d-OPD: Future-Aware On-Policy Distillation for Block Diffusion Language Models

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) typically generate text autoregressively (AR), predicting one token at a time. Block diffusion language models (dLLMs) instead generate blocks sequentially while denoising multiple tokens in parallel within each block, offering a promising way to accelerate generation. Rather than training such models from scratch, recent work adapts strong pretrained AR models into block dLLMs through distillation. On-policy distillation (OPD) has been widely used for LLM training because it supervises the student on states generated by its current policy, rather than only on fixed offline trajectories. By training on the states the student actually visits, it reduces the mismatch between training and generation and can provide more relevant supervision as the student evolves. Recent work has extended this idea to AR-to-block-diffusion conversion. However, this setting introduces a fundamental mismatch in supervision: the block-diffusion student and the causal AR teacher condition on different information at the same training state. The student predicts from the entire partially denoised block, including visible future context, whereas the standard AR teacher target is defined only from the causal prefix. As a result, the teacher distribution used for distillation is not fully aligned with the information available to the student. We therefore introduce d-OPD, a future-aware on-policy distillation method that corrects the AR teacher distribution to better align with the student-visible state by incorporating visible future information within each block, providing supervision that better matches the information used by the student. Across Qwen3 models from 0.6B to 8B, d-OPD improves the six-benchmark average by up to 4.0 points over OPDLM and reduces training time by 1.35–$1.58\times$.

1 Introduction

Autoregressive (AR) language modeling is the standard approach used by most modern large language models (LLMs) (Feng et al., 2025; Kim et al., 2025; Lin et al., 2025; Yang et al., 2025c; Yu et al., 2025; Blondel et al., 2026; Cheng et al., 2026a; Hatamizadeh et al., 2026; Israel et al., 2026; Li et al., 2026b; Tang et al., 2026). AR models generate one token at a time, with each new token depending on the tokens generated before it. This token-by-token dependency makes inference slow, especially for long outputs. To reduce this sequential bottleneck, diffusion language models (dLLMs) generate multiple token positions in parallel for more efficient inference (Bao et al., 2026; Chen et al., 2026c; Fu et al., 2026a; Hu et al., 2026b; Israel et al., 2025; Labs et al., 2025; Seo et al., 2025; Wu et al., 2026b; Ye et al., 2025; Bertolani et al., 2026; Chen et al., 2026b; Gwak et al., 2026; Hersche et al., 2026; Li et al., 2026c; Qi et al., 2026; Qian et al., 2026; Zhu et al., 2026). Among dLLMs, block diffusion has become a particularly important design because it can better preserve the generation quality of AR models while retaining parallel generation within each block (Arriola et al., 2025; Bie et al., 2025; Cheng et al., 2026b; Liu et al., 2025; Cheng et al., 2026a; Hou & Kwok, 2026; Jin et al., 2026; Hu et al., 2026a; Ma et al., 2026a; Mou et al., 2026; Singh et al., 2026). Block dLLMs generate blocks from left to right, preserving the causal structure across blocks while allowing tokens within the current block to interact bidirectionally.

Rather than training such models from scratch, a particularly attractive direction is to convert existing pretrained AR checkpoints into block dLLMs (Bie et al., 2025; Liu et al., 2025; Wu et al., 2026b; Cheng et al., 2026a; Fu et al., 2026b; Ma et al., 2026b; Tian et al., 2026b; Wang et al., 2026b; Zeng et al., 2026). As strong pretrained AR models contain broad linguistic and task knowledge learned

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.

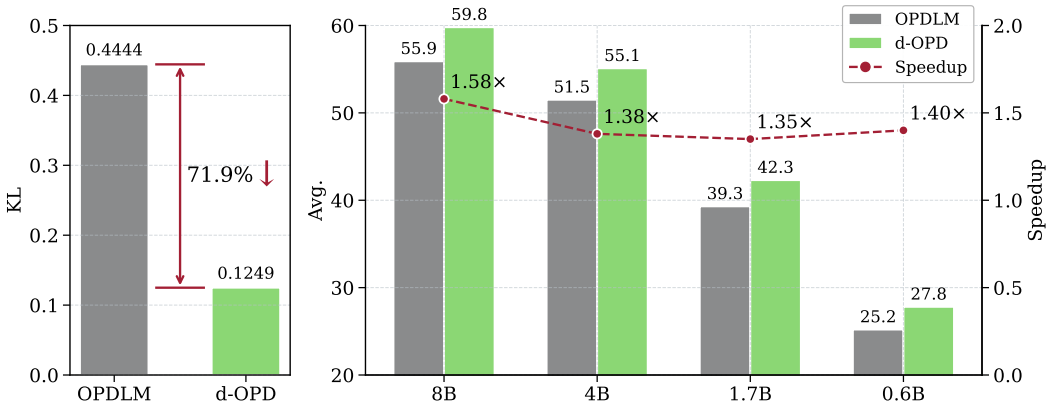


Figure 1: **d-OPD results. Left:** Correcting the AR teacher with visible future information substantially reduces its mismatch with the exact student-state-conditioned target, lowering the mean token-level KL by 71.9%. **Right:** d-OPD consistently improves the six-benchmark average over OPDLM across Qwen3 models from 0.6B to 8B, with training speedup of up to $1.58\times$.

through large-scale pretraining, conversion methods aim to preserve this knowledge while adapting the model to block-wise parallel generation. Distillation provides a natural way to achieve this by using the AR model as a teacher during conversion. However, standard offline distillation supervises the student on states from fixed offline trajectories rather than the partially denoised states it actually visits during block-dLLM generation. On-policy distillation (OPD) addresses this state-distribution shift by training on states generated by the student's current policy (Dat et al., 2026; Fu et al., 2026c; Heo et al., 2026; Li et al., 2026a;d;e; Song & Zheng, 2026; Tan & Hong, 2026; Yang et al., 2026; Zhang et al., 2026; Zhao et al., 2026). OPDLM applies this idea to block-dLLM conversion: it rolls out the block-dLLM student on-policy and uses an AR teacher to provide token-level supervision on the resulting student-generated states (Su et al., 2026).

However, matching the state distribution is not sufficient for block-dLLM distillation. A fundamental mismatch remains in supervision because the block-dLLM student and the causal AR teacher condition on different information at the same training state. Consider a masked position inside a partially denoised block. The student predicts this token from the entire visible block state, which may already contain visible tokens to its right. The standard AR teacher target, in contrast, is defined only from the causal prefix. Consequently, even when teacher supervision is queried on exactly the state visited by the student, the resulting teacher distribution does not fully reflect the information available to the student. OPD therefore only aligns the student's training and generation states, while leaving the teacher target misaligned with the student's available information.

This mismatch suggests that the target distribution should reflect the visible future information inside the current block. The key question is how the teacher distribution at the current position should change given this future context. By the chain rule, the AR teacher's next-token probabilities can be combined to compute the joint probability of a complete sequence. This joint distribution actually allows visible future tokens to provide evidence about the current masked token. This is because different candidate values at the current position can make the observed future more or less likely, which in turn provides a way to reweight the teacher distribution at the current position.

Using this joint-distribution view, we introduce **d-OPD**, a future-aware OPD method for AR-to-block-dLLM conversion. d-OPD uses the visible future context to correct the AR teacher distribution at each masked position. Specifically, we formalize the desired target as the AR teacher posterior conditioned on the complete student-visible state, which naturally combines the causal teacher prediction with how well each candidate agrees with the visible future. Because exact conditioning requires intractable marginalization over undenoised prefix and future tokens, d-OPD constructs a tractable approximation from completed on-policy trajectories. This approximation provides future-aware teacher supervision that better matches the information available to the block-dLLM student.

We evaluate d-OPD across Qwen3 models from 0.6B to 8B. d-OPD consistently improves over OPDLM across model scales and block sizes, with gains of 2.6–4.0 points in the six-benchmark

average. The improvement remains stable as the model scales from 0.6B to 8B, showing that the benefit is not limited to a particular model size. It also persists as the block size increases, reaching the largest improvement of 4.0 points at $N = 8$, where more token positions are modeled jointly within each block. Beyond accuracy, d-OPD improves training efficiency as well, reducing the wall-clock time required to reach the corresponding OPDLM performance levels by 1.35–$1.58\times$ across all evaluated model scales and training settings.

We summarize our contributions as follows:

- We identify a fundamental *target-conditioning mismatch* in AR-to-dLLM OPD. Although OPD aligns the student states used for training with those encountered during generation, the causal AR teacher and block-dLLM student still condition on different information.
- We introduce **d-OPD**, a future-aware OPD method that corrects the AR teacher distribution using visible future information within each block. We formalize the desired future-conditioned target and develop a tractable trajectory-based correction for standard OPD training.
- We evaluate d-OPD across Qwen3 models from 0.6B to 8B. d-OPD consistently improves over OPDLM, achieving up to 4.0 points higher six-benchmark average and 1.35–$1.58\times$ faster time to the corresponding OPDLM performance across the evaluated settings.

2 Related Work

2.1 Diffusion language models

Diffusion language models (dLLMs) generate text by iteratively denoising masked tokens and have recently scaled to general language modeling, reasoning, and instruction following (Bao et al., 2026; Chen et al., 2026c; Fu et al., 2026a; Hu et al., 2026b; Israel et al., 2025; Labs et al., 2025; Seo et al., 2025; Wu et al., 2026b; Ye et al., 2025; Bertolani et al., 2026; Chen et al., 2026b; Gwak et al., 2026; Hersche et al., 2026; Li et al., 2026c; Qi et al., 2026; Qian et al., 2026; Zhu et al., 2026). Block dLLMs further divide generation into causal blocks while allowing bidirectional interaction within the current block (Arriola et al., 2025; Bie et al., 2025; Wang et al., 2026b). Recent work improves block dLLM inference through confidence-aware decoding, cache reuse, alternative block schedules, generation orders, and specialized serving systems (Wu et al., 2026c; Hou & Kwok, 2026; Kedia et al., 2026; Tian et al., 2026a; Wang et al., 2026a). These works mainly improve inference for trained dLLMs, whereas we study the teacher target used during AR-to-block-dLLM conversion.

2.2 AR-to-diffusion adaptation

Recent work converts pretrained AR models into dLLMs through changes in attention, masking, denoising, and block-wise training while preserving pretrained knowledge (Bie et al., 2025; Gong et al., 2025; Liu et al., 2025; Ma et al., 2026b; Wu et al., 2026b; Ye et al., 2025; Cheng et al., 2026a; Fu et al., 2026b; Tian et al., 2026b; Wang et al., 2026b; Zeng et al., 2026). BARD, T*, and related methods refine this conversion through progressive block scaling, stage-wise distillation, trajectory-aware curricula, auxiliary AR objectives, and on-policy adaptation (Bie et al., 2025; Chen et al., 2026a; Su et al., 2026; Tian et al., 2026b; Xia et al., 2026; Wu et al., 2026a).

2.3 On-policy distillation

Knowledge distillation transfers teacher behavior through token- or sequence-level supervision (Mansourian et al., 2025; Wen et al., 2025; Yang et al., 2025b; Fang et al., 2026). On-policy distillation (OPD) trains on student-generated states, reducing the mismatch between training and generation (Fu et al., 2026c; Li et al., 2026d; Song & Zheng, 2026). Recent work has extensively studied and extended OPD for LLM post-training (Dat et al., 2026; Fu et al., 2026c; Heo et al., 2026; Li et al., 2026a;d;e; Song & Zheng, 2026; Tan & Hong, 2026; Yang et al., 2026; Zhang et al., 2026; Zhao et al., 2026). OPDLM applies this paradigm to AR-to-dLLM conversion (Su et al., 2026). It aligns the student states used for training with those encountered during generation, but leaves a target-conditioning mismatch between the causal teacher and block-dLLM student.

3 MOTIVATION

3.1 TARGET-CONDITIONING MISMATCH

Consider a block dLLM with block size N. Let q denote the input question and let z denote the completed causal context preceding the active block. We denote the active block by

$$X = (x_1, \ldots, x_N). \tag{1}$$

At denoising step t, the block dLLM student visits a partially masked state

$$X_t = (x_{t,1}, \ldots, x_{t,N}). \tag{2}$$

Let $p_{\theta,i}$ denote the student distribution at position i, where θ denotes the student parameters. Some entries of X_t have already been revealed while the rest remain masked. For a masked position i, the student predicts from all currently visible tokens in the block,

$$p_{\theta,i}(\cdot \mid X_t, q, z), \tag{3}$$

so visible tokens on either side of position i can influence the prediction.

Let p_T denote the distribution defined by the frozen AR teacher. Let

$$\hat{X} = (\hat{x}_1, \ldots, \hat{x}_N) \tag{4}$$

denote the completed, fully unmasked block produced by the student rollout. Let $\mathcal{V}$ denote the vocabulary. For a candidate token $v \in \mathcal{V}$, standard AR supervision uses the completed causal prefix and defines

$$r_i(v) = p_T(v \mid q, z, \hat{x}_{<i}). \tag{5}$$

Here r_i denotes the causal teacher target at position i. This target depends only on the causal prefix and therefore ignores any visible information to the right of i. If x_i is masked while tokens to its right are already visible, the block dLLM student can use this future context directly, whereas the causal AR target cannot condition on it. The teacher target and student prediction therefore depend on different information at the same student-visited state, creating a target-conditioning mismatch.

3.2 EXACT FUTURE-CONDITIONED TARGET

To derive the target that resolves this mismatch, we start from the teacher's joint distribution over the active block. The frozen AR teacher defines

$$p_T(X \mid q, z) = \prod_{i=1}^{N} p_T(x_i \mid q, z, x_{<i}). \tag{6}$$

Under this joint distribution, each candidate value for x_i assigns a likelihood to the observed future context. Different candidates can assign different likelihoods to the same observed future. By Bayes' rule, the causal teacher distribution at position i is reweighted by these likelihoods, yielding the teacher posterior conditioned on the complete student-visible state.

We first define this posterior at the block level. For a clean block X, write $X \succeq X_t$ if X agrees with every visible token in X_t. Conditioning the teacher joint distribution on this consistency event gives

$$\pi(X \mid X_t, q, z) = \frac{p_T(X \mid q, z)\mathbf{1}\{X \succeq X_t\}}{\sum_{X'} p_T(X' \mid q, z)\mathbf{1}\{X' \succeq X_t\}}. \tag{7}$$

Here X' ranges over all fully unmasked realizations of the active block, and $\mathbf{1}\{\cdot\}$ denotes the indicator function.

We then define the exact teacher target for a masked position i as the corresponding token-level marginal:

$$\pi_i(v \mid X_t, q, z) = \sum_{X : x_i = v} \pi(X \mid X_t, q, z). \tag{8}$$

Here π_i denotes the corresponding token-level posterior at position i.

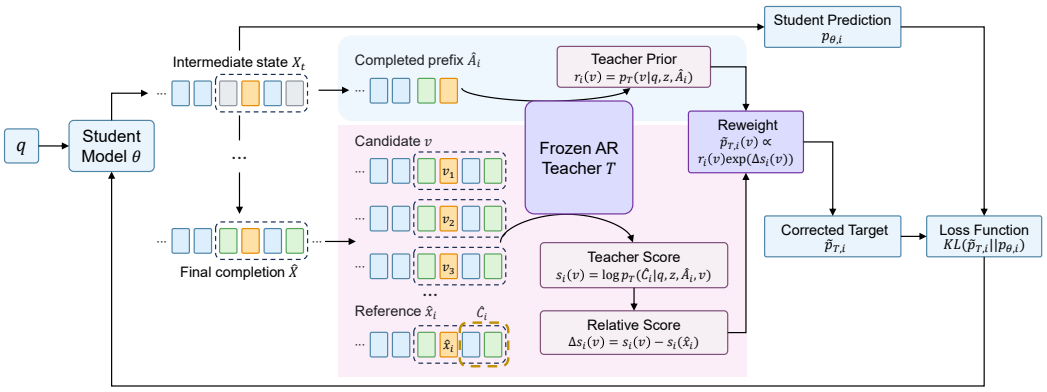


Figure 2: **Method overview.** d-OPD starts from a student-visited partially masked block X_t and its completed block $\hat{X}$ from the same on-policy rollout. For each masked position i, the completed prefix $\hat{A}_i$ gives the causal AR teacher prior r_i. A candidate token v and the reference token $\hat{x}_i$ are scored by how likely the completed future $\hat{C}_i$ is under the frozen teacher. Their relative future scores are then used to reweight r_i to form the future-aware teacher target $\tilde{p}_{T,i}$. The block dLLM student is then trained by minimizing the KL divergence between $\tilde{p}_{T,i}$ and $p_{\theta,i}$.

To expose the structure of this target, let $\mathcal{A}_i(X_t)$ denote the set of left-prefix completions $A_i = x_{<i}$ compatible with X_t. Then

$$\pi_i(v \mid X_t, q, z) = \sum_{A_i \in \mathcal{A}_i(X_t)} \pi(A_i \mid X_t, q, z)\, \pi_i(v \mid X_t, q, z, A_i). \tag{9}$$

Now fix one compatible prefix A_i. Let O_i denote the visible tokens to the right of position i. Applying Bayes' rule to the AR joint distribution gives

$$\pi_i(v \mid X_t, q, z, A_i) = \frac{p_T(v \mid q, z, A_i)\, p_T(O_i \mid q, z, A_i, v)}{\sum_{v' \in \mathcal{V}} p_T(v' \mid q, z, A_i)\, p_T(O_i \mid q, z, A_i, v')}. \tag{10}$$

Equation 10 gives the key observation behind d-OPD. We refer to the first factor, $p_T(v \mid q, z, A_i)$, as the *causal teacher prior*, since it is the ordinary AR teacher distribution over candidate token v. We refer to the second factor, $p_T(O_i \mid q, z, A_i, v)$, as the *future-compatibility term*, since it measures how likely the visible future is under that candidate. Future-aware supervision can therefore be obtained by correcting the causal teacher distribution according to the future-compatibility term.

If no future token is visible, then $O_i = \varnothing$ and the future-compatibility term is constant in v, so the target reduces to the causal teacher prior. If the entire left prefix is visible, $\mathcal{A}_i(X_t)$ contains only one completion and the outer marginalization disappears. In general, the exact target must account for uncertainty in both the masked prefix and the unobserved future.

3.3 Diagnosis

We quantify the target-conditioning mismatch over 49,948 masked-token marginals in a controlled setting described in Appendix D, where the exact future-conditioned posterior can be computed by enumeration. This setting allows us to directly evaluate how closely different teacher targets match the posterior induced by the same partially observed block state, providing a target-level diagnosis of the mismatch rather than inferring it only indirectly from downstream task performance alone. The causal target has a mean token-level KL of 0.4444 to the exact posterior, compared with 0.1249 using our method introduced in Section 4, a 71.9% reduction. The large gap confirms that conditioning only on the causal prefix can substantially misalign the teacher target with the information available to the block dLLM student. Incorporating future information closes most of this gap, moving the teacher target much closer to the exact posterior. This controlled diagnosis therefore provides direct empirical evidence for the target-conditioning mismatch and motivates explicitly correcting the teacher distribution using student-visible future context.

4 METHOD

4.1 FUTURE-AWARE CORRECTION

Section 3 defines the exact future-conditioned teacher target that matches the information available to the block dLLM student. Computing this target is intractable in practice because it requires marginalizing over all compatible left-prefix completions and unobserved future tokens.

d-OPD builds on the completed block $\hat{X}$ already produced by the on-policy rollout. For a masked position i, let $\hat{A}_i = \hat{x}_{<i}$ denote the completed prefix from the same rollout. We use $\hat{A}_i$ as one compatible realization of the uncertain prefix A in Equation 9. The causal teacher prior is therefore

$$r_i(v) = p_T(v \mid q, z, \hat{A}_i). \tag{11}$$

The corresponding fixed-prefix form of Equation 10 is

$$\pi_i(v \mid X_t, q, z, \hat{A}_i) = \frac{r_i(v)\, p_T(O_i \mid q, z, \hat{A}_i, v)}{\sum_{v' \in \mathcal{V}} r_i(v')\, p_T(O_i \mid q, z, \hat{A}_i, v')}. \tag{12}$$

Thus the prefix uncertainty has been resolved using $\hat{A}_i$, and the remaining challenge is to make the future-compatibility term in Equation 12 tractable.

The completed block also provides one compatible realization of the uncertain future. Let

$$\hat{C}_i = (\hat{x}_{i+1}, \ldots, \hat{x}_N) \tag{13}$$

denote the completed suffix after position i. We use the likelihood of this completed future under each candidate token v to estimate the future-compatibility term in Equation 12:

$$s_i(v) = \log p_T(\hat{C}_i \mid q, z, \hat{A}_i, v). \tag{14}$$

Using this completed-future likelihood in Equation 12 gives the completed-future target

$$\tilde{p}_{T,i}(v) = \frac{r_i(v) \exp(s_i(v))}{\sum_{v' \in \mathcal{V}} r_i(v') \exp(s_i(v'))}. \tag{15}$$

Thus, candidates that make the completed future more likely receive larger teacher weight.

Evaluating $s_i(v)$ for every token in the vocabulary is computationally expensive, so in practice we compute future scores only for a small set of high-probability candidates. This leaves the remaining tokens without a future score. Simply leaving these unscored tokens unchanged while multiplying the scored candidates by their absolute future likelihoods $\exp(s_i(v))$ would be inappropriate, because an absolute sequence likelihood is typically much smaller than one, whereas an unchanged token implicitly receives a correction factor of one. This would artificially favor unscored tokens and distort their relative probability mass. We therefore normalize the future likelihoods by a reference candidate, so that a correction factor of one represents a meaningful neutral baseline. Since $\hat{x}_i$ is the token actually produced by the rollout, it provides a natural reference. We express each scored likelihood relative to $\hat{x}_i$:

$$\Delta s_i(v) = s_i(v) - s_i(\hat{x}_i). \tag{16}$$

Equivalently,

$$\exp(\Delta s_i(v)) = \frac{p_T(\hat{C}_i \mid q, z, \hat{A}_i, v)}{p_T(\hat{C}_i \mid q, z, \hat{A}_i, \hat{x}_i)}. \tag{17}$$

Dividing by the reference likelihood does not change the full-vocabulary target, since the same factor cancels during normalization:

$$\tilde{p}_{T,i}(v) = \frac{r_i(v) \exp(\Delta s_i(v))}{\sum_{v' \in \mathcal{V}} r_i(v') \exp(\Delta s_i(v'))}. \tag{18}$$

For a candidate set V_i, we apply the relative future correction only to scored candidates in V_i, while tokens outside V_i are left at the baseline correction weight one.

Table 1: **Main results.** Accuracy across block sizes and Qwen3 model scales on six benchmarks. d-OPD consistently outperforms OPDLM across all evaluated model scales, with improvements remaining stable from 0.6B to 8B. The gains also persist as the block size increases, reaching up to 4.0 points in the six-benchmark average. Best results are in bold. Second-best results are underlined.

Setting	Method	MMLU	MMLU-P	GPQA-D	GSM8K	MATH500	AIME25	Avg.
Qwen3-8B: comparison across block sizes								
	Qwen3-8B	76.1	60.5	44.4	91.7	83.2	16.7	62.1
$N = 4$	SFT	69.1	54.3	**39.9**	87.3	75.4	13.3	56.6
	B-SFT	<u>70.9</u>	52.6	<u>39.4</u>	88.5	72.6	<u>16.7</u>	56.8
	BARD	68.6	<u>55.3</u>	38.4	<u>88.6</u>	76.4	<u>16.7</u>	<u>57.3</u>
	OPDLM	69.2	54.0	38.4	87.2	<u>76.8</u>	10.0	55.9
	d-OPD	**74.3**	**55.9**	38.9	**89.3**	**77.2**	**23.3**	**59.8**
$N = 8$	OPDLM	65.0	45.3	24.2	77.3	59.0	6.7	46.3
	d-OPD	**67.7**	**48.0**	**27.8**	**83.6**	**61.2**	**13.3**	**50.3**
$N = 16$	OPDLM	64.5	43.3	22.7	75.4	49.6	**3.3**	43.1
	d-OPD	**67.1**	**46.5**	**29.8**	**80.6**	**53.8**	**3.3**	**46.9**
Qwen3: generalization across model scales								
4B	Qwen3-4B	72.1	56.2	39.9	90.4	81.8	16.7	59.5
	SFT	65.5	48.9	29.8	82.3	74.0	6.7	51.2
	B-SFT	64.2	<u>50.2</u>	<u>31.3</u>	<u>87.4</u>	74.6	<u>13.3</u>	<u>53.5</u>
	BARD	65.1	<u>50.2</u>	29.3	82.0	<u>75.2</u>	10.0	52.0
	OPDLM	<u>66.2</u>	47.8	30.3	82.9	75.0	6.7	51.5
	d-OPD	**67.4**	**50.5**	**32.3**	**88.3**	**75.4**	**16.7**	**55.1**
1.7B	Qwen3-1.7B	58.6	39.4	28.3	81.5	72.0	6.7	47.7
	OPDLM	42.9	29.9	22.2	73.9	**60.2**	**6.7**	39.3
	d-OPD	**55.2**	**30.6**	**27.8**	**74.3**	59.4	**6.7**	**42.3**
0.6B	Qwen3-0.6B	46.0	27.3	24.2	63.5	52.4	6.7	36.7
	OPDLM	41.1	18.8	19.7	40.6	31.0	**0.0**	25.2
	d-OPD	**46.7**	**19.9**	**21.7**	**44.6**	**33.6**	**0.0**	**27.8**

4.2 METHOD DETAILS

For future scoring, we use the union of the teacher and student top-k sets as the candidate set, since high-probability tokens capture most of the probability mass under both distributions. Taking the union ensures that candidates considered important by either distribution can receive future-aware correction, while avoiding full-vocabulary continuation scoring. As shown in Table 3, the top-4 tokens already cover more than 95% of the probability mass for both the teacher and the student, and the coverage quickly saturates as k increases. This allows future-aware correction to focus on only a small set of plausible candidates, while leaving the remaining vocabulary unchanged.

We also use a correctness gate for the future-aware correction: the completed future is used to reweight the teacher target only when the rollout passes the task verifier, since futures from incorrect rollouts may provide noisy or misleading evidence for reweighting and would also incur unnecessary scoring cost, while standard causal teacher supervision is still applied to all rollouts. More implementation details are provided in Algorithm 1 in Appendix A.

4.3 TRAINING

For each block, we randomly sample one denoising state X_t from the on-policy rollout. This ensures that the student is trained directly on partially revealed states encountered under its current generation policy, rather than on states constructed from a separate offline distribution. Distillation is applied to all masked positions in the sampled state, with the corresponding teacher targets constructed from the completed on-policy rollout. For each masked position i, we optimize the forward KL from the corrected teacher target to the student:

$$\mathcal{L}_i(\theta) = D_{\mathrm{KL}}(\tilde{p}_{T,i} \parallel p_{\theta,i}) \,. \tag{19}$$

Table 2: **Training efficiency.** Wall-clock time to the best six-benchmark average achieved by OPDLM at each model scale. d-OPD consistently reaches the matched OPDLM performance faster, with speedups of 1.35–$1.58\times$ across Qwen3 models from 0.6B to 8B.

Model scale	Qwen3-0.6B	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
OPDLM Time (hr)	18.39	19.92	30.72	28.26
d-OPD Time (hr)	13.15	14.75	22.34	17.91
Speedup	**1.40**$\times$	**1.35**$\times$	**1.38**$\times$	**1.58**$\times$

The token-level losses are averaged over all masked positions collected across the sampled denoising states of a rollout, so each masked prediction contributes directly to the distillation objective. Sampling one state per block also allows a single rollout to provide supervision at different denoising states across its blocks. d-OPD changes only the teacher supervision at these positions, while keeping the student architecture, distillation objective, and overall training procedure unchanged.

5 Experiments

Models. We evaluate Qwen3 checkpoints at 0.6B, 1.7B, 4B, and 8B parameters (Yang et al., 2025a). Model-scale comparisons use block size $N = 4$, and we additionally train Qwen3-8B with $N = \{8, 16\}$ to study larger blocks. Each pretrained AR checkpoint initializes the corresponding block dLLM student and remains frozen as both the causal teacher and future-compatibility scorer.

Training. We use the OPDLM training mixture (Su et al., 2026), containing 61,816 prompts with maximum response length 4,000. Training uses one pass over the full prompt mixture for 483 on-policy iterations on 8 NVIDIA H100 GPUs. Within each comparison setting, methods use the same prompts, optimization budget, random seeds, and state-sampling procedure. Unless otherwise specified, d-OPD uses $k = 16$. Full training and implementation details are provided in Appendix C.

Baselines. OPDLM (Su et al., 2026) is our primary baseline, which uses the same on-policy student states as d-OPD but directly distills the causal AR teacher distribution without future-aware correction. We additionally compare with SFT (Nie et al., 2025), which applies standard masked-token supervised fine-tuning, Blockwise SFT (B-SFT) (Sun et al., 2025), which adapts supervised fine-tuning to the blockwise generation structure, and BARD (Chen et al., 2026a), which converts AR models through progressive blockwise distillation. Together, these methods provide strong representative baselines for AR-to-block-dLLM conversion.

Evaluation. We evaluate MMLU (Hendrycks et al., 2021a;b), MMLU-Pro (Wang et al., 2024), GPQA-Diamond (Rein et al., 2024), GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021c), and AIME25 (Zhang & Math-AI, 2025). Table 1 reports six benchmark scores and their average at the checkpoint with the best six-benchmark average during training, following Appendix C.

5.1 Main results

Table 1 summarizes the main results across model scales and block sizes. d-OPD consistently outperforms OPDLM across all evaluated Qwen3 scales, with gains of 2.6–3.9 points in the six-benchmark average from 0.6B to 8B. The improvement remains stable over this wide range of model capacities, indicating that the benefit of future-aware correction is not specific to a particular student scale. On Qwen3-8B and Qwen3-4B, where we additionally compare against SFT, B-SFT, and BARD, d-OPD also achieves the strongest converted-model average. These results show that the gain is not limited to improving over the corresponding on-policy baseline, but remains competitive against alternative AR-to-block-dLLM conversion strategies.

The advantage also persists as the block size increases. On Qwen3-8B, d-OPD improves the six-benchmark average over OPDLM across all evaluated block sizes, with gains of up to 4.0 points. The improvement remains substantial from block size 4 to 16, where the student can exploit more within-block future context and future-aware correction becomes increasingly important. This is consistent with the target-conditioning mismatch identified in Section 3: as more tokens within the active

Table 3: **Ablation study. Left:** Effect of candidate-set size k, showing the probability mass covered by the selected candidates and the resulting training overhead. Even small candidate sets cover nearly all teacher and student probability mass, while $k = 16$ provides a favorable trade-off between coverage and overhead. **Right:** Effect of correctness gating. The gate improves the six-benchmark average from 58.8 to 59.8 by restricting future-aware correction to verifier-correct rollouts.

Mass (%)	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = 64$
Teacher	97.87	98.88	**99.45**	99.67	99.79
Student	95.43	98.30	**99.44**	99.64	99.77
Overhead	2.44%	2.57%	**2.63%**	3.50%	6.02%

Method	Gate	Avg.
OPDLM	–	55.9
d-OPD	Off	58.8
d-OPD	**On**	**59.8**

block are jointly involved in prediction, the student conditions on increasingly richer information than the causal AR teacher. The sustained gains across larger block sizes therefore further support incorporating visible future information into the teacher target.

5.2 Training efficiency

For each model scale, we use the best six-benchmark average achieved by OPDLM as the target and measure the first wall-clock time at which d-OPD reaches the same performance. This time-to-target metric complements the best achieved performance by measuring how quickly each method reaches a common performance level. By using wall-clock time, it also accounts for the additional teacher computation introduced by future scoring, rather than comparing methods only by the number of optimization steps. As shown in Table 2, d-OPD reaches the matched OPDLM performance 1.35–$1.58\times$ faster across all four Qwen3 model scales, with the largest speedup on Qwen3-8B. The consistent improvement across scales indicates that the benefit is not limited to a particular model size. Thus, although future-aware scoring introduces additional per-step computation, the more informative teacher target enables the student to make faster progress toward the same performance level, yielding a net reduction in wall-clock training time.

5.3 Ablation studies

Table 3 first examines candidate restriction. Even very small candidate sets capture most of the probability mass under both the teacher and student distributions, and the coverage quickly saturates as k increases. By $k = 16$, the candidate set already covers more than 99% of the probability mass for both distributions, while the measured training overhead remains only 2.63%. Increasing k further yields only marginal gains in coverage but noticeably higher overhead, indicating that most of the useful future correction is concentrated on a small number of high-probability tokens. This also suggests that full-vocabulary future scoring is unnecessary in practice: restricting the correction to likely candidates preserves nearly all of the probability mass relevant to the teacher target while substantially reducing computation. We therefore use $k = 16$ as the default coverage–cost tradeoff.

The right side of Table 3 evaluates correctness gating. Applying future-aware correction only to verifier-correct rollouts improves the six-benchmark average from 58.8 to 59.8. This suggests that completed futures from successful trajectories provide more reliable evidence for reweighting, while the gate also avoids unnecessary future scoring on incorrect rollouts and reduces training cost.

6 Conclusion

On-policy distillation for AR-to-block-dLLM conversion supervises student-visited states with a causal AR teacher. We identify a target-conditioning mismatch in this setting: the standard teacher target depends only on the causal prefix, while the block dLLM student uses the full visible state of the active block, including visible future context. We introduce d-OPD, which uses visible future context within each block to reweight the causal AR teacher distribution toward a target that better matches the information available to the student. Across Qwen3 models from 0.6B to 8B, d-OPD consistently improves over OPDLM, with gains of up to 4.0 points in the six-benchmark average and up to $1.58\times$ speedup to matched OPDLM performance. These results highlight the effectiveness of correcting the teacher target with student-visible future context.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with method implementation. We have not used generative AI tools for research ideation, methodology or experiment design, hypothesis development, mathematical claims or proofs, data analysis, or result interpretation, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for language polishing, assisting with evaluation and experiment code, and monitoring experiment execution. We have reviewed all AI-assisted work. All AI-assisted code was manually reviewed, tested, and verified by the authors, and all AI-assisted text was manually checked for technical accuracy and consistency with the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies training methods for language models and does not involve human subjects or the collection of private or personally identifiable information. Our experiments use existing models, benchmarks, and training data following their intended research use. We are not aware of additional ethical concerns specific to the proposed method beyond the broader considerations associated with training and deploying large language models in practical settings.

REPRODUCIBILITY STATEMENT

We provide the main algorithm and training objective in Section 4, with the detailed training procedure summarized in Algorithm 1. Appendix C documents the training setup, rollout and state-sampling procedure, correctness gating, benchmark evaluation, and checkpoint-selection protocol used throughout the experiments. Appendix D gives the complete construction of the controlled enumerable setting used for the exact-target diagnosis, including the restricted support, compared targets, and KL aggregation procedure. Appendix E describes the wall-clock timing protocol used to measure both time to matched performance and the additional training overhead introduced by future-aware scoring. Additional implementation details, including candidate scoring and caching, are provided in Appendix B, while Appendix G reports the full correctness-gating ablation results. Together, these sections specify the main algorithmic, training, evaluation, diagnostic, and timing choices needed to faithfully reproduce the reported results and comparisons.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 50726–50753, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7ede97c3e082c6df10a8d6103a2eebd2-Paper-Conference.pdf.

Wenrui Bao, Zhiben Chen, Dan Xu, and Yuzhang Shang. Learning to parallel: Accelerating diffusion large language models via learnable parallel decoding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=bFJ8Sdr224.

Thomas Bertolani, Davide Bucciarelli, Leonardo Zini, Marcella Cornia, and Lorenzo Baraldi. Diffusion language models: An experimental analysis, 2026. URL https://arxiv.org/abs/2606.19475.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. Llada2.0: Scaling up diffusion language models to 100b, 2025. URL https://arxiv.org/abs/2512.15745.

Mathieu Blondel, Michael Eli Sander, Germain Vivier-Ardisson, Tianlin Liu, and Vincent Roulet. Autoregressive language models are secretly energy-based models: Insights into the lookahead capabilities of next-token prediction. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=997oprE4sh.

Baoyou Chen, Hanchen Xia, Peng Tu, Haojun Shi, Liwei Zhang, Yuxuan Yao, Weihao Yuan, and Siyu Zhu. Bard: Bridging autoregressive and diffusion vision-language models via highly efficient progressive block merging and stage-wise distillation, 2026a. URL https://arxiv.org/abs/2604.16514.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. Dmax: Aggressive parallel decoding for dllms, 2026b. URL https://arxiv.org/abs/2604.08302.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dparallel: Learnable parallel decoding for dllms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 53193–53211, 2026c. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/57250222014c35949476f3f272c322d2-Paper-Conference.pdf.

Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao 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. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22058–22075, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1110. URL https://aclanthology.org/2026.findings-acl.1110/.

Shuang Cheng, Yuhua Jiang, Zineng Zhou, Dawei Liu, Tao Wang, Linfeng Zhang, Biqing Qi, and Bowen Zhou. SDAR-VL: Stable and efficient block-wise diffusion for vision-language understanding. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28882–28901, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1333. URL https://aclanthology.org/2026.acl-long.1333/.

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.

Phuong Tuan Dat, Qi Li, and Xinchao Wang. dopsd: On-policy self-distillation for diffusion language models, 2026. URL https://arxiv.org/abs/2607.04428.

Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zhengliang Liu, Zhenyuan Yang, Haoran Lu, Xilin Gong, Yufang Liu, Terry Ma, Wei Ruan, Ali Abbasi, Jing Zhang, Tao Wang, Ehsan Latif, Weihang You, Hanqi Jiang, Wei Liu, Wei Zhang, Soheil Kolouri, Xiaoming Zhai, Dajiang Zhu, Wenxuan Zhong, Tianming Liu, and Ping Ma. Knowledge distillation and dataset distillation of large language models: Emerging trends, challenges, and future directions, 2026. URL https://arxiv.org/abs/2504.14772.

Zhili Feng, Dhananjay Ram, Cole Hawkins, Aditya Rawal, Jinman Zhao, and Sheng Zha. Sequence-level large language model training with contrastive preference optimization. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4158–4164, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.233. URL https://aclanthology.org/2025.findings-naacl.233/.

Hengyu Fu, Baihe Huang, Virginia Adams, Charles Wang, Junkeun Yi, Mohammad Mahdi Kamani, Venkat Krishna Srinivasan, and Jiantao Jiao. From bits to rounds: Parallel decoding with exploration for diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=CTrA4eNQfv.

Yonggan Fu, Lexington Whalen, Zhifan Ye, Xin Dong, Shizhe Diao, Jingyu Liu, Chengyue Wu, Hao Zhang, Enze Xie, Song Han, Maksim Khadkevich, Jan Kautz, Yingyan Celine Lin, and Pavlo Molchanov. Efficient-DLM: From autoregressive to diffusion language models, and beyond in speed, 2026b. URL https://openreview.net/forum?id=kIbFgoCq86.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan ang Gao, Yudong Wang, Zhiyuan Liu, Ning Ding, and Chaojun Xiao. Rethinking on-policy distillation of large language models ii: One training example, 2026c. URL https://arxiv.org/abs/2609.04172.

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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 5046–5073, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0fa81c3f0d57f95b8776de3a248ef0ed-Paper-Conference.pdf.

Daehoon Gwak, Minhyung Lee, Junwoo Park, and Jaegul Choo. Accelerating masked diffusion large language models: A survey of efficient inference techniques, 2026. URL https://arxiv.org/abs/2607.12829.

Ali Hatamizadeh, Syeda Nahida Akter, Shrimai Prabhumoye, Jan Kautz, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. Rlp: Reinforcement as a pretraining objective. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 41375–41402, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/44a45e27879b8fbab6e123ad8b93afc2-Paper-Conference.pdf.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.

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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021c. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation, 2026. URL https://arxiv.org/abs/2607.15161.

Michael Hersche, Samuel Moor-Smith, Thomas Hofmann, and Abbas Rahimi. Soft-masked diffusion language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 105685–105716, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ac6bd5acdf0c9446389e56e919f62ff0-Paper-Conference.pdf.

Kai Syun Hou and James Kwok. Rethinking the generation order of block diffusion language models, 2026. URL https://arxiv.org/abs/2607.24306.

Wenyang Hu, Yao Shu, See-Kiong Ng, and Bryan Kian Hsiang Low. Adaptive test-time compute allocation via training-free difficulty proxies, 2026a. URL https://openreview.net/forum?id=ztGHhyicWs.

Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed Abdelfattah, Jae-sun Seo, Zhiru Zhang, and Udit Gupta. Flashdlm: Accelerating diffusion language model inference via efficient kv caching and guided diffusion. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 153529–153544, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f930c6e1292a1160189a0734f22b465b-Paper-Conference.pdf.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion llms via adaptive parallel decoding, 2025. URL https://arxiv.org/abs/2506.00413.

Daniel Israel, Tian Jin, Ellie Cheng, Guy Van den Broeck, Aditya Grover, Suvinay Subramanian, and Michael Carbin. Planned diffusion. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 98298–98318, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/9fbae6362539ef7a151303c86cb51c00-Paper-Conference.pdf.

Yijie Jin, Jiajun Xu, Yuxuan Liu, Chenkai Xu, Yi Tu, Jiajun Li, Dandan Tu, Xiaohui Yan, Kai Yu, Pengfei Liu, and Zhijie Deng. Multi-block diffusion language models, 2026. URL https://arxiv.org/abs/2606.29215.

Nitin Kedia, Saurabh Agarwal, Myungjin Lee, and Aditya Akella. Sangam: Efficiently serving diffusion llms with the ar stack, 2026. URL https://arxiv.org/abs/2607.04206.

Minseo Kim, Coleman Hooper, Aditya Tomar, Chenfeng Xu, Mehrdad Farajtabar, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. Beyond next-token prediction: A performance characterization of diffusion versus autoregressive language models, 2025. URL https://arxiv.org/abs/2510.04146.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, and Jinqiao Wang. On-policy distillation with curriculum turn-level guidance for multi-turn agents, 2026a. URL https://arxiv.org/abs/2606.15912.

Jia-Nan Li, Jian Guan, Wei Wu, and Chongxuan Li. Refusion: A diffusion large language model with parallel autoregressive decoding. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=LBtWaUc7FE.

Pengxiang Li, Dilxat Muhtar, Tianlong Chen, Lu Yin, and Shiwei Liu. Why diffusion language models struggle with truly parallel (non-autoregressive) decoding? In *ICLR 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2026c. URL https://openreview.net/forum?id=NAu7pa8vCV.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026d. URL https://openreview.net/forum?id=aaDDXXDGXB.

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision, 2026e. URL https://arxiv.org/abs/2608.06296.

Pengxiao Lin, Zhongwang Zhang, and Zhi-Qin John Xu. Reasoning bias of next token prediction training, 2025. URL https://arxiv.org/abs/2502.02007.

Yangzhou Liu, Yue Cao, Hao Li, Gen Luo, Zhe Chen, Weiyun Wang, Xiaobo Liang, Biqing Qi, Lijun Wu, Changyao Tian, Yanting Zhang, Yuqiang Li, Tong Lu, Yu Qiao, Jifeng Dai, and Wenhai Wang. Sequential diffusion language models, 2025. URL https://arxiv.org/abs/2509.24007.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Linrui Ma, Yufei Cui, Kai Han, and Yunhe Wang. Diffusion in diffusion: Reclaiming global coherence in semi-autoregressive diffusion, 2026a. URL https://arxiv.org/abs/2601.13599.

Xiangyu Ma, Teng Xiao, Zuchao Li, and Lefei Zhang. From AR to diffusion: Efficiently adapting large language models with strictly causal and elastic horizons. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20914–20927, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.958. URL https://aclanthology.org/2026.acl-long.958/.

Amir M. Mansourian, Rozhan Ahmadi, Masoud Ghafouri, Amir Mohammad Babaei, Elaheh Badali Golezani, Zeynab yasamani ghamchi, Vida Ramezanian, Alireza Taherian, Kimia Dinashi, Amirali Miri, and Shohreh Kasaei. A comprehensive survey on knowledge distillation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=3cbJzdR78B.

Xingyu Mou, Zijin Huang, Tianze Zhang, Yuxin Ma, Lanning Wei, Zengfeng Huang, Da Zheng, and Lun Du. Beyond block boundaries: Multi-block editing for diffusion large language models, 2026. URL https://arxiv.org/abs/2607.22663.

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. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 50608–50646. Curran Associates, Inc., 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/48b383b24230e0e6e649d9c98dae4d8c-Paper-Conference.pdf.

Heqiang Qi, Wei Huang, Mingyuan Bai, and Xiangming Meng. Cluster-level attention-guided parallel decoding for masked diffusion language models, 2026. URL https://arxiv.org/abs/2605.29607.

Yu-Yang Qian, Junda Su, Lanxiang Hu, Peiyuan Zhang, Zhijie Deng, Peng Zhao, and Hao Zhang. d3LLM: Ultra-fast diffusion LLM using pseudo-trajectory distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=rzBAQT2Fkg.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3406703. URL https://doi.org/10.1145/3394486.3406703.

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. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. ZeRO-Offload: Democratizing Billion-Scale model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pp. 551–564. USENIX Association, July 2021. ISBN 978-1-939133-23-6. URL https://www.usenix.org/conference/atc21/presentation/ren-jie.

Yeongbin Seo, Dongha Lee, Jaehyung Kim, and Jinyoung Yeo. Fast and fluent diffusion language models via convolutional decoding and rejective fine-tuning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 111394–111429. Curran Associates, Inc., 2025. doi: 10.52202/085713-3719. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a1a1c79d6d3d76d884a8cb70939f67cb-Paper-Conference.pdf.

Vaibhav Singh, Pierre-André Noël, Torsten Scholak, Eugene Belilovsky, and Oleksiy Ostapenko. Fixed state, long reach: What a constant-size cache buys block diffusion at scale, 2026. URL https://arxiv.org/abs/2609.11998.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models, 2026. URL https://arxiv.org/abs/2604.00626.

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.

Bowen Sun, Yujun Cai, Ming-Hsuan Yang, and Yiwei Wang. Blockwise sft for diffusion language models: Reconciling bidirectional attention and autoregressive decoding, 2025. URL https://arxiv.org/abs/2508.19529.

Zhiquan Tan and Yinrong Hong. Self-supervised on-policy distillation for reasoning language models, 2026. URL https://arxiv.org/abs/2605.17497.

Zhiwei Tang, Yuanyu He, Yizheng Han, Wangbo Zhao, Jiasheng Tang, Fan Wang, and Bohan Zhuang. K-forcing: Joint next-k-token decoding via push-forward language modeling, 2026. URL https://arxiv.org/abs/2606.10820.

Bing Tian, Haikun Liu, Xiaocheng Zhong, Zhuohui Duan, Zhaokai Luo, Huayi Jin, Zhiyong Wang, and Xiaofei Liao. Flowblock: Wavefront-parallel decoding for self-correcting diffusion language models, 2026a. URL https://arxiv.org/abs/2607.17652.

Yuchuan Tian, Yuchen Liang, Shuo Zhang, Yingte Shu, Guangwen Yang, Wei He, Sibo Fang, Tianyu Guo, Kai Han, Chao Xu, Hanting Chen, Xinghao Chen, and Yunhe Wang. From next-token to next-block: A principled adaptation path for diffusion llms, 2026b. URL https://arxiv.org/abs/2512.06776.

Jinhao Wang, Zhexin Hu, Kangjie Zhou, Xin Zhou, and Fangfang Liu. Groupkv: Hierarchical kv cache management for long-context diffusion llm inference. In *Proceedings of the 55th International Conference on Parallel Processing*, ICPP '26, pp. 553–563, New York, NY, USA, 2026a. Association for Computing Machinery. ISBN 9798400726576. doi: 10.1145/3832810.3832896. URL https://doi.org/10.1145/3832810.3832896.

Ruiheng Wang, Shuanghao Bai, Haoran Zhang, Badong Chen, and Xiangyu Xu. Blockvla: Accelerating autoregressive vla via block diffusion finetuning, 2026b. URL https://arxiv.org/abs/2605.13382.

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. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Yuqiao Wen, Freda Shi, and Lili Mou. Knowledge distillation for language models. In Maria Lomeli, Swabha Swayamdipta, and Rui Zhang (eds.), *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts)*, pp. 25–29, Albuquerque, New Mexico, May 2025. Association for Computational Linguistics. ISBN 979-8-89176-193-3. doi: 10.18653/v1/2025.naacl-tutorial.4. URL https://aclanthology.org/2025.naacl-tutorial.4/.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Chengyue Wu, Shiyi Lan, Yonggan Fu, Sensen Gao, Jin Wang, Jincheng Yu, Jose M. Alvarez, Pavlo Molchanov, Ping Luo, Song Han, Ligeng Zhu, and Enze Xie. Fast-dvlm: Efficient block-diffusion vlm via direct conversion from autoregressive vlm, 2026a. URL https://arxiv.org/abs/2604.06832.

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. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 128353–128370, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d0865cbe51d35ec322f9af9db7806fc7-Paper-Conference.pdf.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 57027–57051, 2026c. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5d8d4e6061c3ba96c240b7fa1ae3471d-Paper-Conference.pdf.

Hanchen Xia, Baoyou Chen, Yutang Ge, Guojiang Zhao, and Siyu Zhu. T*: Progressive block scaling for masked diffusion language models through trajectory aware reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 808–816, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-391-3. doi: 10.18653/v1/2026.acl-short.67. URL https://aclanthology.org/2026.acl-short.67/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Trans. Intell. Syst. Technol.*, 16(6), November 2025b. ISSN 2157-6904. doi: 10.1145/3699518. URL https://doi.org/10.1145/3699518.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Haobo Wang, Mingxuan Xia, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, Junbo Zhao, and Gang Chen. Oprd: On-policy representation distillation, 2026. URL https://arxiv.org/abs/2606.06021.

Shu-Wen Yang, Byeonggeun Kim, Kuan-Po Huang, Qingming Tang, Huy Phan, Bo-Ru Lu, Harshavardhan Sundar, Shalini Ghosh, Hung-Yi Lee, Chieh-Chi Kao, and Chao Wang. Generative audio language modeling with continuous-valued tokens and masked next-token prediction. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 70793–70812.

PMLR, 13–19 Jul 2025c. URL https://proceedings.mlr.press/v267/yang25n.html.

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

Runpeng Yu, Qi Li, and Xinchao Wang. Discrete diffusion in large language and multimodal models: A survey, 2025. URL https://arxiv.org/abs/2506.13759.

Lunbin Zeng, Jingfeng Yao, Bencheng Liao, Hongyuan Tao, Wenyu Liu, and Xinggang Wang. Diffusionvl: Translating any autoregressive models into diffusion vision language models, 2026. URL https://arxiv.org/abs/2512.15713.

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models, 2026. URL https://arxiv.org/abs/2604.17535.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Ying Zhu, Jiaxin Wan, Xiaoran Liu, Siyang He, Qiqi Wang, Xu Guo, Tianyi Liang, Zengfeng Huang, Ziwei He, and Xipeng Qiu. Dirl: An efficient post-training framework for diffusion language models, 2026. URL https://arxiv.org/abs/2512.22234.

A Training Algorithm

Algorithm 1 summarizes one on-policy training iteration of d-OPD. Given student-visited denoising states and the completed trajectories produced by the same rollout, d-OPD constructs a future-aware teacher target for each masked prediction position during student training. The correction is applied only to a small candidate set and only when the completed rollout passes the task verifier. Otherwise, training falls back to the ordinary causal teacher target used for standard distillation.

B Additional Implementation Notes

Candidate scoring and caching. For each masked position eligible for future-aware correction, we score only the union of the teacher and student top-k sets. The teacher prefix $(q, z, \hat{A}_i)$ is shared across all candidate branches for a given position and can be cached and reused. Only the candidate token differs across branches, and the same completed suffix $\hat{C}_i$ is then scored under each candidate. This avoids recomputing the shared causal prefix for every candidate and keeps teacher computation focused on the part of the sequence that depends on the candidate choice. Tokens outside the candidate set are left unchanged by the future-aware correction before the final normalization, so the sparse scoring procedure modifies only the selected high-probability candidates.

For each eligible position, we construct one scoring job containing a reference branch for the realized rollout token $\hat{x}_i$ and one branch for every distinct token in the union of the teacher and student top-k sets. With the default $k = 16$, each job therefore contains at most 33 branches, including the reference branch. All branches for the same position are evaluated together in a single scorer batch in the default implementation. This organization allows most of the shared causal-prefix computation to be reused rather than recomputed independently for every candidate.

For each training example, we run the frozen AR scorer once on the clean prompt–response sequence to construct a Hugging Face `DynamicCache` (Wolf et al., 2020). For an individual scoring job, this cache is truncated to the causal prefix preceding the candidate position. At every Transformer layer, the cached key and value tensors have logical shape $[1, n_{\mathrm{kv}}, L_{\mathrm{prefix}}, d_{\mathrm{head}}]$ and are expanded

Algorithm 1 One on-policy training iteration of d-OPD

Require: Student p_θ, frozen AR teacher p_T, candidate size k
Generate responses with p_θ and record visited denoising states
Obtain each completed response $\hat{y}$ and its correctness label $R(\hat{y})$
Sample one visited denoising state for each generated block
for each selected active-block state X_t **do**
  Let z be the completed causal context preceding the active block
  Let $\hat{X} = (\hat{x}_1, \ldots, \hat{x}_N)$ be its completed block from the same rollout
  Run p_θ on X_t to obtain $p_{\theta,i}$ for all masked positions i
  for each masked position i in X_t **do**
    Form completed prefix $\hat{A}_i = (\hat{x}_1, \ldots, \hat{x}_{i-1})$
    Compute causal prior $r_i(v) = p_T(v \mid q, z, \hat{A}_i)$
    if $R(\hat{y}) = 1$ and X_t contains visible future context for i **then**
      Construct $V_i^{(k)}$ as the union of the teacher and student top-k sets
      Form completed suffix $\hat{C}_i = (\hat{x}_{i+1}, \ldots, \hat{x}_N)$
      Compute reference score $s_i(\hat{x}_i) = \log p_T(\hat{C}_i \mid q, z, \hat{A}_i, \hat{x}_i)$
      for $v \in V_i^{(k)}$ **do**
        Compute $s_i(v) = \log p_T(\hat{C}_i \mid q, z, \hat{A}_i, v)$
        Compute $\Delta s_i(v) = s_i(v) - s_i(\hat{x}_i)$
      end for
      Reweight candidates in $V_i^{(k)}$ by $\exp(\Delta s_i(v))$ and normalize to obtain $\tilde{p}_{T,i}$
    else
      Set $\tilde{p}_{T,i} = r_i$
    end if
  end for
end for
Average the token-level losses over all masked positions in each rollout
Update θ using Equation 19

only along the candidate-branch batch dimension. The suffix beginning at the candidate position is then evaluated separately for each branch. The same trajectory-level cache is reused across all eligible positions in that training example, but caches are not shared across different examples. To avoid excessive memory growth from cache expansion, the candidate-branch batch is further reduced whenever the estimated expanded prefix-cache size would exceed 16 GiB.

Fallback behavior and numerical safeguards. The corrected target reduces exactly to the causal teacher target whenever the response fails the correctness gate or when no future token is visible to the right of the masked position in the sampled student state. In both cases, there is no future-aware reweighting and we directly set $\tilde{p}_{T,i} = r_i$.

The scorer is loaded in the checkpoint's native dtype, which is bfloat16 for the reported Qwen3 experiments. Scorer logits, teacher and student log probabilities, suffix log-likelihood sums, relative score differences, correction factors, and final normalization are computed in float32. Candidate scores are expressed relative to the reference score $s_i(\hat{x}_i)$ before exponentiation. The future-aware correction for a position is committed only when the resulting full-vocabulary normalizer is finite and strictly positive. Otherwise, the causal target r_i is retained for that position. We do not otherwise clip finite score differences. Exact zero-probability entries remain zero after correction and are represented as $-\infty$ when converted back to log probabilities.

C Training and Evaluation Details

C.1 Data

We use the OPDLM training mixture (Su et al., 2026), which contains 61,816 prompts in total. We follow the same data preprocessing and prompt formatting used by OPDLM unless

otherwise specified. The original task information required for training and verification is retained throughout preprocessing. Each prompt is rendered with the Qwen chat template using `add_generation_prompt=True` and `enable_thinking=False`. We perform no additional deduplication or filtering beyond the preprocessing inherited from the original OPDLM setup.

C.2 TRAINING

Within each comparison setting, methods use the same prompt mixture, rollout configuration, state-selection procedure, optimization budget, and random seeds. Training makes one pass over the full prompt mixture, corresponding to 483 on-policy iterations on eight NVIDIA H100 80 GB GPUs. Unless otherwise specified, d-OPD uses candidate size $k = 16$.

Each on-policy iteration processes up to 128 prompts and generates one rollout per prompt with temperature 1.0, top-p $= 1.0$, and top-k $= 0$. Rollout decoding uses four denoising steps per block and a maximum generated response length of 4,000 tokens. At on-policy iteration s, a global response-length limit M_s increases from 100 to 4,000 tokens over the first 100 iterations according to a cosine schedule. We otherwise follow the rollout configuration used by OPDLM.

For state selection, we record the first-unmask time of every generated token and reconstruct the denoising states actually visited during rollout. For each response block, we uniformly sample one visited denoising round in which that block contains at least one masked prediction position. The round is sampled independently across blocks, and the selected blockwise states are then combined into one trajectory-level training example. Different blocks within the same example may therefore correspond to different denoising rounds, but every selected state was visited during the same on-policy rollout. Completed preceding blocks provide the corresponding causal context z, while the final completed rollout provides $\hat{X}$, $\hat{A}_i$, and $\hat{C}_i$ used for teacher-target construction.

We optimize the student with AdamW (Loshchilov & Hutter, 2019) using learning rate 1×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and zero weight decay. The learning rate is linearly warmed up during the first five on-policy iterations and then cosine-decayed to 10% of its peak value. Gradients are clipped to a global norm of 1.0. We use BF16 mixed precision, TF32 matrix operations, gradient checkpointing, and DeepSpeed ZeRO Stage 3 (Rajbhandari et al., 2020; Rasley et al., 2020; Ren et al., 2021) with parameters and optimizer states offloaded to CPU.

Each on-policy iteration generates up to 128 rollouts, one for each sampled prompt. Every retained rollout yields one trajectory-level training example containing the sampled blockwise denoising states described above. These examples are then used for student optimization under the training configuration described above. Keeping each rollout as a separate training example preserves the direct correspondence between its visited states and final completed trajectory throughout target construction. We do not pack multiple trajectory-level examples into a single sequence.

C.3 VERIFIER AND CORRECTNESS GATING

We follow the task-specific verification procedure used in the OPDLM training setup and use the resulting Boolean correctness signal as $R(\hat{y})$. Verification is applied only to determine whether the future-aware correction is enabled for a rollout. A verifier-correct rollout receives the future-aware teacher correction described in Section 4, whereas a verifier-incorrect rollout continues to use the standard causal teacher target. Thus, correctness gating does not discard unsuccessful trajectories from training and affects only the additional future-aware reweighting. Examples without an applicable correctness verifier receive standard causal teacher supervision.

C.4 BENCHMARK EVALUATION

We evaluate MMLU, MMLU-Pro, GPQA-Diamond, GSM8K, MATH500, and AIME25 using the repository-local OPDLM evaluation pipeline from our released code snapshot (Su et al., 2026). We follow the corresponding OPDLM benchmark formatting, answer extraction, and verification procedures. All six benchmarks are evaluated zero-shot using the Qwen3 chat template.

Evaluation uses the same block length as the corresponding model setting, four denoising steps per block, low-confidence static remasking, temperature 1.0, top-$p = 1.0$, and top-$k = 1$. We generate one completion per problem with at most 4,096 new tokens. Evaluation is performed on the full

benchmark sets: 14,042 MMLU examples, 12,032 MMLU-Pro examples, 198 GPQA-Diamond examples, 1,319 GSM8K examples, 500 MATH500 examples, and 30 AIME25 examples.

We evaluate checkpoints after iterations 50, 100, ..., 450 and once more at the final iteration 483. For each evaluated checkpoint, we compute the unweighted average accuracy over the six benchmarks. We then select the checkpoint with the highest six-benchmark average and report all six benchmark scores from this same checkpoint. The reported aggregate is the corresponding six-benchmark average. We do not independently select the best checkpoint for each benchmark.

D Enumerable Teacher Construction and Exact-Target Diagnosis

We construct a finite teacher distribution induced by Qwen3-8B so that the exact future-conditioned target can be evaluated by exhaustive enumeration. The purpose of this experiment is diagnostic: it isolates the target-conditioning mismatch in a controlled setting where the complete teacher-induced posterior can be computed exactly, allowing candidate teacher targets to be compared directly against the exact posterior rather than only through downstream task accuracy.

D.1 Restricted enumerable support

We construct a restricted enumerable vocabulary sequentially from left to right under a fixed causal context. Suppose the first j positions have already been assigned restricted vocabularies $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_j$ of size V, defining a set $\mathcal{A}_j$ of V^j possible restricted prefixes. For each prefix $A \in \mathcal{A}_j$, we query Qwen3-8B for the full next-token distribution and assign the prefix the normalized teacher weight

$$\omega(A) = \frac{p_T(A \mid q)}{\sum_{A' \in \mathcal{A}_j} p_T(A' \mid q)}. \tag{20}$$

We then form the probability-weighted mixture

$$\bar{p}_{j+1}(v) = \sum_{A \in \mathcal{A}_j} \omega(A) p_T(v \mid q, A). \tag{21}$$

The top-V tokens under $\bar{p}_{j+1}$ define the restricted vocabulary $\mathcal{S}_{j+1}$ at position $j+1$. Repeating this procedure yields a finite support that preserves high-probability local choices under the original AR teacher while remaining small enough for exhaustive enumeration.

D.2 Exact teacher posterior

After constructing the restricted vocabulary at every position, we enumerate every complete block sequence X in the resulting finite support. Each sequence is scored using the original AR factorization,

$$p_T(X \mid q) = \prod_{j=1}^{N} p_T(x_j \mid q, x_{<j}), \tag{22}$$

and the resulting probabilities are renormalized over the restricted support. For a partially observed state X_t, exhaustive enumeration then gives the exact block posterior in Equation 7 and the corresponding exact token marginal in Equation 8. Because every compatible completion is explicitly enumerated, this construction provides a direct reference target against which approximate teacher targets can be measured consistently under the same restricted support.

D.3 Compared targets

For each partially observed state X_t, we sample one compatible completion

$$\hat{X} = (\hat{x}_1, \ldots, \hat{x}_N) \tag{23}$$

from the corresponding completion distribution. For each masked position i, this completion provides the completed prefix $\hat{A}_i = \hat{x}_{<i}$ and completed suffix

$$\hat{C}_i = (\hat{x}_{i+1}, \ldots, \hat{x}_N). \tag{24}$$

The causal target is

$$r_i(v) = p_T(v \mid q, \hat{A}_i), \tag{25}$$

matching the ordinary causal AR supervision used by OPDLM. The future-aware target additionally scores how compatible the same completed future is with each candidate token,

$$s_i(v) = \log p_T(\hat{C}_i \mid q, \hat{A}_i, v), \tag{26}$$

and forms the relative score

$$\Delta s_i(v) = s_i(v) - s_i(\hat{x}_i). \tag{27}$$

The resulting future-aware target is obtained by reweighting the causal teacher distribution with $\exp(\Delta s_i(v))$ and normalizing over the restricted support. Because the enumerable support is small, this diagnostic applies the correction to the full restricted vocabulary rather than introducing the practical top-k approximation used in large-scale training. This allows the experiment to focus directly on whether incorporating completed future information moves the causal teacher target closer toward the exact future-conditioned posterior under controlled conditions.

D.4 KL METRIC AND AGGREGATION

For each masked position i, we measure target quality using

$$D_{\mathrm{KL}}\left(\pi_i(\cdot \mid X_t, q) \,\|\, \hat{p}_i(\cdot \mid X_t, q)\right), \tag{28}$$

where π_i is the exactly enumerated future-conditioned target from Equation 8, and $\hat{p}_i$ denotes either the causal or future-aware target defined above. The KL divergence is computed over the restricted candidate support at position i, providing a consistent token-level measure of target quality.

We evaluate block lengths $N \in \{4, 6, 8\}$ and restrict each position to V candidate tokens drawn from its position-specific enumerable support. This gives the ten configurations

$$(N, V) \in \{(4,6), (4,12), (4,24), (4,48), (6,6), (6,8), (6,12), (6,16), (8,6), (8,8)\}. \tag{29}$$

For each configuration, we sample 400 complete blocks from the restricted teacher distribution and independently retain each position with probability 0.45, masking the remaining positions. If all positions are retained, we uniformly select one position to mask so that every sampled state contains at least one prediction target. The exact target for every masked position is then obtained by exhaustive enumeration over the corresponding finite support.

To vary the discrepancy between the student and teacher completion distributions, we define a perturbed student distribution over the same finite complete-block support:

$$p_S^{(\sigma)}(X) \propto \exp\left(\log p_T(X) + \sigma Z_X\right), \qquad Z_X \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{30}$$

with $\sigma \in \{0, 0.5, 1.0, 1.5\}$. At $\sigma = 0$, the student distribution coincides with the restricted teacher distribution, while larger σ introduces stronger perturbations. The same 400 partially observed states are reused across all four noise levels. For each state and noise level, we restrict $p_S^{(\sigma)}$ to compatible complete blocks, renormalize it, and sample one completion $\hat{X}$ used to construct the causal and future-aware targets above, enabling a controlled comparison across perturbation strengths.

When no future token is visible to the right of position i, the exact future-compatibility term is constant across candidates, so the future-aware target reduces to the causal teacher target. These positions therefore provide a natural limiting case in which the two compared targets should coincide.

Across all configurations, noise levels, sampled states, and masked positions, the sweep contains 49,948 masked-token marginals. We pool the token-level KL values over the full sweep and report their arithmetic mean as the aggregate diagnostic statistic used in our analysis.

E WALL-CLOCK TIMING PROTOCOL

Training efficiency is measured using time to matched performance. For each model scale, we first identify the best six-benchmark average achieved by OPDLM under the checkpoint-selection protocol described in Appendix C. This value is then used as the matched-performance target for the

Table 4: Mean KL divergence to the exactly enumerated future-conditioned target, averaged over 49,948 masked-token marginals across all controlled configurations. Lower is better.

Target	KL	Reduction
Causal teacher	0.4444	–
Future-aware	**0.1249**	**71.9%**

corresponding comparison. We measure the elapsed wall-clock time at which OPDLM reaches this value and the first elapsed time at which d-OPD reaches or exceeds the same threshold.

We separately measure the wall-clock overhead introduced by future-aware scoring during training for Qwen3-0.6B, 1.7B, 4B, and 8B. Each measured duration covers the complete on-policy iteration and therefore includes rollout generation, verifier computation, teacher scoring, student optimization, model serialization, and reloading the updated student weights into the rollout engine. Periodic benchmark evaluation is performed outside the timed region and is excluded from the measurement, while routine training and checkpointing costs remain included so that the reported timing reflects the actual end-to-end training workflow under the same experimental setup as closely as possible.

All measurements use eight NVIDIA H100 80 GB GPUs and the same distributed configuration for OPDLM and d-OPD at each model scale. For each method and model scale, we compute the mean wall-clock duration per on-policy iteration and use this quantity to compare their per-iteration training cost. The relative overhead is defined as the percentage increase in mean iteration time of d-OPD over OPDLM, and we report its unweighted average across the four evaluated model scales. Under the default $k = 16$ configuration, this yields the 2.63% average per-iteration overhead reported in the main text. Candidate restriction and correctness gating are both enabled in this measurement, matching the default d-OPD training configuration used in the main experiments.

F Limitations and Future Work

d-OPD currently uses a single completed on-policy trajectory to construct the future-aware target. A natural extension is to sample multiple compatible trajectories for the same student-visited state and aggregate their future evidence when reweighting the teacher distribution. Averaging across several possible futures may reduce the variance introduced by relying on a single sampled completion and provide a more stable estimate of the future-conditioned teacher target. In addition, d-OPD is complementary to supervised fine-tuning because it modifies the teacher target during on-policy distillation rather than the student architecture or the underlying training objective. This makes it straightforward to combine the two paradigms in a two-stage training pipeline, using SFT for initial adaptation followed by d-OPD for on-policy refinement. Studying how these two stages interact, and whether SFT provides a stronger initialization for subsequent future-aware distillation, is an interesting direction for future work. More broadly, evaluating d-OPD across additional model families and block sizes is also an important direction for future work.

G Additional Ablation Results

Table 5 provides the full per-benchmark results for the correctness gate ablation reported in Table 3.

Table 5: Full results for the correctness-gating ablation on Qwen3-8B. The gate applies future-aware correction only to verifier-correct rollouts, while ordinary causal teacher supervision remains active for all rollouts. Avg. denotes the unweighted average over the six benchmarks. Higher is better.

Method	Gate	MMLU	MMLU-P	GPQA-D	GSM8K	MATH500	AIME25	Avg.
OPDLM	–	69.2	54.0	38.4	87.2	76.8	10.0	55.9
d-OPD	Off	74.2	**56.9**	37.4	87.9	76.2	20.0	58.8
d-OPD	**On**	**74.3**	55.9	**38.9**	**89.3**	**77.2**	**23.3**	**59.8**