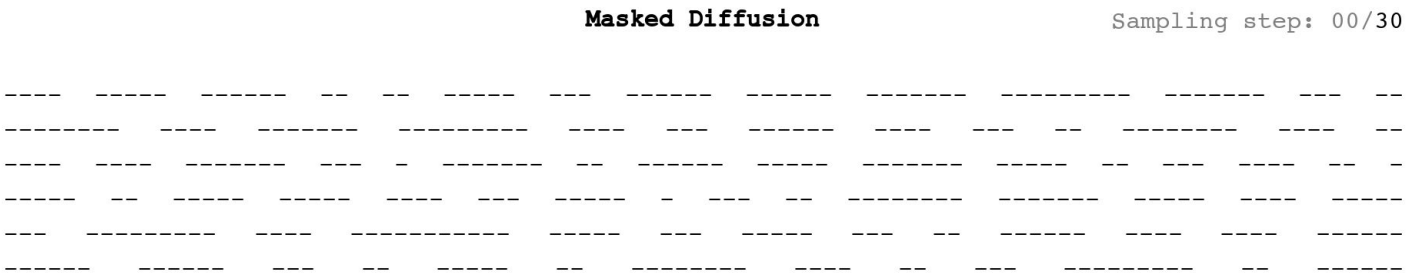
TiDAR: Think in Diffusion, Talk in Autoregression

Jingyu Liu & Xin Dong

Zhifan Ye, Rishabh Mehta, Yonggan Fu, Vartika Singh,
Jan Kautz, Ce Zhang, Pavlo Molchanov

Project page: https://tidarlm.github.io/
Paper link: https://arxiv.org/abs/2511.08923v1

The dilemma



(Gif from diffusion duality, Sahoo et al.)

- AR models excel at quality due to autoregressive language modeling
  - But generating one token at a time might not be the fastest approach due to under-utilized GPU compute during memory-bound regime
- Diffusion models hold the promise of parallel decoding
  - But usually resulting in degraded quality with more tokens generated per NFE
- Can we combine the advantages from both worlds?

TiDAR

- TiDAR is a sequence-level hybrid architecture that simultaneously conducts drafts (**thinking**) with **diffusion** and sample (**talking**) with **autoregression** within a single model and forward

- The efficiency comes from using the ``free token slots'' that are available on modern GPUs
  - In a single forward, sending one token (AR + bs = 1) results in similar latency as sending K tokens (our case) when K is chosen wisely.

Free Tokens Slots

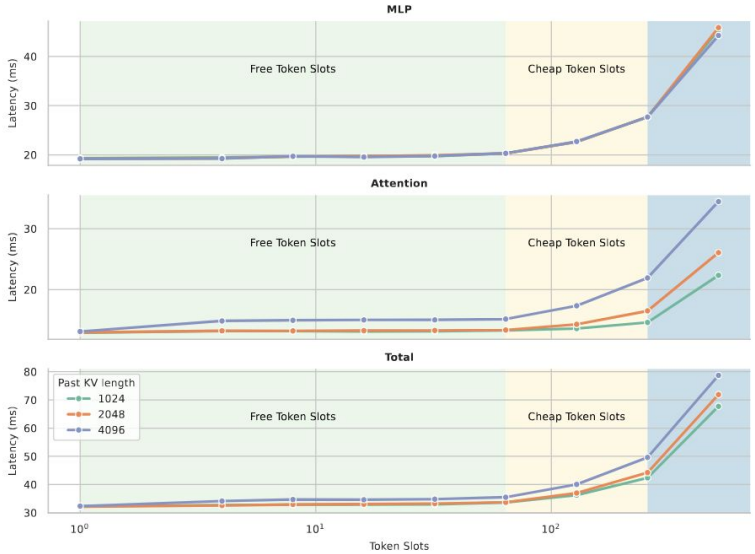
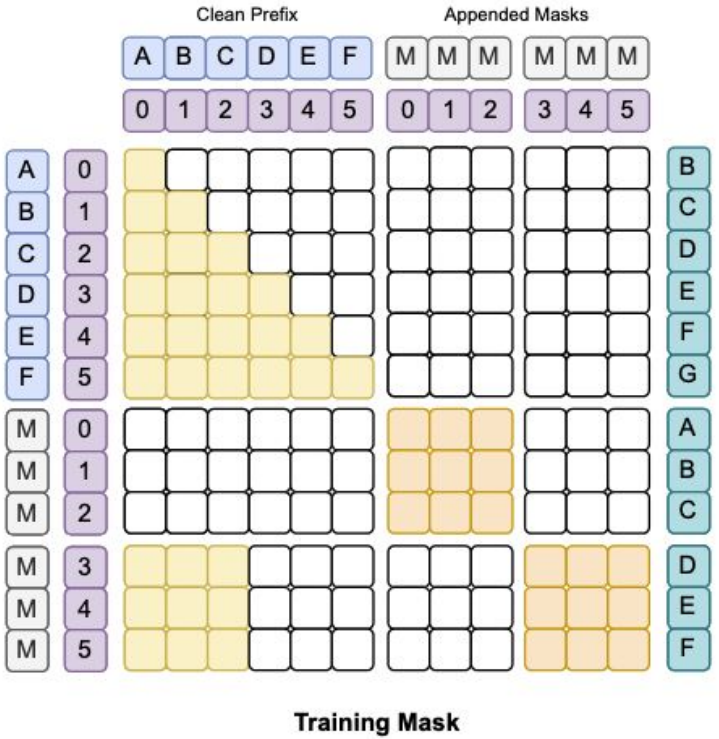


Figure 1 | **Latency Scaling over Token Slots:** We plot the latency of Qwen3-32B decoding on NVIDIA H100 with batch size=1 and Flash Attention 2 [15] over different prefix lengths. Latency stays relatively the same with a certain amount of tokens sent to forward (free + cheap token slots), before transitioning to the compute-bound regime. We leverage this characteristic to achieve almost free parallelled drafting and sampling for TiDAR.

Main Architecture and Training

- TiDAR is trained by initializing from an AR model (e.g. Qwen2.5 1.5B and Qwen3 8B)
- We continually pre-train it with a combination of AR and diffusion loss
- We adopt a similar training mask to SBD and Block Diffusion but with the **corrupted sequence being full masks**



Training Mask

Main Architecture and Training

- TiDAR's training objective provides many advantages:
  - No need to find a sophisticated masking strategy
  - Can take signals from the AR part
    - Normal diffusion < AR < TiDAR (loss density)
  - Easy loss balancing between two terms due to equi-variance from the same number of loss tokens

$$\mathcal{L}_{TiDAR}(\theta) = \frac{1}{1+\alpha}\left(\sum_{i=1}^{S-1}\frac{\alpha}{S-1}\cdot\mathcal{L}_{AR}(x_i, x_{i+1};\theta) + \sum_{i=1}^{S-1}\frac{1}{S-1}\cdot\mathcal{L}_{Diff}([mask], x_i;\theta)\right)$$

Inference

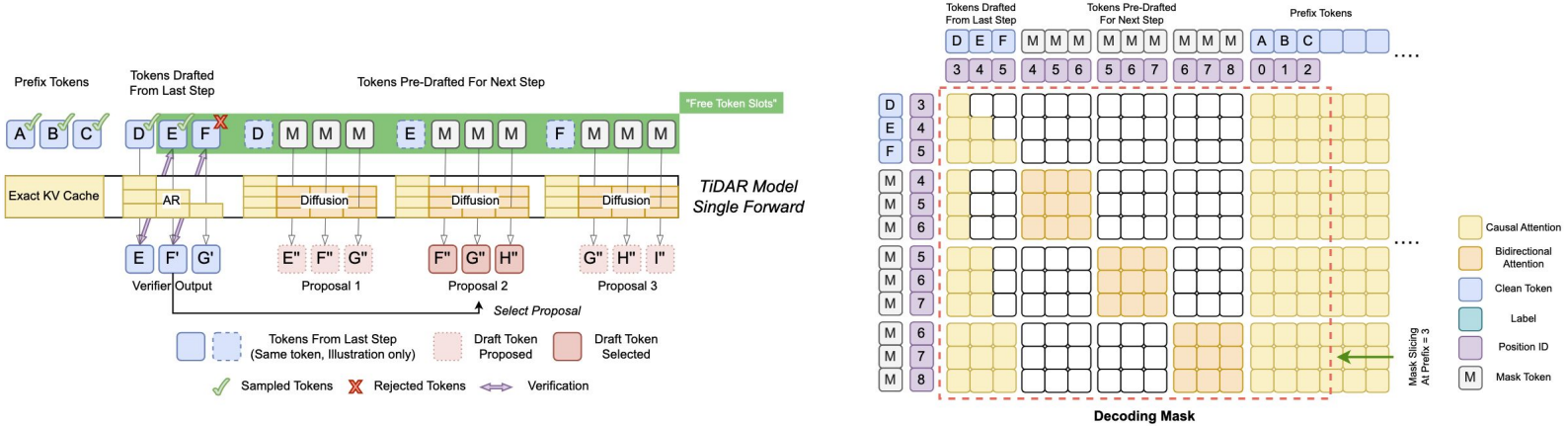


Figure 2 | **TiDAR Architecture:** TiDAR uses a single model forward to sample drafted tokens from the last step and pre-draft tokens for the next step in parallel. By switching the attention pattern among different parts of the sequence, TiDAR encodes the clean tokens drafted from last step causally and mask tokens block-causally (bidirectional within each block) for one-step diffusion pre-drafting. Upon accepting a prefix, the corresponding pre-drafts (proposal) can be selected. The KV cache for tokens forwarded causally will be stored and later evicted if the corresponding tokens are rejected. We illustrate this with a draft length of 3 and an accepted length of 2. Figure 3 shows the exact decoding mask for this example.

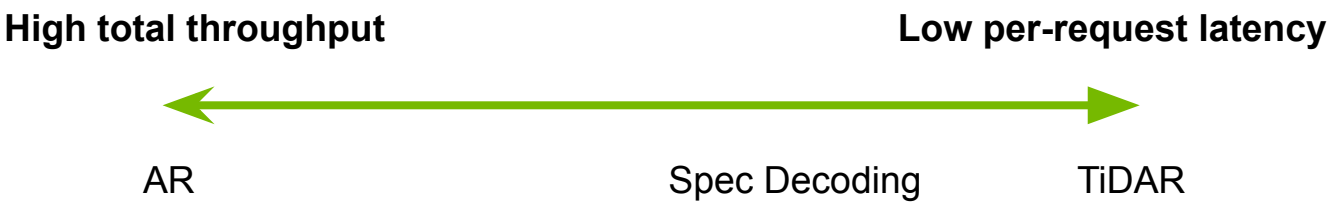
Demo: https://tidarlm.github.io/

Related Works

- Pure diffusion models: Dream, Llada, MDLM, Block Diffusion.
  - These methods usually get worse results when we increase the parallelism
- Speculative decoding

	Draft Model		Drafting Process	
Model	**Shared with Base**	**Drafting Capacity**	**Parallel Decoding**	**Parallel to Verification**
Classic Spec. Decoding [16]	✗	Low	✗	✗
APD [17]	✗	High (Weak Verifier)	✓	✗
EAGLE-3 [18] & DeepSeek-V3 [19]	✓*	Mid	✗	✗
Apple MTP [20]	✓*	Mid	✗	✓
TiDAR	✓	High	✓	✓

Grand Spectrum of Serving



Experiment Setup

- Initializing from Qwen series
- Full-weight continual-pretrain with 50B tokens (1.5B) and 150B tokens for (8B)

Main Results - Likelihood

- The way to evaluate likelihood for diffusion models has always been under debate:
  - E.g. Calculated on masked positions averaged over Monte Carlo samples
  - Not directly comparable to AR models (can compare multi-choice questions)

- TiDAR uses a single forward pass in AR mode (by setting the attention mask to causal) to get likelihood results.

Main Results - Likelihood

		Knowledge	Commonsense Reasoning					
Model Arch	**Size**	**MMLU**	**ARC-e**	**ARC-c**	**Hellaswag**	**PIQA**	**Winogrande**	**Avg**
Llama3.2	1B	30.98%	65.28%	36.35%	63.76%	74.43%	63.30%	55.68%
SmolLM2	1.7B	49.99%	77.82%	47.44%	71.44%	77.64%	67.88%	65.37%
Qwen2.5	0.5B	47.65%	64.77%	31.83%	52.25%	70.02%	57.70%	54.04%
Qwen2.5	1.5B	60.96%	75.17%	45.05%	67.90%	76.12%	65.75%	65.16%
Qwen3	1.7B	62.53%	73.32%	44.62%	66.43%	75.63%	64.96%	64.58%
Block Diff	1.5B*	57.94%	74.41%	45.73%	56.26%	70.13%	61.80%	61.05%
TiDAR	1.5B*	58.99%	77.78%	45.39%	65.26%	75.52%	63.61%	64.43%
Qwen3	4B	73.00%	79.00%	52.00%	74.00%	78.00%	72.00%	71.33%
Qwen3	8B	76.93%	81.90%	53.16%	78.59%	79.22%	75.69%	74.25%
Block Diff	4B†	71.53%	81.48%	55.63%	65.48%	74.92%	70.96%	70.00%
Dream	7B	67.00%	82.20%	59.13%	73.73%	75.52%	73.56%	71.86%
LLaDA	8B	65.86%	73.78%	49.15%	71.05%	73.88%	74.66%	68.06%
TiDAR	8B‡	76.57%	84.18%	58.53%	76.36%	80.25%	76.48%	75.40%

Table 3 | **Likelihood Evaluation Results:** We compare our method on factual knowledge and common sense reasoning tasks using likelihood estimation. For Llada, Dream, and Block Diffusion, we follow the standard diffusion likelihood evaluation using Monte Carlo (MC) sampling. For our model, we evaluate the likelihood using pure causal mask like AR models, as natively supported by our architecture. Models initialized from QWEN2.5 1.5B, QWEN3 4B, and QWEN3 8B are super-scripted by $\star$, $\dagger$, and $\ddagger$ respectively. Models with *italicized* names are using MC with 128 steps for likelihood evaluation.

Main Results - Generative

		Coding				Math		Avg
Model Arch	**Size**	**HumanEval**	**HumanEval+**	**MBPP**	**MBPP+**	**GSM8k**	**Minerva Math**	
Llama3.2	1B	17.68%	14.63%	26.60%	38.89%	5.67%	1.92%	17.57%
SmolLM2	1.7B	0.61%	0.61%	35.40%	47.62%	28.20%	11.28%	20.62%
Qwen2.5	0.5B	27.44%	25.61%	29.60%	44.97%	37.45%	14.48%	29.92%
Qwen2.5	1.5B	35.98%	29.88%	43.60%	59.23%	54.74%	26.40%	41.64%
Qwen3	1.7B	48.17%	41.46%	55.80%	71.43%	66.72%	29.74%	52.22%
Block Diff	1.5B*	39.02%	34.76%	34.00%	48.15%	52.99%	21.56%	38.41%
TiDAR	1.5B*	43.29% (6.50)	39.02% (6.50)	41.40% (9.25)	61.11% (9.43)	53.90% (5.07)	25.48% (7.92)	44.03% (7.45)
Qwen3	4B	57.32%	50.61%	67.00%	80.69%	77.48%	47.10%	63.37%
Qwen3	8B	64.63%	56.71%	69.40%	83.07%	81.80%	52.94%	68.09%
LLaDA	8B	32.32%	27.44%	40.80%	51.85%	70.96%	27.30%	41.78%
Dream	7B	54.88%	49.39%	56.80%	74.60%	77.18%	39.60%	58.74%
Block Diff	4B†	56.10%	51.22%	54.60%	69.84%	82.87%	47.02%	60.27%
TiDAR (Trust AR)	8B‡	55.49% (7.46)	52.44% (7.44)	65.40% (9.96)	79.63% (10.13)	79.83% (6.90)	50.58% (7.48)	63.90% (8.23)
TiDAR (Trust Diff)	8B‡	57.93% (7.30)	55.49% (7.29)	65.40% (10.00)	80.95% (10.13)	80.44% (7.07)	51.64% (7.68)	65.31% (8.25)

Table 2 | **Generative Evaluation Results:** We evaluate the generative sample quality of TiDAR over several coding and math tasks. For all diffusion models we decode one token per forward for the best quality and report the average number of tokens (T/NFE) generated by TiDAR in parenthesis. Models initialized from QWEN2.5 1.5B, QWEN3 4B, and QWEN3 8B are super-scripted by $\star$, $\dagger$, and $\ddagger$ respectively.

Main Results - Efficiency and Quality Trade-offs

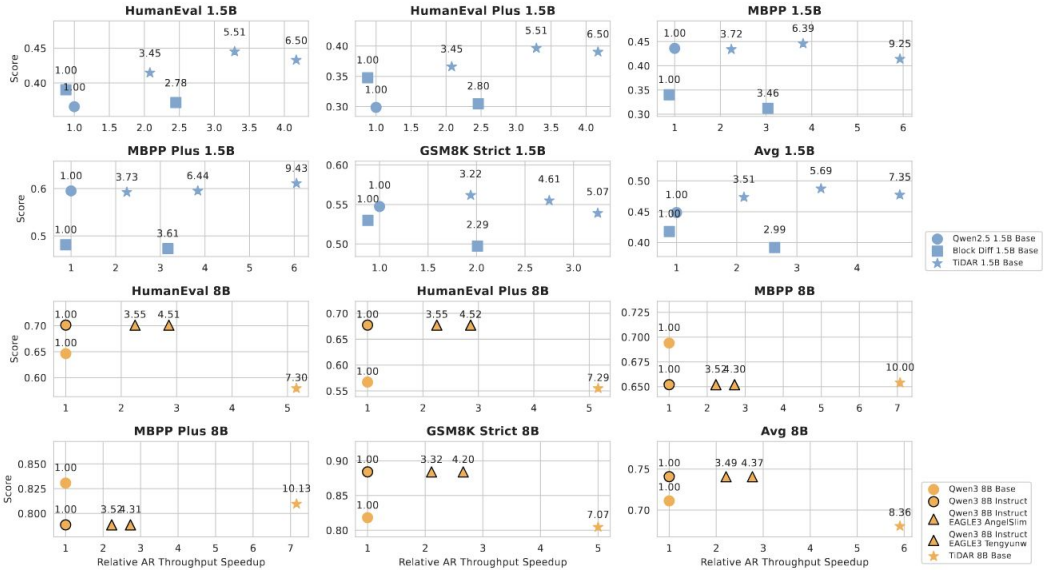


Figure 4 | **Efficiency-Quality Benchmarking:** We compare TiDAR on 1.5B and 8B with AR, AR with speculative decoding (EAGLE-3), and Block Diffusion. Points colored the same indicate the same model sizes while markers suggest different methods. On the y-axis we have individual task scores. On the x-axis, we showcase the relative decoding throughput speedup measured in tokens per second, with the baseline being the AR model within the same size group (● Qwen2.5 1.5B Base, ● Qwen3 8B Base and ● Qwen3 8B Instruct). On top of each point, we report the average tokens per NFE. For 1.5B models, we showcase two and three different settings for Block Diffusion (threshold = max, 0.8, illustrated from left to right) and TiDAR (training block size = 4, 8, 16, illustrated from left to right) respectively.

What does the speedup look like in SGLang?

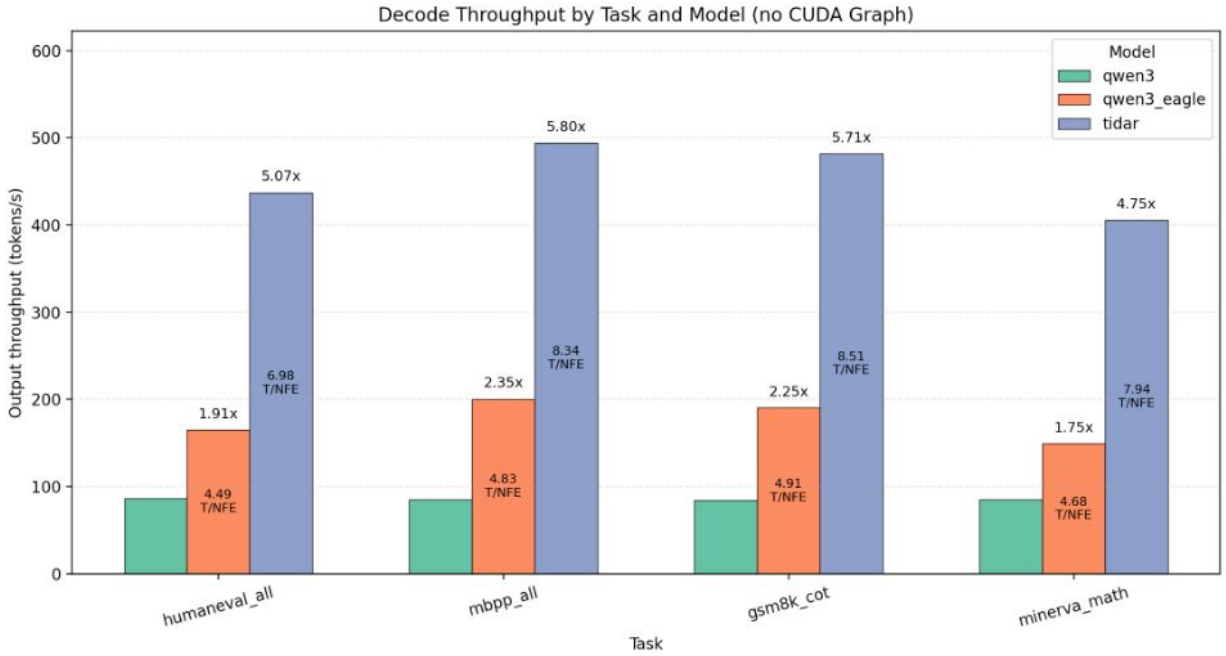


Figure 4: TiDAR 8B decoding speed on SGLang using a single H100. TiDAR achieves up to 5.8x higher decoding T/s than AR and 2.5x+ higher than EAGLE-v3.

Ablations

- We want to understand several questions:
  - Pareto frontier of different models under the same training recipe
    - *TiDAR is the best among OSS DLLMs*
  - Comparing TiDAR with other decoding strategies
    - *TiDAR's outperforms other methods*
  - Trusting AR and Diffusion for sampling
    - *Conclusion the advantages of incorporating "autoregression" into generation*
  - How useful is our full-mask strategy
    - *Proves the usefulness both in terms of quality and efficiency*

Ablation - Pareto Frontier under the Same Training Recipe

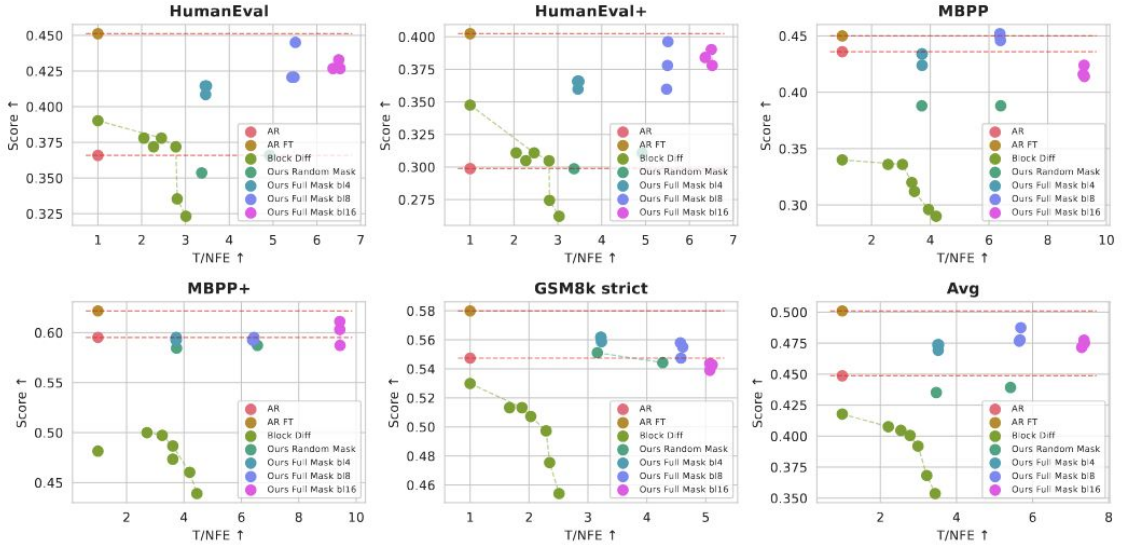


Figure 5 | **Pareto Frontier of Different Architectures with the Same Recipe:** We report the performance-efficiency trade-offs on 1.5B scale among AR model, fine-tuned AR model, Block Diffusion under different decoding thresholds, and TiDAR using different drafting lengths. Our model achieves the best Pareto Frontier compared to Block Diffusion and AR and is approaching the quality of fine-tuned AR with 7x more tokens per NFE.

Ablation - Comparing TiDAR with other decoding strategies

Strategy	Avg T/NFE	HumanEval Avg	MBPP Avg	GSM8k
Confidence Max	1.00	34.45%	43.92%	53.07%
	2.00	23.78%	22.19%	38.06%
Left to right (AR)	1.00	36.28%	46.51%	53.37%
	2.00	21.95%	18.61%	41.32%
Confidence > 0.9	2.63	32.01%	42.50%	51.40%
Confidence > 0.8	3.06	28.96%	39.28%	47.54%
Confidence > 0.7	3.42	27.74%	33.90%	43.44%
Confidence > 0.6	3.81	22.56%	26.47%	37.60%
TiDAR (4 drafts)	3.47	38.42%	50.96%	55.87%
TiDAR (8 drafts)	5.49	39.94%	52.13%	54.74%
TiDAR (16 drafts)	6.97	41.16%	51.26%	53.90%

Table 4 | **Comparing Different Decoding Strategies:** We showcase the superiority of our proposed parallel draft and sampling process (on full mask models) compared to using standard confidence/negative entropy based decoding, as well as the autoregressive decoding schemes.

Ablation - Trusting AR and Diffusion for Sampling

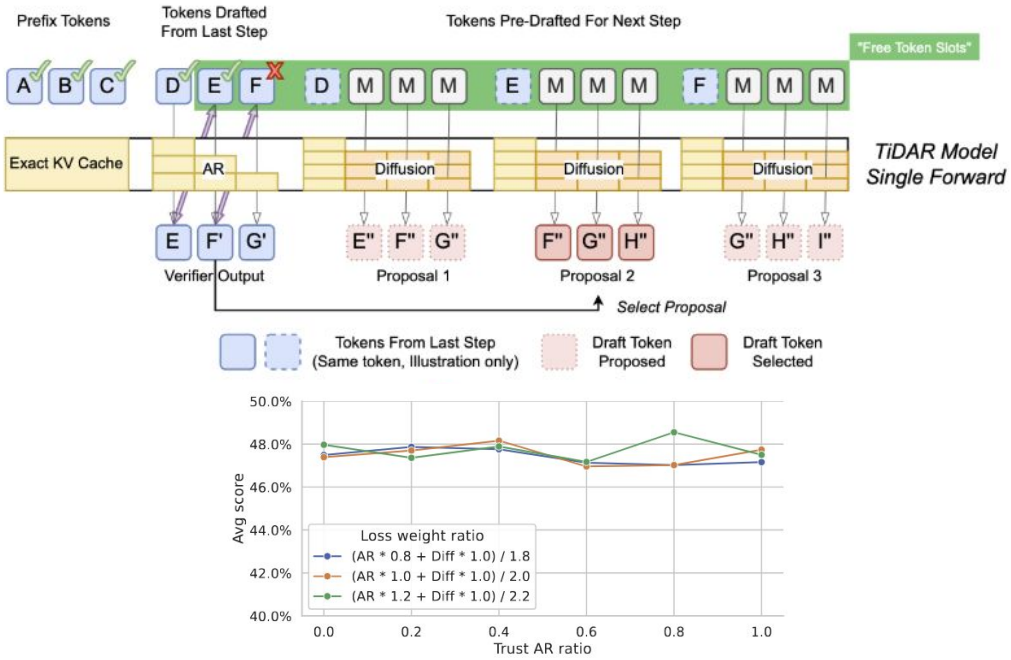


Figure 6 | **Trusting AR v.s. Diffusion Outputs for Sampling:** TiDAR is trained to be highly balanced in the sense that no matter whether we trust the prediction from the diffusion or AR parts, the autoregressive sampling, along with the high drafting model capacity, can guarantee quality under almost the same speedup.

Ablation - How Useful is our Full-mask Strategy

Masking	Draft	HumanEval Avg	MBPP Avg	GSM8k	Avg
Random	4	32.62% (3.37)	48.63% (3.72)	55.11% (3.16)	45.45% (3.42)
	8	33.85% (4.92)	48.77% (6.48)	54.43% (4.27)	45.68% (5.22)
Full	4	38.42% (3.46)	50.96% (3.72)	55.87% (3.23)	48.42% (3.47)
	8	39.94% (5.49)	52.13% (6.40)	54.74% (4.58)	48.94% (5.49)

Table 5 | **Quality-efficiency Improvement from Full Masking:** Turning random corruption strategy to full masking on the model, we substantially improve both efficiency and quality due to less train-test discrepancy and richer diffusion loss signals. Average T/NFE is shown in the parenthesis.

TiDAR's implication from hardware perspective

- Tensor parallelism:
  - we don't need to batch up if we don't have to
- RL rollout:
  - Scaling GPUs -> making the waiting time per gradient update shorter
- FP8/FP4 inference
- Efficient kernels: going beyond FlashInfer with custom boolean mask to squeeze more performance

Limitations

- Quality gap at larger scale (8B) vs AR:
  - We notice some quality gap between TiDAR and AR models when we fine-tune TiDAR from AR model checkpoint. Better data and fine-tuning recipe should help here.

- Batch size:
  - Experiments done with batch size 1, which is generally the norm when comparing dLLM tokens/sec.
  - Possible to accommodate higher batch sizes via adjusting the block (draft) length in a zero-shot manner.

- Long context extension:
  - Current implementation requires doubling the sequence length with appended mask tokens during training. In a long context, the model may be less memory bound and certain advantages of TiDAR may diminish.

- System optimization:
  - In future, writing custom attention kernels and scheduling algorithms may maximize the use of the “free token slots” specific to the serving hardware in use.

Key Take-aways

- We introduce TiDAR, a sequence-level hybrid architecture that attempts to take advantage of both AR quality and diffusion efficiency by leveraging the free token slots on GPUs.
- We show that with our architecture and modified training mask, TiDAR can significantly close the gap from AR with decent speedup.
- TiDAR outbeats open-source DLLMs like Dream and Llada in both quality and speedup.
- TiDAR supports exact KV cache without extra computation and is easy to compute likelihood
- We’re currently working on addressing some of the limitations.

Thanks!

- Let’s keep connected!

Wechat Linkedin Website TiDAR

Appendix: NFE Distribution Analysis

- HE/MBPP/GSM8K