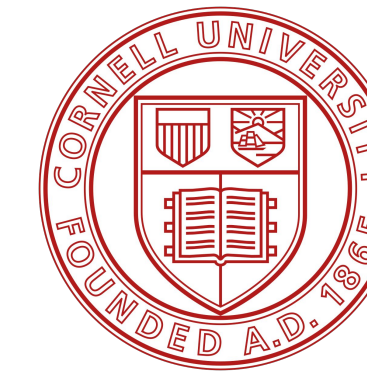


Block Diffusion: Interpolating between Autoregressive and Diffusion Language Models

Marianne Arriola¹, Aaron Gokaslan¹, Justin T. Chiu², Zhihan Yang¹,
Zhixuan Qi¹, Jiaqi Han³, Subham Sahoo¹, Volodymyr Kuleshov¹

Cornell Tech¹, Cohere², Stanford University³



1 AR or Diffusion for language? Better together!

Autoregression

Generation

They're

They're speculating

They're speculating about ...

✗ Sequential
✗ Causal context only
✓ High quality
✓ Flexible-length
✓ KV caching

Diffusion

The diagram illustrates the evolution of a sentence over three generations of a language model. A vertical arrow on the left, labeled "Generation", points downwards from the top row to the bottom row. The three rows show the following sentences:

- Generation 1: Hirsh need account
- Generation 2: Hirsh need to account data to released.
- Generation 3: Hirsh will need to take account data that's to be released.

☒ Parallel
 ☒ Global context
 ☐ Low quality
 ☐ Fixed-length
 ☐ No KV caching

Block Diffusion (Ours)

Generation
↓

Anatoly well for

Anatoly is well-known for his plays, like

Anatoly is well-known for his witty chess plays, like bold Gambit...

- ✓ Parallel
- ✓ Semi-global context
- ✓ High quality
- ✓ Flexible-length
- ✓ KV caching

2 Unifying AR and Diffusion Objectives

$$-\log p_\theta(\mathbf{x}) = -\sum_{b=1}^B \log p_\theta(\mathbf{x}^b | \mathbf{x}^{<b}) \leq \sum_{b=1}^B \mathcal{L}_{\text{diff}}(\mathbf{x}^b | \mathbf{x}^{<b}; \theta)$$

Clean context $\mathbf{x}^{<b}$

Anatoly is well-known for

Clean block \mathbf{x}^b

his witty chess plays, like

BERT-style training within each block

1) Mask with probability $t \sim [0, 1]$

2) Unmask

$$p_{\theta}(\mathbf{x}^b | \mathbf{x}_t^b, \mathbf{x}^{<b})$$

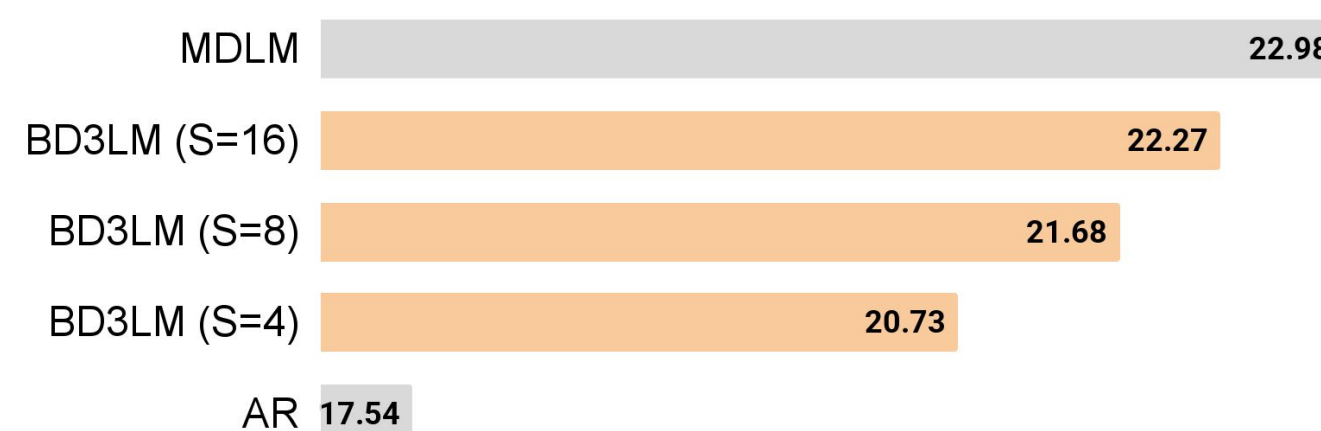
Diffusion loss $\mathcal{L}_{\text{diff}}$

$$\mathbb{E}_{t, \mathbf{x}_t \sim q} -\frac{1}{t} \log p_{\theta}(\mathbf{x}^b | \mathbf{x}_t^b, \mathbf{x}^{<b})$$
Noised block \mathbf{x}_t^b

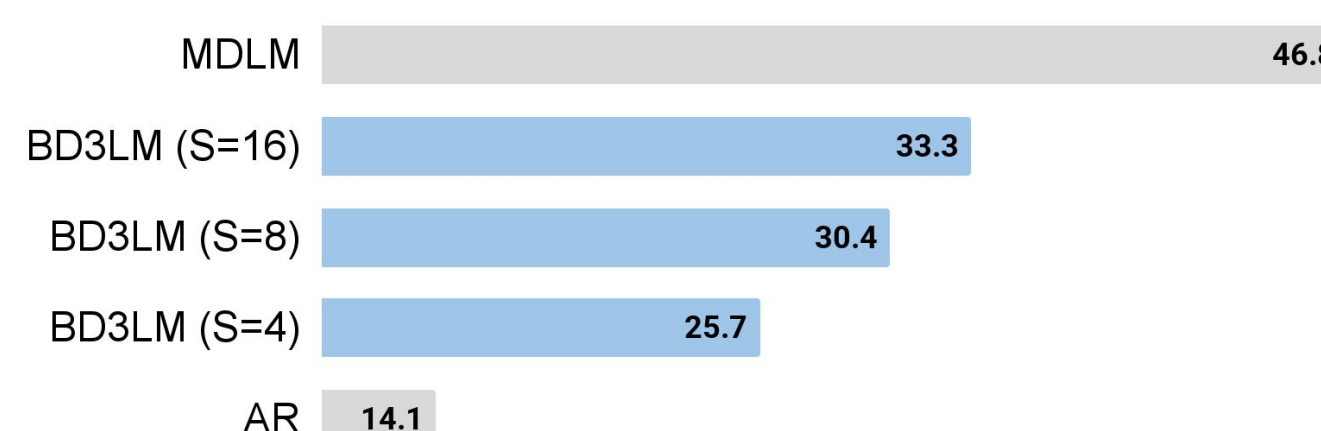
5 SOTA Likelihoods & Long Document Gen.

S = Block size

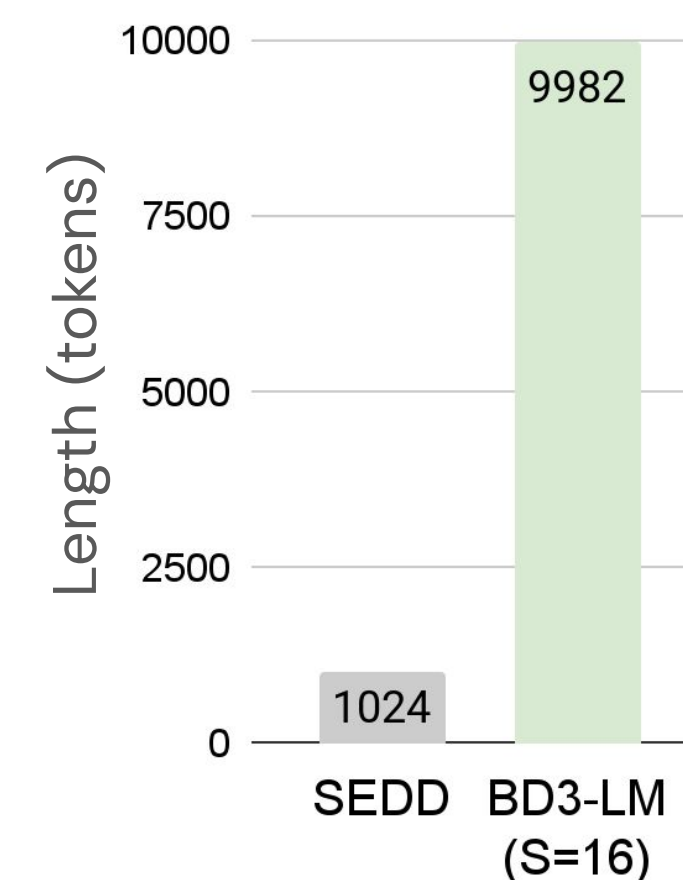
Test Perplexity (↓) on OpenWebText



Perplexity (↓) of Generated Samples under GPT2



Maximum Document
Generation Length (↑)



4 Arbitrary-Length Generation + KV caching

Block 1

Samples: At [?] the [?]

Inputs: [?] [?] [?] [?]

Sampler

x Generation steps

Block 2

Samples: told the [?] that

Inputs: [?] [?] [?] [?]

Sampler

x Generation ... steps

Accumulated Samples: At 9AM, the coach

3 Predict all blocks in 1 pass

Goal: Predict B blocks

Block 1

Diagram illustrating a Transformer model architecture. The input sequence, labeled "Inputs", consists of a question mark icon and the text "World!". These inputs are processed by the "Transformer" block. The output of the Transformer is a bar chart labeled "Predictions", showing three bars of varying heights.

Block 2

The diagram illustrates a Transformer encoder. A green bar labeled "Transformer" receives input from a sequence of tokens: "Hello", "World!", "Today", and a question mark "?". The tokens "Hello" and "World!" are in a blue box, while "Today" and "?" are in an orange box. Arrows point from each token to the Transformer bar. Above the Transformer bar, a small bar chart shows the output representation for each token, with the representation for the question mark being the highest.

Solution: Use 1 pass with a custom attention pattern

Diagram illustrating the input to a Transformer model. The input consists of three types of blocks:

- Noised blocks** (orange): "World! Today" followed by three masked tokens (represented by question marks).
- Clean blocks** (blue): "Hello World! Today I'm going to".
- Transformer** (green): The model processes the input, with three histograms above it showing the distribution of tokens.

The input sequence is represented as a sum of vectors: $\mathbf{x}_{t_1}^1 \oplus \dots \oplus \mathbf{x}_{t_R}^B$.

Attention Pattern

