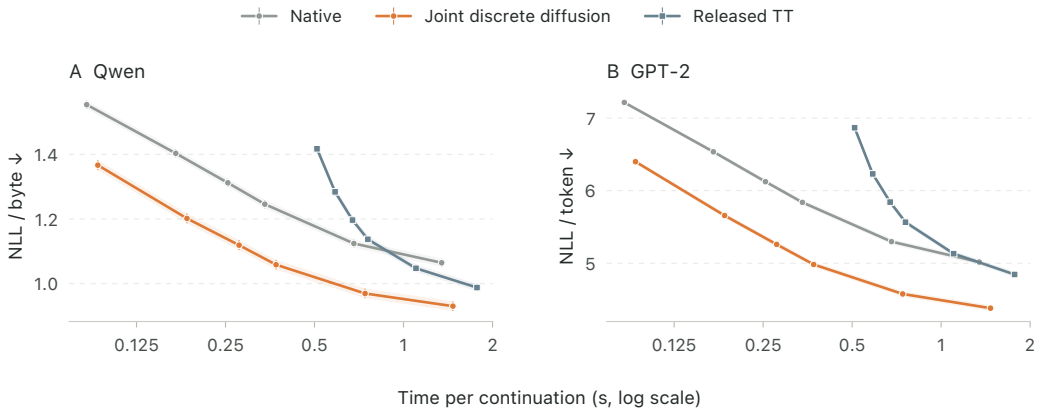


Figure 8: **Matched-decoder comparison with tensor-train inference.** Native, Joint discrete diffusion and TT use the same 1,024 ordered prefixes, 32 prefix tokens, 256 continuation tokens, temperature one, batch eight and Bernoulli reveal kernel, at 4/8/12/16/32/64 calls. Each prefix batch runs all three methods on the same GPU in randomized order after warmup. The 64-call extension uses the same selected heads in a separate session. Vertical intervals resample paired prefixes and horizontal intervals resample measured batches, both pointwise 95%; neither includes training-seed or between-session variation.

D.2 Tensor-train comparison on SDLLM-MDLM

We adapt the authors' released rank-four tensor-train head (Kim & Sung, 2026) to frozen SDLLM-MDLM and FineWeb. The primary rerun uses their probability, loss, contracted-core construction and sampling routines; it excludes our earlier custom execution optimizations. TT receives 49,152 batch-eight Adam updates at learning rate 10^{-5}, with auxiliary contracted-core coefficient one. Its 393,216 document presentations match the cumulative input exposure of the retained CRF lineage; the last 24,576 updates replay our reconstruction-stage documents and masks. Objectives, initialization histories, optimizers and compute still differ. Under the fixed 49,152-update budget, minimum exact validation NLL selects the final checkpoint.

TT has 85,303,568 trainable parameters, versus 71,810 for our head, and couples all missing positions rather than 16-token blocks. Its fit takes 4.82 hours on eight MI250X GCDs, including scheduled validation and checkpointing. Our final reconstruction stage takes 1.62 hours on one GCD, excluding inherited stages.

TT improves Native at every shared budget under both evaluators. Joint discrete diffusion improves TT at all six budgets, with paired pointwise 95% intervals excluding zero, and has lower measured runtime. At 16 calls, the TT-minus-CRF differences are $0.0785\,[0.0664, 0.0905]$ Qwen nats/byte and $0.5828\,[0.5412, 0.6250]$ GPT-2 nats/token; runtimes are 0.758 and 0.371 seconds per continuation. At 64 calls, the corresponding differences are $0.0575\,[0.0436, 0.0715]$ and $0.4645\,[0.4163, 0.5127]$, with runtimes 1.772 and 1.469 seconds. The comparison demonstrates the efficiency of the CRF implementation under common evaluation settings. All outputs, budgets and paired contrasts are retained.

D.3 A separately trained factorized adapter

The exact-marginal control removes dependence at sampling time, but its marginals were learned through a joint objective. We therefore also fit an independent predictor directly. On frozen SDLLM-MDLM we train a fresh rank-ρ, block-16 CRF alongside a factorized residual adapter, whose per-position weight is

$$\tilde{u}_i(c; z) \;\propto\; u_i(c; z) \exp\left\{ \tfrac{\eta_i}{\sqrt{\rho}} (h_i W_h)(\bar{e}_c W_e)^\top \right\} \tag{40}$$

with η_i the gate of Equation (15), h_i the frozen contextual hidden state and $\bar{e}_c$ the normalized frozen output embedding. The two $d \times \rho$ projections and the scalar gate give each head exactly 71,810 parameters at $d = 2176$ and $\rho = 16$, and the adapter samples positions independently. Its

zero-initialized W_h recovers Native exactly, whereas the CRF starts with weak positive coupling ($s = -3$).

Both heads receive identical batches, mask counts, masked positions and backbone outputs for 24,576 updates at global batch eight, using the same AdamW settings as the main heads. Both minimize complete reconstruction NLL per missing token (Equation (17)); for the factorized law, this is independent cross entropy. Each selects its own earliest minimum exact FP32 loss on the same 1,024 fixed validation corruptions, evaluated at updates 0/1024/4096/8192/12288/16384/24576. Both select the final update. This is one fresh paired seed, separate from the inherited main-curve heads. Combined fitting consumes 8.26 GCD-hours, including validation, the discarded smoke test and work repeated after a storage interruption.

For Table 1B, we additionally train a fresh rank-four tensor-train head using the released parameterization and optimizer in Appendix D.2 (85,303,568 parameters). It receives the CRF/adapter pair's exact examples, corruptions and 24,576 batch-eight updates. The same validation records and checkpoint grid select the final update.

Table 1B compares the fresh CRF with the trained factorized adapter and tensor-train head. All five methods are regenerated and timed together using the same 1,024 curve documents, 32-token prefixes, 256-token outputs, temperature one, batch eight and Bernoulli reveal rule. Each batch runs warmed arms in randomized order on the same GPU. Scores are means over all outputs; time is seconds per continuation. "CRF independent marginals" samples the fresh CRF's exact marginals independently.

The joint CRF beats both the trained adapter and its own product decoder at every tested budget under both evaluators, with paired pointwise 95% intervals excluding zero. At 16 calls, adapter-minus-joint differences are $0.1243\,[0.1105, 0.1375]$ Qwen nats/byte and $0.7341\,[0.6912, 0.7760]$ GPT-2 nats/token. This controls parameter count, data exposure and optimization settings. The joint also has lower corpus token entropy than the adapter. Bundled paired score and timing vectors, validation losses and compute receipts reproduce all comparisons and checkpoint selection.

D.4 Training lineage and checkpoint selection

The CRF experiments use batch size eight, AdamW with learning rate 0.003, zero weight decay, and gradient clipping at norm five. Head factors and messages are evaluated in FP32. Where training uses BF16 vocabulary contractions, fixed checkpoint audits compare objective values and exact marginals against FP32. A discrepancy above 0.005 in the objective or 0.01 in marginal ℓ_1 error disables the approximation for subsequent training. No gradient enters the pretrained backbone or its frozen output embeddings.

Paired iLLaDA objective experiment. The two initial head seeds are 921101 and 921102. Each had previously received 12,288 mixed-grid ELBO updates. Each is copied into two disjoint parameter sets, yielding an ELBO continuation and a joint reconstruction continuation with identical paired initialization. The four heads share clean training examples, corruptions, two reveal-subset draws per state, and backbone predictions. Each receives 12,288 additional updates and a fresh optimizer. Only the four-step strengths participate in this continuation. For each head, checkpoint selection minimizes the same fixed four-step validation NELBO, with earliest-checkpoint tie breaking. All four select the final update. Generated-text evaluators play no role in selection. The experiment has two independently initialized head lineages, each continued under both objectives.

General iLLaDA variants. Both variants start from the selected four-step joint reconstruction continuation of the first lineage, which inherits the earlier ELBO stage. The mixed-step head copies the complete 60-entry strength table for four budgets and their stages. The step-free head replaces it with one log strength initialized to the arithmetic mean of the parent's four active K4 log strengths. Both retain the parent's factors and scalar contextual gate and receive 24,576 additional updates. This projection initializes the step-free variant. The mixed-step objective averages uniformly over $K \in \{4, 8, 16, 32\}$ and stages; step-free training draws a positive masked count uniformly and then a uniform subset of that size, scoring the complete missing-clean joint NLL divided by the count. Each checkpoint is selected by its own fixed reconstruction validation objective, using 1,024 corruption records on the 128 held-out validation documents. Those objectives have different numerical scales and are not compared as likelihood bounds. There is one fit per general variant.

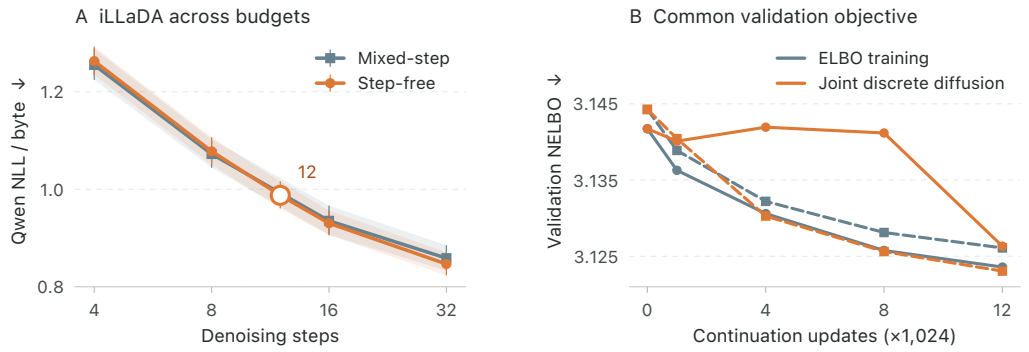


Figure 9: **Reuse across budgets and validation-ELBO trajectories. A:** Mixed-step and step-free Joint discrete diffusion on iLLaDA over 4–32 steps. The hollow marker is the additional step-free 12-step evaluation. Bars and shading show pointwise 95% intervals for individual Qwen means; differences between the variants at shared budgets are unresolved. **B:** The original paired objective experiment's common four-step validation NELBO. Circles with solid lines and squares with dashed lines distinguish the two initial head lineages. Both objectives start from the same weights within each pair and receive 12,288 continuation updates.

Dream, MDLM and Duo heads. Dream starts a fresh near-independent head with random centers drawn from its frozen output embeddings. It uses the masked-count reconstruction objective and receives 49,152 updates. The official one-token output shift is preserved. MDLM and Duo inherit factors and the scalar contextual gate of their earlier ELBO heads and replace their four log strengths by their arithmetic mean. They receive 24,576 additional updates. MDLM uses masked-count reconstruction. Duo samples the uniform-corruption level in $[0.001, 1)$, scores the complete clean sequence joint NLL divided by sequence length, and applies the clean-posterior evidence correction exactly once. Generation samples a clean joint proposal and then the retained uniform reverse channel. The head has no requested inference-budget input; the uniform-corruption posterior depends on the noise level. All three select the earliest minimum fixed reconstruction validation loss and select their final checkpoint.

The general curves evaluate the reconstruction heads produced by these training procedures, including their inherited training where applicable. The controlled comparison of objectives is the paired continuation experiment in Figure 5A; comparing general reconstruction curves with earlier ELBO curves would also compare their different training histories.

D.5 Sampling, scoring, and timing

For absorbing models, stage $j \in \{0, \dots, K-1\}$ reveals each unresolved position independently with probability $1/(K-j)$; the final stage reveals all remaining positions. For Duo, all methods use the same retained finite uniform-diffusion schedule and reverse channel. Native uses the backbone's factorized predictions. The marginal control first computes the CRF's exact singleton marginals, then draws each token independently; it still runs CRF inference. Joint sampling draws from the same fitted CRF. Runtime includes the full computation of each decoder.

Native budgets are $4, 8, 12, 16, 24, 32, 64$, with an additional 128 for Duo. Joint discrete diffusion and marginal controls use $4, 8, 12, 16, 32, 64$ for every backbone. Mixed-step iLLaDA is evaluated only at its explicitly supported $4, 8, 16, 32$ budgets, and the mixed-step comparison shows the 4–32-step range. The generation frontiers retain every declared point. Curves connect measurements without smoothing, interpolation-based speedup estimates, temperature changes, or output filtering.

The main iLLaDA and Dream frontiers measure all 19 declared arms in their fresh confirmation sessions. The original development-cohort 64-step extensions remain archived with the earlier curves. MDLM's joint and marginal 64-step points extend its original curve cohort in a separate timing session, using the same selected checkpoint and sampling settings.

GPT-2-large and Qwen2.5-7B score the complete generated continuation conditional on its prefix, using rolling context and including judge tokens crossing the prefix boundary. GPT-2 is normalized by scored tokens, Qwen by UTF-8 bytes. The underlying pretrained judge weights and tokenizers are fixed and SHA256-audited. Full output tokens, including special symbols, are retained. Score intervals use 10,000 bootstrap draws over paired documents; timing intervals resample generation batches. They are pointwise rather than multiplicity-corrected. Inference is conditional on the fitted heads.

Runtime covers the warmed, synchronized backbone/head/sampling loop on one AMD Instinct MI250X graphics compute die on LUMI. Arm order is randomized within batches of eight. Loading, compilation warmup, static caches, decoding, and I/O are excluded. The displayed seconds divide generation time by the number of completed continuations. Consequently a ratio such as $1.9\times$ is a throughput improvement at this batch size and hardware configuration.

D.6 Confidence-based reveal policies

We test whether the iLLaDA comparison depends on choosing positions at random. The joint iLLaDA head is evaluated on the first 256 documents of the shared curve cohort, with 32 prefix tokens, 256 output tokens, temperature one and batch eight. At 8/16/32/64 calls, we compare Native and joint sampling under Bernoulli reveals and under a maximum-confidence policy. The latter chooses exactly $256/K$ unknown positions per call using the original backbone's maximum token probability, before drawing any tokens. This gives both methods the same rule for choosing the reveal set at a common input state. A fifth arm samples Native candidates first, then ranks their backbone probabilities. Selecting positions after seeing their candidates also biases which token values are committed; it is outside the value-independent reveal model of Appendix A. This adapts the iLLaDA sampler's candidate-ranking rule, retaining our common FP64 categorical draws, full token support except mask, and global rather than windowed updates. It tests that policy in the continuation setting, rather than reproducing the publisher's complete evaluation protocol.

Maximum-confidence revealing worsens both evaluator scores relative to Bernoulli revealing at every budget. The CRF still improves Native under this same policy, with paired intervals excluding zero under both judges, but its scores remain worse than under Bernoulli revealing. Candidate ranking gives the lowest NLL and the clearest degeneration: 230–252 of 256 outputs are repetition-heavy across the four budgets. Here repetition-heavy means more than 20% repeated word/punctuation four-grams. At 16 calls it yields GPT-2 NLL 0.972 and Qwen NLL 0.311, with 252 repetitive outputs; Bernoulli Native yields 5.778 and 1.236, with none. Every output remains included. These results document sensitivity to the decoder and a failure of NLL as a standalone quality measure.

D.7 Temperature and lexical diversity

Equal nominal temperature does not imply equal output diversity. Figure 3 sweeps backbone-logit temperature on iLLaDA at 32 denoising steps, using the joint iLLaDA head, 256 development prefixes and 128-token continuations. Native uses $T \in \{0.92, 0.95, 0.98, 1.00\}$; both CRF decoders use these temperatures plus 1.02 and 1.05. We divide backbone log probabilities by T before renormalization and exact CRF inference. Learned pair factors and gates stay fixed at a common corrupted state; this does not raise the entire joint distribution to the power $1/T$. The independent decoder uses exact marginals of that same tempered CRF. Token support, reveal schedules, prefixes and seeds remain unchanged. Each decoder reproduces its first eight original $T = 1$ outputs exactly.

Cooling Native substantially improves its evaluator scores. Conversely, warming Joint discrete diffusion to $T = 1.05$ raises its token entropy above Native at $T = 1$ (7.321 versus 7.268 nats), while preserving lower Qwen NLL (0.961 versus 1.035) and GPT-2 NLL (4.621 versus 4.944). The paired gains are $0.075\,[0.040, 0.110]$ nats/byte for Qwen and $0.322\,[0.201, 0.440]$ nats/token for GPT-2. Thus lower corpus unigram entropy alone cannot explain the observed score advantage. Every measured temperature is shown; no exactly entropy-matched result is inferred by interpolation.

Independent-marginal sampling follows a similar tradeoff: at $T = 1.05$ it also has higher token entropy and lower NLL under both evaluators than Native at $T = 1$. At this 32-step budget, joint–marginal score differences are unresolved at five of the six temperatures. The temperature sweep therefore primarily supports improved CRF predictions, rather than an additional coordination gain.

Their singleton laws agree at a common state and temperature, but subsequent sampled states can differ.

We compute entropy from pooled generated model-token frequencies, retaining special tokens. Repetition is the mean per-document fraction of duplicate token four-grams. At the warmer operating point, repetition is 0.472% for Joint discrete diffusion versus 0.334% for Native at $T = 1$. The control uses one development cohort and one fitted head at 32 steps. Intervals condition on the fitted models and specified temperatures; uncertainty from temperature selection is excluded.

D.8 Generated-text diagnostics

The main frontiers compare denoising laws under a shared schedule; Appendix D.6 tests two additional reveal rules. The evaluation measures external-model likelihood, corpus token entropy and surface repetition. Raw continuations include grammatical errors and topic drift. General-head results use one fit per configuration; paired-document intervals describe evaluation uncertainty for those heads.

Table 3: **Descriptive text checks at the headline operating points.** Entries are Native/joint, using the budgets and full cohorts of Table 1A. Repetition is the document mean of $1 -$ unique four-grams/all four-grams after case folding and splitting text into word runs and individual punctuation marks. Digit-heavy means more than 20% of characters are decimal digits, using the previously chosen post-hoc Duo diagnostic.

Backbone	Documents	Repeated 4-grams (%)	Digit-heavy outputs
iLLaDA	1,024	0.397 / 0.359	0 / 5
Dream	1,024	0.537 / 0.282	0 / 5
SDLLM-MDLM	1,024	0.393 / 0.315	1 / 2
SDLLM-Duo	1,024	0.618 / 0.600	3 / 38

Table 3 applies the same text checks to all four backbones. Mean repeated four-grams remain below 0.62%, and no output is empty after removing whitespace.

The Duo transfer has a visible numeric-output failure mode. Under a post-hoc diagnostic threshold of more than 20% decimal characters, 38/1,024 Joint discrete diffusion outputs at 64 steps are digit-heavy versus 3/1,024 Native128 outputs. The affected outputs raise the mean evaluator NLL and remain in every reported score. At high Duo budgets, the joint-versus-marginal contrast is unresolved.

Dream's vocabulary includes 397 output-only padded identifiers that its official slow tokenizer cannot decode. Support is identical across all methods and excludes only mask. For text scoring, each contiguous mapped span is decoded with the original tokenizer and each unmapped identifier is rendered as one Unicode replacement character. Exact identifiers and positions are recorded; outputs are never discarded or resampled. Across the 19 arms of the main Dream evaluation, 17 of 19,456 continuations contain one such identifier each.

D.9 Held-out likelihood of the general reconstruction head

We evaluate the joint iLLaDA head used in the main frontier, without fitting, on the first 256 reserved density-test documents. These are disjoint from fitting and selection; the pool was previously evaluated by earlier ELBO heads. Eight fixed contexts per document balance the 2,048 states over masked counts $k = 1, \ldots, 128$. Each context supplies four shared Bernoulli reveal subsets at $r \in \{1/8, 1/4, 1/2\}$; $r = 1$ scores the complete missing sequence. Dividing each subset NLL by rk preserves the expected singleton scale. Intervals use 10,000 paired document-bootstrap draws and condition on this head.

At full reveal, the marginal contribution is -0.0732 $[-0.0781, -0.0684]$ and the dependence contribution is 0.2867 $[0.2805, 0.2933]$, giving a net improvement of 0.2135 $[0.2072, 0.2199]$ nats per missing token. The signed dependence contribution measures the change in target cross-entropy from scoring the joint instead of its independent marginals. This observation connects the joint iLLaDA head to the local supervision analysis. The bundle also reports each contrast by corruption level. Diagnostic contractions use FP64 with the learned factors fixed; representative FP32–FP64

Table 4: **Dependence changes the ranking as more tokens are revealed.** N, P, J are the normalized NLLs of Native, the product of exact CRF marginals, and the CRF joint. The first two differences sum to the third; positive $N - J$ favors the joint. Intervals are pointwise 95%.

Reveal r	$N - P$	$P - J$	$N - J$ [95% interval]
$1/8$	-0.0751	0.0404	-0.0347 $[-0.0401, -0.0293]$
$1/4$	-0.0721	0.0797	0.0076 $[0.0028, 0.0124]$
$1/2$	-0.0723	0.1544	0.0821 $[0.0771, 0.0871]$
1	-0.0732	0.2867	0.2135 $[0.2072, 0.2199]$

differences are below 6.5×10^{-5} nats per missing token. Exact-marginal, singleton, empty-subset and neutral-law checks pass.

D.10 LOCAL DEPENDENCE IN THE FITTED HEAD

We measure dependence under the fitted joint using full-vocabulary mutual information (Figure 4A), and its predictive benefit by revealing one correct token and measuring the change in another token's log probability (Figure 4B). Conditioning the existing joint helps nearby predictions, but the gain falls rapidly with distance. Rerunning the backbone uses the same evidence over longer distances. Splitting the fitted chains into shorter blocks reduces their likelihood advantage gradually (Figure 7C). These observations support a role for the head in coordinating nearby predictions between backbone evaluations.

Figure 4 uses the same joint iLLaDA head and 256 reserved density documents as the preceding diagnostic. We balance target positions across the four original 32-token blocks and average the two reveal directions within each document. The 10,000 paired document-bootstrap draws condition on this fitted head. No-evidence target NLL is 9.695 bits for Native and 9.736 for the joint marginal. Exact CRF conditioning holds the original backbone fields and learned factors fixed; Native uses one additional backbone evaluation. Each document has a visible 32-token prefix and an initially masked 128-token continuation. All evidence–target pairs remain within the same original block, with intervening tokens masked. Every integer separation from 1 through 15 is measured; the main figure shows 1–8. Separation three therefore has two masked tokens between the evidence and target positions.

Mutual information under the model. For a fixed corrupted context z, let p_{ij} be the pair marginal of the frozen joint, p_i and p_j its singletons, and $\Lambda_{ij} = p_{ij}/(p_i p_j)$. Marginalization integrates all intermediate tokens using the full-vocabulary low-rank contractions. In bits,

$$I(X_i; X_j \mid z) = \frac{1}{\log 2} \mathbb{E}_{p_i p_j} \left[\Lambda_{ij} \log \Lambda_{ij} - \Lambda_{ij} + 1 \right] \tag{41}$$

using independent samples from the product marginals, without vocabulary truncation. The estimate uses 65,536 draws per pair, except at separation one, where an independent 262,144-draw refinement meets the prespecified precision criterion. Monte Carlo error is checked separately from document-bootstrap uncertainty; at separation one its aggregate standard error is 0.000330 bits, 7.5% of the document standard error. The two directions are averaged within each document before resampling, and all distances and curves use the same 10,000 document-bootstrap draws. The own-product control has exactly zero mutual information and reveal gain. Mutual information averages token pairs under the model; the reveal-gain diagnostic evaluates held-out true tokens, so their magnitudes need not agree.

How far the dependence reaches. Two independent measurements agree. Under the frozen joint, the mutual information between positions at separation d falls by about an order of magnitude per position, with 91% of it at $d = 1$ (Figure 4A). Cutting a trained chain into shorter blocks at inference time removes adjacent edges and costs gain in proportion: eight-token blocks retain 90.0% of the gain over Native and 90.3% of the original adjacent edges (Figure 7C). These cuts are inference interventions on one fitted head, not separately trained models.

Training without boundaries gives the same answer. From the same width-32 step-free iLLaDA checkpoint we continued width-32 and full-width heads for 4,096 updates each (one seed, 128-token continuations, 256 documents per evaluation). Generation-score differences were unresolved on both evaluators at 4, 8, 16, 32 and 64 steps. Full width cost 2.2–2.3% more generation time and improved full-mask reconstruction NLL from 5.598 to 5.570 nats per token, and its mean mutual information gain still fell to 1.3×10^{-4} bits at distance eight despite having no internal block boundaries. The short range is therefore a property of the head rather than of the boundaries, and blocks trade no measurable generation quality for a small speed gain.

D.11 HELD-OUT RESULTS FOR ELBO-TRAINED HEADS

An earlier iLLaDA experiment evaluates two ELBO heads on 1,024 independently held-out density documents. Figure 7A–B reports the complete finite-grid likelihood decomposition. At each stage, all laws use the same forward-corrupted state and two independent reveal subsets. Scores sum over all stages and divide by 128 continuation tokens. Head-independent schedule constants cancel in every reported difference. The native-minus-joint gain is positive under family-adjusted intervals

at 4, 8, and 16 steps; the 32-step total gain is unresolved. At four steps the total gains are 0.05274 and 0.05155 nats/token. The decomposition is informative: the CRF's own independent marginals score slightly worse than Native on these held-out corrupted data states, but the joint dependence contribution more than compensates. This is a density evaluation on real text, distinct from external evaluator losses on generated text. The intervals use 50,000 paired-document bootstrap draws and a fixed Bonferroni family of 24 contrasts: four budgets, two heads, and three gains.

D.12 Complete reconstruction-head results

Using GPT-2-large as the scorer for the same generations as our Qwen main text curves. We observe robust strong gains by using Joint Discrete Diffusion.

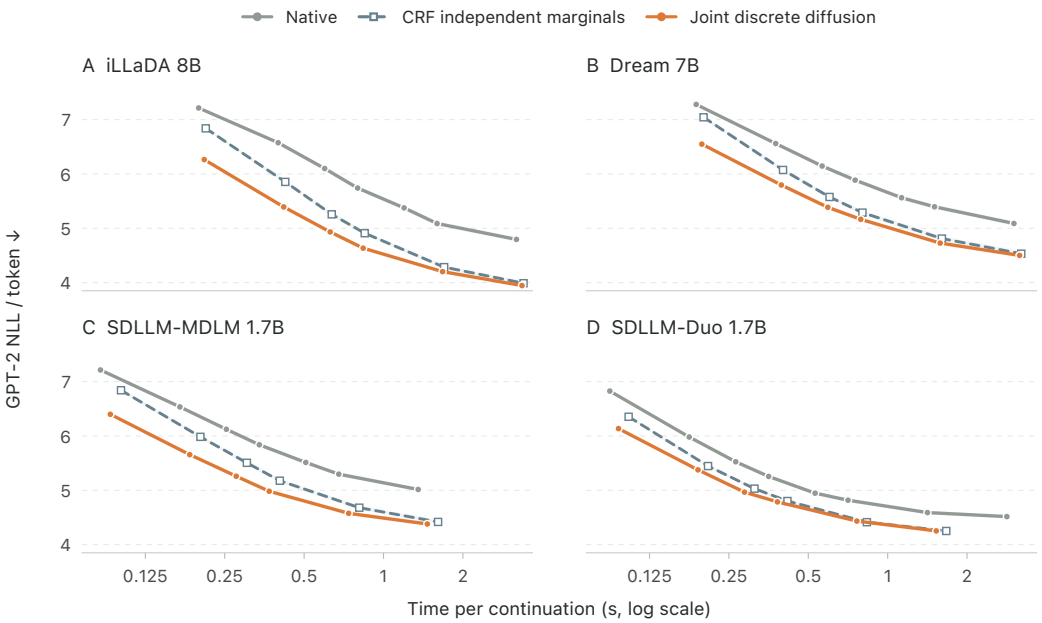


Figure 10: Native and Joint discrete diffusion with independent or joint sampling under GPT-2-large, using exactly the same generations and measured runtimes as Figure 2. Error bars show pointwise 95% document-bootstrap intervals; light shading connects their endpoints.

D.13 EXAMPLE GENERATIONS

Prompt. A lot of school subjects teach students to seek out "The Right Answer." But when we engage in a creative project, "The Right Answer" doesn't

Method	Steps	Continuation
Native	64	always exist. Thus we give students a E, E & Execution. , Every practice in coaching is an imaginative alchemy, Discipline is not learning is a "EvaL Itsp", yet at the same time it can also be. It can become 'evil' when we deny creative freedom and force iron-fast ideations upon ourselves and others. It crystalises thought, can't be said to be good to creativity. But we service the curiosity of students, we show people what interests open their minds …
CRF independent marginals	32	exist. To understand the focus, no clearly need a few parts: Every practice you engage in affects and will eventually affect others. " I sent a word, who set itself free" Last week we spoke about how we naturally respond to fighting against "evil," but we also naturally respond to retracting our energy back into ourselves. People want calmness, stability, a firmness. When they respond to our waywardness we will likely recede. Perhaps we can invite out money of their …
Joint discrete diffusion	32	exist. A good project will challenge and stimulate our students, prompting them to stretch towards a deeper understanding. In an artistic project, there won't be "The Right Solution." When you mix the subjects of learning, the result will be even more absorbing and engaging. We know that teaching a subject by using art or design is a powerful match. But that mix needs to be carefully balanced and that creativity and technical skills, starting with writing, needs to be infused. …

Table 5: Example generations with iLLaDA 8B at temperature 1.0. We show the shared 32-token prompt and the first 80 words of each 256-token continuation.