PREFIX-CONDITIONED DIFFUSION: REDUCING THE PRETRAINING–GENERATION MISMATCH IN DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (dLLMs) generate text through iterative denoising, allowing multiple tokens to be predicted in parallel. However, pretraining may mask tokens throughout a sequence, whereas prompt continuation conditions on an intact prefix. This difference remains in conversion pipelines that denoise entire sequences during the stable stage. We propose *Prefix-Conditioned Diffusion* (PCD), which samples a boundary, preserves the prefix, and denoises the suffix. The training recipe also applies autoregressive supervision to the prefix. We evaluate PCD in the stable stage of a warmup, stable, and decay conversion pipeline, with inference unchanged. Matched experiments across model families show improvements in reasoning and coding over native diffusion training. A matched continuation study further shows that the advantage persists after a shared decay stage. In a separate reconstruction diagnostic, the full PCD recipe's advantage over native diffusion training reverses as more evaluation prefix tokens are masked. Controlled experiments show lower suffix reconstruction loss with a clean training prefix, an intact evaluation prefix, and no autoregressive loss.

1 INTRODUCTION

Diffusion language models (dLLMs) generate text through iterative denoising, predicting multiple tokens in parallel (Nie et al., 2025). For prompt continuation, they condition on an intact prefix. Autoregressive (AR) training reflects this structure by predicting each token from its preceding context (Vaswani et al., 2017; Brown et al., 2020). Diffusion pretraining over entire sequences, however, may mask tokens in this conditioning prefix. This difference motivates testing whether preserving prefix context during pretraining improves prompt continuation.

Existing block diffusion methods already condition on clean preceding blocks (Arriola et al., 2025; Jain, 2026). This conditioning structure need not be maintained throughout the conversion of an AR model into a dLLM. For example, LLaDA2.0 uses a warmup, stable, and decay block-size schedule, analogous to WSD learning-rate scheduling (Wen et al., 2025; Bie et al., 2025). Its stable stage denoises entire sequences before returning to block diffusion during decay. We use this pipeline to test whether preserving a clean prefix during stable training improves downstream performance and whether the benefit persists after the same decay.

Our approach constructs the conditioning structure of prompt continuation directly from unlabeled text. *Prefix-Conditioned Diffusion* (PCD) samples a boundary within each sequence, preserves the prefix, and applies diffusion corruption only to the suffix (Figure 1). Sampling the boundary exposes the model to different amounts of visible context without requiring annotated prompt and response pairs. The full recipe also applies causal AR supervision to the prefix, retaining the objective of predicting the next token alongside suffix denoising. The objective does not require WSD. Here, we apply it during stable training while retaining the warmup, decay, and inference procedures.

Matched comparisons across two model families show reasoning and coding gains over native diffusion training. On LLaDA2-Mini, the advantage persists after both checkpoints undergo the same decay procedure. Separate public controls use block diffusion at a smaller budget. With attention and suffix supervision fixed and no AR loss, preserving the training prefix lowers reconstruction loss

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.

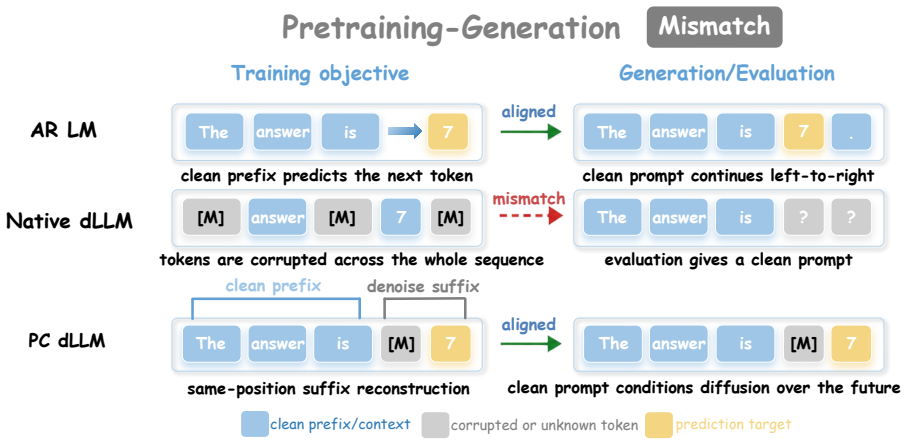


Figure 1: Motivation for Prefix-Conditioned Diffusion (PCD). Native diffusion pretraining can corrupt the prefix even though prompt continuation provides it intact at evaluation. The full PCD recipe applies an AR loss to the prefix and an MDM loss to the suffix. Its defining interface change is that suffix denoising conditions on a clean prefix.

under evaluation with an intact prefix. A different diagnostic shows that the full recipe's reconstruction advantage over native training reverses as more evaluation prefix tokens are masked.

Our contributions are threefold:

- **Prefix-conditioned training.** PCD partitions each sequence at a sampled boundary, preserving the prefix for suffix denoising to reduce the mismatch with prompt continuation. The full recipe combines causal prefix AR supervision with same-position suffix reconstruction.
- **Task gains and persistence through decay.** Matched comparisons across two model families show task gains for the complete recipe. In the LLaDA2-Mini comparison, an average advantage remains after both checkpoints undergo the same decay procedure.
- **Controlled evidence for prefix visibility.** A separate public control fixes attention and suffix supervision without AR loss. Clean training prefixes lower reconstruction loss under clean-prefix evaluation, isolating a visibility benefit within the tested block-causal setting.

2 Related Work

Diffusion language models. Diffusion models generate data through iterative denoising (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021), with discrete, categorical, and continuous formulations for language (Hoogeboom et al., 2021; Austin et al., 2021a; Li et al., 2022; Gong et al., 2023). Discrete dLLMs improve objectives and sampling (Lou et al., 2024; Sahoo et al., 2024), while LLaDA and Dream demonstrate large-scale pretraining and supervised fine-tuning (Nie et al., 2025; Ye et al., 2025). Earlier continuous-diffusion work reports prefix-masking trade-offs across conditioning settings (Lo Cicero Vaina et al., 2023). WSD supports branching and learning-rate decay in language-model pretraining (Wen et al., 2025); LLaDA2.0 uses a WSD-like recipe to preserve AR knowledge, train with full-sequence diffusion, and decay to block-diffusion checkpoints (Bie et al., 2025). Nemotron-Labs-Diffusion jointly trains AR and diffusion objectives with a strictly causal clean stream and a noisy stream, supporting AR, diffusion, and self-speculative decoding (Fu et al., 2026). Joint AR/denoising training is thus established and does not itself distinguish PCD.

Block diffusion. Block diffusion combines autoregression across blocks with denoising within blocks for arbitrary-length generation; BD3-LM trains noisy blocks conditioned on clean preceding blocks (Arriola et al., 2025). Blockwise SFT aligns one active response block with semi-autoregressive inference by freezing preceding response blocks and hiding future ones (Sun et al., 2025); prompt-infilling work instead extends masking over prompts during SFT (Fujinuma & Sak-

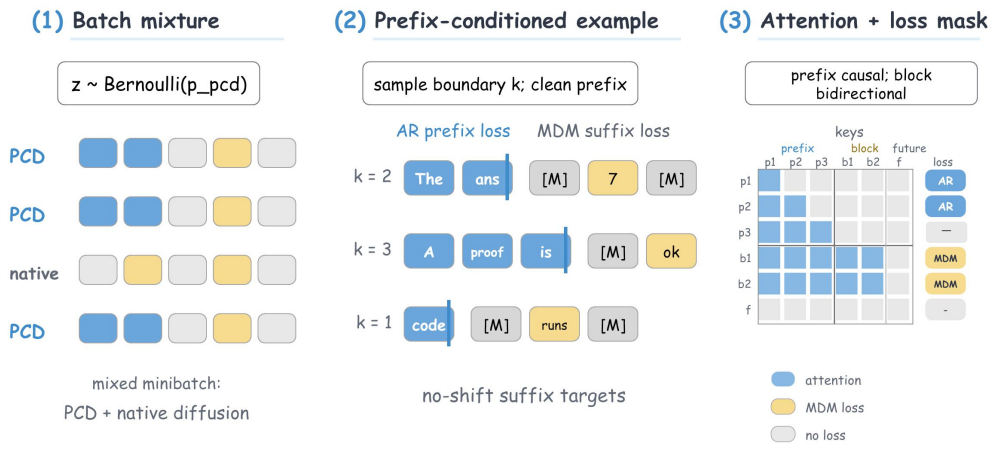


Figure 2: PCD training. Panel (3) shows block-causal attention for prefix and active-block queries; future queries are omitted. Full-scale suffix attention is bidirectional. AR logits end at $k-1$ (none for $k=1$); MDM loss uses masked suffix positions only.

aguchi, 2026). Adaptive Block Diffusion samples prefix-window configurations and denoises an active window given clean prefix context (Jain, 2026), while training-free analyses expose a left-to-right generation bias in block-diffusion checkpoints (Hou & Kwok, 2026). Decoding efficiency is also addressed through confidence-based horizon adaptation (Lu et al., 2026a) and on-policy transition distillation (Lu et al., 2026b). PCD uses a single clean-prefix/noisy-suffix sequence, rather than Nemotron-Labs-Diffusion's dual streams. Its sampled prefix and remaining suffix form a restricted case of Adaptive Block Diffusion's prefix-window configurations; the full PCD recipe also includes prefix AR supervision. Our WSD comparisons ask whether applying PCD during stable training leaves a task advantage after the same decay; the public intervention tests prefix visibility at fixed attention and supervision. Without matched training comparisons, our results do not rank these methods.

3 Method

PCD preserves a sampled prefix during denoising, independently of the training schedule. We define its corruption rule, attention pattern, and joint objective before describing the training procedure and its application to WSD (Figure 2).

3.1 Prefix-Preserving Denoising

Let $x = (x_1, \ldots, x_T)$ be a sequence and $m_i = 1$ mark a prediction target. Masked diffusion replaces selected tokens with `[MASK]` and leaves the others unchanged, producing $\tilde{x}$. Native training allows targets at any position and minimizes

$$\mathcal{L}_{\text{native}} = -\mathbb{E}_{x,m} \frac{\sum_{i:m_i=1} \log p_\theta(x_i \mid \tilde{x}, m)}{\max(\sum_i m_i, 1)}. \tag{1}$$

Conditioning denotes inputs accessible under the attention mask. This is the per-example normalized cross-entropy used here, not a general diffusion likelihood bound.

PCD samples $\rho \sim q(\rho)$ and sets $k = \min(T-1, \max(1, \lfloor \rho T \rfloor))$, keeping both regions nonempty. Sampling the boundary varies the amount of conditioning context without requiring annotated prompt/response pairs. With $m_i^{\text{pcd}} = \mathbb{I}[i > k] m_i$, its input is

$$\tilde{x}_i^{\text{pcd}} = \begin{cases} \texttt{[MASK]}, & m_i^{\text{pcd}} = 1, \\ x_i, & \text{otherwise.} \end{cases} \tag{2}$$

Only masked suffix positions receive denoising targets. Noise sampling is a recipe choice; the public controls draw a masking rate and suffix masks, ensuring at least one target (Appendix B.2). These settings are not assigned to the original full-scale runs.

3.2 Attention and Joint Supervision

Let $A_{ij} = 1$ allow query position i to read key position j. PCD uses

$$A_{ij} = \begin{cases} \mathbb{I}[j \le i], & i \le k, \\ 1, & i > k,\ j \le k, \\ S_{ij}, & i > k,\ j > k. \end{cases} \tag{3}$$

The causal prefix cannot read suffix tokens, preventing leakage into its AR predictions. Every suffix query can read the intact prefix. Full-scale stable training uses bidirectional suffix attention, $S_{ij} = 1$. The public controls instead use $S_{ij} = \mathbb{I}[b(j) \le b(i)]$, where $b(i) = \lfloor (i - k - 1)/B \rfloor$ for $i > k$ and block size B. Thus their suffix blocks begin at $k + 1$ and attend to current and preceding blocks.

The complete recipe adds AR supervision to train on the preserved prefix. Its losses are

$$\begin{aligned} \ell_{\text{suf}} &= -\frac{1}{N_m} \sum_{i: m_i^{\text{pcd}} = 1} \log p_\theta(x_i \mid \tilde{x}^{\text{pcd}}, m^{\text{pcd}}, k), \\ \ell_{\text{ar}} &= -\frac{1}{N_{\text{ar}}} \sum_{i=2}^{k} \log p_\theta(x_i \mid x_{<i}), \\ \mathcal{L}_{\text{pcd}} &= \mathbb{E}_{x,\rho,m}[\lambda_{\text{mdm}} \ell_{\text{suf}} + \lambda_{\text{ar}} \ell_{\text{ar}}], \end{aligned} \tag{4}$$

where $N_m = \max(\sum_i m_i^{\text{pcd}}, 1)$ and $N_{\text{ar}} = \max(k - 1, 1)$. Prefix logits at positions $1, \ldots, k - 1$ predict $x_2, \ldots, x_k$; position k has no AR target. Suffix logits reconstruct tokens at the same positions ("no-shift"). If $k = 1$, the AR loss is zero. The boundary k specifies attention and supervision, not a learned input.

Each region is normalized by its own target count before weighting, so a longer region does not dominate merely by having more targets. Unit weights assign equal coefficients to region averages, not individual tokens. The full-scale recipe uses $\lambda_{\text{ar}} = \lambda_{\text{mdm}} = 1$ unless varied in a sweep. Setting $\lambda_{\text{ar}} = 0$ removes prefix supervision while retaining the same PCD inputs, attention, and suffix targets; the downstream benefit of the AR term remains an empirical question.

3.3 Training Recipe and WSD Evaluation Setting

The joint objective provides *intra-sample* mixing of AR and denoising. Optional *inter-sample* mixing selects PCD with probability p_{pcd} and native diffusion otherwise:

$$\mathcal{L} = p_{\text{pcd}} \mathcal{L}_{\text{pcd}} + (1 - p_{\text{pcd}}) \mathcal{L}_{\text{native}}. \tag{5}$$

For each example, select the objective branch; for PCD, sample the boundary, corrupt the suffix, and construct Equation 3. Compute the selected branch's normalized losses, average over examples, and update the model. The prefix distribution, mixing probability, and AR weight are independent choices. At prefix range $[1, 1]$, clamping gives $k = T - 1$, leaving one denoising token; we therefore call its mixture with native training *AR-dominant/native mixing*.

WSD is our evaluation setting, not a requirement of PCD. We apply PCD during stable training after a shared AR-to-block-diffusion warmup. Matched decay then tests whether the task advantage persists under the same subsequent training and inference. PCD requires no AR decoding, verification, or self-speculation at inference. Appendix A separates retained full-scale settings from public-control configurations.

4 Experiments

4.1 Setup

We evaluate LLaDA2-Mini and Qwen-1.7B checkpoints across reasoning, coding, math, and knowledge benchmarks. Within each native/PCD pair, no-shift SGLang settings, 8k context, prompts,

Table 1: Main results with external scale references. Avg.6 averages the six displayed tasks. External rows contextualize model scale; the LLaDA2-Mini rows provide the controlled comparison, with 400B stable tokens and a matched 50B-token decay. Shading marks PCD; bold marks the best LLaDA2-Mini score in each metric. Table 2 reports the remaining six tasks.

Model	Params	Avg.6	GSM8K	MATH	BBH	KorBench	MBPP	HumanEval-FIM
External dLLM references								
LLaDA base	8B	50.12	73.62	31.48	47.18	32.72	40.60	75.12
LLaDA-MoE base	7B-A1B	51.94	66.41	36.10	52.71	31.20	52.40	72.80
Dream base	7B	57.48	81.58	41.50	53.38	37.68	53.20	77.54
External AR scale references								
LLaMA3-8B base	8B	48.50	58.07	17.74	58.83	33.04	48.20	75.12
Qwen2.5-7B base	7B	61.87	86.28	51.10	56.79	35.92	62.40	78.70
Representative matched LLaDA2-Mini comparison								
LLaDA2-Mini native stable	16B-A1B	60.30	75.82	56.38	66.38	33.12	59.40	70.67
LLaDA2-Mini native decay	16B-A1B	62.87	80.82	56.92	68.96	**37.36**	59.40	73.77
LLaDA2-Mini **PCD stable**	16B-A1B	62.86	80.44	57.64	68.40	35.84	60.20	**74.64**
LLaDA2-Mini **PCD decay**	16B-A1B	**64.54**	**83.47**	**59.52**	**69.63**	37.28	**63.20**	74.15

Table 2: Remaining six tasks in the matched LLaDA2-Mini comparison. Together with Table 1, these rows display all 12 task scores; Avg.12 averages the complete suite. Bold marks the better value within each matched budget pair. All intermediate checkpoints appear in Table 5.

Checkpoint	Avg.12	NQ	TriviaQA	MMLU-Pro	OmniMath	LCBench	AutoLogi
Native stable (400B)	47.26	25.84	54.27	44.74	19.70	27.54	33.27
PCD stable (400B)	**49.74**	**29.17**	**57.46**	**46.08**	**20.90**	**30.72**	**35.37**
Native decay (+50B)	49.30	27.67	55.80	43.94	18.50	29.56	**38.95**
PCD decay (+50B)	**50.94**	**29.53**	**57.99**	**46.10**	**20.80**	**31.41**	38.21

initialization, data, and token budget are matched. The controlled baseline is the same-family native stable objective, which corrupts the sequence without preserving a clean prefix. Public dLLMs such as LLaDA, LLaDA-MoE, and Dream (Nie et al., 2025; Zhu et al., 2025a; Ye et al., 2025), together with AR checkpoints, provide scale references rather than controlled comparisons. We use the LLaDA2-Mini 400B stable pair as the primary recipe comparison, matched +50B decay as a continuation check, and Qwen-1.7B at 50B tokens as a second-backbone task study. These comparisons evaluate the complete PCD objective relative to native stable training, rather than isolating each ingredient. For PCD examples, suffix targets are unshifted and losses use per-region target normalization. Appendix A separates the retained full-scale settings, public-control configuration, and reported checkpoint summaries; the objective sweeps are reported in Appendix A. Separate public-data reconstruction controls use Qwen3-1.7B-Base and FineWeb-Edu at a smaller budget and with block-causal suffix attention. For seeds 42–44, four recipes branch from one 20M-token warmup for 50M stable tokens: native MDM, PCD, clean-prefix/no-AR, and full-sequence AR plus native MDM in separate views. A follow-up with seeds 52–54 changes only training-prefix corruption and uses a fresh 2M-token holdout. Both public studies pair evaluator draws across checkpoints and measure masked-token reconstruction cross-entropy. Their 50M-token stable budgets are distinct from the 50B-token Qwen task study; they test reconstruction behavior rather than reproduce its task gains.

4.2 Benchmarks

The LLaDA2-Mini suite covers knowledge/QA (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), MMLU-Pro (Wang et al., 2024)), math (OmniMath (Gao et al., 2024),

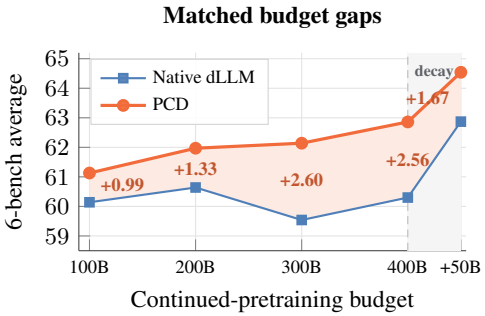


Figure 3: Matched budget gaps. PCD stays above native dLLM across stable budgets and matched +50B decay.

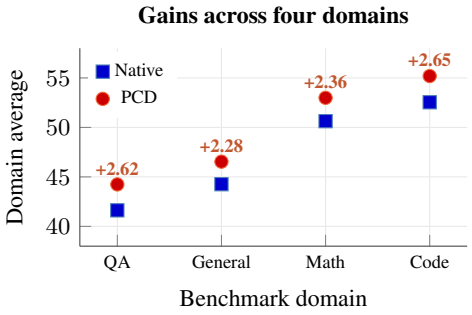


Figure 4: 12-benchmark domain averages at the final stable stage. Orange labels report PCD gains.

Table 3: Qwen-1.7B objective variants at 50B continued pretraining. HE denotes standard HumanEval. All use matched initialization, data, budget, and evaluation. Bold marks the best value in this controlled set; the close Avg.6 values do not identify a unique causal component.

Setting	p_{pcd}	Prefix range	Avg.6	HE	GSM8K	MATH	BBH	KorBench	MBPP
Native baseline	0.00	–	34.28	25.00	59.14	26.84	39.72	20.96	34.00
Mixed PCD	0.50	[0.0, 1.0]	38.94	28.05	66.94	33.36	45.87	22.00	37.40
Intra-sample only	1.00	[0.0, 1.0]	**39.14**	25.61	**67.70**	**33.68**	**46.20**	21.84	**39.80**
AR-dominant/native mix	0.50	[1.0, 1.0]	38.98	**29.88**	62.93	**33.68**	45.84	**23.12**	38.40

MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021)), code (MBPP (Austin et al., 2021b), HumanEval-FIM, LiveCodeBench (LCBench) (Jain et al., 2024)), and general reasoning (BBH (Suzgun et al., 2022), KorBench (Ma et al., 2024), AutoLogi (Zhu et al., 2025b)). We report an infilling variant of HumanEval (Chen et al., 2021) as HumanEval-FIM. The Qwen-1.7B suite uses the six benchmarks listed in its tables, including standard HumanEval rather than HumanEval-FIM. Task scores use a 0–100 scale (higher is better). Avg.6 and Avg.12 are unweighted means of the specified task scores; gains are absolute score-point differences. Chat-SFT averages five tasks. Public reconstruction controls report masked-suffix cross-entropy in nats (lower is better).

4.3 Main Results

Table 1 reports external dLLM and AR references alongside the matched LLaDA2-Mini stable and decay results. The stable pair tests the objective replacement; the +50B pair tests whether adapting both parents through the same decay removes the resulting advantage. At the matched 400B stable budget, PCD improves the native dLLM stable baseline by +2.56 Avg.6 points. Figure 3 shows that the advantage persists along the paired training trajectories: PCD improves over the same-family native baseline by +0.99 Avg.6 at 100B, +1.33 at 200B, +2.60 at 300B, and +2.56 at 400B. At 400B, PCD improves all 12 tasks in the wider suite, raising Avg.12 from 47.26 to 49.74 (+2.48); the gain spans knowledge/QA, math, code, and general reasoning. Tables 1 and 2 report all 12 tasks; Appendix Table 5 gives the complete trajectories.

After matched +50B decay, PCD retains +1.67 Avg.6 and +1.64 Avg.12 points, improving 10/12 tasks. The exceptions are KorBench (-0.08) and AutoLogi (-0.74); these are descriptive differences on the reported task scores. These scores describe one paired full-scale trajectory; Section 5 discusses uncertainty and attribution. Chat-SFT transfer and AR-decoding compatibility are examined in Section 5 (Figures 8 and 9).

4.4 Recipe Controls and Prefix Visibility

Warmup and continuation. Under the tested Qwen recipe, stable PCD performs much better after AR-to-BD warmup than when initialized directly from an AR checkpoint (Appendix Table 10). We

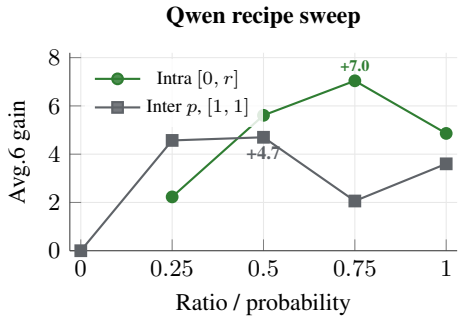


Figure 5: Qwen-1.7B 50B-token Avg.6 gains. Intra varies r; Inter varies p at the AR-dominant prefix endpoint $[1, 1]$. Full scores are in Tables 3 and 8.

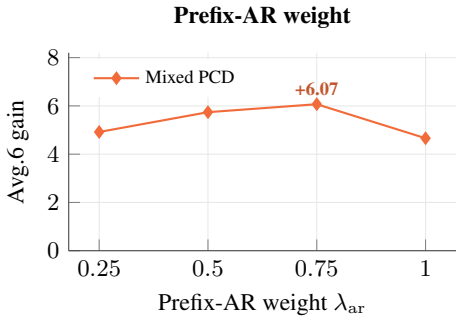


Figure 6: Qwen-1.7B 50B-token Avg.6 gains over native training. Mixed PCD fixes $p_{pcd} = 0.5$, prefix range $[0, 1]$, and $\lambda_{mdm} = 1$; only λ_{ar} varies. Full scores are in Table 9.

Table 4: Single-factor training-prefix intervention on Qwen3-1.7B. Values are per-example normalized masked-suffix cross-entropy in nats under the same clean-prefix evaluation interface (lower is better). Each row averages 976 blocks and three paired evaluator seeds; SD is across training seeds sharing one warmup. The holdout and FP32 reduction differ from Table 15, so absolute losses should not be compared across tables.

Training seed	Clean-prefix training	Corrupted-prefix training	Δ (clean minus corrupted)
52	3.9492	4.0173	-0.0681
53	3.9514	4.0200	-0.0686
54	3.9497	4.0205	-0.0708
Mean $\pm$ SD	3.9501 ± 0.0012	4.0193 ± 0.0017	-0.0692 ± 0.0015

therefore retain that warmup in the studied pipeline. Without a direct AR-to-native control, this diagnostic cannot identify a PCD-specific dependence on warmup. Longer PCD-only continuations appear in Appendix Table 7; without corresponding native runs, they do not extend the matched decay comparison.

Intra- vs. inter-sample mixing. Under matched Qwen-1.7B settings, intra-sample-only PCD gives a +4.86 Avg.6 gain; the AR-dominant/native mixture (+4.70) and mixed PCD (+4.66) yield similar reported averages (Table 3). All three variants exceed native training; Section 5 discusses their attribution limits. The finer ratio/probability sweep in Figure 5 shows that the strongest reported prefix range is $[0, 0.75]$. Its numerical maxima describe the completed sweep, rather than a recipe selected by an independently documented validation rule.

Objective design. At 100B tokens, same-position PCD suffix targets achieve 61.13 Avg.6 versus 54.43 for shifted targets; mismatching training and evaluation shifts reduces the score to 0.64 (Appendix Table 6). In the Qwen mixed recipe, all tested prefix-AR weights from 0.25 to 1.00 improve over the native baseline; weight 0.75 reaches 40.35 Avg.6 (Figure 6; full scores in Table 9). This sweep measures weight sensitivity; a matched zero-AR task comparison remains absent.

Clean-prefix/no-AR yields the lowest reconstruction loss in the public four-recipe screen (Appendix Table 15). We test prefix visibility directly below by fixing attention and suffix supervision.

Matched training-prefix intervention. We pair hybrid attention, suffix boundaries, masks, targets, normalization, data order, and optimization, with no AR loss; only training-prefix corruption differs. The corrupted-prefix arm therefore retains the clean arm's attention and suffix-only supervision, unlike the native-MDM recipe in Appendix Table 15. Both arms are evaluated with a clean prefix on the same fresh holdout. Across three new training seeds sharing the warmup parent, the clean-minus-corrupted difference is -0.0692 nats (paired training-seed 95% interval $[-0.0728, -0.0655]$); every seed improves, exceeding the prespecified 0.02 reduction threshold.

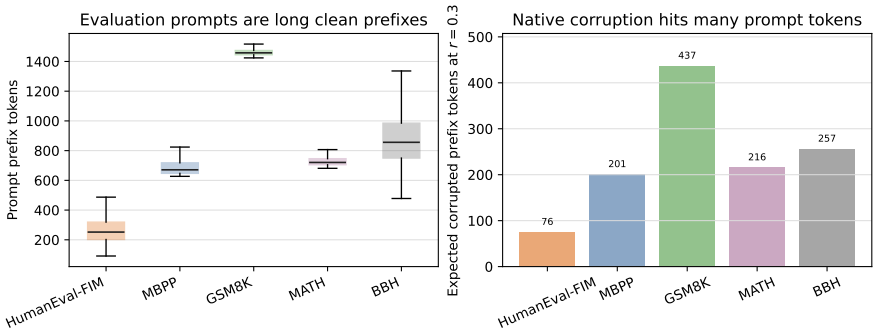


Figure 7: Prompt-length distributions for five benchmarks (left) and expected masked-prefix counts for their median-length prompts at $r = 0.3$ (right). HumanEval-FIM uses a length proxy; see Table 13.

Table 4 reports the individual seeds; Appendix B.2 gives the protocol and frozen decision rule. This isolates a training-prefix benefit under the fixed block-causal reconstruction interface, not a decomposition of the full-scale task gains.

Appendix Table 14 lists the changed factors: native-to-PCD comparisons combine attention, targets, and conditioning, whereas the matched pair isolates training-prefix visibility under one reconstruction interface.

5 Analysis

Benchmark and budget dependence. At 400B stable, the gains cover all four domains: +2.62 in knowledge/QA, +2.28 in general reasoning, +2.36 in math, and +2.65 in code (Figure 4). The aggregate trend coexists with task-specific variation. For example, MBPP is lower for PCD at 100B (57.60 versus 58.80), but higher at 400B (60.20 versus 59.40). Tables 1 and 2 expose the final stable and decay differences, while Appendix Table 5 retains every intermediate checkpoint.

Clean-prefix structure of benchmark prompts. Across the five benchmarks in Figure 7, 14,363 public examples rendered with the evaluation templates and a fixed Qwen tokenizer have median prefix and target lengths of 754 and 12 tokens (Table 13). HumanEval-FIM uses a length proxy. At $r = 0.3$, a median-length prefix has 226.2 expected masked tokens, illustrating the clean context available at evaluation.

Evaluation-time prefix corruption. In the public reconstruction study, PCD's advantage disappears and reverses as evaluation prefixes are masked, indicating dependence on visible context (Appendix Figure 10).

What survives the common decay? The stable checkpoint measures objective replacement; matched decay tests what remains after both branches return to block diffusion. Over +50B, native gains +2.57 Avg.6 versus +1.68 for PCD, compressing the gap from +2.56 to +1.67. PCD still leads on average, but this catch-up and two task reversals preclude extrapolation beyond the tested decay.

Transfer under chat SFT. We also compare the reported native-derived and PCD-derived chat-SFT runs. The exact starting checkpoints are not recorded, so this comparison is descriptive and separate from the matched-decay study. Under the same SFT recipe, the final reported epoch scores 67.39 for the PCD-derived checkpoint versus 65.84 for native SFT, a +1.55-point difference (Figure 8). The five corresponding-epoch gaps range from -0.08 to $+2.01$, so the advantage is not uniform along the trajectory. This comparison uses one pair of SFT runs and a shared final epoch, without selecting separate best checkpoints. HumanEval-FIM is excluded because chat tuning changes the infilling interface; full task scores remain in Appendix Table 11.

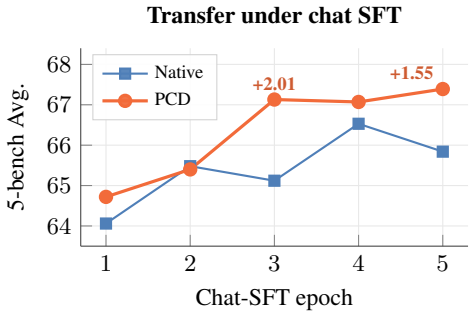


Figure 8: Chat-SFT transfer on LLaDA2-Mini. HumanEval-FIM is excluded because chat tuning changes the infilling interface. All epochs and task scores are in Appendix Table 11.

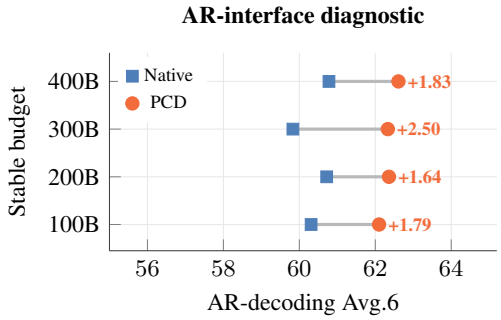


Figure 9: AR-interface diagnostic. Avg.6 uses standard HumanEval rather than HumanEval-FIM; orange labels report gains. Details are in Appendix Table 12.

Compatibility with AR decoding. Figure 9 evaluates the matched stable checkpoints through their AR interface; Avg.6 replaces HumanEval-FIM with standard HumanEval for causal continuation. PCD remains above native stable training at all four budgets, with Avg.6 gains of +1.79, +1.64, +2.50, and +1.83 at 100B, 200B, 300B, and 400B, respectively. These results show that the observed diffusion gains coexist with stronger AR-decoding scores than the matched native baseline. The per-task scores and consistent MBPP judging protocol appear in Appendix Table 12.

What can be attributed to prefix visibility? The full-scale comparisons establish a task advantage for the complete recipe, which changes attention, corruption support, and supervision. The separate no-AR intervention isolates training-prefix visibility and finds a reconstruction benefit; evaluation-prefix corruption instead reveals the full recipe's dependence on supplied context. These effects are not an additive explanation of the full-scale task gains. Auxiliary AR's task-level contribution remains unresolved, and the AR-interface comparison uses native stable checkpoints rather than the original AR parent, so it does not establish preservation of the parent's capabilities.

The public generation checks did not meet the paired readiness criterion under the tested samplers (Appendix B.4). The public controls therefore support reconstruction claims; their effects on generated-answer accuracy and arbitrary infilling remain unestablished.

Comparison unit and uncertainty. The 100B–400B points form one paired native/PCD trajectory; matched decay adds 50B tokens to each 400B parent, and chat SFT compares corresponding epochs. The full-scale averages therefore describe completed trajectories rather than independent-training variability. Public reconstruction intervals quantify variation across stable-stage seeds conditional on a shared warmup parent and their declared evaluation interface; they do not estimate uncertainty in full-scale task accuracy.

6 Conclusion

PCD constructs clean-prefix conditioning from unlabeled sequences by sampling a boundary and restricting corruption to the suffix. Its full recipe combines suffix denoising with causal prefix supervision. The treatment of conditioning context during stable training remains consequential in the studied AR-to-diffusion conversion pipeline: PCD's task advantage remains after the matched +50B-token decay studied here, with inference unchanged. In the primary comparison, PCD improves all 12 tasks at the final stable checkpoint and retains gains on 10 after matched decay. Separate public studies identify a reconstruction benefit from training-prefix visibility without AR supervision and a context-dependent specialization cost for the full recipe. These findings support evaluating stable-stage conditioning through both the resulting task behavior and its dependence on visible context. The downstream value of auxiliary AR and the transfer of the isolated visibility effect to full-scale task gains remain unresolved.

Reproducibility Statement

The corruption rule, attention, and objectives are specified in Sections 3.1 and 3.2, with the comparison protocol in Appendix A. Appendix C states the assumptions and proof of the information diagnostic. The public studies report pinned model/data revisions, training and evaluation settings, and the fresh-holdout construction in Appendix B.2.

References

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. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

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, 2021b. URL https://arxiv.org/abs/2108.07732.

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. *arXiv preprint arXiv:2512.15745*, 2025. URL https://arxiv.org/abs/2512.15745.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, 2021. URL https://arxiv.org/abs/2107.03374.

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, 2021. URL https://arxiv.org/abs/2110.14168.

Yonggan Fu, Lexington Whalen, Abhinav Garg, Chengyue Wu, Maksim Khadkevich, Nicolai Oswald, Enze Xie, Daniel Egert, Sharath Turuvekere Sreenivas, Shizhe Diao, Chenhan Yu, Ye Yu, Weijia Chen, Sajad Norouzi, Jingyu Liu, Shiyi Lan, Ligeng Zhu, Jin Wang, Jindong Jiang, Morteza Mardani, Mehran Maghoumi, Song Han, Ante Jukić, Nima Tajbakhsh, Jan Kautz,

and Pavlo Molchanov. Nemotron-Labs-Diffusion: A tri-mode language model unifying autoregressive, diffusion, and self-speculation decoding. Technical report, NVIDIA, 2026. URL https://research.nvidia.com/publication/2026-05_nemotron-labs-diffusion-tri-mode-language-model-unifying-autoregressive.

Yoshinari Fujinuma and Keisuke Sakaguchi. Unlocking prompt infilling capability for diffusion language models, 2026. URL https://arxiv.org/abs/2604.03677.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark For Large Language Models, 2024. URL https://arxiv.org/abs/2410.07985.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jQj-_rLVXsj.

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, 2021. URL https://arxiv.org/abs/2103.03874.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, 2021. URL https://papers.neurips.cc/paper/2021/hash/67d96d458abdef21792e6d8e590244e7-Abstract.html.

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

Gagan Jain. Adaptive block diffusion: Resolving training-inference mismatch in diffusion language models, 2026. URL https://arxiv.org/abs/2606.29275.

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, 2024. URL https://arxiv.org/abs/2403.07974.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/1be5bc25d50895ee656b8c2d9eb89d6a-Abstract-Conference.html.

Sofia Maria Lo Cicero Vaina, Nikita Balagansky, and Daniil Gavrilov. Diffusion language models generation can be halted early. *arXiv preprint arXiv:2305.10818*, 2023. URL https://arxiv.org/abs/2305.10818.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2310.16834.

Xiaocheng Lu, Shuhan Guo, Ziyue Ma, Jie Zhang, Jian Liu, Jingcai Guo, Haoxuan Che, and Song Guo. PACE-dLLM: Elastic block decoding via confidence cliff estimation for diffusion language models, 2026a. URL https://arxiv.org/abs/2609.26249.

Xiaocheng Lu, Hualei Zhang, Shuhan Guo, Jie Zhang, Xiaoyi Pang, Jian Liu, Haoxi Li, Bohai Gu, Haoxuan Che, Jingcai Guo, and Song Guo. OPTD: On-policy transition distillation with consistency-guided adaptive compression for few-step diffusion language models, 2026b. URL https://arxiv.org/abs/2608.02942.

Kaijing Ma, Xinrun Du, Yunran Wang, Haoran Zhang, Zhoufutu Wen, Xingwei Qu, Jian Yang, Jiaheng Liu, Minghao Liu, Xiang Yue, Wenhao Huang, and Ge Zhang. KOR-Bench: Benchmarking Language Models on Knowledge-Orthogonal Reasoning Tasks, 2024. URL https://arxiv.org/abs/2410.06526.

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.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.07524.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

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.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them, 2022. URL https://arxiv.org/abs/2210.09261.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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, 2024. URL https://arxiv.org/abs/2406.01574.

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=m51BgoqvbP.

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.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tieyuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, Xiaolu Zhang, Zhenzhong Lan, Junbo Zhao, Da Zheng, Chongxuan Li, Jianguo Li, and Ji-Rong Wen. LLaDA-MoE: A sparse MoE diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025a. URL https://arxiv.org/abs/2509.24389.

Qin Zhu, Fei Huang, Runyu Peng, Keming Lu, Bowen Yu, Qinyuan Cheng, Xipeng Qiu, Xuanjing Huang, and Junyang Lin. AutoLogi: Automated Generation of Logic Puzzles for Evaluating Reasoning Abilities of Large Language Models, 2025b. URL https://arxiv.org/abs/2502.16906.

AI USE STATEMENT

Generative AI tools were used for manuscript editing, literature-positioning assistance, refinement of the token-wise information argument, design, implementation, and interpretation of the public small-model controls and diagnostics, and record-based statistical verification. They also assisted with LaTeX preparation and formatting checks. The authors are responsible for verifying AI-assisted text, code, mathematical arguments, and experimental interpretations and for the final content of this work.

A Full-Scale Experimental Details and Results

This appendix groups the full-scale task results in Section A, the public reconstruction studies in Section B, and the information diagnostic in Section C.

A.1 Comparison Protocol

The full-scale experiments use proprietary continued-pretraining data and a shared AR-to-block-diffusion warmup within each native/PCD pair. Initialization, data stream, token budget, evaluation prompts, and inference are matched. LLaDA2-Mini uses 100B–400B stable tokens followed by matched +50B decay; Qwen-1.7B uses 50B stable tokens for its six-task recipe study. These comparisons evaluate the complete objective replacement, including changes to attention, corruption support, and supervision.

Retained settings and reporting limits. Unless varied in a sweep, PCD uses unit AR/MDM weights, per-region target normalization, causal prefix attention, and bidirectional suffix attention. Evaluation uses matched no-shift SGLang settings and an 8k context. The original records do not recover the main runs' prefix distribution, PCD/native mixing probability, training sequence length, global batch, optimizer or learning-rate schedule. Complete decoding settings (block size, steps, temperature, top-p, remasking, and output limits) and exact task-specific scoring configurations are also unavailable. Sweep settings apply only to their rows; the separate public configuration in Section B does not fill these gaps. The reported sweeps have no independently documented validation-selection rule.

A.2 LLaDA2-Mini Trajectories and Target Alignment

Table 5 gives all 12 task scores at every reported stable and matched-decay checkpoint. These are checkpoints along one paired trajectory, not independent training repetitions. Table 6 compares same-position and shifted suffix targets at 100B.

Table 5: Matched LLaDA2-Mini 12-benchmark results. Avg.12 averages all benchmarks; longer PCD-only decay is in Table 7. Corrected LCBench and AutoLogi settings are used.

Method	Budget	Avg.12	MATH	GSM8K	MBPP	HumanEval-FIM	LCBench
AR reference	–	52.88	63.88	82.94	66.40	74.54	40.70
Native dLLM stable	100B	47.16	53.52	77.56	58.80	73.57	29.37
Native dLLM stable	200B	47.34	54.14	77.63	59.40	73.96	26.22
Native dLLM stable	300B	46.50	56.84	76.72	54.20	71.35	25.62
Native dLLM stable	400B	47.26	56.38	75.82	59.40	70.67	27.54
Native dLLM decay	+50B	49.30	56.92	80.82	59.40	73.77	29.56
PCD stable	100B	48.46	54.54	81.05	57.60	72.89	28.96
PCD stable	200B	48.63	55.22	80.59	60.00	72.12	28.12
PCD stable	300B	49.12	56.16	80.74	60.40	72.41	29.30
PCD stable	400B	49.74	57.64	80.44	60.20	**74.64**	30.72
PCD decay	+50B	**50.94**	**59.52**	**83.47**	**63.20**	74.15	**31.41**

Method	Budget	NQ	TriviaQA	MMLU-Pro	OmniMath	BBH	KorBench	AutoLogi
AR reference	–	30.42	59.28	48.11	23.70	74.14	36.24	34.25
Native dLLM stable	100B	27.12	53.33	44.18	17.30	66.01	31.36	33.82
Native dLLM stable	200B	26.59	53.37	43.76	18.20	66.48	32.24	36.05
Native dLLM stable	300B	25.76	52.72	43.92	18.10	65.43	32.72	34.64
Native dLLM stable	400B	25.84	54.27	44.74	19.70	66.38	33.12	33.27
Native dLLM decay	+50B	27.67	55.80	43.94	18.50	68.96	**37.36**	**38.95**
PCD stable	100B	28.70	55.36	45.45	20.50	66.87	33.84	35.81
PCD stable	200B	27.98	56.55	45.06	19.20	69.09	34.80	34.78
PCD stable	300B	28.37	56.38	45.34	20.10	68.81	34.32	37.08
PCD stable	400B	29.17	57.46	46.08	20.90	68.40	35.84	35.37
PCD decay	+50B	**29.53**	**57.99**	**46.10**	**20.80**	**69.63**	37.28	38.21

Table 6: No-shift suffix-target diagnostic at 100B. Same-token MDM is stronger than shifted suffix labels, and PCD improves over the native no-shift MDM control under the same budget.

Method	Target	Train shift	Eval shift	Budget	HumanEval-FIM	MBPP	GSM8K	MATH	BBH	KorBench	Avg.6
Native MDM	Same token	×	×	100B	73.57	58.80	77.56	53.52	66.01	31.36	60.14
PCD	Same token	×	×	100B	72.89	57.60	81.05	54.54	66.87	33.84	61.13
Shifted suffix	Next token	✓	✓	100B	72.31	51.80	76.72	29.46	62.99	33.28	54.43
Shift mismatch	Next token	✓	×	100B	0.58	0.00	0.53	0.32	2.39	0.00	0.64

Longer PCD-only decay. Table 7 extends the 400B PCD checkpoint without corresponding native continuations. The highest reported Avg.12 is at +200B, but the change from +50B is uneven: LCBench and OmniMath increase while HumanEval-FIM, MBPP, NQ, and TriviaQA decrease. These rows do not extend the matched +50B comparison.

Table 7: PCD-only decay continuation from the 400B PCD stable checkpoint. Avg.12 averages all listed benchmarks.

Extra tokens	Avg.12	MATH	GSM8K	MBPP	HumanEval-FIM	LCBench
0B	49.74	57.64	80.44	60.20	**74.64**	30.72
+50B	50.94	59.52	83.47	63.20	74.15	31.41
+100B	51.21	59.14	83.62	61.20	74.35	34.66
+150B	51.05	**60.24**	**84.69**	**63.40**	73.86	31.39
+200B	**51.24**	59.26	84.46	60.60	71.93	**40.39**

Extra tokens	NQ	TriviaQA	MMLU-Pro	OmniMath	BBH	KorBench	AutoLogi
0B	29.17	57.46	46.08	20.90	68.40	35.84	35.37
+50B	**29.53**	57.99	46.10	20.80	69.63	**37.28**	38.21
+100B	29.45	**58.16**	**46.91**	21.00	**70.44**	36.96	38.62
+150B	28.84	56.38	46.23	21.40	69.95	37.12	**39.12**
+200B	27.78	56.41	46.62	**22.40**	69.88	36.48	38.69

A.3 Qwen Objective Sweeps

Tables 8 and 9 expand Figures 5 and 6. The intra-sample gain peaks at prefix range $[0, 0.75]$ and falls from $+7.04$ to $+4.86$ at $[0, 1]$; varying the range also changes the mean prefix length. At the AR-dominant endpoint $[1, 1]$, inter-sample gains are nonmonotonic: $p = 1$ is below $p = 0.25/0.50$ but above $p = 0.75$.

All tested AR weights from 0.25 to 1.00 exceed the native baseline, with the highest reported Avg.6 at 0.75. Without an otherwise matched zero-AR row, this sweep measures weight sensitivity rather than the necessity or task-level contribution of AR supervision.

Table 8: Qwen-1.7B ratio/probability sweep at 50B tokens; HE denotes standard HumanEval. Avg.6 averages all six listed benchmarks; ΔAvg.6 is measured against the native baseline in Table 3. Figure 5 summarizes these sweeps. Native is not a measured zero-prefix PCD endpoint: clamping retains a prefix with different attention and supervision.

Setting	p_{pcd}	Prefix range	Avg.6	ΔAvg.6	HE	GSM8K	MATH	BBH	KorBench	MBPP
Intra range sweep	1.00	[0.0, 0.25]	36.51	**+2.23**	32.93	67.70	31.14	42.22	20.64	24.40
Intra range sweep	1.00	[0.0, 0.50]	39.89	**+5.61**	34.15	69.07	33.48	43.54	20.88	38.20
Intra range sweep	1.00	[0.0, 0.75]	**41.32**	**+7.04**	**35.37**	**69.52**	**35.68**	44.46	22.88	**40.00**
Inter prob. sweep	0.25	[1.0, 1.0]	38.85	**+4.57**	33.54	64.22	33.30	44.33	21.92	35.80
Inter prob. sweep	0.50	[1.0, 1.0]	38.98	**+4.70**	29.88	62.93	33.68	45.84	**23.12**	38.40
Inter prob. sweep	0.75	[1.0, 1.0]	36.34	**+2.06**	26.83	53.37	34.12	44.66	21.84	37.20
Inter prob. endpoint	1.00	[1.0, 1.0]	37.88	**+3.60**	26.22	62.62	33.88	**47.45**	21.68	35.40

Table 9: Qwen-1.7B prefix-AR loss weight sweep at 50B continued-pretraining. The PCD rows vary only the prefix-AR component weight within the mixed recipe. Avg.6 averages the six tasks; Figure 6 plots gains over the native reference.

Setting	Prefix-AR weight	Avg.6	HumanEval	GSM8K	MATH	BBH	KorBench	MBPP
Native baseline	–	34.28	25.00	59.14	26.84	39.72	20.96	34.00
Mixed PCD	0.25	39.20	35.37	65.50	32.18	42.77	21.76	37.60
Mixed PCD	0.50	40.02	**37.80**	65.35	33.36	43.09	23.12	37.40
Mixed PCD	0.75	**40.35**	31.71	**67.78**	**35.38**	44.45	**24.56**	**38.20**
Mixed PCD	1.00	38.94	28.05	66.94	33.36	**45.87**	22.00	37.40

Warmup diagnostic. Table 10 compares direct AR initialization with the studied warmup pipeline. The missing direct-AR-to-native control prevents identifying a PCD-specific warmup dependence.

Table 10: Qwen-1.7B curriculum diagnostic at 50B continued-pretraining. The direct-AR row starts stable training from an AR checkpoint and is not a controlled baseline in the main comparison. Warmup improves PCD under this recipe; the absent direct AR-to-native row prevents attributing this dependence specifically to PCD. The controlled native/PCD comparison uses the same warmup initialization.

Initialization and stable recipe	Avg.6	HumanEval	GSM8K	MATH	BBH	KorBench	MBPP
Direct AR init → stable PCD	10.06	2.44	9.17	11.28	23.20	7.44	6.80
AR-to-BD warmup init → native stable	34.28	25.00	59.14	26.84	39.72	20.96	34.00
AR-to-BD warmup init → PCD stable	**41.32**	**35.37**	**69.52**	**35.68**	**44.46**	**22.88**	**40.00**

A.4 Chat SFT and AR Decoding

Table 11 reports every SFT epoch with the same answer-extraction rule for both runs. The exact starting checkpoints and whether they precede or follow decay are not recorded. At the shared final epoch, PCD is higher on GSM8K, MATH, BBH, and MBPP, but lower on KorBench; the epoch-2 average is also slightly lower. The separate maxima, 67.39 at epoch 5 for PCD and 66.53 at epoch 4 for native, do not constitute a comparison at a common stopping point.

Table 12 expands the AR-interface diagnostic. It uses standard HumanEval and a consistent 120s MBPP execution timeout. The reference is native stable training, not the original AR parent.

Table 11: Chat-SFT transfer on LLaDA2-Mini. Avg. averages GSM8K, MATH, BBH, KorBench, and MBPP, excluding HumanEval-FIM because chat SFT changes the infilling interface. BBH uses the same answer-extraction rule for both SFT runs. Bold marks the best score in each column, and shaded rows mark PCD-derived checkpoints.

Model	Epoch	Avg.	GSM8K	MATH	BBH	KorBench	MBPP
Native-SFT baseline							
LLaDA2-Mini native SFT	1	64.06	88.63	73.26	59.30	39.12	60.00
LLaDA2-Mini native SFT	2	65.48	88.40	73.26	61.42	41.52	62.80
LLaDA2-Mini native SFT	3	65.12	87.49	73.64	62.09	40.96	61.40
LLaDA2-Mini native SFT	4	66.53	89.69	74.28	62.30	42.40	64.00
LLaDA2-Mini native SFT	5	65.84	88.63	74.18	62.67	42.32	61.40
PCD-derived SFT							
LLaDA2-Mini PCD SFT	1	64.72	87.79	75.42	59.46	36.72	**64.20**
LLaDA2-Mini PCD SFT	2	65.40	88.78	74.88	62.37	39.76	61.20
LLaDA2-Mini PCD SFT	3	67.13	89.39	75.00	65.29	**42.56**	63.40
LLaDA2-Mini PCD SFT	4	67.07	**90.07**	75.58	**67.16**	39.52	63.00
LLaDA2-Mini PCD SFT	5	**67.39**	89.76	**76.00**	66.69	40.32	**64.20**

Table 12: Detailed AR-decoding diagnostic for completed LLaDA2-Mini stable checkpoints. Avg.6 averages the six listed benchmarks. MBPP is judged with a consistent 120s execution timeout for the paired native and PCD rows.

Model	Budget	Avg.6	GSM8K	MATH	BBH	KorBench	MBPP	HumanEval
Native stable AR trend								
LLaDA2-Mini native stable	100B	60.31	77.79	53.74	66.01	31.36	59.20	73.77
LLaDA2-Mini native stable	200B	60.72	77.79	54.08	66.47	32.24	59.60	74.15
LLaDA2-Mini native stable	300B	59.83	77.33	56.90	65.46	32.72	55.20	71.35
LLaDA2-Mini native stable	400B	60.78	75.97	56.62	66.39	33.12	59.40	73.17
PCD stable AR trend								
LLaDA2-Mini PCD stable	100B	62.10	**80.97**	54.58	66.92	34.08	58.00	**78.05**
LLaDA2-Mini PCD stable	200B	62.36	80.74	55.18	**69.13**	34.72	60.00	74.39
LLaDA2-Mini PCD stable	300B	62.33	80.74	56.42	68.75	34.32	**60.60**	73.17
LLaDA2-Mini PCD stable	400B	**62.61**	80.36	**57.54**	68.32	**35.84**	59.20	74.39

A.5 Prompt-Length Diagnostic

Table 13 gives the rendered prompt statistics behind Figure 7, using the evaluation templates and a fixed Qwen tokenizer. HumanEval-FIM uses a deterministic length proxy from HumanEval solutions, not the model-scoring inputs. Expected corrupted-prefix counts describe median-length prompts at masking rate $r = 0.3$.

Table 13: Rendered prompt statistics for five benchmarks from the main suite. Lengths are token counts after applying the evaluation templates; expected corrupted-prefix counts refer to median-length prompts at $r = 0.3$. HumanEval-FIM uses a deterministic infilling-length proxy built from HumanEval solutions for this diagnostic, not for model scoring.

Benchmark	Examples	Median prefix	P90 prefix	Median target	Prefix ratio	Exp. corrupted prefix at $r = 0.3$
HumanEval-FIM	1033	252	431	7	97.3%	75.6
MBPP	500	671	779	46	93.9%	201.3
GSM8K	1319	1458	1493	115	92.7%	437.4
MATH	5000	720	804	153	82.7%	216.1
BBH	6511	856	1273	2	99.8%	256.8
Aggregate	14363	754	1448	12	96.8%	226.2

B Public Reconstruction Controls

These studies use Qwen3-1.7B-Base and FineWeb-Edu at a 50M-token stable budget with block-causal suffix attention. They test reconstruction behavior separately from the 50B-token Qwen task study. Table 14 distinguishes the four-recipe screen from the single-factor visibility intervention.

B.1 Recipe Comparison

The model and data revisions are `ea980cb0a6c2` and `87f09149ef47`. All recipes share a 20,054,016-token warmup, then train for 382 updates (50,069,504 tokens) on two B200 GPUs, with sequence length 2,048 and 131,072 tokens per update. Each branch consumes 24,448 of the 29,296 packed blocks in the 60M-token training snapshot. Data-order protocol, optimizer, and budget are fixed across objectives within each of seeds 42–44.

Evaluation uses a separate 2M-token validation snapshot, prefix ratio $U[0, 0.75]$, block size 128, and three evaluator seeds with 32 sequences each. Checkpoints share examples, boundaries, suffix masks, and nested prefix-corruption draws within each evaluator seed.

Table 14: Training factors in the public studies. The first four rows correspond to Table 15; the final pair isolates prefix visibility (Table 4) using a fresh holdout and FP32 reduction. All suffix targets are at the same position.

Public training arm	Prefix inputs	Attention	MDM targets	AR targets
Native MDM	May be masked	Fixed block grid	Whole sequence	None
AR + native MDM	Separate clean AR view	Native grid + causal AR view	Whole sequence	Whole sequence
Full PCD	Clean	Hybrid; suffix anchored at k	Suffix only	Prefix
Clean prefix, no AR	Clean	Hybrid; suffix anchored at k	Suffix only	None
Matched clean	Clean	Same hybrid mask	Same suffix targets	None
Matched corrupted	May be masked	Same hybrid mask	Same suffix targets	None

Table 15: Public Qwen3-1.7B recipe controls at 50M stable tokens. Values average three evaluator seeds per training seed; $\pm$ is SD over three training seeds. Benchmark-continuation results appear in Table 17.

Stable-stage objective	AR supervision	Suffix loss $\downarrow$	Δ vs. native
Native MDM	None	4.6476 ± 0.0017	–
AR + native MDM (dual view)	Full-sequence	4.6828 ± 0.0005	$+0.0352$
Full PCD recipe	Prefix	4.2636 ± 0.0018	-0.3840
Clean prefix, no AR	None	$\mathbf{4.2134 \pm 0.0034}$	$\mathbf{-0.4342}$

Table 15 reports the clean-prefix endpoint. Full PCD reduces native-MDM loss by 0.3840 nats, while the separate full-sequence AR view is 0.0352 worse. Clean-prefix/no-AR has the lowest loss. Comparisons with native MDM change multiple factors; Section B.2 isolates training-prefix visibility.

B.2 Matched Training-Prefix Visibility Control

For each of seeds 52–54, two new branches start from the same warmup parent. Both use causal prefix attention, suffix blocks anchored at the sampled boundary, and masked-suffix-only supervision without AR loss. Suffix queries read the prefix, their own block, and preceding blocks. Data order, boundaries, suffix masks, target counts, optimization, and budget are paired; only training-prefix corruption differs. The corrupted-prefix arm is therefore distinct from native MDM.

Table 16 specifies the configuration. For $T = 2,048$, the boundary is $k = \max(1, \min(T - 1, \lfloor T\rho \rfloor))$. The corrupted arm independently masks prefix tokens at the same sampled rate t as the suffix, without forcing a prefix mask. Separate random streams for boundaries, suffix masks, and prefix corruption preserve pairing. Both arms are retrained with FP32 cross-entropy and reduction because the earlier clean-prefix checkpoint used a BF16 scalar reduction. Warmup and stable training share the 60M-token snapshot; no packed block repeats within a stable branch.

Table 16: Training and evaluation configuration for the paired prefix-visibility intervention in Table 4.

Setting	Completed visibility-control runs
Initialization	Shared 20,054,016-token warmup; weights loaded, optimizer/scheduler/RNG reset
Stable budget	382 updates; 50,069,504 tokens per branch
Sequence / batch	2,048 tokens; 2 GPUs $\times$ 32 sequences; no accumulation; 131,072 tokens/update
Prefix / suffix	$\rho \sim U[0, 0.75]$; suffix blocks of 128 anchored at the sampled boundary
Corruption	$t \sim U[0.001, 1]$; paired suffix masks with at least one target; prefix clean or independently masked at the same t
Supervision / precision	Suffix-only, same-position targets; no AR; per-example FP32 loss reduction, BF16 model
Optimizer	AdamW; LR 2×10^{-5}; $\beta = (0.9, 0.95)$; weight decay 0.1; gradient norm clip 1.0
LR schedule	20 warmup updates, then cosine decay to 0.1 of peak LR
Evaluation	976 holdout blocks; seeds 2028–2030; batch size 16; paired clean-prefix hybrid interface
Uncertainty	First average paired block/draw differences per training seed; t interval across seeds 52–54 ($n = 3$, df=2)

Fresh holdout. The earlier validation snapshot informed development, so a new 2M-token holdout is frozen before evaluating these branches. Replaying the pinned FineWeb-Edu stream reproduces the historical train/validation token-file hashes over 63,226 documents. Later validation-assigned documents are collected after excluding historical document IDs and full-text SHA256 hashes and deduplicating the new set likewise. The 2,002 retained documents yield 976 nonoverlapping 2,048-token blocks; the last 1,152 tokens are discarded. This excludes exact overlap, not semantic near-duplicates. All six checkpoints share these blocks, boundaries, and suffix masks under clean-prefix evaluation. The two eight-update operational smokes are excluded.

Endpoint and uncertainty. Paired block/draw differences are averaged within each training seed, then across seeds. Table 4 reports the resulting -0.0692-nat mean and 95% t interval $[-0.0728, -0.0655]$ ($n = 3$, df=2), conditional on the common warmup. The prospectively fixed rule requires every seed difference to be negative, the mean to be at most -0.02 nats, and the interval upper bound to be below zero; all three conditions hold. The experiment tests reconstruction at fixed attention and supervision. Total compute, including smoke runs and evaluation, is 5.19 B200 GPU-hours, below the prespecified 16 GPU-hour ceiling.

B.3 Reconstruction on Benchmark Prompts

Using frozen templates and dataset revisions, a fixed hash rank (seed 3407) selects 128 valid prompt/gold-continuation pairs each from BBH, GSM8K, MATH-lighteval, and sanitized MBPP before checkpoint evaluation. There is no result-dependent filtering. All 12 checkpoints share the 512 tokenized examples and three evaluator draws, with suffix masking rate $U[0.001, 1]$, block size 128, and at least one target per example. The clean-prefix interface masks only suffix tokens with hybrid attention; the native diagnostic masks the whole sequence with block-causal attention.

PCD improves clean-prefix reconstruction in all four domains but worsens the native-corruption endpoint (Table 17). These scores reconstruct partially observed gold text, rather than measure generated-answer accuracy.

Table 17: Benchmark masked-token conditional cross-entropy relative to native MDM (Δ = objective minus native; lower is better). Each domain has 128 gold continuations. Entries average three paired evaluator seeds within each of three training seeds; $\pm$ is training-seed standard deviation. For PCD's macro clean Δ, a paired example bootstrap gives 95% CI $[-0.5101, -0.4688]$. These values are neither full-continuation likelihoods nor standard perplexities.

Objective	BBH clean Δ	GSM8K clean Δ	MATH clean Δ	MBPP clean Δ	Macro clean Δ	Macro native Δ
PCD	-0.4687	-0.3904	-0.4305	-0.6678	$\mathbf{-0.4894 \pm 0.0296}$	$+0.8214 \pm 0.0033$
Clean prefix, no AR	-0.5267	-0.4300	-0.4837	-0.7533	$\mathbf{-0.5484 \pm 0.0349}$	$+0.7838 \pm 0.0059$
AR + native MDM (dual view)	-0.0307	$+0.0010$	$+0.0252$	$+0.0278$	$+0.0058 \pm 0.0118$	$+0.0145 \pm 0.0005$

No-AR contrast. Table 18 is a post hoc comparison of the stored no-AR and full-PCD losses, not the prospectively specified primary endpoint. These recipes share warmup, data, budget, prefix distribution, suffix attention, and optimizer settings; no AR disables prefix supervision. For each example, the three paired draw differences are averaged, followed by within-domain means and an equal-weight domain average. All 12 domain/seed means favor no AR. The macro difference is -0.0591 nats, with seed SD 0.0058 and 95% t interval $[-0.0736, -0.0446]$ ($n = 3$, df=2). This interval conditions on the shared warmup, fixed examples/draws, and stored BF16 losses; it excludes warmup variation and numerical-precision error. No new FP32 evaluation or generation test is performed. The result concerns reconstruction, not the AR term's downstream task value.

Table 18: Post hoc paired no-AR minus full-PCD masked-token cross-entropy (nats; negative favors no AR) from the same benchmark evaluation as Table 17. Each domain has 128 examples with three evaluator draws. Means are computed from stored per-example losses before rounding, rather than by subtracting the rounded entries in that table.

Domain	Seed 42	Seed 43	Seed 44	Mean $\pm$ seed SD
BBH	-0.0882	-0.0453	-0.0405	-0.0580 ± 0.0262
GSM8K	-0.0358	-0.0403	-0.0426	-0.0396 ± 0.0035
MATH	-0.0496	-0.0520	-0.0579	-0.0532 ± 0.0043
MBPP	-0.0875	-0.0770	-0.0921	-0.0855 ± 0.0078
Macro	-0.0653	-0.0537	-0.0583	-0.0591 ± 0.0058

B.4 Context Corruption and Generation Checks

Prefix corruption. The four-recipe evaluator holds examples, boundaries, suffix masks, and attention fixed while masking prefix tokens at $r \in \{0, 0.1, 0.3, 0.5, 0.7, 1\}$. PCD-minus-native loss rises from -0.3840 ± 0.0013 at $r = 0$ to $+0.7573 \pm 0.1661$ at $r = 1$ (Figure 10; Table 19). Each training seed crosses near 50% corruption. This measures reliance on visible context, not robustness to natural prompt errors or distribution shift.

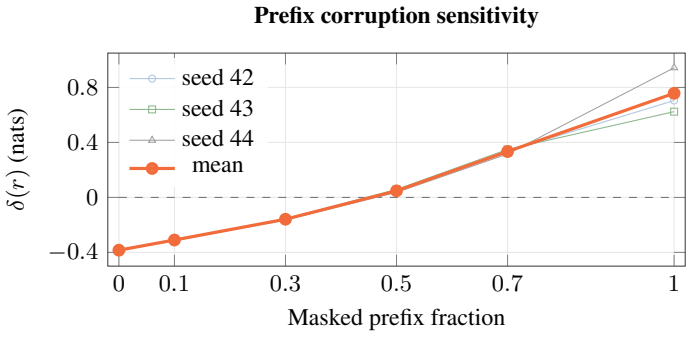


Figure 10: Qwen3-1.7B, 50M stable tokens (seeds 42–44): $\delta(r) = \mathcal{L}_{\text{PCD}}(r) - \mathcal{L}_{\text{native}}(r)$; negative favors PCD. Thin lines average three paired evaluator draws; the thick line is their mean. Only prefix corruption varies.

Table 19: Training-seed replication of the prompt-corruption endpoint. Each entry averages three paired evaluator seeds; uncertainty is across training seeds.

Training seed	$\delta(0)$	$\delta(1)$	$\delta(1) - \delta(0)$
42	-0.3840	$+0.7048$	$+1.0888$
43	-0.3828	$+0.6238$	$+1.0066$
44	-0.3853	$+0.9432$	$+1.3285$
Mean $\pm$ SD	$\mathbf{-0.3840 \pm 0.0013}$	$\mathbf{+0.7573 \pm 0.1661}$	$\mathbf{+1.1413 \pm 0.1673}$

Generation readiness. The seed-52 matched clean/corrupted checkpoints are tested on 16 GSM8K and 16 BBH development prompts each, reused from reconstruction evaluation. Greedy confidence-ranked decoding uses clean-prefix hybrid attention, 128-token blocks, and GSM8K/BBH output limits of 512/128 tokens. Both models score zero on both tasks at 32 steps per block. At 128 steps, clean-prefix training gives 3/16 GSM8K and 1/16 BBH, versus 3/16 and 0/16 for corrupted-prefix training; the latter has only 6/16 nonempty BBH outputs. The prospective readiness rule requires each model to achieve at least one correct answer and 75% nonempty outputs per task. The pair fails, so the reserved generation evaluation is not run. These development checks establish neither generation nor efficiency gains and do not invalidate the separate large-budget comparison.

C TOKEN-WISE INFORMATION DIAGNOSTIC

For an oracle supplied with the corruption metadata R, the information diagnostic is

$$\mathcal{R}^*_{\text{corrupt}} - \mathcal{R}^*_{\text{clean}} = \mathbb{E}_{r\sim R}\left[\sum_i w_i(r) I(X_i; C \mid \tilde{C}, Z_i, R = r)\right] \geq 0. \tag{6}$$

This standard identity compares normalized token-wise oracle risks at fixed suffix inputs, target support, attention, and weights. It is neither a diffusion likelihood bound nor the multi-factor native/PCD comparison.

Setup and scope of the oracle. Let R contain the sampled prefix boundary k, suffix target mask, attention pattern, and any shared corruption-rate variable. For a realization r, define $S(r) = \{i > k : m_i = 1\}$ and $w_i(r) = \mathbb{I}[i \in S(r)]/\max(|S(r)|, 1)$. Let $C = X_{1:k}$, and let Z_i contain only the corrupted suffix inputs accessible when predicting X_i. For block-causal attention, these are the current and preceding suffix blocks, not later blocks; Z_i is identical in both conditions. Corrupt the prefix through a kernel satisfying

$$\tilde{C} \perp (X_i, Z_i) \mid (C, R). \tag{7}$$

Independent prefix and suffix corruption draws conditional on their shared rate satisfy this assumption. Supplying R to the oracle makes this conditioning explicit; a trained network need not observe all of this metadata. The oracle also receives the allowed raw inputs, not capacity-limited hidden representations.

For $a \in \{\text{clean}, \text{corrupt}\}$, set $C_{\text{clean}} = C$ and $C_{\text{corrupt}} = \tilde{C}$, and define

$$\mathcal{R}^*_a = \inf_{(q_i)} \mathbb{E}\left[\sum_i w_i(R)\{-\log q_i(X_i \mid C_a, Z_i, R)\}\right]. \tag{8}$$

The unrestricted per-position infimum is attained by the true token posterior: conditional cross-entropy equals conditional entropy plus a nonnegative KL divergence. Since the weights depend only on r, this gives

$$\mathcal{R}^*_a = \mathbb{E}_{r\sim R}\left[\sum_i w_i(r) H(X_i \mid C_a, Z_i, R = r)\right]. \tag{9}$$

By Equation 7, each target's entropy difference is

$$\begin{aligned} &H(X_i \mid \tilde{C}, Z_i, R = r) - H(X_i \mid C, Z_i, R = r) \\ &\quad = H(X_i \mid \tilde{C}, Z_i, R = r) - H(X_i \mid C, \tilde{C}, Z_i, R = r) \\ &\quad = I(X_i; C \mid \tilde{C}, Z_i, R = r) \geq 0. \end{aligned} \tag{10}$$

Weighting and averaging yields Equation 6; examples with no targets contribute zero.

Why joint suffix entropy is insufficient. Let Y_1, Y_2 be independent fair bits, let $C = Y_1 \oplus Y_2$, and mask both suffix bits completely while erasing the corrupted prefix. Observing C reduces joint suffix entropy from $2\log 2$ to $\log 2$. However, each individual bit remains uniformly distributed given C, so the normalized token-wise denoising risk remains $\log 2$ in both conditions. A joint-suffix information gain therefore need not yield a positive gain in the token-wise objective.

Interpretation. The identity concerns unrestricted token-wise oracle risks, not finite-model losses or generation accuracy. It does not establish AR utility, an optimal mixture, or a decomposition of objectives that change attention and supervision. Table 4 and Figure 10 provide separate empirical tests of training-prefix visibility and evaluation-time context dependence; neither follows from the identity.