From Next-Token to Next-Block: A Principled Adaptation Path for Diffusion LLMs

Anonymous ACL submission

Abstract

Diffusion Language Models (DLMs) enable fast generation, yet training large DLMs from scratch is costly. As a practical shortcut, adapting off-the-shelf Auto-Regressive (AR) model weights into a DLM could quickly equip the DLM with strong long-context generation capabilies. Prior "adaptation" attempts either modify logits or randomly grow attention masks to Full-Sequence diffusion, or simply transplant AR weights into a Block-Diffusion recipe, leaving two key questions unaddressed: where is the final destination of adaptation, and how to adapt better? For manifold benefits, we reframe the whole AR-to-DLM adaptation under the Block-Diffusion paradigm, transitioning from block size 1 to the final Block-Diffusion state. Concretely, the principled pathway of adaptation is designed as follows: we keep a context-causal path where causal attention is kept in the prefix, an efficient parallel adaptation procedure where an AR guidance is maintained, and gradual increment of the generation block size for a smoother transition. Built on these components, the adaptation is proved competitive on various models at different scales. With better adaptation, we propose NBDIFF-7B that could inherit the long-context modeling and reasoning capabilities, and achieve state-of-the-art performance among the 7B-class DLMs.

1 Introduction

Large language models (LLMs) are rapidly permeating real-world applications because of their strong generative capability. However, the dominance of AutoRegressive (AR) LLMs is built on a fundamental trade-off: powerful left-to-right causal generation at the cost of strictly sequential, token-by-token decoding. This trade-off creates an inference bottleneck that limits the decoding speed of AR LLMs. In contrast, Diffusion Language Models (DLMs) offer a promising alternative by enabling parallel generation, reducing sequential dependencies and yielding substantially higher throughput and lower wall-clock latency in practice.

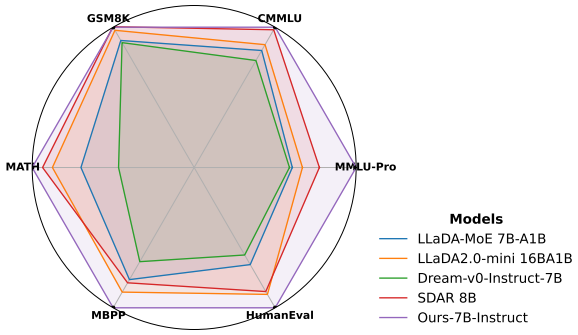

Figure 1: **Comparison of our model with baselines.** After adaptation an open-sourced AR LLM, our model has good long-sequence and reasoning capabilities and shows outstanding performance in various benchmarks.

Current diffusion approaches for language have largely converged on masked diffusion, with two dominant paradigms. Full-Sequence Diffusion (Nie et al., 2025; Ye et al., 2025) starts from a fully masked sequence and denoises to a complete output where all tokens attend bidirectionally. Block-Diffusion (Arriola et al., 2025; Cheng et al., 2025) decodes one block at a time: tokens are bidirectional within the active block, while blocks themselves follow left-to-right causal order, yielding a semi-autoregressive workflow. Training masked diffusion is intrinsically harder than AR pretraining because, unlike AR–where every token contributes a next-token-prediction loss–only masked tokens provide supervision, which slows optimization. Yet masked diffusion and AR models are strikingly similar in input–output format and transformer architecture. This naturally motivates the question: can we leverage powerful off-the-shelf AR checkpoints and rapidly adapt them into diffusion models, preserving their knowledge while avoiding the cost of training a DLM from scratch?

Existing adaptation methods are lacking. Early attempts used logit shifts and random attention

mask growth to Full-Sequence Diffusion (Gong et al., 2025a; Ye et al., 2025). More recent block-wise adaptations simply 'transplant' the AR model into a Block-Diffusion training setup 'as is.' (Cheng et al., 2025)–they do not investigate the core mismatch between AR and Block-Diffusion. These methods leaves a clear gap: where should be our destination of adaptation, and how to adapt an AR model to Diffusion for better performance?

Our approach is grounded in a key insight: for longer-sequence training, efficient inference, and adaptation benefits, Block-Diffusion should be either the pathway and the destination of adaptation. AR generation could be viewed as a special case of Block-Diffusion with a $blocksize$ of 1, reframing adaptation not as a crude switch, but as a smooth transition across a spectrum. Under this unified view, we look for a principled and smooth transition path from AR to Block-Diffusion. Our design consists of a context-causal attention mask that preserves AR inductive bias in committed context, parallel training with an auxiliary AR guidance that regularizes the path of adaptation, and gradual growth of block size. The design provides an efficient adaptation strategy from AR to DLM that progressively unlocks bidirectional attention and parallel decoding within the generating block while maintaining strict train–inference alignment.

Our contributions are as follows:

1. After investigation, we propose to view the whole adaptation process under Block-Diffusion for its natural training, inference, and adaptation benefits. The transition from AR to DLM is then simply blocksize growth from causal ($blocksize = 1$) to target size.

2. We propose the Context-Causal mechanism tailored for this adaptation, which preserves AR knowledge in the context while enabling efficient bidirectional intra-block generation. We develop an efficient parallel training strategy that aligns with inference and incorporates an auxiliary AR loss. We also develop a gradual block growth approach that alleviates the gap between AR and Block-Diffusion models. These measures markedly improve adaptation performance.

3. We demonstrate the effectiveness of our approach with various models. We also propose NBDIFF-7B, which, after efficient adaptation from its strong AR counterparts, could model **long contexts** (up to 32K sequence lengths) and perform **reasoning**. Both NBDIFF-7B-BASE and NBDIFF-7B-INSTRUCT outperform strong baselines like LLaDA (Nie et al., 2025; Zhu et al., 2025; inclusionAI, 2025), Dream (Ye et al., 2025), and SDAR (Cheng et al., 2025) on general, math, and code benchmarks (Figure 1), achieving state-of-the-art performance.

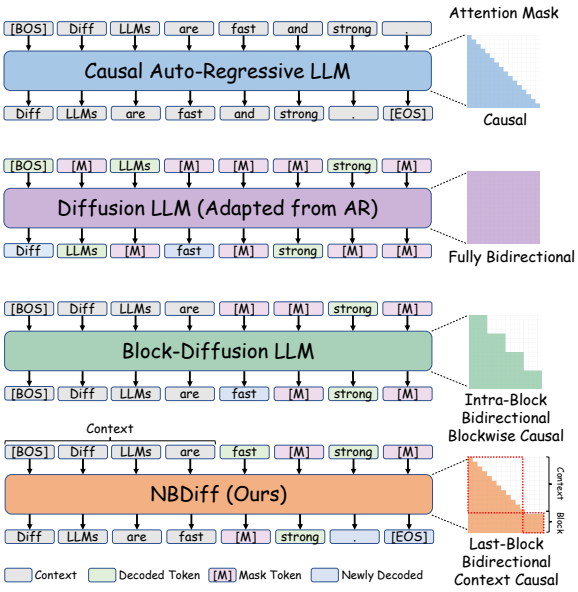

Figure 2: **The diffusion paradigm of our NBDIFF model.** We compare popular language generation paradigms. Diffusion LLMs adapted from AR adopt logit shift and attention mask growth; Block-Diffusion uses block-wise autoregressive and maintains an intra-block bidirectional mask; Our model adopts Block-Diffusion where bidirectional attention is used intra-block, but features a causal context.

2 Rethinking DLM Adaptation from AR: to Where, and How?

2.1 Revisiting Previous Adaptation

Prior adaptation work (Gong et al., 2025a) mainly focuses on adapting a Full-Sequence Diffusion model from an AR model. The authors observe the difference in attention mechanism, and proposes random "annealing", or random growth, of the attention mask from a lower-triangular causal attention mask to a full, bidirectional attention mask.

While the work (Gong et al., 2025a) is trying to bridge the AR and Diffusion generation paradigms, we hold different opinions on random growth of attention masks: its transition is not "natural." In practice, training sees unknown future corpora; spo-

radically granting early tokens access to a random subset of future tokens yields incomplete and potentially misleading context, thus limiting adaptation potentials.

Hence, we try to answer two major questions regarding this transition, *i.e.* to **where**, and **how**. Firstly, what should be the destination of this transition? Secondly, is there a smoother and better way to transition from an AR model to a DLM model?

2.2 Block-Diffusion and its Advantages

Unlike previous adaptation methods (Gong et al., 2025a) that focuses merely on Full-Sequence Diffusion models, recent diffusion LLMs (Cheng et al., 2025; Wu et al., 2025a) increasingly adopt Block-Diffusion (Arriola et al., 2025), which is both more efficient and performant than Full-Sequence Diffusion and conceptually sits between Full-Sequence Diffusion and AR: generation proceeds left-to-right across blocks while remaining bidirectional within a block (Figure 2). Specifically, tokens within the same block attend to each other bidirectionally, whereas attention across blocks is strictly causal. Decoding is performed block by block, with all tokens in a block capable of generating in parallel.

We analyze the advantages from several aspects: either from training, from inference, and also from the perspective of adaptation from AR models.

Training advantages: stabler training at longer sequences. AR models excel at long-sequence reasoning, yet most diffusion LLMs still operate at modest context lengths (e.g., LLaDA (Nie et al., 2025)=1K, Dream (Ye et al., 2025)=512), raising the question of how sequence length impacts DLM training. We therefore pretrain Full-Sequence Masked Diffusion and Block-Diffusion under identical corpora at 1K/2K/4K/8K sequence lengths and compare their training losses (Figure 3). For fair comparison, we adjust batchsize accordingly with sequence length to ensure that each training iteration consumes the same number of tokens. As context grows, the full-sequence model exhibits increasingly large loss oscillations, whereas Block-Diffusion remains consistently stable. Notably, at long lengths the block-diffusion loss is also lower, suggesting that longer contexts provide tangible generation benefits when the training dynamics are well-conditioned.

We hypothesize the instability of masked Full-Sequence Diffusion stems from the combinatorial explosion of the masking space: as sequence length

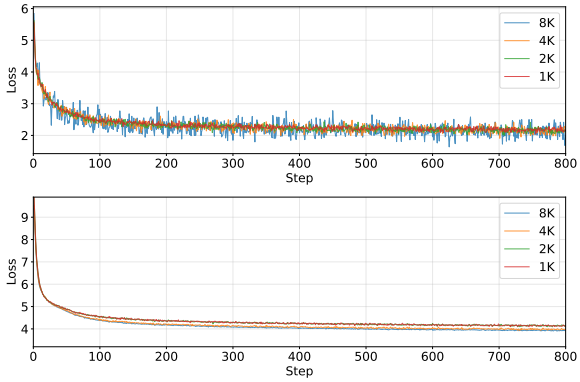

Figure 3: **Loss curves at different sequence lengths (Upper: Full-Sequence; Lower: Block-Diffusion).** As sequence length increases, Full-Sequence Diffusion suffers loss fluctuation while Block-Diffusion loss remains stable. Block-Diffusion is a better choice for long-sequence generation.

increases, (i) *the number of masked tokens per step can vary drastically*, and (ii) *the effective decoding/denoising orders proliferate, complicating optimization* (Kim et al., 2025). In contrast, Block-Diffusion constrains the space by fixing a constant block size, which regularizes the denoising schedule, preserves left-to-right structure, and stabilizes gradients, thereby enabling long-sequence advantages to manifest in masked DLMs.

Inference advantages: KV-Cache reuse. Different from Full-Sequence Diffusion where the whole sequence has to pass through the model together for each inference step, Block-Diffusion keeps previous tokens fixed while only performs decoding in the last block of the generated tokens. Thus it could re-use the KV-Cache from previous block generations and only the last block to be generated needs to be passed into the model, reducing significant inference costs. Besides, though Block-Diffusion has designated the causal (left-to-right) block generation sequence, the use of bidirectional attention within the last generating block still enables parallel token-decoding. In practice, we use the $blocksize$ of 32 tokens (larger than previous same-scale DLMs (Cheng et al., 2025)) to tap the speed potential of the proposed model to the full.

Adaptation advantages: analogous paradigms and easier adaptation from AR. Apart from the performance advantages, Block-Diffusion helps easy adaptation from AR. Instead of forcing a global jump from causal to full-sequence bidirectional attention, we treat Block-Diffusion as the

destination. By preserving left-to-right semantics at the block level and relaxing bidirectionality inside the active block, we try to partially align with the AR inductive bias for better adaptation, which greatly reduces the difficulty for alignment. In addition, the blockwise semi-AR manner could also enable parallel training, improving data utility and model convergence.

In the next section, we will stick to the paradigm of Block-Diffusion and seek a way for fast transition from AR model.

3 Designing Transition Paradigms

3.1 How Should the Unmasked Context Attend?

Comparing the attention mechanisms of Block-Diffusion and AR (which could be roughly viewed as Block-Diffusion of $blocksize = 1$, as introduced in Sec. 1), the key difference that requires our adaptation efforts lies in the bidirectional attention within the last active block; namely, we have to grow the attention mask at the end of the generating sequence from $blocksize = 1$ to target block size. However, apart from the attention within the active block region, different transition solutions arises from the decoded context: how tokens in the unmasked context ($x_{<s_K}$) should attend to each other? Here we analyze two possible attention pathways of Block-Diffusion as follows:

- **Block-Causal (widely used in Block-Diffusion (Arriola et al., 2025) / D2F (Wang et al., 2025c) / SDAR (Cheng et al., 2025)).** Tokens have *bidirectional* attention within every block (both past/committed blocks and the active block), and *causal* flow across blocks (each token can see all tokens in earlier blocks). This maximizes intra-block interaction everywhere, not only in the active block.

- **Context-Causal (our preferred setting).** The context (prompt + already generated/-committed blocks) remains *strictly causal*: each token only attends to itself and predecessors. Only the last (active) block is given *bidirectional* attention to support diffusion-style refinement; future blocks are hidden.

To examine the two schemes, we adapt two Block-Diffusion models from an AR model (based on Block-Causal and Context-Causal, respectively) by training 2000 iterations, and examine their performance on popular math and coding benchmarks. The results are shown in Table 1.

Table 1: **Comparison of Block-Causal and Context-Causal attention schemes.** Context-Causal gains a clear advantage in adaptation from AR.

Scheme	GSM8K	MATH	HumanEval	MBPP	Avg
Block-Causal	60.1	1.6	24.4	39.4	31.4
Context-Causal	**68.8**	**36.8**	**41.5**	**47.4**	**48.6**

Empirical takeaway and intuition. In these preliminary adaptation experiments, **Context-Causal** consistently outperforms Block-Causal by large margins: the accuracy is significantly higher when the context keeps strict causality and only the active block is bidirectional.

We attribute this to: (i) *Inductive-bias alignment* with AR pretraining, which reduces the gap between AR and Block-Diffusion by preserving causal self-attention in the context; and (ii) *Generation-paradigm consistency*: although the active block is refined bidirectionally (no fixed order inside the block), the overall decoding remains left-to-right *across* blocks. Keeping the context causal does not reduce the visibility required for the block being generated and avoids introducing spurious, partially bidirectional signals into earlier (already "finalized") content.

3.2 Training Parallelism

The naive block-diffusion recipe is data-inefficient: random cropping wastes the remaining tokens of each sequence, and only a small subset of masked tokens inside the last block contributes to the loss. Unlike AR pretraining—where every token can supervise next-token prediction—switching to next-block prediction sharply reduces token utilization. We therefore restructure training so that all blocks provide learning signal in a single forward pass.

To reach this goal, the clean (unmasked) sequence is concatenated after the noised sequence and enforce a structured attention mask that lets the clean side provide context to the noised side. The mask (shown in Figure 4) has four quadrants: (upper-left, $\mathbf{M}_{\mathrm{BD}}$) block-diagonal self-attention within the noised view (bidirectional inside each noised block); (upper-right, $\mathbf{M}_{\mathrm{OBC}}$) attention from noised tokens to earlier clean blocks, so denoising conditions on stable context; (lower-left) zeros, preventing the clean sequence from reading

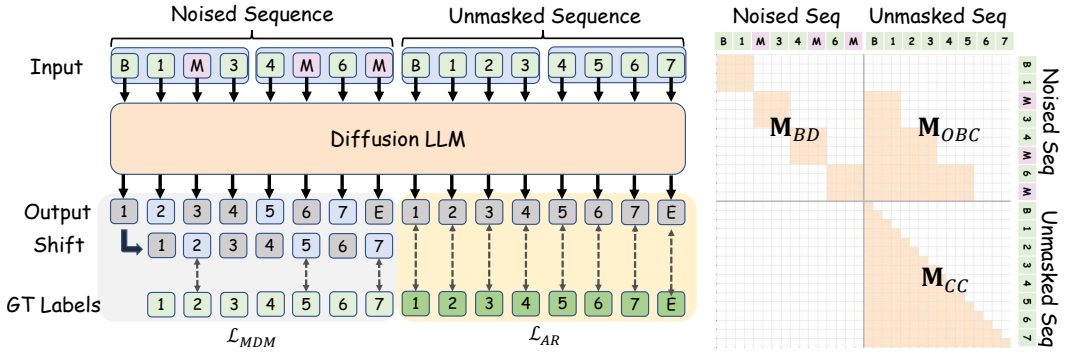

Figure 4: **Our Parallel Training Diagram.** The diagram shows the parallel training form of our Context-Causal setting (we use $blocksize = 4$ as an example; the actual $blocksize$ is 32). We concatenate a clean, unmasked token sequence to the noised sequence. The attention mask $\mathbf{M}_{\text{all}}$ is designed (shown in the right) such that strictly-causal attention is applied in the unmasked input; for the masked input, each token has bidirectional attention intra-block, but causal attention to past inter-block tokens that are unmasked. AR loss $\mathcal{L}_{\text{AR}}$ is introduced in addition to the canonical masked loss $\mathcal{L}_{\text{MDM}}$ for faster adaptation.

the noised view (matching inference); and (lower-right, $\mathbf{M}_{\text{CC}}$) strictly causal self-attention within the clean sequence. Relative to prior block-diffusion training masks (Arriola et al., 2025), we replace block-causal with fully causal (context-causal) in this lower-right quadrant, preserving the AR inductive bias in the context while still enabling intra-block bidirectionality only where generation happens. The detailed formulation of training and the structured attention mask in enclosed in Appendix A. This design optimizes per-step token utilization, and empirically stabilizes training.

3.3 AR Loss Guidance

While our one-pass parallel recipe enables efficient Block-Diffusion training, the diffusion loss is only applied to logits on the noised (active) blocks; logits on the clean-context branch of the concatenated sequence mainly serve as conditioning signals. Meanwhile, our adaptation follows a path from Block-Diffusion with $blocksize = 1$ (i.e., AR) to a target block size, and we would like this path to remain anchored to the AR behavior rather than drifting too far away. To this end, we introduce an auxiliary AR objective as a lightweight constraint, which is naturally attached to the clean-context branch because it already follows strictly causal self-attention. This turns otherwise unused context predictions into supervised next-token targets, improving token utilization without changing diffusion-side conditioning.

Let $\mathcal{C}$ index tokens on the clean context, x_i be the ground-truth token at position i, and $\mathbf{M}_{\text{CC}}$ denote the context-causal mask; we attach a standard LM head and define an autoregressive objective over the context as

$$\mathcal{L}_{\text{AR}}(\theta) = \mathbb{E}\left[-\sum_{i \in \mathcal{C}} \log p_\theta\big(x_{i+1} \mid x_{\le i};\, \mathbf{M}_{\text{CC}}\big) \right]. \tag{1}$$

Let $\mathcal{L}_{\text{MDM}}(\theta)$ be the masked/block-diffusion denoising loss computed on the noised view under $\mathbf{M}_{\text{all}}$ (as in Eq. (5)); we then train with an affine combination controlled by $\lambda \ge 0$:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{MDM}}(\theta) + \lambda\, \mathcal{L}_{\text{AR}}(\theta). \tag{2}$$

In practice, we set $\lambda = 0.5$ so that the number of tokens participating in the MDM and AR losses is kept at a comparable scale throughout adaptation. Overall, $\mathcal{L}_{\text{AR}}$ provides a simple yet effective guidance signal along the AR→Block-Diffusion path, while being "free" to compute from the clean-context branch in our parallel training formulation.

3.4 Gradual Block Growth

After determining the path for transition and guidance, we pursue a smoother adaptation by growing blocks in Block-Diffusion (Figure 5). As noted, AR could be viewed as Block-Diffusion of $blocksize = 1$; hence, the transition could be viewed under the Block-Diffusion paradigm, where we start from block size of 1 and end at the target block size. Naturally, we gradually increase the generation block size from AR's single-token steps toward larger blocks, so that the model transitions continuously from next-token prediction to next-block refinement. This monotonic growth retains left-to-right causality while progressively introducing intra-block bidirectionality, narrowing the procedural gap and easing optimization.

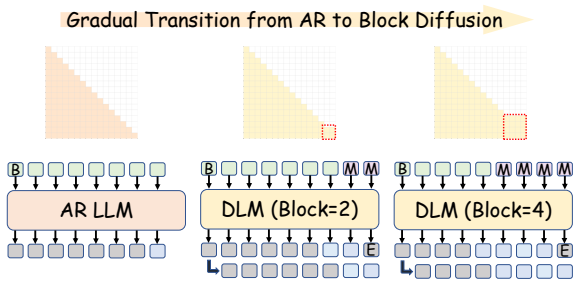

Figure 5: **The Diagram for Gradual Block Growth.** Starting from an AR model (which could be viewed as a Block-Diffusion model of $blocksize = 1$), we gradually double the $blocksize$ during training until reaching the target size, mitigating the adaptation gap between AR and Block-Diffusion.

Starting from blocksize b=1 (AR), we interpolate to larger bidirectional blocks by growing b in *integer powers* of a user-chosen base $r \in \{2, 4, \dots\}$ (on a normal basis the power of 2 is adopted) at fixed training intervals. Let s be the global training step, Δ the interval (in steps) between growth events, s_0 an optional warmup before the first growth, b_0=1 the starting size, and $b_{\max}$ the inference target. The schedule is as follows.

$$b(s) \;=\; \min\Big\{\, b_{\max},\; b_0 \cdot r^{\left\lfloor \frac{\max(0,\, s-s_0)}{\Delta} \right\rfloor} \Big\}, \quad (3)$$

which holds b constant on plateaus $[s_0 + k\Delta,\; s_0 + (k{+}1)\Delta)$ and multiplies it by r whenever s crosses a multiple of Δ (e.g., $1 \to r \to r^2 \to \dots$) until capped by $b_{\max}$. This integer-power curriculum reduces the AR $\to$ diffusion adaptation gap by aligning early optimization with the AR inductive bias (small b) and gradually unlocking intra-block bidirectionality and parallel supervision as b increases. In practice, we keep train-inference semantics matched at each plateau via the same context-causal mask family, optionally co-schedule compute by shrinking the refinement steps per block $T(s) \propto 1/b(s)$ to maintain a roughly stable token-update budget, and anneal the AR-loss weight λ as blocks grow to reallocate gradient capacity toward the diffusion objective.

4 Experiments

In this section, we empirically demonstrate the success of our proposed adaptation methods via manifold experiments. We test our method on various off-the-shelf model weights of different sizes, including Qwen3-4B-Base, Qwen3-8B-Base (Yang et al., 2025), and openPangu-Embedded-7B (Chen et al., 2025).

Experiment setup. For all experiments, we use sequence length ℓ=8k and global batch size B=1024; lr is set as $1e-5$ with the Adam optimizer. we train for 4000 iterations that consumes approximately 30B of the training data. Adaptation iteration for gradual growth is set as 2500. For the Qwen3 series, we use the FineWeb-100B (Penedo et al., 2024) dataset. For openPangu, we use a large, high-quality 700B internal dataset.

Comparison with existing baselines. On the basis of logit shift introduced in DiffuLLaMA (Gong et al., 2025a), we adopt two methods as baselines. "Annealed Attention Mask" (Gong et al., 2025a) proposes random growth from the auto-regressive causal attention mask to the targeted Diffusion attention mask. In our setting, since we are using structured attention masks for parallel training, "Annealed Attention Mask" is applied as the random interpolation between structured attention masks for $blocksize = 1$ and $blocksize = 32$. We are aware that $\mathbf{M}_{\mathrm{BD}}$ and $\mathbf{M}_{\mathrm{OBC}}$ are closely dependent; hence, we "chain" the randomness of $\mathbf{M}_{\mathrm{BD}}$ and $\mathbf{M}_{\mathrm{OBC}}$ such that each token will not view each position twice in attention. "Plain Finetuning" directly uses the targeted attention mask for training.

As shown in Table 2, our method consistently outperforms both baselines across model scales and evaluation benchmarks. Compared to annealed attention masks and plain finetuning, our approach achieves the highest average performance on Qwen3-4B, Qwen3-8B, and openPangu-7B with particularly notable gains on GSM8K, MATH, and MBPP. These results indicate that a structured and progressive adaptation strategy is more effective than either random interpolation or directly applying the target attention mask, leading to more stable and reliable performance improvements across diverse tasks and model sizes.

Contribution of adaptation. In Table 2, we also ablate our two adaptation components: AR loss and gradual block-size growth—starting from a plain fine-tuning baseline. Adding AR loss lifts the overall Avg from 48.95 to 52.97 (+4%), with the largest gains on math and MBPP and modest improvements elsewhere. Stacking gradual block-size growth further raises Avg to 54.94, indicating that smoother progression from next-token to next-block generation improves stability and yields con-

Table 2: **Comparison of Adaptation Methods.** We demonstrate the effectiveness of our method across different models and settings.We also show the contribution of each component in our adaptation methods.

Method	GSM8K	MATH	HumanEval	MBPP	Avg
Qwen3-4B-Base					
Annealed Attention Mask (Gong et al., 2025a)	76.57	28.66	25.61	59.40	47.56
Plain Finetuning	72.18	24.84	**31.10**	56.80	46.23
Ours	**79.83**	**32.26**	27.44	**61.60**	**50.28**
Qwen3-8B-Base					
Annealed Attention Mask (Gong et al., 2025a)	80.67	25.76	39.02	49.80	48.81
Plain Finetuning	78.32	24.22	**42.68**	58.20	50.85
Ours	**82.26**	**34.68**	31.71	**64.40**	**53.26**
openPangu-7B					
Annealed Attention Mask (Gong et al., 2025a)	70.05	35.46	26.83	45.00	44.34
Plain Finetuning	72.63	36.14	39.63	47.40	48.95
+ AR loss	75.13	43.30	40.24	53.20	52.97
+ AR loss & Gradual Growth (**Ours**)	**76.57**	**44.06**	**44.51**	**54.60**	**54.94**

sistent, additional benefits—especially for coding and multi-step reasoning.

5 NBDiff-7B

We further demonstrate the effectiveness of our adaptation method via intensive data scaling. We start from the openPangu-Embedded-7B (Chen et al., 2025) base checkpoint and adapt it into a diffusion language model (DLM) using the training dataset in that release. With these efforts, we launch NBDIFF-7B, a State-of-the-Art diffusion model.

5.1 Setup

Training. The pretraining adaptation stage uses a two-phase learning-rate schedule: we keep the learning rate constant for the first 24,000 iterations and then apply a learning-rate cooldown over the final 60,000 iterations, for a total of 84,000 iterations. We train with sequence length ℓ=8k and global batch size B=1024. The effective tokens processed per iteration are 8M tokens, so across 84,000 iterations the total token volume is approximately 700B tokens. NBDIFF-7B-BASE, the State-of-the-Art base model with 8K context, is completed after this stage. Then, to equip the model with long-sequence generative capabilities, we extend the pretraining sequence length ℓ=32k and train for 23,800 iterations (approximately 100B tokens), equipping the model with long-sequence modeling capabilities. Finally, we use 10B-token SFT data of sequence length ℓ=32k to finetune the model for 10 epochs (approximately 17,000 iterations) and equip it with reasoning capabilities.

We use a uniform masking strategy over the diffusion step $t \sim \mathrm{Uniform}[0, 1]$ (sampled and mapped to the discrete step index), and keep the inference/training mask families matched at each curriculum plateau. All other optimizer and system-level settings follow the default configuration of the openPangu-Embedded-7B (Chen et al., 2025) release.

Inference. For sampling during inference, we build on Fast-DLLM-v2 (Wu et al., 2025b), i.e., a Block-Diffusion (Arriola et al., 2025) instantiation of Fast-DLLM (Wu et al., 2025b): at the macro level the sequence is generated left-to-right by blocks (causal across blocks), while inside each block we permit bidirectional attention to refine tokens jointly. For speed, the inner refinement can follow the v2 "small-block" schedule or be collapsed into a single full-block bidirectional pass when latency matters. Compared with the vanilla recipe, we replace confidence-based scheduling with an entropy-based parallel decoding rule. For code benchmarks, we enable sampling and set $top\_p = 0.9$ and $T = 0$ for better performance. To balance train-inference consistency and throughput, we set the macro block size to 32, which aligns masking between training and decoding and yields substantial parallelism, and follow Fast-DLLM-v2 (Wu et al., 2025a) for the small block size of 8 during intra-block refinement. The experiment results are all single-run outcomes.

5.2 Evaluation

We primarily evaluate NBDIFF-7B across three capability areas—code, math, and general knowl-

Table 3: **Comparison between NBDIFF-7B-INSTRUCT and the latest SFT (Instruct) version diffusion language models.** Our model demonstrates strong performance on general, math, and coding tasks, and outcompetes latest diffusion baselines by large margins. * indicates non-official replications.

Benchmark	LLaDA-MoE 7B-A1B	LLaDA2.0-mini preview 16BA1B	Dream-v0 Instruct-7B	SDAR 8B	Ours-7B Instruct
General					
MMLU	67.2	72.5	67.0	78.6	**82.1**
MMLU-Pro	44.6	49.2	43.3	56.9	**73.6**
CMMLU	64.3	67.5	58.8	75.7	**77.1**
CEval	63.9	66.5	58.0	72.7*	**74.3**
IFEval	59.3	**62.5**	**62.5**	61.4	60.6
Math					
GSM8K	82.4	89.0	81.0	**91.3**	91.0
MATH	58.7	73.5	39.2	78.6	**84.0**
Coding					
MBPP	70.0	77.8	58.8	72.0	**87.6**
HumanEval	61.6	80.5	55.5	78.7	**89.0**
Avg	61.1	71.0	58.2	74.0	**79.9**

edge—and compare its performance against several baseline models to understand relative strengths and trade-offs. We evaluate general capabilities on MMLU (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024), CEVAL, CMMLU (Li et al., 2023), and IFEval (Zhou et al., 2023); mathematical reasoning on GSM8K (Cobbe et al., 2021) and MATH (Lewkowycz et al., 2022); and coding performance on MBPP (Austin et al., 2021b) and HumanEval (Chen et al., 2021).

We present the SFT (Instruct) results (*i.e.* NBDIFF-7B-INSTRUCT) in Table 3. Due to page limits, the Base version, NBDIFF-7B-BASE, is presented in Appendix B. NBDIFF-7B-INSTRUCT delivers the highest macro average (79.9) among SFT baselines, substantially outperforming SDAR-8B (Cheng et al., 2025) and LLaDA (Nie et al., 2025) variants (Zhu et al., 2025; inclusionAI, 2025). On general knowledge, it sets the pace on MMLU (82.1), MMLU-Pro (73.6), and CMMLU (77.1), and ranks second on CEval (74.3) while remaining competitive on IFEval (60.6) despite ties among baselines. For math, NBDIFF-7B-INSTRUCT achieves good GSM8K performance (91.0) and State-of-the-Art MATH performance (84.0), indicating strong multi-step and competition-style reasoning under instruction following. In coding, it tops both MBPP (87.6) and HumanEval (89.0), narrowing and in most cases reversing the AR-favoring gap seen in some base models. Taken together, these results show that instruction tuning on a diffusion LLM not only preserves the Base model's breadth, but amplifies performance across general, math, and coding by large margins, establishing NBDIFF-7B-INSTRUCT as a strong, balanced SFT model in the 7B class.

6 Conclusion

In this work, we propose a principled adaptation framework that bridges the gap between Autoregressive (AR) and Block-Diffusion models. By reframing adaptation as a continuous interpolation–viewing AR as a Block-Diffusion model with a block size of one–we introduce the context-causal attention mechanism and an efficient parallel training recipe with auxiliary AR supervision, which maximally preserves the pre-trained knowledge of the source model. We also propose a block-size growth curriculum that smoothly transitions the model from sequential to parallel generation.

Our resulting model, NBDIFF-7B, has achieves state-of-the-art performance among 7B-parameter diffusion models, outperforming strong baselines on math, code, and general reasoning benchmarks. These results demonstrate that expensive pre-training from scratch is not necessary to build high-quality diffusion LLMs. Instead, our method offers a compute-efficient pathway to unlock parallel generation capabilities in existing open-source AR checkpoints, potentially accelerating the deployment of faster and more flexible generative models.

Limitations

Our study is constrained by compute: we did not scale beyond the 7–8B regime. We also focused deliberately on adaptation; we did not explore reinforcement-learning–based objectives (e.g., policy optimization for reasoning/code) or more advanced decoding policies beyond our entropy-guided block refinement. Extending the approach to larger model scales, integrating RL pipelines, and systematically evaluating richer decoding and routing strategies (e.g., adaptive block sizing, uncertainty-aware early exit) are promising directions. **Risks:** The NBDIFF model may occasionally generate offensive or harmful content (e.g., toxic language, stereotypes, or culturally insensitive statements). We recommend deploying with content filtering, user feedback loops, and continual safety fine-tuning to mitigate residual risks. **AI Usage:** AI is used to polish academic papers and assist with formatting.

References

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. In *ICLR*.

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021b. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Hanting Chen, Yasheng Wang, Kai Han, Dong Li, Lin Li, Zhenni Bi, Jinpeng Li, Haoyu Wang, Fei Mi, Mingjian Zhu, Bin Wang, Kaikai Song, Yifei Fu, Xu He, Yu Luo, Chong Zhu, Quan He, Xueyu Wu, Wei He, and 5 others. 2025. Pangu embedded: An efficient dual-system llm reasoner with metacognition. *Preprint*, arXiv:2505.22375.

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, and 39 others. 2021. Evaluating large language models trained on code.

Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. 2025. Sdar: A synergistic diffusion–autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*.

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

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. 2025a. Scaling diffusion language models via adaptation from autoregressive models. *Preprint*, arXiv:2410.17891.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. 2025b. Diffucoder: Understanding and improving masked diffusion models for code generation. *Preprint*, arXiv:2506.20639.

Xu Han, Xiaoya Li, Shuhe Wang, Tianwei Zhang, Boshi Wang, Yuxian Meng, Jiwei Li, and Fei Wu. 2023. Ssd-lm: Semi-autoregressive simplex-based diffusion language modeling. In *ACL*.

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

inclusionAI. 2025. Llada2.0-mini-preview. Hugging Face Model Card.

Xiangqi Jin, Yuxuan Wang, Yifeng Gao, Zichen Wen, Biqing Qi, Dongrui Liu, and Linfeng Zhang. 2025. Thinking inside the mask: In-place prompting in diffusion llms. *Preprint*, arXiv:2508.10736.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. 2025. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *Preprint*, arXiv:2502.06768.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *NeurIPS*.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. 2025. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*.

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. dkv-cache: The cache for diffusion language models. *arXiv preprint arXiv:2505.15781*.

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

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2024. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Preprint*, arXiv:2406.17557.

Subhendra Sekhar Sahoo, Quentin Anthony, Mengzhou Xia, Wojciech Stokowiec, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *NeurIPS*.

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. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. 2025. wd1: Weighted policy optimization for reasoning in diffusion language models. *Preprint*, arXiv:2507.08838.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025a. Remasking discrete diffusion models with inference-time scaling. *Preprint*, arXiv:2503.00307.

Guanghan Wang, Yair Schiff, Gilad Turok, and Volodymyr Kuleshov. 2025b. d2: Improved techniques for training reasoning diffusion language models. *Preprint*, arXiv:2509.21474.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. 2025c. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. *Preprint*, arXiv:2508.09192.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. 2025a. Fast-dllm v2: Efficient block-diffusion large language model. *arXiv preprint arXiv:2509.26328*.

Chengyue Wu and 1 others. 2025b. Fast-dllm: Training-free acceleration of diffusion llm by block-wise kv caching and dependency repair. *arXiv preprint arXiv:2505.22618*.

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, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. 2024. Diffusion of thoughts: Chain-of-thought reasoning in diffusion language models. *Preprint*, arXiv:2402.07754.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *Preprint*, arXiv:2508.15487.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. 2025. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2025. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *Preprint*, arXiv:2504.12216.

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

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, and 7 others. 2025. Llada-moe: A sparse moe diffusion language model. *Preprint*, arXiv:2509.24389.

A Methodology Details

Attention mask for parallel training. The naive block-diffusion recipe is data-inefficient: random cropping wastes the remaining tokens of each sequence, and only a small subset of masked tokens inside the last block contributes to the loss. Unlike AR pretraining—where every token can supervise next-token prediction—switching to next-*block* prediction sharply reduces token utilization. We therefore restructure training so that all blocks provide learning signal in a single forward pass.

We seek to model all blockwise conditionals in parallel using a single transformer call. Instead of invoking the denoiser B times, we concatenate a *noised* view x_t (partitioned into blocks) with the *clean* sequence x:

$$x_{\text{all}} = x_t \oplus x \quad \text{(length } 2L\text{)}.$$

A structured attention mask $\mathbf{M}_{\text{all}} \in \{0,1\}^{2L\times 2L}$ updates all token representations in one shot:

$$\mathbf{M}_{\text{all}} = \begin{bmatrix} \mathbf{M}_{\text{BD}} & \mathbf{M}_{\text{OBC}} \\ \mathbf{0} & \mathbf{M}_{\text{CC}} \end{bmatrix}. \tag{4}$$

Within the noised view x_t, attention is restricted to be block-wise (block-diagonal):

$$[\mathbf{M}_{\text{BD}}]_{ij} = \begin{cases} 1, & i \text{ and } j \text{ are in the same block,} \\ 0, & \text{otherwise.} \end{cases}$$

From noised tokens to the clean context, we allow only *earlier* clean blocks as conditioning (offset block-causal):

$$[\mathbf{M}_{\text{OBC}}]_{ij} = \begin{cases} 1, & \text{clean position } j \text{ lies in a block} \\ & \text{strictly before the block of } i, \\ 0, & \text{otherwise.} \end{cases}$$

Inside the clean context, we keep strict left-to-right causality (context-causal):

$$[\mathbf{M}_{\text{CC}}]_{ij} = \begin{cases} 1, & j \le i, \\ 0, & j > i. \end{cases}$$

The lower-left tile is zero so the clean context never reads from the noised view, matching inference-time semantics.

Let $\mathcal{B}$ index all blocks and $\mathcal{M}_t$ be the step-dependent visibility inside the noised view. Under Eq. (4), one forward pass supplies gradients for all masked tokens across all blocks:

$$\mathcal{L}_{\text{parallel}}(\theta) = \mathbb{E}\Bigg[-\sum_{B\in\mathcal{B}} \sum_{i\in B:\, \mathcal{M}_t(i)=0} \log p_\theta\big(x_i \mid x_t, x;\, \mathbf{M}_{\text{all}}\big) \Bigg]. \tag{5}$$

Processing x_t and x jointly amortizes KV-cache construction, maximizes per-step token utilization, and empirically stabilizes training compared with randomly growing global masks. An example for L=16 and block size b=4 is illustrated in Fig. 4; but in reality, we use b=32.

B Other Experiments

NBDiff-7B-Base. We also measure the performance of NBDIFF-7B-BASE. The comparative results for our NBDIFF-7B-BASE against strong 7B baselines is summarized in Table 4. Apart from the introduced benchmarks, we also include BBH (BIG-Bench Hard) (Suzgun et al., 2022), which is a curated set of particularly difficult tasks targeting abstraction, compositionality, and complex reasoning. Overall, NBDIFF-7B-BASE attains the highest macro average, surpassing Dream-v0-Base-7B and both LLaDA bases. On general knowledge, it leads on MMLU-Pro (52.7), CMMLU (76.9), CEval (75.9), and BBH (69.4), and remains competitive on MMLU (69.1, second only to Dream's 69.5). In math, NBDIFF-7B-BASE ranks first on both GSM8K (79.6) and MATH (46.0), indicating strong multi-step and competition-style reasoning. In coding, it is consistently runner-up, slightly behind Dream-v0 (Ye et al., 2025) but ahead of the LLaDA (Nie et al., 2025) baselines. Taken together, these results show that a diffusion-style LLM can match or outperform autoregressive bases across diverse evaluations, with particularly clear gains on harder general-reasoning and Chinese benchmarks.

Base vs. SFT AR weight initialization. We attempted adapting both a pretrained Base AR checkpoint and an instruction-tuned (SFT) checkpoint into our DLM, with the comparison summarized in Table 5. Somewhat surprisingly, the Base-initialized model achieves the stronger overall balance (Though Avg 53.26 vs. 54.39 for SFT, SFT's edge is driven largely by HumanEval's small-set volatility; for other benchmarks, the Base model performs slightly better). We hypothesize two causes: 1. Objective alignment: the Base model is trained purely on next-token prediction, which better complements our masked-diffusion objective and auxiliary AR-loss, whereas SFT shifts the likelihood landscape toward instruction formats and response conventions; 2. Format priors: SFT injects stylistic and safety priors (headings, disclaimers, verbosity) that are beneficial for chat but act as spurious targets for diffusion.

Table 4: **Comparison between NBDIFF-7B-BASE and latest base-version diffusion language models.** Our base model shows strong performance on general, math, and coding benchmarks.

Benchmark	LLaDA-8B Base	LLaDA-MoE-7B A1B-Base	Dream-v0 Base-7B	Ours-7B Base
General				
MMLU	65.9	64.6	**69.5**	70.1
MMLU-Pro	41.8	39.2	48.2	**59.1**
CMMLU	69.9	65.7	60.9	**77.3**
CEval	70.5	65.6	59.2	**73.0**
BBH	49.8	52.7	57.9	**77.3**
Math				
GSM8K	70.7	66.4	77.8	**78.8**
MATH	27.3	36.1	39.6	**46.0**
Coding				
MBPP	38.2	52.4	**56.2**	55.8
HumanEval	33.5	45.7	**57.9**	50.0
Avg	52.0	54.3	60.1	**65.3**

Table 5: **Comparing DLMs adapted from Base and SFT version of loaded weights.** Context-Causal gains a clear advantage in adaptation from AR.

Scheme	GSM8K	MATH	HumanEval	MBPP	Avg
Qwen3-8B-Base	**82.26**	**34.68**	31.71	**64.40**	53.26
Qwen3-8B	81.05	32.24	**42.68**	61.60	**54.39**

C Related Work

Discrete diffusion language models. Diffusion has been extended to categorical spaces, showing that discrete denoising objectives can effectively model text and clarifying connections to classical LM training, including transition design and absorbing states (Austin et al., 2021a). Two dominant paradigms have since emerged. Masked diffusion iteratively reveals tokens, enabling controllable generation with competitive likelihoods without left-to-right decoding (Li et al., 2022), while recent MDLM variants substantially close the perplexity gap to AR LMs using simplified training recipes (Sahoo et al., 2024). Absorbing-state diffusion instead corrupts tokens toward a sink symbol; recent analyses relate its objective to conditional modeling, calibration, and sampling behavior (Ou et al., 2024). At scale, systems trained from scratch such as LLaDA demonstrate that masked-diffusion-style pretraining can rival strong AR baselines and extend naturally to multimodal instruction tuning (Nie et al., 2025; You et al., 2025), establishing the viability of DLMs at billion-parameter scale.

Recent trend: Block-Diffusion, adaptation from AR, and test-time scaling. While Full-Sequence Diffusion provides fully bidirectional context, it is computationally inefficient for long texts and misaligned with left-to-right inductive biases. Attempts to amortize this cost via intermediate-state caching improve efficiency but do not fundamentally resolve the issue (Liu et al., 2025; Ma et al., 2025; Wu et al., 2025b). Block-Diffusion addresses this by fixing past context while denoising the current block bidirectionally, enabling parallel token updates and unbounded-length generation with tunable quality–efficiency trade-offs (Han et al., 2023; Arriola et al., 2025). Beyond training from scratch, several works adapt pretrained AR models into diffusion-style decoders, often at the block level, reporting objective connections, practical conversion recipes, and hybrid AR–diffusion paradigms that preserve AR quality while enabling parallel generation (Gong et al., 2025a; Cheng et al., 2025; Wu et al., 2025b,a). In parallel, diffusion-based reasoning systems explore inference-time scaling or reinforcement learning to improve multi-step reasoning, particularly for math and code, but remain limited by short contexts or underutilized AR priors (Ye et al., 2024; Wang et al., 2025a; Jin et al., 2025; Gong et al., 2025b; Zhao et al., 2025; Tang et al., 2025; Wang et al., 2025b). In contrast, our approach adapts strong AR models into block-diffusion generators via a smooth way, enabling longer context and better performance.