DISTILLING LANGUAGE KNOWLEDGE ACROSS GENERATIVE PROCESSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models enable parallel text generation. Prior work has shown that distillation from pretrained autoregressive (AR) models offers a promising route to acquiring strong language capabilities. However, the causal-prefix supervision used in prior work does not fully account for the mismatch between the two generation processes: an AR teacher predicts from a left prefix, while a diffusion student reconstructs missing text using observed context on both sides. To address this mismatch, we introduce CrossDLM, a framework for distilling language knowledge across these generative processes. CrossDLM uses the teacher's sequence-level scores to evaluate candidate tokens in the context of subsequent text, providing supervision for bidirectional reconstruction. This supervision incorporates visible evidence while accounting for uncertainty in unresolved content. Shared teacher computation across token alternatives and aggregation over a small number of student-generated completions make this supervision practical. With on-policy self-distillation, CrossDLM converts pretrained AR checkpoints into DLMs using approximately 0.10B generated rollout tokens per model. It outperforms causal-prefix distillation across a range of benchmarks and model families.

1 INTRODUCTION

Diffusion language models (DLMs) generate text through iterative denoising, allowing multiple tokens to be predicted in parallel (Austin et al., 2021a; Lou et al., 2024; Sahoo et al., 2024). However, training DLMs from scratch to acquire strong language capabilities can require substantial data and computation (Nie et al., 2025). Distillation from pretrained autoregressive (AR) models offers a promising alternative. Recent work, particularly OPDLM (Su et al., 2026), demonstrates data-efficient AR-to-DLM conversion through on-policy self-distillation: a pretrained AR checkpoint initializes both the frozen teacher and the diffusion student, which learns at its own denoising states.

Despite this progress, the two generation processes require different conditional predictions (Figure 1a). An AR teacher predicts the next token from a left prefix, whereas a diffusion student reconstructs missing text using observed context on both sides. Prior work, such as OPDLM (Su et al., 2026), constructs teacher targets from the causal prefixes of final rollout completions (Figure 1b). These queries can include completed values for positions that remain unknown at the current denoising step, while excluding observed tokens to the right. Consequently, causal-prefix supervision does not fully reflect the information available to the diffusion student. This mismatch can limit the effectiveness of knowledge transfer: strong autoregressive predictions do not necessarily translate into effective guidance at partially observed denoising states.

Although the AR teacher generates text from left to right, its joint distribution captures dependencies across the full sequence. This sequence-level knowledge, which is not directly exposed by causal-prefix supervision, supports conditioning on observed evidence and marginalizing unresolved content. Building on this observation, we introduce CrossDLM, a framework for distilling language knowledge across generative processes. For each missing position, CrossDLM conditions the teacher's joint distribution on the visible text and marginalizes the remaining unknown tokens to obtain a conditional token marginal (Figure 1c). These targets address both aspects of the mismatch: visible tokens on either side inform the prediction, while unresolved positions are integrated out rather than fixed to their final rollout values. This construction applies to both offline and on-policy distillation, separating the choice of teacher supervision from the source of training states.

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.

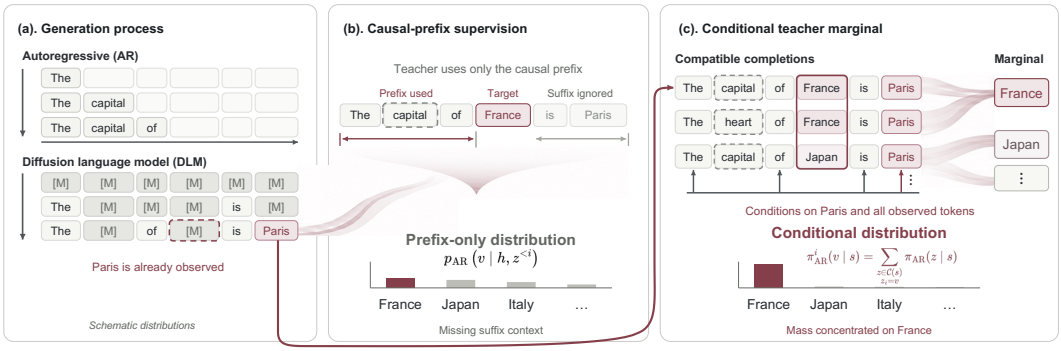


Figure 1: Generation and supervision. (a) AR extends a prefix; diffusion fills partially observed text. (b) Causal-prefix supervision fixes unresolved left tokens to their final rollout values and omits visible right context. (c) CrossDLM conditions on all visible tokens and marginalizes unresolved content. Distributions are schematic. Here h is the fixed prompt and completed blocks; z is the terminal completion in (b) and a compatible completion in (c).

Computing these conditional marginals exactly requires summing over exponentially many completions. CrossDLM makes this inference practical through local token scoring and aggregation over a small number of student-generated completions (Figure 2). Within each completion, it combines the teacher's exact next-token scores with a shared gradient-based approximation of how token substitutions affect subsequent text. The resulting local token distributions account for surrounding context without rescoring each alternative. Importance-weighted aggregation then combines these distributions to approximate marginalization over the remaining unknown tokens. In on-policy self-distillation experiments, CrossDLM converts pretrained AR checkpoints into DLMs using approximately 0.10B generated rollout tokens per model and outperforms causal-prefix distillation across a range of benchmarks and model families.

2 Knowledge Distillation across Generation Processes

Distilling from causal to causal models. In autoregressive distillation, the teacher's next-token distribution can directly supervise the student because both models condition on the same prefix. Generalized Knowledge Distillation (GKD) builds on this shared prediction task while allowing the training sequences and fitting divergence to vary (Agarwal et al., 2024). For a prompt c, response y with prefix $y^{<i}$, AR teacher p_{AR}, and AR student p_θ^{AR}, the objective is

$$\mathcal{L}_{\mathrm{GKD}}(\theta) = \mathbb{E}_{y\sim\mu_\lambda(\cdot|c)}\left[\frac{1}{|y|}\sum_i \mathcal{D}\big(p_{\mathrm{AR}}(\cdot \mid c, y^{<i}) \,\big\|\, p_\theta^{\mathrm{AR}}(\cdot \mid c, y^{<i})\big)\right], \tag{1}$$

where $\mu_\lambda = (1-\lambda)\mu_{\mathrm{off}} + \lambda\mu_{\mathrm{on}}$ combines the offline response distribution $\mu_{\mathrm{off}}(\cdot \mid c)$ with $\mu_{\mathrm{on}}(\cdot \mid c) = p_{\bar\theta}^{\mathrm{AR}}(\cdot \mid c)$, the generation distribution of the current student snapshot $\bar\theta$. The mixing coefficient $\lambda \in [0,1]$ controls the source of prefixes, while $\mathcal{D}$ selects the fitting divergence.

Challenges in distilling from causal to noncausal models. When the student uses masked diffusion, its input is partially revealed text with observations at arbitrary positions. Here noncausal refers to conditioning on visible tokens on either side of a prediction site. This creates a mismatch between causal-prefix queries and the student's denoising task.

Let $s = (x_t, t)$ denote a denoising state,[1] where x_t is the partially revealed block of length L and t indexes the denoising schedule. Visible positions form $\mathcal{O}_t$, and masked positions form $\mathcal{M}_t = [L]\setminus\mathcal{O}_t$. The student head $F_\theta^i(v \mid s)$ predicts a clean token $v \in \mathcal{V}$ at a masked position i. OPDLM (Su et al., 2026) obtains teacher supervision from the terminal completion y of the student rollout. Its teacher

[1] In blockwise generation, blocks are generated sequentially, with denoising applied within the current block. The prompt and previously completed blocks form a fixed prefix for this block. Conditioning on this prefix is implicit in all teacher and student distributions below.

query and the student's prediction are therefore

$$\underbrace{p_{\mathrm{AR}}(X^i = v \mid y^{<i})}_{\text{teacher: completed causal prefix}} \qquad \text{and} \qquad \underbrace{F_\theta^i(v \mid s)}_{\text{student: partially revealed block}} \quad . \tag{2}$$

An unresolved position before i is unknown at state s but fixed to its terminal value in the teacher query. Conversely, a visible token after i is available to the student but absent from that query. For example, causal-prefix supervision at the missing country in *The capital of [MASK] is Paris* excludes the observed city. These queries can provide useful supervision, but they generally answer a different conditional prediction problem from the one presented to the student.

Main idea: align supervision with the student's observed evidence. In principle, aligning the teacher and the student to use the same set of information requires the teacher to provide a joint distribution over the *masked tokens* conditioned on the student's *unmasked* evidence. For a candidate assignment y to the masked positions, we seek the following conditional probability from the AR teacher

$$p_{\mathrm{AR}}(X^{\mathcal{M}_t} = y \mid X^{\mathcal{O}_t} = x_t^{\mathcal{O}_t}), \qquad \text{abbreviated as } p_{\mathrm{AR}}(X^{\mathcal{M}_t} = y \mid s). \tag{3}$$

Let z denote the resulting sequence of replacing $x_t^{\mathcal{M}_t}$ in x_t with y. Conditioning is equivalent to restricting and renormalizing the teacher measure on the subset of sequences compatible with the unmasked tokens: $\mathcal{C}(s) = \{z \in \mathcal{V}^L : z^{\mathcal{O}_t} = x_t^{\mathcal{O}_t}\}$. With that, we can now define a teacher policy that the student aims to match:

$$\pi_{\mathrm{AR}}(z \mid s) \triangleq \begin{cases} p_{\mathrm{AR}}(z)/Z(s), & z \in \mathcal{C}(s), \\ 0, & \text{otherwise}, \end{cases} = \frac{\mathbf{1}[z \in \mathcal{C}(s)] p_{\mathrm{AR}}(z)}{Z(s)}. \tag{4}$$

where $Z(s) = \sum_{z \in \mathcal{C}(s)} p_{\mathrm{AR}}(z)$ sums teacher probability mass over compatible completions. Visible token values define the target at a given state; the reveal policy determines which states are visited (Appendix A.1).

The conditional teacher policy $\pi_{\mathrm{AR}}(z \mid s)$ is defined over complete sequences compatible with a given state. Its utility is especially highlighted when supervising the student's *instantaneous* denoising distribution at that state. Specifically, we construct the following distribution from the student model using its current parameter θ:

$$Q_\theta(z \mid s) = \mathbf{1}[z \in \mathcal{C}(s)] \prod_{i \in \mathcal{M}_t} F_\theta^i(z^i \mid s). \tag{5}$$

With forward KL divergence, we take advantage of the factorization of the Q, arriving at the following distillation loss for a given state s

$$\ell_s(\theta) = \frac{D_{\mathrm{KL}}(\pi_{\mathrm{AR}}(\cdot \mid s) \| Q_\theta(\cdot \mid s))}{|\mathcal{M}_t|} = \frac{1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} D_{\mathrm{KL}}(\pi_{\mathrm{AR}}^i(\cdot \mid s) \| F_\theta^i(\cdot \mid s)) + \text{const}, \tag{6}$$

Here π_{AR}^i is the conditional teacher marginal at position i. The factorization of the loss follows from the product form of the student's heads at a fixed state, not independence between denoising steps. Appendix A.3 gives the proof and distinguishes this pointwise optimum from what a shared neural network or its full rollout can realize.

We simplify further the loss as follows (up to a constant with respect to θ):

$$\ell_s(\theta) = -\frac{1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \mathbb{E}_{v \sim \pi_{\mathrm{AR}}^i(\cdot \mid s)}[\log F_\theta^i(v \mid s)] + \text{const}. \tag{7}$$

This objective maximizes the expected student log probability under the teacher marginal at each position. Section 3 uses importance sampling over the other unresolved tokens to construct the marginals, followed by an explicit vocabulary sum for this fitting loss. Averaging over a chosen state distribution gives

$$\mathcal{J}_\rho(\theta; \bar{\theta}) = \mathbb{E}_{s \sim \rho}[\ell_s(\theta)]. \tag{8}$$

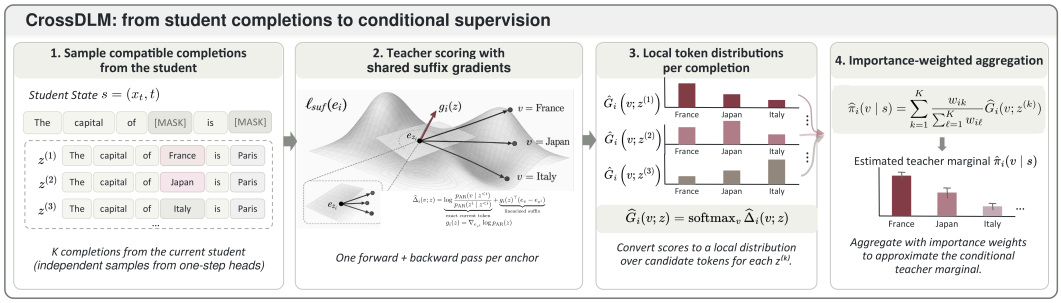


Figure 2: Overview of CrossDLM. Compatible completions are sampled from the current denoiser's one-step prediction, keeping visible tokens fixed. For each completion, an exact current-token term and an embedding-space approximation to the suffix likelihood score token substitutions. Normalization yields a local conditional; importance-weighted aggregation across completions approximates the teacher's token marginal for student training. Scores and distributions are schematic.

This separates conditional supervision from state collection, as in GKD. States can come from masked offline text, student trajectories, or a mixture. For comparison with OPDLM (Su et al., 2026), our experiments also use on-policy $\rho = d_{\bar{\theta}}$, induced by a current-student trajectory $\tau = (s_0, \ldots, s_T)$ and a rule $\nu(\cdot \mid \tau)$ for selecting a nonterminal state, with sampling held fixed during each update.

Under this forward-KL objective, the optimal unrestricted categorical heads are the conditional marginals:

$$F^i_*(v \mid s) = \pi^i_{\mathrm{AR}}(v \mid s) = p_{\mathrm{AR}}(X^i = v \mid X^{\mathcal{O}_t} = x_t^{\mathcal{O}_t}). \tag{9}$$

A key challenge is that merely computing these marginals requires a sum over an exponential number of compatible completions. We address that next.

3 CrossDLM: Efficient Teacher Supervision

For a fixed state s, the teacher marginal at a masked position i is obtained by summing the scores of compatible completions:

$$\pi^i_{\mathrm{AR}}(v \mid s) = \frac{\sum_{z \in \mathcal{C}(s),\, z^i = v} p_{\mathrm{AR}}(z)}{\sum_{z \in \mathcal{C}(s)} p_{\mathrm{AR}}(z)}. \tag{10}$$

Computing this marginal involves two nested computations. With the other masked tokens fixed, comparing all candidates at position i naively requires up to $|\mathcal{V}|$ teacher sequence evaluations. The marginal additionally sums over $|\mathcal{V}|^{|\mathcal{M}_t|-1}$ assignments to those tokens. CrossDLM first uses embedding-space linearization to estimate the local conditional distribution and its normalizer, sharing teacher computation across token alternatives (Section 3.1). Importance sampling over student-generated completions then combines these local quantities to approximate the conditional marginal (Section 3.2).

3.1 Local token scoring with shared suffix gradients

Consider a compatible completion z. Holding all positions except i fixed gives a local conditional over candidate tokens:

$$G_i(v; z) = \frac{p_{\mathrm{AR}}(z^{<i}, v, z^{>i})}{C_i(z)}, \qquad C_i(z) = \sum_{a \in \mathcal{V}} p_{\mathrm{AR}}(z^{<i}, a, z^{>i}). \tag{11}$$

Here $(z^{<i}, v, z^{>i})$ is the same completed block with token i replaced by v. The normalizer $C_i(z)$ is the teacher mass of the remaining tokens after summing over position i. Both exact quantities depend only on those remaining tokens, not on the original value z^i. Fixing a completion is a computational step: sampled tokens remain unknown at state s, and Section 3.2 integrates over them.

Separating the current token from its suffix. Define $\Delta_i(v; z) \triangleq \log p_{\mathrm{AR}}(z^{<i}, v, z^{>i}) - \log p_{\mathrm{AR}}(z)$, the change in log likelihood from replacing token i by v. Autoregressive factorization gives

$$\Delta_i(v; z) = \underbrace{\log \frac{p_{\mathrm{AR}}(v \mid z^{<i})}{p_{\mathrm{AR}}(z^i \mid z^{<i})}}_{\text{current token}} + \underbrace{\log \frac{p_{\mathrm{AR}}(z^{>i} \mid z^{<i}, v)}{p_{\mathrm{AR}}(z^{>i} \mid z^{\leq i})}}_{\text{suffix}}. \tag{12}$$

Earlier likelihood terms cancel. One teacher forward pass provides the current-token term for all candidates. The suffix term accounts for how each candidate changes the likelihood of subsequent text, including visible right context, but would require a separate suffix evaluation for each replacement. The complete scores recover the local conditional as $G_i(v; z) = \mathrm{softmax}_v \, \Delta_i(v; z)$.

Sharing the suffix computation. A first-order expansion in embedding space approximates the suffix term while preserving the exact current-token term:

$$\widehat{\Delta}_i(v; z) = \log \frac{p_{\mathrm{AR}}(v \mid z^{<i})}{p_{\mathrm{AR}}(z^i \mid z^{<i})} + g_i(z)^\top (e_v - e_{z^i}), \qquad g_i(z) = \nabla_{e_{z^i}} \log p_{\mathrm{AR}}(z). \tag{13}$$

Here e_v is token v's input embedding. With likelihood target IDs held fixed, causal attention makes the embedding at i affect only subsequent likelihood terms, so g_i captures the suffix response. Related gradient approximations appear in prior work (Grathwohl et al., 2021; Zhang et al., 2025). One teacher forward pass and one input-gradient backward pass provide the quantities for all positions in z. The vocabulary projection still costs $O(|\mathcal{M}_t||\mathcal{V}|d)$ arithmetic for embedding dimension d, but avoids a new teacher transformer evaluation for each replacement. Appendix A.6 details the derivation and approximation error.

The resulting scores estimate both the local conditional and its normalizer:

$$\widehat{G}_i(v; z) = \mathrm{softmax}_v \, \widehat{\Delta}_i(v; z), \qquad \widehat{C}_i(z) = p_{\mathrm{AR}}(z) \sum_{a \in \mathcal{V}} e^{\widehat{\Delta}_i(a; z)}. \tag{14}$$

The original completion supplies the expansion point. Unlike their exact counterparts in Equation (11), these approximations may depend on its reference token z^i.

3.2 Integrating the unresolved context

Each local conditional above fixes a different completion of the unknown text. To estimate the marginal in Equation (10), these conditional predictions must be combined according to the teacher mass of the surrounding tokens. The student provides a tractable proposal: a single forward pass at s yields the product distribution $Q_{\bar{\theta}}(\cdot \mid s)$ in Equation (5), from which K independent completions $z^{(k)}$ are drawn. Each preserves the visible tokens and independently fills the masked positions from the current heads. This differs from continuing the denoising trajectory to its terminal output, whose probability generally is not this product.

Because these completions are drawn from the student, their frequencies require importance correction. For position i, the proposal probability of the surrounding unknown tokens is the product of the other masked heads. The approximate teacher marginal is

$$\widehat{\pi}_i(v \mid s) = \sum_{k=1}^{K} \frac{w_{ik}}{\sum_{\ell=1}^{K} w_{i\ell}} \widehat{G}_i(v; z^{(k)}), \tag{15}$$

$$w_{ik} = \frac{\widehat{C}_i(z^{(k)})}{\prod_{j \in \mathcal{M}_t \setminus \{i\}} F_{\bar{\theta}}^j(z^{(k),j} \mid s)}. \tag{16}$$

The numerator estimates the teacher mass after summing over token i; the denominator corrects the probability of drawing the other unknown tokens. Position i is excluded because its alternatives have already been summed explicitly in $\widehat{C}_i$. Although a full completion is sampled to obtain a Taylor expansion point, no sampled-token correction at i is needed. Appendix A.5 derives this change of measure.

At $K = 1$, the normalized weight is one and the target reduces to a single local conditional. With exact local scores and full proposal support, the estimator converges to the teacher marginal as K grows. Appendices A.8 and A.10 provide the estimator analysis and numerical details.

Fitting the inferred marginals. Replacing the ideal marginals in Section 2 by these estimates yields the student fitting loss

$$\widehat{\ell}_s(\theta) = -\frac{1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \sum_{v \in \mathcal{V}} \widehat{\pi}_i(v \mid s) \log F_\theta^i(v \mid s). \tag{17}$$

The constructed targets, including proposal-dependent weights, are held fixed in this update; gradients flow only through the fitted student heads. Teacher input gradients are computed earlier to form the targets. Importance correction enters through $\widehat{\pi}_i$, equivalently weighting the conditional cross-entropies by the same normalized w_{ik} (Appendix A.5). The vocabulary expectation is evaluated explicitly, so no additional token-level importance ratio is required. Teacher entropy is constant for fitting and need not be estimated. Algorithm 1 summarizes the update; generation after training uses only the student and its sampler.

4 Experiments

The primary baseline is OPDLM (Su et al., 2026), which uses on-policy causal-prefix supervision from terminal rollout completions. LLaDA, Dream, SDAR, and Fast-dLLM v2 provide additional open-source references (Nie et al., 2025; Ye et al., 2025; Cheng et al., 2025; Wu et al., 2025).

4.1 Setup and overall performance

Experiments on Qwen3.5-4B and Qwen3.5-9B (Qwen Team, 2026) compare the two supervision targets under matched initialization, data, optimization, generation, and evaluation settings (Table 1). Each checkpoint initializes the frozen teacher and trainable student; each student collects its own on-policy states for forward-KL fitting. Both use OPDLM's 61,816 prompts for one epoch (7,728 updates). Defaults and result sources are in Appendices B.1 and B.2.

Both methods denoise four-token blocks conditioned on the prompt and previously completed blocks. Even one visible suffix token or unresolved prefix token can substantially change the teacher's conditional prediction. CrossDLM applies this supervision across denoising states throughout on-policy training, shaping the student policy used to generate the full response.

At 4B, CrossDLM exceeds OPDLM on all 12 benchmarks in Table 1, including MMLU (71.4% to 78.5%), GSM8K (85.8% to 92.5%), and MATH-500 (68.2% to 78.8%). SDAR also achieves strong performance, although its report does not fully specify the training data mixture (Cheng et al., 2025). At 9B, CrossDLM improves on 10 of the 12 benchmarks, with one tie, gaining 7.05 points on GSM8K, 9.60 on MATH-500, and 8.50 on ZebraLogic. The 4B code-generation comparison in Appendix B.10 additionally shows gains over OPDLM on all four benchmarks. Rollout-token counts increase by only 2.17% and 0.84% at 4B and 9B; Section 4.5 reports measured training time.

Figure 3 compares these rollout-token budgets with larger reference DLMs, distinguishing training stages. CrossDLM-4B exceeds the larger models shown; Appendix B.2 specifies token accounting.

4.2 Generality across base model families

On-policy self-distillation is performed independently on Qwen3-4B, Phi-4-mini-reasoning-3.8B, and Ministral-3-8B-Instruct-2512 (Yang et al., 2025; Xu et al., 2025a; Liu et al., 2026). Across the eight benchmarks in Figure 5, CrossDLM improves on all eight for Qwen3, seven for Ministral, and five for Phi, with one Phi tie. All three improve on GSM8K, MATH-500, and AIME-25; Phi gains 13.27 points on GSM8K and Ministral gains 6.61 on LCB-v6. CrossDLM improves mathematical reasoning across these families; gain magnitudes vary with the teacher–student adaptation process (Appendix B.8).

4.3 Ablation of teacher supervision

Table 2 compares Qwen3-4B supervision configurations under matched initialization, prompts, optimization, update count, and decoding. Each student collects its own on-policy trajectories; Appendix B.4 details the protocol.

Table 1: Capability comparison at 4B and 7B–9B scales. OPDLM and CrossDLM train the same base models under matched settings. Their scores and token counts are from our runs; other baseline values are taken from Table 1 of Su et al. (2026). Bold marks the unique highest benchmark score within each scale group.

	4B scale			7B / 8B / 9B scale					
Benchmark	SDAR 4B	OPDLM 4B	**CrossDLM 4B**	LLaDA 8B	Dream 7B	SDAR 8B	Fast-dLLM v2-7B	OPDLM 9B	**CrossDLM 9B**
Tokens (B)	55	0.10	0.10	1500	580	55	1	0.10	0.10
General Knowledge & Instruction Following									
MMLU	74.9	71.4	**78.5**	65.5	67.0	78.6	66.6	80.0	**80.1**
MMLU-Pro	50.9	54.7	**60.3**	37.0	43.3	56.9	41.5	57.8	**59.5**
GPQA-Diamond	33.0	50.5	**56.6**	31.8	32.1	40.2	27.3	42.4	**45.5**
IFEval	56.6	75.2	**79.9**	59.9	62.5	61.4	65.4	72.8	**76.3**
CEval	62.9	69.8	**77.3**	–	–	70.2	70.3	**78.2**	78.1
LiveBench	25.3	40.4	**47.4**	–	–	28.6	9.5	42.2	**44.1**
Mathematics & Reasoning									
GSM8K	89.9	85.8	**92.5**	78.6	81.0	91.3	83.7	85.6	**92.6**
MATH-500	72.8	68.2	**78.8**	26.6	39.2	78.6	65.6	70.6	**80.2**
AIME-24	10.0	10.0	**16.7**	2.1	0.0	10.0	10.0	16.7	16.7
AIME-25	7.5	10.0	**13.3**	0.4	0.0	10.0	0.0	16.7	**23.3**
LMB-Hard	6.9	17.8	**20.0**	–	–	8.9	8.9	8.9	**17.8**
ZebraLogic	6.3	10.0	**10.5**	–	–	7.8	3.5	4.1	**12.6**

A dash denotes an unreported baseline value. Our token counts measure generated rollout responses, excluding prompts, padding, and auxiliary completion proposals.

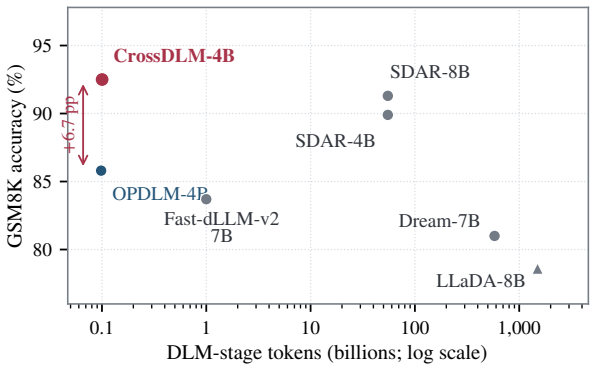


Figure 3: GSM8K versus DLM-stage tokens. Circles denote AR-initialized adaptation; the triangle denotes from-scratch pretraining. AR pretraining is excluded. Values follow Table 1; our counts exclude auxiliary proposals.

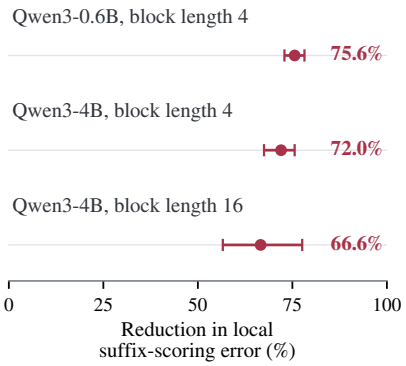


Figure 4: Local suffix-scoring error reduction under small embedding perturbations, relative to causal-prefix scoring. Bars show paired 95% confidence intervals over 256 validation questions per setting.

Removing supervision components. Starting from CrossDLM, replacing one-step samples $z^j \sim F_\theta^j(\cdot \mid s)$ with terminal completions obtained after further denoising lowers all three scores. Both preserve visible tokens; one-step proposals use the current heads and provide tractable probabilities for aggregation. Further removing Taylor suffix scoring recovers OPDLM's causal-prefix target $p_{\mathrm{AR}}(v \mid z^{<i})$ and lowers all three scores again. CrossDLM's total gains over OPDLM are 1.74, 3.20, and 2.11 percentage points on GSM8K, MATH-500, and MBPP-base, respectively.

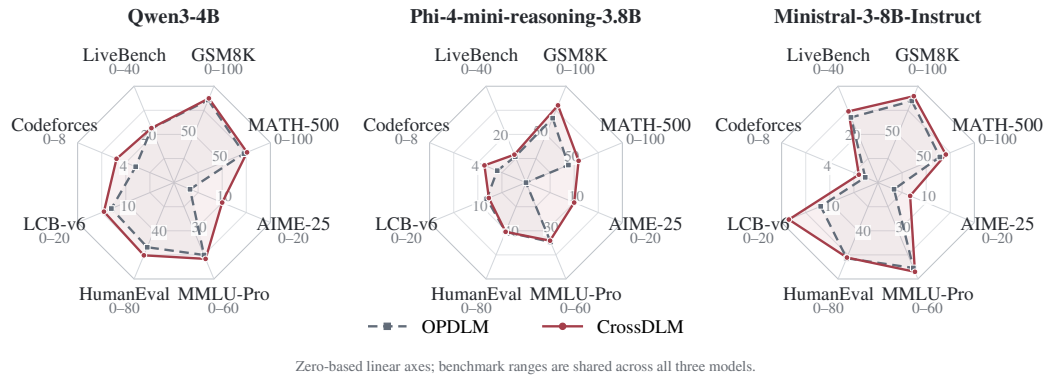


Figure 5: Independent self-distillation on three base models. Axes start at zero and use the labeled ranges shared across panels. HumanEval denotes HumanEval-base; LiveBench uses official active-1000. Appendix B.8 gives tabulated scores and settings.

Configuration	GSM8K	MATH-500	MBPP-base
CrossDLM	**87.41**	**75.80**	**46.03**
− One-step proposals (terminal)	87.13	73.46	44.44
Further − Suffix scoring (OPDLM)	85.67	72.60	43.92

Table 2: Component ablation on Qwen3-4B. Rows cumulatively replace one-step proposals with terminal completions, then remove suffix scoring. Scores are percentages; bold marks the best per benchmark.

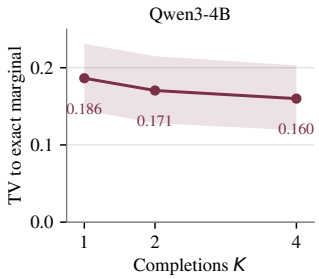


Figure 6: Qwen3-4B marginal estimation. TV to the exact marginal on restricted support, with 95% question-bootstrap intervals.

Aggregating completions. Figure 6 examines the effect of K on marginal estimation at frozen Qwen3-4B states. Enumerating a restricted candidate space provides an exact reference. Increasing K from one to four lowers mean total variation (TV) distance from 0.186 to 0.160, a reduction of approximately 14%. This diagnostic measures target accuracy independently of downstream training; Appendix B.6 gives the protocol and separates sampling error from the scoring approximation.

4.4 Local accuracy of the suffix approximation

Figure 4 probes the teacher's local suffix response using small embedding perturbations while holding the completion fixed. Taylor scoring reduces weighted error relative to causal-prefix scoring by 75.6%, 72.0%, and 66.6% across the three settings, with paired 95% confidence intervals excluding zero. These measurements characterize local response accuracy. The downstream ablation in Table 2 separately evaluates the effect of suffix scoring with complete token replacements. Appendices B.5 and B.6 examine complete token substitutions and completion aggregation.

4.5 Training cost

On-policy rollouts dominate computation in both methods, while CrossDLM batches its additional teacher scoring across proposals. On eight H100 GPUs, 4B training takes 90.73 hours with $K = 4$, versus OPDLM's 87.15 hours at the same update budget: only 4.1% overhead (Appendix B.7).

5 Discussion and Limitations

CrossDLM turns an AR teacher's sequence scores into supervision for partially observed text, separating the target from the source of training states. On-policy sampling determines which denoising states the student encounters; conditional inference determines the teacher prediction at each state. Local scoring and completion aggregation address complementary aspects of this prediction. The former accounts for how a candidate token affects surrounding text, while the latter integrates over alternative values of unresolved tokens. Together, they expose sequence-level teacher dependencies to the student's parallel prediction heads. The additional teacher computation is confined to training, so the student retains its diffusion sampler and requires no teacher access during generation.

Its target conditions on the committed prefix and visible tokens within the active block. Finite proposal coverage and importance-weight concentration limit marginal estimation, while first-order scoring can misestimate large token substitutions. Matching token marginals does not preserve all joint dependencies under parallel sampling. The teacher supplies model preferences rather than external correctness, and forward KL need not optimize every downstream metric. Appendix C discusses these limitations and broader extensions.

6 Related Work

Diffusion language models and AR adaptation. Discrete diffusion learns to reverse corruption over categorical variables (Austin et al., 2021a). SEDD estimates discrete probability ratios (Lou et al., 2024), while masked diffusion admits simplified masked-prediction objectives (Sahoo et al., 2024). LLaDA scales this paradigm through training from scratch (Nie et al., 2025); block diffusion combines sequential dependence across blocks with parallel denoising within them (Arriola et al., 2025). DiffuLLaMA and Dream instead adapt pretrained AR checkpoints through continued training (Gong et al., 2025; Ye et al., 2025). SDAR and Fast-dLLM v2 develop blockwise adaptation strategies (Cheng et al., 2025; Wu et al., 2025). These approaches develop model architectures and training recipes for parallel generation. CrossDLM addresses the complementary question of how an AR teacher can supervise denoising predictions under a different conditioning pattern.

Knowledge distillation and on-policy learning. Knowledge distillation transfers predictive distributions (Hinton et al., 2015), while sequence-level distillation trains on teacher-generated outputs (Kim & Rush, 2016). GKD combines offline and student-generated data with different fitting divergences (Agarwal et al., 2024); MiniLLM develops on-policy reverse-KL optimization for AR distillation (Gu et al., 2024). The state distribution determines where the student learns, while the teacher target specifies the prediction encouraged at each state. OPDLM applies on-policy self-distillation to AR-to-DLM conversion using causal-prefix targets (Su et al., 2026). CrossDLM retains this experimental setting but constructs conditional marginals reflecting the visible and unresolved content at each state. The resulting targets incorporate visible right context and integrate over unresolved tokens, aligning teacher supervision with the conditioning information available to the diffusion student. Its target construction also applies to offline states.

Sequence scoring and conditional inference. ENGINE trains an inference network using AR sequence energy (Tu et al., 2020), while energy-based diffusion uses sequence-level corrections during generation (Xu et al., 2025b). Gradient-based local approximations support efficient discrete sampling (Grathwohl et al., 2021). TCSM (Zhang et al., 2025) connects local conditional matching to pseudolikelihood and studies AR-to-diffusion distillation using clean-space scores and Taylor approximations for token substitutions. CrossDLM directly targets the teacher's token marginals conditioned on the current visible content, integrating over the remaining unresolved tokens. These marginals are estimated by aggregating local conditional predictions across student-generated completions, with importance weights accounting for teacher probability mass and proposal frequency. Whereas TCSM (Zhang et al., 2025) studies AR-to-diffusion distillation in an offline language-modeling setting, CrossDLM separates supervision construction from the source of training states and extends naturally to on-policy distillation. The inferred targets are learned during training, so generation requires no teacher scoring. Appendix D provides the extended comparison.

Reproducibility Statement

Section 3 and Algorithm 1 describe target construction and training. Appendices A.4–A.10 provide the estimator derivations, teacher input-gradient computation, and numerical implementation details. Appendix B.1 specifies the main model initializations, training data, optimization, generation schedules, proposal sampling, target truncation, and evaluation settings. The remaining sections of Appendix B document benchmark sources, ablation and diagnostic protocols, model-family comparisons, and training-cost accounting.

AI Use Statement

Generative AI tools were used primarily for language editing, manuscript organization, and LaTeX formatting, with additional assistance in interpreting experimental results. The authors take responsibility for the final content.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.09573.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021a.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021b. URL https://arxiv.org/abs/2108.07732.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. SDAR: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025. URL https://arxiv.org/abs/2510.06303.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Codeforces. Codeforces problemset, n.d. URL https://codeforces.com/problemset. Official problem archive; accessed September 25, 2026.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.17891.

Will Grathwohl, Kevin Swersky, Milad Hashemi, David Duvenaud, and Chris Maddison. Oops i took a gradient: Scalable sampling for discrete distributions. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3831–3841, 2021. URL https://proceedings.mlr.press/v139/grathwohl21a.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-Eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023. URL https://arxiv.org/abs/2305.08322.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the scaling limits of LLMs for logical reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37889–37905, 2025. URL https://proceedings.mlr.press/v267/lin25i.html.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026. URL https://arxiv.org/abs/2601.08584.

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your LLMs capable of stable reasoning? *arXiv preprint arXiv:2412.13147*, 2024. URL https://arxiv.org/abs/2412.13147.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Mathematical Association of America. American invitational mathematics examination (AIME): 2024 and 2025 competitions, 2025. URL https://maa.org/maa-invitational-competitions/. Official competition source.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. URL https://arxiv.org/abs/2502.09992.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

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. *arXiv preprint arXiv:2406.07524*, 2024. URL https://arxiv.org/abs/2406.07524.

Xingyu Su, Jacob Helwig, Shubham Parashar, Atharv Chagi, Lakshmi Jotsna, Degui Zhi, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Data-efficient autoregressive-to-diffusion language models via on-policy distillation, 2026. URL https://arxiv.org/abs/2606.06712.

Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. ENGINE: Energy-based inference networks for non-autoregressive machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2819–2826. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.251. URL https://aclanthology.org/2020.acl-main.251/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. URL https://arxiv.org/abs/2406.01574.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. *arXiv preprint arXiv:2406.19314*, 2024. URL https://arxiv.org/abs/2406.19314.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dLLM v2: Efficient block-diffusion LLM. *arXiv preprint arXiv:2509.26328*, 2025. URL https://arxiv.org/abs/2509.26328.

Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, Shuohang Wang, Weijian Xu, Jianfeng Gao, and Weizhu Chen. Phi-4-Mini-Reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*, 2025a. URL https://arxiv.org/abs/2504.21233.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *International Conference on Learning Representations*, 2025b. URL https://arxiv.org/abs/2410.21357.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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.

Ruixiang Zhang, Shuangfei Zhai, Yizhe Zhang, James Thornton, Zijing Ou, Joshua Susskind, and Navdeep Jaitly. Target concrete score matching: A holistic framework for discrete diffusion, 2025. URL https://arxiv.org/abs/2504.16431.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

A Derivations and Implementation Details

The notation follows Sections 2–3: π_{AR} is the conditional teacher joint, F_θ^i is a student head, G_i is a fixed-background teacher conditional, and $\widehat{G}_i$ is its local approximation. Within fixed-state derivations, π_i abbreviates π_{AR}^i and the conditioning state s is suppressed when unambiguous.

As in the main text, probabilities implicitly condition on the fixed prompt and completed blocks. For the background-based derivations only, let $u = z^{\mathcal{M}_t \setminus \{i\}}$ denote the other masked values and use $r = Q_{\bar{\theta}}(\cdot \mid s)$ for the completion proposal. Its background marginal is $r_{-i}(u) = \prod_{j \in \mathcal{M}_t \setminus \{i\}} F_{\bar{\theta}}^j(u^j \mid s)$. The notation $p_{\mathrm{AR}}(v, u; s)$ scores the full block assembled from token v at i, the other masked values u, and the visible tokens in s. When written as $C_i(u)$ and $G_i(v \mid u)$ below, the exact local quantities are the same $C_i(z)$ and $G_i(v; z)$ from the main text: any compatible z with this background gives the same value. Approximate quantities retain the full completion argument because their Taylor expansion also depends on z^i.

A.1 Conditioning on visible tokens

Here z contains clean tokens, whereas x_t is the masked observation. A mask symbol in x_t marks an unknown clean value; it is not an additional clean-token value to condition z on. The compatible completion space and frozen teacher score are

$$\mathcal{C}(s) = \{z \in \mathcal{V}^L : z^{\mathcal{O}_t} = x_t^{\mathcal{O}_t}\}. \tag{18}$$

$$p_{\mathrm{AR}}(z) = \prod_{j=1}^{L} p_{\mathrm{AR}}(z^j \mid z^{<j}). \tag{19}$$

The conditional joint whose token marginals appear in Equation (9) is

$$\pi_{\mathrm{AR}}(z \mid s) = \frac{p_{\mathrm{AR}}(z)\mathbf{1}[z \in \mathcal{C}(s)]}{Z(s)}, \qquad Z(s) = \sum_{z \in \mathcal{C}(s)} p_{\mathrm{AR}}(z). \tag{20}$$

For a fixed teacher and prefix, this distribution depends only on the visible positions and their token values. States with the same visible content therefore have the same exact teacher target, regardless of the reveal policy or denoising step that produced them. The reveal policy instead affects the distribution ρ of training states in Equation (8). Confidence-based revealing can change which states receive supervision without changing the target definition at a fixed state.

The prefix and visible tokens may be student-generated. We condition the teacher on their realized values and score only the active block.

A.2 What causal-prefix target averaging estimates

The causal-prefix target used for comparison is

$$q_i^{\mathrm{OPDLM}}(v; y) = p_{\mathrm{AR}}(v \mid y^{<i}). \tag{21}$$

Fix a state s and let R_s be the conditional distribution of terminal completions under the student trajectory and state-selection procedure. Write $q_i(v; y) = p_{\mathrm{AR}}(v \mid y^{<i})$ and define

$$\overline{q}_i(v \mid s) = \mathbb{E}_{y \sim R_s}[q_i(v; y)]. \tag{22}$$

For a fixed student prediction $F_i(\cdot \mid s)$, expanding the logarithms gives

$$\mathbb{E}_{R_s} D_{\mathrm{KL}}(q_i(\cdot; y) \| F_i) = D_{\mathrm{KL}}(\overline{q}_i \| F_i) + \mathbb{E}_{R_s} D_{\mathrm{KL}}(q_i(\cdot; y) \| \overline{q}_i). \tag{23}$$

The last term is independent of the fitted prediction. Thus repeated causal-prefix supervision fits $\overline{q}_i$ at the fixed state, rather than necessarily fitting π_{AR}^i. This statement concerns forward-KL fitting with the sampling distribution fixed during the update.

For example, let a teacher on two unknown binary tokens have $P(AA) = P(BB) = 0.45$ and $P(AB) = P(BA) = 0.05$. Its second-token marginal is $P(X^2 = A) = 0.5$. If the terminal student completion has $R_s(Y^1 = A) = 0.9$, causal-prefix averaging instead gives $\overline{q}_2(A) = 0.9(0.9) +$

$0.1(0.1) = 0.82$. This analytic example isolates the discrepancy due to unresolved prefix values. Conditioning on visible right context can additionally change the teacher's conditional probabilities.

The two targets can coincide in special cases. If the only observed content is a complete causal prefix and the predicted position immediately follows it, Equation (9) is the teacher's native next-token distribution. More generally, equality depends on the teacher and the completion distribution; it is not ruled out by the different constructions.

A.3 Optimal parallel predictions and the choice of fitting criterion

Fix a state with $m = |\mathcal{M}_t| > 0$, and let P be its conditional teacher joint over the masked variables. Let P_i be the marginals and $Q(x) = \prod_i q_i(x^i)$ range over all product distributions. Forward KL and expected log loss differ only by the fixed teacher entropy. To characterize their common optimum, write the normalized log-loss risk as

$$R(Q) = -\frac{1}{m}\mathbb{E}_P \log Q(X) = \frac{1}{m}\sum_i \left[H(P_i) + D_{\mathrm{KL}}(P_i \| q_i)\right], \tag{24}$$

where H is Shannon entropy. Each KL is nonnegative and vanishes exactly when $q_i = P_i$. Thus the unique optimum over categorical factors is

$$Q^* = \prod_i P_i. \tag{25}$$

This proves Equation (9). The result uses both the product family and the forward-KL criterion; it does not follow from factorization alone. For an unrestricted joint student, log loss instead has optimum $Q^* = P$. A sum of token losses can also be defined for a nonfactorized model, but it need not fit its joint distribution.

Equivalent forward projection. Log loss satisfies $mR(Q) = H(P) + D_{\mathrm{KL}}(P\|Q)$. The joint divergence further decomposes as

$$D_{\mathrm{KL}}(P\|Q) = \mathbb{E}_P\left[\log P(X) - \sum_i \log q_i(X^i)\right] \tag{26}$$

$$= \mathbb{E}_P\left[\log P(X) - \sum_i \log P_i(X^i)\right] + \sum_i \mathbb{E}_{P_i}\left[\log \frac{P_i(X^i)}{q_i(X^i)}\right] \tag{27}$$

$$= D_{\mathrm{KL}}\left(P \,\middle\|\, \prod_i P_i\right) + \sum_i D_{\mathrm{KL}}(P_i\|q_i). \tag{28}$$

The first term is the teacher's total correlation and does not depend on Q. For $P = \pi_{\mathrm{AR}}(\cdot \mid s)$ and $Q = Q_\theta(\cdot \mid s)$, the constant in Equation (6) equals $|\mathcal{M}_t|^{-1} D_{\mathrm{KL}}(\pi_{\mathrm{AR}} \| \prod_i \pi^i_{\mathrm{AR}})$. Equation (8) averages the normalized joint forward KL under the fixed state distribution ρ. Dropping this constant gives the same fitting gradient as the normalized sum of token forward KLs. Replacing each token KL by cross-entropy additionally drops its fixed teacher entropy. The teacher can have arbitrary dependencies; shared student parameters may prevent attaining the unrestricted optimum at every training state.

For the conditional teacher joint, write $Z_i(v; s) = \sum_u p_{\mathrm{AR}}(v, u; s)$ for the unnormalized mass with token v fixed at i, summing over the other masked positions. Normalizing these masses gives

$$\pi^i_{\mathrm{AR}}(v \mid s) = \frac{Z_i(v; s)}{\sum_{a \in \mathcal{V}} Z_i(a; s)}. \tag{29}$$

At a pointwise optimum of the token fitting objective, the student's log odds therefore satisfy

$$\log \frac{F^i_\theta(v \mid s)}{F^i_\theta(v' \mid s)} = \log \frac{Z_i(v; s)}{Z_i(v'; s)}. \tag{30}$$

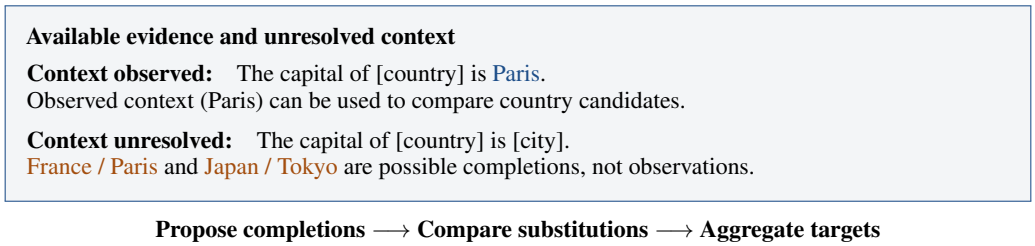


Figure 7: Constructing supervision for partially observed text. Visible tokens are fixed; unresolved tokens are integrated approximately using candidate completions. Slots are schematic, without assuming a particular tokenization or measured teacher probabilities. Comparisons use the current block and committed prefix.

Joint reverse projection. Holding all factors except q_i fixed, minimizing $D_{\mathrm{KL}}(Q\|P)$ yields

$$q_i(v) \propto \exp\left(\mathbb{E}_{x^{-i}\sim q_{-i}}[\log P(v, x^{-i})]\right). \tag{31}$$

This mean-field update generally differs from the marginal in Equation (25). For example, if $P(00) = P(11) = 0.49$ and $P(01) = P(10) = 0.01$, both marginals are uniform. Their product places probability 0.25 on every pair. Under joint reverse KL, a distribution concentrated on 00 already has lower divergence than that uniform product. This also illustrates why matching token marginals alone cannot guarantee joint sample coherence.

Token forward versus reverse KL. Once a fixed token target P_i has been selected, both $D_{\mathrm{KL}}(P_i\|q_i)$ and $D_{\mathrm{KL}}(q_i\|P_i)$ have optimum $q_i = P_i$. For logits a_i with $q_i = \mathrm{softmax}(a_i)$, their gradients differ:

$$\frac{\partial D_{\mathrm{KL}}(P_i\|q_i)}{\partial a_i(v)} = q_i(v) - P_i(v), \tag{32}$$

$$\frac{\partial D_{\mathrm{KL}}(q_i\|P_i)}{\partial a_i(v)} = q_i(v)\left[\log\frac{q_i(v)}{P_i(v)} - D_{\mathrm{KL}}(q_i\|P_i)\right]. \tag{33}$$

Consequently, a comparison between token-level fitting losses should not be interpreted as a comparison between the two joint projections.

Why this criterion, and what on-policy does not determine. We use forward KL to preserve the teacher's conditional token probabilities in the parallel heads. The argument above establishes the optimum of this criterion, not universal superiority of forward KL for generation quality. Reverse KL is a valid alternative objective; at the joint level it changes the product-family optimum, and at the token level it changes the fitting dynamics. Finite capacity, approximate targets, and the downstream sampler can affect which loss performs better.

On-policy specifies the source of training states and does not require either KL direction. GKD explicitly separates student-generated training sequences from the choice of divergence (Agarwal et al., 2024). One reason reverse KL is natural in AR policy optimization is the chain rule: for two AR distributions q and p on the same fixed-length sequence space,

$$D_{\mathrm{KL}}(q(Y)\|p(Y)) = \sum_n \mathbb{E}_{Y^{<n}\sim q} D_{\mathrm{KL}}\left(q(\cdot \mid Y^{<n}) \,\middle\|\, p(\cdot \mid Y^{<n})\right). \tag{34}$$

Thus a student-prefix expectation of reverse token KL can represent a reverse sequence KL in that setting. A student-prefix expectation of forward token KL generally is not forward sequence KL, whose chain rule uses teacher-prefix weights. Neither identity makes reverse KL mandatory for on-policy imitation. Our statewise forward-KL objective fits a factorized clean prediction conditional on a diffusion state; it does not assert an identity with either KL between the terminal teacher and student generation distributions. Differentiating the changing state distribution is a separate issue addressed in Appendix A.9.

A.4 IMPORTANCE IDENTITIES AND THE COMPLETE-SAMPLE UPDATE

For a fixed state, put $H(z) = p_{\mathrm{AR}}(z)$ on $\mathcal{C}(s)$ and let r cover this support. Write $W(z) = H(z)/r(z)$. Multiplying and dividing by r gives

$$\sum_z H(z)f(z) = \sum_z r(z)\frac{H(z)}{r(z)}f(z) = \mathbb{E}_r[W(z)f(z)]. \tag{35}$$

Applying this equality with $f = 1$ yields $\mathbb{E}_r W = Z(s)$ and the normalized identity

$$\mathbb{E}_\pi[f(z)] = \frac{\mathbb{E}_r[W(z)f(z)]}{\mathbb{E}_r[W(z)]}, \qquad W(z) = \frac{p_{\mathrm{AR}}(z)}{r(z)}, \qquad \mathbb{E}_r[W] = Z(s). \tag{36}$$

Ratios of sample averages approximate the corresponding ratio of expectations, but are generally biased for finite K.

For the per-state forward KL ℓ_s in Equation (6), write $r_\theta = Q_\theta$ for the product in Equation (5). Since the conditional teacher is fixed, differentiating the forward KL gives

$$\nabla_\theta \ell_s = -\frac{1}{|\mathcal{M}_t|}\mathbb{E}_\pi[\nabla_\theta \log r_\theta(z \mid s)] = -\frac{1}{|\mathcal{M}_t| Z(s)}\mathbb{E}_r[W(z)\nabla_\theta \log r_\theta(z \mid s)]. \tag{37}$$

Self-normalizing the sampled weights gives the direct complete-sample estimate

$$\widehat{g}_{\mathrm{seq}} = -\frac{1}{|\mathcal{M}_t|}\sum_{k=1}^{K} \overline{W}_k \nabla_\theta \log r_\theta(z^{(k)} \mid s), \qquad \overline{W}_k = \frac{W(z^{(k)})}{\sum_{\ell=1}^{K} W(z^{(\ell)})}. \tag{38}$$

The proposal is fixed during this update, even when $r = r_{\bar{\theta}}$ comes from the current student. Differentiating a finite-sample weighted loss through these weights would add terms absent from Equation (37).

At $K = 1$, the self-normalized weight is one. If $r = r_\theta$ at the update point, the conditional expectation of the resulting complete-sample gradient is zero by $\mathbb{E}_{r_\theta}\nabla_\theta \log r_\theta = 0$. Individual samples can still produce noisy updates. This observation concerns a per-state self-normalized estimator with weights held fixed during differentiation. It does not apply to an estimator using the true $Z(s)$, or to all possible single-sample training objectives.

A.5 CONDITIONAL INTEGRATION AND RAO–BLACKWELLIZATION

At a fixed state, define the local conditional and background correction by

$$C_i(u) = \sum_{a \in \mathcal{V}} p_{\mathrm{AR}}(a, u; s), \qquad G_i(v \mid u) = \frac{p_{\mathrm{AR}}(v, u; s)}{C_i(u)}, \qquad W_i(u) = \frac{C_i(u)}{r_{-i}(u)}. \tag{39}$$

Here u assigns the other masked tokens, $C_i(u)$ is their total teacher mass after summing over token i, and $r_{-i}(u)$ is their proposal probability. For the unnormalized marginal mass introduced in Appendix A.3,

$$Z_i(v) = \sum_u p_{\mathrm{AR}}(v, u; s) = \mathbb{E}_{u \sim r_{-i}}\left[\frac{p_{\mathrm{AR}}(v, u; s)}{r_{-i}(u)}\right]. \tag{40}$$

Proposition 1 (Conditional marginal inference)**.** *Assume positive teacher and proposal probabilities on the finite compatible support. Then*

$$\pi_i(v) = \frac{\mathbb{E}_{r_{-i}}[W_i(u)G_i(v \mid u)]}{\mathbb{E}_{r_{-i}}[W_i(u)]}. \tag{41}$$

For independent backgrounds, the corresponding mass and marginal estimates are

$$\widehat{Z}_i^{\mathrm{exact}}(v) = \frac{1}{K}\sum_k \frac{p_{\mathrm{AR}}(v, u_k; s)}{r_{-i}(u_k)}, \qquad \widehat{\pi}_i^{\mathrm{exact}}(v) = \sum_k \frac{W_i(u_k)}{\sum_\ell W_i(u_\ell)} G_i(v \mid u_k). \tag{42}$$

Let $u = z^{\mathcal{M}_t \setminus \{i\}}$ and write $H_i(v, u) = p_{\mathrm{AR}}(v, u; s)$. With the visible positions fixed,

$$W_i(u)G_i(v \mid u) = \frac{H_i(v, u)}{r_{-i}(u)}. \tag{43}$$

Taking an expectation under r_{-i} gives $Z_i(v)$; summing over v gives $\mathbb{E}_{r_{-i}} W_i = Z(s)$. These establish Equation (41) and the unbiasedness of $\widehat{Z}_i^{\text{exact}}$. The strong law of large numbers applied to the numerator and denominator gives almost-sure convergence of their ratio to $\pi_i(v)$.

The link to the complete-sample estimator follows by conditioning on u. More generally, let $r(v \mid u)$ denote the proposal conditional. Then

$$\mathbb{E}_r[W(z)\mathbf{1}\{z^i = v\} \mid u] = \frac{H_i(v,u)}{r_{-i}(u)}, \tag{44}$$

$$\mathbb{E}_r[W(z) \mid u] = \frac{\sum_a H_i(a,u)}{r_{-i}(u)} = \frac{C_i(u)}{r_{-i}(u)}. \tag{45}$$

The factorized proposal makes $r(v \mid u) = F_{\bar{\theta}}^i(v \mid s)$, but the conditional-expectation identity itself does not require factorization. Sampling a full completion from $Q_{\bar{\theta}}$ induces precisely this background marginal r_{-i}. With exact scores, both $C_i(z)$ and $G_i(v; z)$ are independent of the sampled z^i, so averaging over that reference token contributes a factor of one. This justifies using full completions in the main-text estimator while excluding head i from the proposal denominator. Whenever second moments exist, the law of total variance shows that each conditionally integrated component has no greater variance than its complete-sample counterpart. This is a statement about the unnormalized components. Finite-K ratio error and the cost of obtaining the local scores require separate analysis.

For the gradient of head i, the same calculation replaces the sampled token score by

$$\mathbb{E}_r[W(z)\nabla_\theta \log F_\theta^i(z^i \mid s) \mid u] = W_i(u) \sum_v G_i(v \mid u)\nabla_\theta \log F_\theta^i(v \mid s). \tag{46}$$

This connects the conditional soft target directly to the forward-KL gradient being estimated. In terms of the replacement scores from Equation (12),

$$G_i(v \mid u) = \text{softmax}_v\, \Delta_i(v; z), \qquad C_i(u) = p_{\text{AR}}(z) \sum_{a \in \mathcal{V}} e^{\Delta_i(a;z)}. \tag{47}$$

Both expressions are independent of the original token at position i when the scores are exact.

The same correction in the fitting loss. Multiplying Equation (41) by the negative student log probability and summing over the vocabulary gives

$$-\sum_v \pi_i(v) \log F_\theta^i(v \mid s) = \frac{\mathbb{E}_{r_{-i}}\left[W_i(u)\left(-\sum_v G_i(v \mid u) \log F_\theta^i(v \mid s)\right)\right]}{\mathbb{E}_{r_{-i}}[W_i(u)]}. \tag{48}$$

Thus importance correction can equivalently be viewed as estimating the teacher-averaged cross-entropy through weighted conditional losses. Only the backgrounds are sampled in this expression; the token expectation is summed explicitly. Substituting self-normalized sample weights and local approximations gives Equation (17). No second token-level teacher-to-student ratio is required. This identity is exact for the population quantities; a finite-sample ratio and approximate local scores need not give an unbiased loss or gradient.

An analytic illustration. For the binary teacher in Appendix A.2, let both tokens be unknown and $r_2(A) = 0.9$. Averaging the exact first-token conditionals without correction gives $0.9(0.9) + 0.1(0.1) = 0.82$. Background importance correction recovers the teacher marginal 0.5 in the population calculation.

A.6 Suffix differentiation and local likelihood approximation

Local gradient approximations have been used for discrete sampling (Grathwohl et al., 2021) and for clean-space score estimation (Zhang et al., 2025). We give the derivation here to specify the exact current-token term, the approximate suffix term, and the computations shared across supervised positions.

Fix the anchor token labels $z^1, \ldots, z^L$, and view the teacher-forced log likelihood as a function of its input embeddings. In a causal decoder, the likelihood term for z^j depends on embeddings at positions

strictly before j. Therefore

$$\nabla_{e_{z^i}} \log p_{\mathrm{AR}}(z) = \nabla_{e_{z^i}} \sum_{j>i} \log p_{\mathrm{AR}}(z^j \mid z^{<j}) = g_i(z). \tag{49}$$

The target IDs used in the likelihood are constants in this differentiation. The exact change of the label at position i is evaluated separately by the current-token likelihood ratio.

Let $\ell_{>i}(e; z)$ be the suffix log likelihood with the input embedding at position i replaced by e and all other inputs fixed. For $\delta = e_v - e_{z^i}$,

$$\ell_{>i}(e_{z^i} + \delta; z) - \ell_{>i}(e_{z^i}; z) = g_i(z)^\top \delta + R_i(v; z). \tag{50}$$

The exact substitution score consequently equals

$$\Delta_i(v; z) = \widehat{\Delta}_i(v; z) + R_i(v; z). \tag{51}$$

If the suffix function is twice differentiable along the segment and its Hessian operator norm is bounded there by L_i, the standard Taylor remainder satisfies $|R_i(v; z)| \le L_i \|\delta\|^2/2$. Discrete token embeddings need not be close, so the approximation alone does not guarantee a small remainder or preservation of the exact substitution ranking.

If $\widehat{\Delta}_i = \Delta_i + \xi_i$, the estimated probability ratio differs from the exact ratio by the multiplicative factor e^{ξ_i}. Local softmax normalization also couples errors across vocabulary entries. Increasing K averages over more anchors but does not eliminate systematic errors in these local scores.

Computational accounting. Let m be the number of supervised positions, $V = |\mathcal{V}|$, and d the teacher embedding dimension. For one anchor, a naive exact procedure separately evaluates up to mV single-token replacements. The cached prefix can be shared, but suffixes affected by distinct replacements generally require distinct computations. A last-position replacement is a special case: its suffix is empty and the exact teacher logits already suffice.

Let C_{F} denote the cost of a full-block teacher forward pass and C_{B} the cost of a backward pass to its input embeddings. Recomputing each replacement in full costs $O(KmVC_{\mathrm{F}})$ for K anchors. The gradient approximation instead costs

$$O(K(C_{\mathrm{F}} + C_{\mathrm{B}} + mVd)) \tag{52}$$

for local scoring, followed by $O(KmV)$ arithmetic for normalization and aggregation. These expressions exclude student rollout and proposal generation, and do not represent a measured wall-clock speedup. Batching, prefix caching, and vocabulary truncation affect practical costs. Computing all input gradients uses one backward pass per anchor, with teacher parameters frozen. The mVd term is the projection onto vocabulary embeddings; the method avoids repeated transformer evaluations without removing vocabulary-dependent arithmetic.

A.7 The factorized local surrogate

For a fixed anchor $z^{(k)}$, abbreviate the approximate local conditional as $\widehat{G}_{ik}(v) = \widehat{G}_i(v; z^{(k)})$. On compatible completions, define the unnormalized score

$$\widetilde{H}_k(x) = p_{\mathrm{AR}}(z^{(k)}) \exp\left[\sum_{i \in \mathcal{M}_t} \widehat{\Delta}_i(x^i; z^{(k)})\right]. \tag{53}$$

Normalizing this score gives

$$\widetilde{P}_k(x) = \frac{\prod_{i \in \mathcal{M}_t} \exp \widehat{\Delta}_i(x^i; z^{(k)})}{\prod_{i \in \mathcal{M}_t} \sum_{a \in \mathcal{V}} \exp \widehat{\Delta}_i(a; z^{(k)})} = \prod_{i \in \mathcal{M}_t} \widehat{G}_{ik}(x^i). \tag{54}$$

The anchor likelihood cancels. This surrogate agrees with the chosen approximate single-token log-likelihood changes around $z^{(k)}$. Simultaneous changes omit interactions across positions. The method can supervise many positions through singleton marginals. An explicit joint target over M positions has $|\mathcal{V}|^M$ entries; factorization of this local surrogate does not recover the original teacher's

joint dependencies. Its marginals are therefore marginals of the local surrogate, not necessarily of the original teacher.

Even with exact single-token scores, constructing this additive log-score surrogate at one anchor is distinct from projecting the original joint distribution. Moreover, the weights w_{ik} depend on the supervised position. The marginal aggregation in Equation (15) is not generally a mixture of these joint surrogates with one common set of anchor weights.

A.8 Aggregation, tempering, and approximation limits

The practical mass estimate behind Equation (15) follows from replacing the teacher score in Equation (40) by the local approximation:

$$\widehat{Z}_i(v) = \frac{1}{K}\sum_k \exp\Big[\log p_{\mathrm{AR}}(z^{(k)}) + \widehat{\Delta}_i(v; z^{(k)}) - \log r_{-i}(u_k)\Big]. \tag{55}$$

For this derivation, let $B_{ik} = \sum_{a\in\mathcal{V}} e^{\widehat{\Delta}_i(a;z^{(k)})}$ denote the local softmax normalizer. The estimated background mass from Section 3.1 is $\widehat{C}_i(z^{(k)}) = p_{\mathrm{AR}}(z^{(k)})B_{ik}$, and $\widehat{G}_{ik}$ abbreviates the corresponding local conditional. Therefore

$$\widehat{Z}_i(v) = \frac{1}{K}\sum_k \frac{p_{\mathrm{AR}}(z^{(k)})B_{ik}}{r_{-i}(u_k)}\frac{e^{\widehat{\Delta}_i(v;z^{(k)})}}{B_{ik}} = \frac{1}{K}\sum_k w_{ik}\widehat{G}_{ik}(v). \tag{56}$$

Normalizing $\widehat{Z}_i(v)$ over v gives the mixture in Equation (15). The factor B_{ik} must remain in the background weight even though it cancels within each unnormalized summand.

Finite-budget variants. A possible weight-concentration variant is

$$\widehat{\pi}_i^{(\beta)}(v) = \sum_k \mathrm{softmax}_k(\beta \log w_{ik})\widehat{G}_{ik}(v), \qquad 0 \le \beta \le 1. \tag{57}$$

Here $\beta = 1$ gives Equation (15), while $\beta = 0$ averages the candidate probabilities. Separately, a multiplier $\gamma \in [0, 1]$ controls the suffix correction:

$$\widehat{\Delta}_i^{(\gamma)}(v; z) = \log \frac{p_{\mathrm{AR}}(v \mid z^{<i})}{p_{\mathrm{AR}}(z^i \mid z^{<i})} + \gamma\, g_i(z)^\top (e_v - e_{z^i}). \tag{58}$$

The main construction uses $\gamma = 1$. At $K = 1$ and $\gamma = 0$, the normalized local target is $p_{\mathrm{AR}}(v \mid z^{<i})$. Using the terminal active-block completion as that single candidate recovers the OPDLM per-position target. This comparison concerns target construction, not the full published training schedule. A terminal completion does not have the one-step proposal probability in Equation (5).

For exact scores, the original anchor token cancels from both G_i and W_i. In the large-K limit, tempering gives

$$\pi_i^{(\beta)}(v) = \frac{\mathbb{E}_{r_{-i}}[W_i(u)^\beta G_i(v \mid u)]}{\mathbb{E}_{r_{-i}}[W_i(u)^\beta]} = \mathbb{E}_{u\sim\nu_{i,\beta}}[G_i(v \mid u)], \tag{59}$$

$$\nu_{i,\beta}(u) \propto r_{-i}(u)^{1-\beta}\pi_{-i}(u)^\beta. \tag{60}$$

Thus $\beta = 0$ uses proposal backgrounds and $\beta = 1$ uses teacher backgrounds. Intermediate values interpolate their negative log probabilities, up to normalization. This changes the population target, not only the sampling variance.

With Taylor scores, the local approximation can depend on the original token z^i, even when the background u is unchanged. Writing $w_i(z) = \widehat{C}_i(z)/r_{-i}(z^{\mathcal{M}_t\setminus\{i\}})$ for that anchor's background weight, the limiting target must be expressed as an expectation over the full anchor:

$$\widetilde{\pi}_i^{(\beta)}(v) = \frac{\mathbb{E}_{z\sim r}[w_i(z)^\beta \widehat{G}_i(v; z)]}{\mathbb{E}_{z\sim r}[w_i(z)^\beta]}. \tag{61}$$

At fixed proposal and finite support this is the limit of the practical estimator. It need not equal the teacher marginal at $\beta = 1$, nor obey the exact-score background interpolation in Equation (60).

At $\beta = 0$, averaging fixed probability targets gives the same student gradient as averaging the corresponding K single-anchor forward-KL losses, for the same positions and normalization. Averaging logits is a different operation. At $K = 1$, normalized anchor weights are always one.

A.9 FIXED-TARGET FITTING AND STATE-DISTRIBUTION DERIVATIVES

The practical fitting loss, allowing the weighting variant above, is

$$\widehat{\mathcal{L}}(\theta; s) = -\frac{1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \sum_{v \in \mathcal{V}} \mathrm{sg}\left[\widehat{\pi}_i^{(\beta)}(v \mid s)\right] \log F_\theta^i(v \mid s). \tag{62}$$

Here sg denotes stop-gradient: all quantities inside the constructed target are treated as constants for the student fitting update. Freezing teacher parameters alone does not impose this convention on proposal probabilities computed by the student; these probabilities enter the importance weights. Teacher input gradients used to construct the local scores are computed before this fitting step. Equation (8) holds the state distribution ρ fixed while optimizing θ. The current snapshot $\bar{\theta}$, used for proposals in Section 3.2, is fixed in the same update. The proposal is $Q_{\bar{\theta}}(\cdot \mid s)$ regardless of the source of s. In the on-policy instantiation, write $d_{\bar{\theta}}$ for the state distribution obtained from that snapshot; then $\rho = d_{\bar{\theta}}$. With the targets held fixed, Equation (62) is cross-entropy and has the same student gradient as token forward KL. Equation (62) implements the resulting fitting update by differentiating only the student log probabilities. Self-normalization and approximate local scores mean its finite-sample gradient need not be unbiased for the ideal loss.

Let Γ_θ denote the distribution of student decoding trajectories τ and $\nu(\cdot \mid \tau)$ a rule for selecting a nonterminal state from a trajectory. On-policy state collection samples $\tau \sim \Gamma_{\bar{\theta}}$ followed by $s \sim \nu(\cdot \mid \tau)$. If one instead defines an objective with a parameter-dependent trajectory distribution,

$$J_{\mathrm{full}}(\theta) = \mathbb{E}_{\tau \sim \Gamma_\theta} \mathbb{E}_{s \sim \nu(\cdot \mid \tau)}[\ell_s(\theta)], \tag{63}$$

then, for a fixed state-selection rule ν and under regularity conditions,

$$\nabla_\theta J_{\mathrm{full}} = \mathbb{E}_{\tau,s}[\nabla_\theta \ell_s(\theta)] + \mathbb{E}_\tau \left[\nabla_\theta \log \Gamma_\theta(\tau)\, \mathbb{E}_{s \sim \nu(\cdot \mid \tau)} \ell_s(\theta)\right]. \tag{64}$$

Our update omits the second term. If ν itself depends explicitly on the parameters, its derivative also has to be included in a full derivative. Pointwise constants in the joint-to-token KL decomposition, and the teacher entropies dropped when using cross-entropy, are harmless for fixed-state fitting but may matter when the state distribution is differentiated.

A.10 SUPPORT, TRUNCATION, AND NUMERICAL COMPUTATION

The exact identities use a declared common token space and proposals with positive probability wherever the target mass is positive. Mask symbols are observations, not candidate clean tokens. EOS handling and the active block's support must agree across scoring, proposals, and supervision. If the teacher scores are restricted to a declared token space, Equation (20) normalizes those scores on compatible completions.

The basic proposal samples the current denoiser independently at each masked position at unit temperature. Any change to temperature, top-token sampling, or other support rules changes the proposal and must be reflected in its probability. A terminal rollout generally has a different density from Equation (5); its probability cannot be replaced by the product of the current head probabilities.

Vocabulary truncation is an optional approximation to the local sum. For the anchor $z^{(k)}$, let $\widehat{G}_{ik}(v) = \widehat{G}_i(v; z^{(k)})$ and let B_{ik} be the local normalizer defined in Appendix A.8. If only a set $\mathcal{S}_{ik}$ is evaluated, both the local conditional and its normalizer must use that same set:

$$B_{ik}^{\mathcal{S}} = \sum_{v \in \mathcal{S}_{ik}} e^{\widehat{\Delta}_i(v; z^{(k)})}, \qquad \widehat{G}_{ik}^{\mathcal{S}}(v) = \frac{\mathbf{1}[v \in \mathcal{S}_{ik}] e^{\widehat{\Delta}_i(v; z^{(k)})}}{B_{ik}^{\mathcal{S}}}. \tag{65}$$

The background weight must use the same restricted normalizer multiplied by the anchor likelihood. A fixed restricted completion space defines a restricted target. Anchor-dependent candidate sets generally introduce additional approximation error and do not automatically correspond to one fixed restriction of the original teacher. More anchors cannot recover support that the procedure never evaluates or proposes.

In the main experiments, top-16 sparsification is applied to the final token target used in the loss, after completion aggregation for CrossDLM. Retained target probabilities preserve their original mass, while student probabilities retain normalization over the full valid vocabulary. This sparse

forward-KL approximation differs from restricting and renormalizing each anchor's local conditional as above; the experimental settings are given in Appendix B.1.

For numerical evaluation, compute block likelihoods, proposal probabilities, and local normalizers in log space. Equation (16) becomes $\log w_{ik} = \log p_{\mathrm{AR}}(z^{(k)}) + \log B_{ik} - \log r_{-i}(u_k)$. Normalize these log weights across k with log-sum-exp. Teacher logits and input gradients can be batched over anchors. Vocabulary projections still require arithmetic over the evaluated candidates; the saving is the avoidance of a separate teacher transformer evaluation for every replacement.

A.11 Relation to the general distillation framework

The GKD objective in Equation (1) separates sequence sourcing from the fitting divergence (Agarwal et al., 2024). Its mixture μ_λ combines offline responses μ_{off} with responses generated by the current student, μ_{on}. In Section 2, the analogous sampling choice is the distribution ρ of masked states, while the statewise loss ℓ_s specifies the fitting criterion. A response distribution alone does not specify ρ: constructing masked states also requires a masking scheme or a student decoding process. The conditional-target estimator introduces a third distribution, the one-step completion proposal $Q_{\bar{\theta}}$ in Section 3.2 (abbreviated r in these derivations). It supplies compatible completions at an already selected state, whereas μ_λ supplies training sequences and ρ supplies training states. The present experiments choose on-policy states and forward KL; the conditional target construction itself does not require that state source.

A.12 Training procedure

Algorithm 1 CrossDLM training update

Inputs: student θ, frozen AR teacher, state source ρ, number of completions K.

1. Set $\bar{\theta} \leftarrow \theta$; select a state $s \sim \rho$ (on-policy in the experiments).
2. Evaluate the student at s; draw $z^{(1)}, \ldots, z^{(K)} \sim Q_{\bar{\theta}}(\cdot \mid s)$ independently.
3. For each completion, compute its teacher likelihood and Equation (13).
4. Form $\widehat{G}_i$ and $\widehat{C}_i$; aggregate using Equations (15)–(16).
5. Fit the student using Equation (17), holding states and targets fixed.

B Experimental Protocols and Additional Diagnostics

B.1 Main experiment setup

The main comparisons train Qwen3.5-4B and Qwen3.5-9B with OPDLM's causal-prefix targets or CrossDLM's conditional targets. At each scale, initialization, training data, optimization, rollout settings, and evaluation are matched between the two methods. The pretrained checkpoint initializes both the frozen AR teacher and the diffusion student; each student collects its own on-policy states. Table 3 summarizes the shared settings and the differences in target construction. Model-family comparisons and Qwen3-4B ablations are described separately in Appendices B.8 and B.4.

The training corpus follows OPDLM (Su et al., 2026) and contains 61,816 prompts: 20,222 mathematics examples from DAPO and Nemotron-v2-Math; 21,594 coding examples from TACO, KodCode-Light-RL, and AceCode; 10,000 science examples from Nemotron-v2-STEM; and 10,000 chat examples from Nemotron-v2-Chat. Only the prompts are retained; responses are generated on-policy.

Table 3: Main-experiment settings. Unless specified otherwise, entries apply to both Qwen3.5-4B and Qwen3.5-9B. A merged entry is shared by OPDLM and CrossDLM.

Setting	OPDLM	CrossDLM
Optimization and training budget		
Training prompts	OPDLM mixture (61,816 prompts)	
Optimizer	AdamW, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$	
Weight decay / gradient clipping	0 / clipping threshold 1	
Learning rate	3×10^{-7}, constant after warmup	
Warmup	2 optimizer updates	
Global optimization batch size	8	
Prompts per rollout batch	128	
Updates per rollout batch	16	
Training budget	1 epoch; 7,728 optimizer updates	
Generation and supervision		
Block size	4	
Training reveal threshold	0.9; highest-confidence fallback	
Teacher target	Causal-prefix	Bidirectional (Taylor)
Completion source	Terminal rollout	One-step, $K = 4$
Target temperature	1	
Sparse loss support	Top-16 of the final token target	
Importance-weight tempering	—	None
Evaluation		
Decoding	Greedy; reveal threshold 0.95	
Prompting	Zero-shot; native Qwen3.5 chat template	

Rollout and denoising. The rollout-length cap increases from 100 to 4,000 tokens over the first 100 rollout batches, then remains at 4,000. Domain-specific caps are 4,000 for mathematics, 2,048 for code and chat, and 1,024 for science; the smaller applicable cap is used. Generation stops at the first chat end-of-sequence token. At each training step, positions with confidence above 0.9 are revealed; if none qualifies, the highest-confidence position is revealed.

Completion proposals and target construction. CrossDLM draws four independent one-step completions at temperature 1 from the full valid vocabulary, without top-k or nucleus truncation. Both the 4B and 9B runs use untempered importance weights and unscaled suffix corrections ($\beta = \gamma = 1$ in Appendix A.8), with no clipping of either quantity. The gradient-clipping threshold of 1 applies only to model parameter gradients.

Sparse teacher supervision. Both methods retain the 16 highest-probability entries of the final teacher target at each supervised position. For OPDLM, these are selected from the causal teacher

prediction; for CrossDLM, selection follows aggregation of the four Taylor-based completion targets. The retained probabilities are not renormalized, and student probabilities are normalized over the full valid vocabulary. The resulting loss is a sparse approximation to the token-level forward KL. This final-target truncation is distinct from restricting the local candidate sums before aggregation (Appendix A.10).

Evaluation. Evaluation uses four-token blocks with greedy token selection. Positions above confidence 0.95 are revealed together; otherwise, the highest-confidence position is revealed. Mathematics prompts request a boxed final answer, and multiple-choice prompts request the answer letter directly. HumanEval and MBPP use EvalPlus base pass@1; LCB-v6 and Codeforces use execution-based pass@1. Other tasks use their benchmark scorers. Table 4 lists output limits, and Appendix B.3 provides benchmark references.

Table 4: Evaluation output limits by benchmark group, shared across both methods and model sizes in the Qwen3.5 main comparisons.

Benchmarks	**Maximum tokens**
GSM8K, LiveBench	4,096
MATH-500, AIME-24/25, GPQA-Diamond, LMB-Hard	2,048
Other multiple-choice benchmarks	1,000
IFEval	1,280
HumanEval-base, MBPP-base	1,024
LCB-v6, Codeforces	16,000

Token accounting. The token counts in Table 1 sum the response tokens actually generated for training prompts. They exclude prompts and padding, do not count a generated response again when reused for optimization, and exclude the additional completion proposals used to construct CrossDLM targets. Counts are rounded to 0.10B for all four runs. Appendix B.7 reports the available timing measurements.

B.2 Baseline provenance and training-token scope

For the OPDLM baseline, Qwen3.5-4B and Qwen3.5-9B are trained with causal-prefix targets derived from terminal rollout completions (Su et al., 2026). These runs use the same pretrained initialization, prompt data and order, optimization schedule, rollout settings, and evaluation protocol as the corresponding CrossDLM runs. Each student collects its own on-policy states; Table 3 specifies the target construction for each method. OPDLM and CrossDLM scores and token counts in Table 1 are from these experiments. All other baseline values—SDAR-4B/8B, LLaDA-8B, Dream-7B, and Fast-dLLM-v2-7B—are transcribed from Table 1 of Su et al. (2026). The independent model-family comparisons in Figure 5 are also our experiments.

Figure 3 uses the same GSM8K scores and reference token counts as Table 1. The plotted OPDLM-4B and CrossDLM-4B counts are 0.098198B and 0.100333B generated response tokens, respectively, displayed as approximately 0.10B in the table. LLaDA is trained from scratch (Nie et al., 2025); Dream, SDAR, and Fast-dLLM v2 adapt pretrained AR models (Ye et al., 2025; Cheng et al., 2025; Wu et al., 2025). For the AR-initialized models, the displayed token budget covers the reported adaptation or distillation stage and excludes the original AR pretraining. The LLaDA value is the published pretraining count transcribed from OPDLM. These stage-specific counts illustrate different routes to obtaining a DLM, rather than total lifecycle training cost. Our rollout counts exclude auxiliary proposals, as detailed in Appendix B.7; the measured training time includes the target-construction workload.

B.3 Benchmark sources

Table 5 collects the original papers and official sources for the benchmarks in Table 1. The same references apply to the cross-model comparisons, component ablations, and diagnostic question sets. Benchmark names and variants follow the reported evaluations: MATH-500 is the 500-problem

MATH evaluation subset, LMB-Hard denotes LiveMathBench-Hard, and LCB-v6 denotes LiveCodeBench release v6.

Table 5: Benchmark references. AIME and Codeforces are cited through their official competition sources.

Benchmark	Source
MMLU	Hendrycks et al. (2021a)
MMLU-Pro	Wang et al. (2024)
GPQA-Diamond	GPQA (Rein et al., 2023)
IFEval	Zhou et al. (2023)
CEval	C-Eval (Huang et al., 2023)
LiveBench	White et al. (2024)
GSM8K	Cobbe et al. (2021)
MATH-500	MATH (Hendrycks et al., 2021b); evaluation subset (Lightman et al., 2024)
AIME-24 / AIME-25	Mathematical Association of America (2025)
LMB-Hard	LiveMathBench (Liu et al., 2024)
ZebraLogic	Lin et al. (2025)
HumanEval-base	HumanEval (Chen et al., 2021)
MBPP-base	MBPP (Austin et al., 2021b)
LCB-v6	LiveCodeBench (Jain et al., 2024)
Codeforces	Codeforces (n.d.)

B.4 Component-ablation setup

The three configurations in Table 2 use Qwen3-4B, with the same teacher checkpoint, student initialization, prompt pool, optimization schedule, training-update count, and rollout and evaluation settings. Each student collects its own on-policy states. The table starts from CrossDLM and removes components cumulatively: first replace one-step proposals with terminal completions while retaining Taylor scoring, then remove suffix scoring to recover OPDLM. The effect of completion count is examined separately using the frozen-state marginal-estimation diagnostic in Figure 6 and Appendix B.6.

Completion sources. Both sources preserve the visible tokens at the selected state. A terminal completion incorporates the student's subsequent denoising decisions; a one-step completion samples the missing positions from $Q_{\bar{\theta}}(\cdot \mid s)$ without further denoising. In both cases, sampled values for missing tokens provide computational backgrounds rather than additional observed evidence. One-step completions additionally have tractable product probabilities for importance-weighted aggregation. Appendix A.2 gives the formal relation between terminal completions, state selection, and the teacher target.

Detailed component comparisons. Replacing one-step proposals with terminal completions changes GSM8K, MATH-500, and MBPP-base by -0.28, -2.34, and -1.59 percentage points, respectively. Further removing Taylor suffix scoring changes these scores by -1.46, -0.86, and -0.52 points, recovering OPDLM. CrossDLM improves over OPDLM by $+1.74$, $+3.20$, and $+2.11$ points overall. The gains occur at both stages of the evaluated comparison. These sequential comparisons do not estimate interactions across every possible target–completion combination.

B.5 Evaluating the suffix approximation

The gradient in Equation (13) describes the teacher's response near the original token embedding. A complete token replacement can move far from that point, where higher-order terms may matter. Small perturbations therefore test the accuracy of the first-order approximation near the sampled completion; larger displacements examine how this accuracy changes toward complete token substitution.

Evaluation setup. The diagnostic uses frozen Qwen3-0.6B and Qwen3-4B checkpoints with four-token blocks, and a Qwen3-4B checkpoint with 16-token blocks. Each setting uses the same 256 GSM8K training-side validation questions, with four saved states per question. With four-token blocks, all masked prediction positions are evaluated. With 16-token blocks, one position per state is selected, balanced across fixed suffix lengths of 0, 3, 7, and 15 tokens. Suffix length counts all subsequent tokens in the fixed completion, including sampled tokens. Positions with an empty suffix are excluded from the error comparison because both scoring rules are exact there. The checkpoints and state distributions differ across settings; their results test the approximation in several configurations rather than isolate the effect of block length.

Embedding perturbations. For completion z, let $\ell_i(e; z)$ be the teacher log-likelihood of the fixed suffix $z^{>i}$ when the input embedding at position i is e, holding all other inputs fixed. For candidate token v, the perturbed embedding and exact suffix response are

$$e_i(\alpha, v) = e_{z^i} + \alpha(e_v - e_{z^i}), \qquad d_i(v, \alpha; z) = \ell_i(e_i(\alpha, v); z) - \ell_i(e_{z^i}; z). \tag{66}$$

Here α controls only the diagnostic perturbation: $\alpha = 0$ leaves the original embedding unchanged and $\alpha = 1$ gives a complete token substitution. Intermediate values measure the local approximation without changing the training algorithm. Causal-prefix scoring omits the suffix response, whereas Taylor scoring estimates it using the input gradient:

$$\widehat{d}_i^{\mathrm{CP}}(v, \alpha; z) = 0, \qquad \widehat{d}_i^{\mathrm{Taylor}}(v, \alpha; z) = \alpha g_i(z)^\top (e_v - e_{z^i}). \tag{67}$$

The current-token log-probability is evaluated exactly by both rules, so their scoring errors concern only the suffix. Exact rescoring uses FP32. The displacement grid is $\alpha \in \{0, .01, .03, .1, .25, .5, .75, 1\}$, with $\alpha = .1$ shown in Figure 4.

Error metric. Each position uses the 16 candidates with the largest recorded target probabilities. Let $\mathcal{S}_{qi}$ denote this set for question q and position i, and let ω_{qiv} be those probabilities without renormalizing within the set. The weighted error for scoring rule m is

$$L_{m,qi}(\alpha) = \left[\sum_{v \in \mathcal{S}_{qi}} \omega_{qiv} \big(\widehat{d}_i^m(v, \alpha; z) - d_i(v, \alpha; z) \big)^2 \right]^{1/2}. \tag{68}$$

Errors are averaged first over eligible positions within a question, then equally over questions. Relative reduction is $1 - \overline{L}_{\mathrm{Taylor}} / \overline{L}_{\mathrm{CP}}$. Paired 95% bootstrap intervals resample questions, keeping their states and positions together. The metric measures suffix log-likelihood error on the selected candidates, rather than a distance between full-vocabulary token distributions.

From small perturbations to token substitution. Figure 8 compares Taylor and causal-prefix suffix-scoring errors across displacement sizes. Taylor has lower error at small displacements in all three settings. Training uses complete token substitutions with the standard, unscaled first-order suffix correction. Calibrating the strength of this correction to account for nonlinear suffix responses is a potential direction for improving approximation accuracy for complete token substitutions.

B.6 Accuracy of completion aggregation

This diagnostic examines how the number of completion proposals affects the estimated token marginal. Unlike the embedding perturbations above, it uses complete token replacements and enumerates a small candidate space to obtain reference distributions.

States and reference distributions. The diagnostic is conducted separately for Qwen3-0.6B and Qwen3-4B, using frozen states collected from each model's own student on GSM8K validation prompts.

Each position in a four-token block has three retained candidate tokens. Enumerating all 3^4 block completions, scoring them with the teacher, and marginalizing over the other positions gives the exact token marginal on this restricted support. Exhaustive aggregation with Taylor scores gives a second reference that retains the scoring approximation but removes completion-sampling error. Finite-K targets use the same Taylor scoring and importance aggregation as the method. For each prediction state, 512 repeated sets of four completions are drawn from the factorized proposal on the retained support; their first one, two, or four completions give paired estimates for $K = 1, 2, 4$.

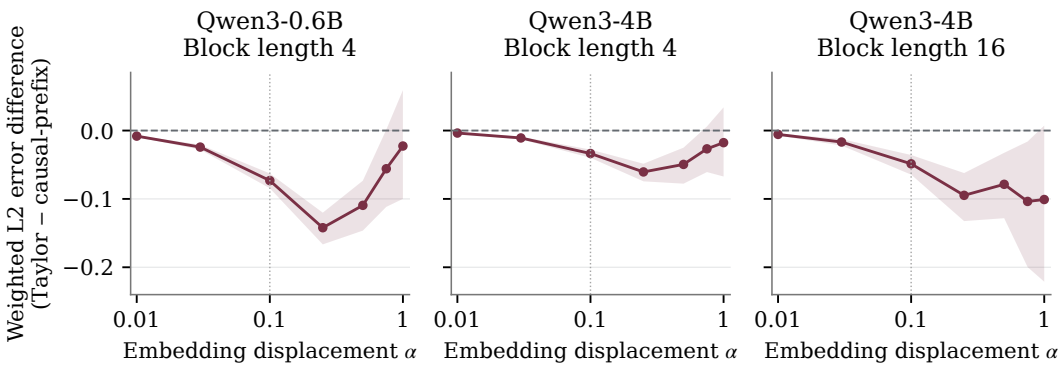


Figure 8: Difference in suffix-scoring error between Taylor and causal-prefix scoring across embedding displacements. Values are differences in weighted L2 error (Taylor minus causal-prefix), not percentage reductions; negative values favor Taylor. Shaded bands show paired 95% question-bootstrap intervals. The horizontal dashed line marks zero difference; vertical dotted lines mark the main-text setting ($\alpha = .1$). At $\alpha = 1$, the token is completely substituted. Horizontal axes are logarithmic and vertical axes are linear. The zero-displacement control is omitted.

Sampling and total error. Figure 9 measures error using total variation (TV), one half of the summed absolute differences between token probabilities. Sampling error compares a finite-K Taylor target with exhaustive Taylor aggregation. Total error compares the same target with the restricted exact teacher marginal. The two distances have different references and are not additive. Results are averaged over repeated draws within each state and then over questions; uncertainty intervals use 20,000 question-level bootstrap replicates.

Both errors decrease as more completions are included. The Qwen3-4B total-error curve is shown separately in the main text (Figure 6): mean TV falls from 0.1864 at $K = 1$ to 0.1705 at $K = 2$ and 0.1600 at $K = 4$, a reduction of approximately 14%. Figure 9 additionally shows the sampling-error reference and the 0.6B diagnostic. Increasing K improves the sampling approximation while leaving the discrepancy between the exhaustive Taylor and exact references unchanged.

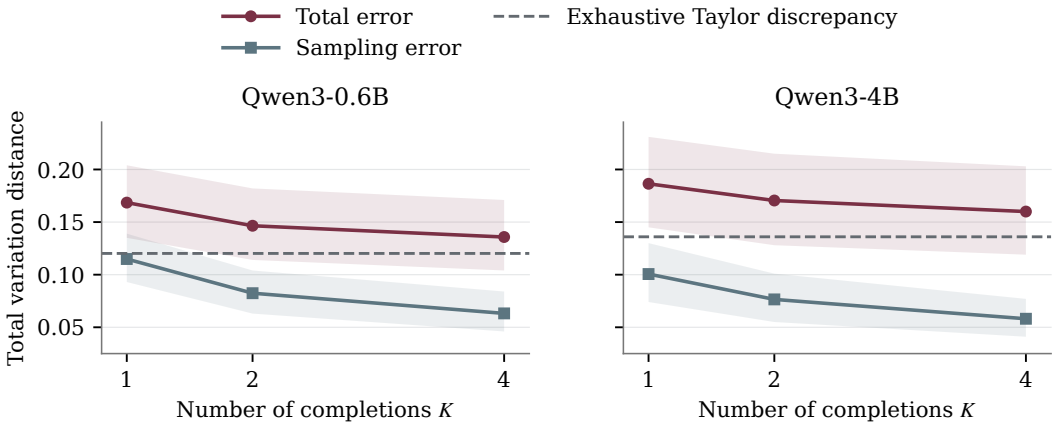


Figure 9: Completion aggregation on states collected separately from each student. Total error uses the restricted exact teacher marginal as reference; sampling error uses exhaustive Taylor aggregation. Shading gives 95% question-bootstrap intervals. The dashed line shows the discrepancy between the two exhaustive references, which does not depend on K and is not a lower bound for every finite-K estimate. All distances are measured on the restricted candidate support.

B.7 Training cost and token accounting

The Qwen3.5-4B runs take 87.15 hours for OPDLM and 90.73 hours for CrossDLM with $K = 4$, a 4.1% increase. Both runs use eight NVIDIA H100 GPUs, BF16 forward computation, and 7,728 optimizer updates. Multiplying elapsed time by eight GPUs gives approximately 697 and 726 allocated GPU-hours, respectively.

The token counts in Table 1 measure generated rollout responses, excluding prompts, padding, repeated optimization, and auxiliary completion proposals. CrossDLM's teacher scoring and input-gradient computations are used during training; generation requires only the distilled student.

B.8 Base model comparisons: scores, protocol, and figure scales

Table 6 reports selected benchmark scores for the three model pairs shown in Figure 5. Each model is distilled independently from its own pretrained AR checkpoint, which initializes the student and supplies the frozen teacher for both objectives. The Qwen3-4B comparison uses four-token blocks.

All three model pairs use a learning rate of 3×10^{-7} for both objectives. For all three model pairs, initialization, training data and order, optimization, decoding, prompts, output limits, and scoring are matched between OPDLM and CrossDLM. The LiveBench results use official active-1000.

Table 6: Selected benchmark scores for the model comparisons in Figure 5. Scores are on a 0–100 scale; Δ denotes CrossDLM minus OPDLM, computed from unrounded scores. Bold marks the higher score within each pair; ties are unbolded.

	Qwen3-4B			**Phi (3.8B)**			**Ministral (8B)**		
Benchmark	OPDLM	CrossDLM	Δ	OPDLM	CrossDLM	Δ	OPDLM	CrossDLM	Δ
GSM8K	85.67	**87.41**	+1.74	66.87	**80.14**	+13.27	84.69	**89.69**	+5.00
MATH-500	72.60	**75.80**	+3.20	44.00	**54.60**	+10.60	64.20	**70.40**	+6.20
AIME-25	3.33	**10.00**	+6.67	0.00	**10.00**	+10.00	3.33	**6.67**	+3.33
MMLU-Pro	45.06	**47.49**	+2.43	**36.96**	36.15	−0.81	53.13	**55.53**	+2.39
LCB-v6	13.00	**14.54**	+1.54	**8.15**	7.71	−0.44	11.89	**18.50**	+6.61
Codeforces	3.18	**4.77**	+1.59	2.39	**3.45**	+1.06	1.06	**1.59**	+0.53
LiveBench	22.50	**22.62**	+0.12	10.79	**11.61**	+0.82	27.06	**29.57**	+2.51

Qwen3-4B: four-token blocks. All three model pairs: matched learning rate 3×10^{-7}. LiveBench: official active-1000. Differences are computed from unrounded scores and may differ slightly from subtracting the displayed values.

Variation across backbones. The effect of conditional supervision depends on the interaction between the pretrained teacher and the adapting student. In self-distillation, each backbone defines both the teacher's joint distribution and the student's predictions at partially observed states. These predictions shape subsequent on-policy training states, coupling target construction with the adaptation trajectory. Consequently, CrossDLM's performance gains vary across base models, reflecting the interaction between each teacher's conditional predictions and its student's adaptation process.

Axis ranges. Each benchmark is mapped linearly from zero to the upper limit in Table 7, using the same range across all three model panels. These ranges make low-scoring tasks such as AIME-25 and Codeforces legible alongside GSM8K; polygon area is not used as an aggregate performance metric. Table 6 provides the numerical scores.

B.9 Detailed Qwen3.5-9B results

Table 8 provides detailed paired scores for the 12 Qwen3.5-9B benchmarks in Table 1, with $K = 4$ completion proposals for CrossDLM.

Table 7: Linear radar ranges, shared across all model panels.

Benchmark	Range	Benchmark	Range
GSM8K	0–100	HumanEval-base	0–80
MATH-500	0–100	LCB-v6	0–20
AIME-25	0–20	Codeforces	0–8
MMLU-Pro	0–60	LiveBench	0–40

Table 8: Detailed Qwen3.5-9B results, with $K = 4$ for CrossDLM. Scores and differences are shown to two decimal places; Δ is CrossDLM minus OPDLM, computed from unrounded scores. Bold marks the higher score within each pair; ties are unbolded.

Benchmark	OPDLM	CrossDLM ($K = 4$)	Δ
MMLU	80.02	**80.15**	+0.13
MMLU-Pro	57.84	**59.48**	+1.65
GPQA-Diamond	42.42	**45.45**	+3.03
IFEval	72.83	**76.34**	+3.51
CEval	**78.23**	78.08	−0.15
LiveBench	42.17	**44.07**	+1.90
GSM8K	85.60	**92.65**	+7.05
MATH-500	70.60	**80.20**	+9.60
AIME-24	16.67	16.67	0.00
AIME-25	16.67	**23.33**	+6.67
LiveMathBench-Hard	8.89	**17.78**	+8.89
ZebraLogic	4.10	**12.60**	+8.50

B.10 Code-generation comparison at the 4B scale

SDAR-4B provides a strong published reference, although its exact training-data mixture is not fully specified (Cheng et al., 2025). The controlled comparison in Table 9 trains the same Qwen3.5-4B model using OPDLM's causal-prefix supervision or CrossDLM's conditional targets under matched settings (Appendix B.1). CrossDLM improves over OPDLM on all four code benchmarks, with gains of 16.5, 10.6, 15.2, and 7.4 percentage points on HumanEval-base, MBPP-base, LCB-v6, and Codeforces, respectively.

Table 9: Code-generation performance at the 4B scale. OPDLM and CrossDLM train Qwen3.5-4B under matched settings; SDAR-4B is a published reference from Table 1 of Su et al. (2026). Scores are percentages; bold marks the best score per benchmark.

Benchmark	SDAR-4B	OPDLM-4B	CrossDLM-4B
HumanEval-base	**76.8**	57.9	74.4
MBPP-base	**80.7**	60.6	71.2
LCB-v6	12.6	14.5	**29.7**
Codeforces	4.0	1.9	**9.3**

C Extended Discussion and Limitations

The central design choice is how to extract useful supervision from an AR teacher when the student predicts from partially observed context. Completed text provides a shared object for scoring; conditioning and marginalization translate that score into token targets. This interpretation separates target construction from the selection of on-policy training states.

The teacher remains causal, and its likelihood is a model preference rather than an external correctness signal. The current method considers the committed prefix and the active block, not all future response

blocks. At a given state, the exact teacher target conditions on visible token values; the reveal schedule determines the distribution of states used for training.

The estimator also has distinct limitations. Factorized instantaneous proposals may poorly cover the teacher's correlated completions. Finite-K importance weights may concentrate, while the Taylor approximation can misestimate large discrete substitutions. More candidates do not guarantee better targets or downstream performance, and weight tempering generally changes the limiting target. Matching token marginals does not by itself preserve joint correlations under simultaneous sampling.

Teacher input gradients and multiple completion proposals add computation during target construction. Section 4.5 reports end-to-end training time at a matched update budget. Forward KL motivates a particular product-family projection; it does not establish that this projection is optimal for every downstream metric.

The broader idea of scoring shared completed objects can inform transfer between different generation processes. Extensions to other corruption kernels or model families require their own observation models, target estimators, and experiments. We restrict the present method to masked/block diffusion and treat these broader settings as future work.

D Extended Related Work

Discrete diffusion language modeling. Discrete diffusion models learn to reverse corruption processes over categorical variables (Austin et al., 2021a). SEDD estimates discrete probability ratios (Lou et al., 2024), while masked diffusion admits a simplified objective expressed through masked prediction losses (Sahoo et al., 2024). LLaDA demonstrates that this paradigm can support large-scale language pretraining and instruction tuning (Nie et al., 2025). Block diffusion combines autoregressive dependence across blocks with parallel denoising within each block (Arriola et al., 2025). These developments establish the modeling and sampling framework within which CrossDLM transfers knowledge from an AR teacher.

Adapting pretrained autoregressive models. Reusing AR checkpoints offers a route to diffusion language modeling without training from scratch. DiffuGPT and DiffuLLaMA use continual pretraining to adapt AR models to diffusion objectives (Gong et al., 2025), and Dream scales AR-initialized diffusion modeling with an adapted training recipe (Ye et al., 2025). SDAR and Fast-dLLM v2 develop blockwise generation and adaptation strategies that combine AR initialization with parallel token prediction (Cheng et al., 2025; Wu et al., 2025). CrossDLM addresses the supervision used during such a conversion: the AR teacher's joint distribution supplies conditional predictions for the student's partially observed text.

Knowledge distillation and on-policy learning. Knowledge distillation transfers predictive distributions (Hinton et al., 2015), while sequence-level distillation trains on teacher-generated outputs (Kim & Rush, 2016). GKD allows offline and student-generated sequences to be combined with different fitting divergences (Agarwal et al., 2024); MiniLLM develops on-policy optimization of reverse KL for AR language-model distillation (Gu et al., 2024). OPDLM applies on-policy self-distillation to AR-to-DLM conversion, supervising student trajectories with causal-prefix predictions from a frozen AR teacher (Su et al., 2026). CrossDLM retains this experimental setting but changes the teacher target to reflect the visible context and unresolved content at each denoising state. The target construction also applies when states are obtained offline.

Sequence scoring and conditional inference. AR sequence scores have previously supported non-autoregressive generation. ENGINE trains an inference network to minimize an AR teacher's sequence energy (Tu et al., 2020), while energy-based diffusion models use sequence-level corrections and importance sampling during generation (Xu et al., 2025b). Gradient-based local approximations enable efficient discrete sampling (Grathwohl et al., 2021). TCSM (Zhang et al., 2025) connects local conditional matching to pseudolikelihood and studies AR-to-diffusion distillation using clean-space scores and Taylor approximations for token substitutions. Its AR-to-diffusion study focuses on offline language modeling (Zhang et al., 2025, Section 5.4 and Appendix H). CrossDLM directly targets the teacher's conditional token marginals at partially observed states. Local conditional predictions are intermediate quantities: the remaining unknown tokens must be integrated out according to their

teacher probability mass. When completions are drawn from the student, an unweighted average follows the proposal rather than the teacher's conditional distribution over unresolved contexts. CrossDLM uses importance-weighted aggregation to account for both teacher mass and proposal frequency when constructing the marginal targets. By separating supervision construction from the source of training states, CrossDLM extends naturally to on-policy distillation. The student learns these targets during training, so generation requires no teacher scoring.