Presenting The Duo Framework – Diffusion Duality for Language Modeling

Introduction: The State of Discrete Diffusion

What are USDMs?

Uniform-state Discrete Diffusion Models (USDMs) treat text generation as a Markov process where tokens "flip" into a uniform noise state.

Advantages over Autoregressive (AR):

Unlike AR models that generate tokens sequentially (slow), USDMs generate sequences in parallel.

Advantages over Masked Diffusion Language Models (MDLM):

Unlike MDLMs, USDMs have the ability to self-correct.

The Current Gap:

Despite their potential, USDMs have historically fallen behind in quality compared to AR models and MDLMs.

The One-Hot Advantage: Breaking the Embedding Bottleneck

Previous Approaches:

Early attempts at continuous-time diffusion for text injected Gaussian noise into **pre-trained embedding vectors**.

The Embedding Trap:

Relying on fixed embeddings limits the model to a pre-defined geometric space, which may not be optimal for the specific diffusion process and lacks a formal mathematical proof for "word-flipping" logic.

One-Hot Duality:

Duo operates directly on **one-hot token representations**.

Why it Wins:

This approach allows the model to learn its own semantic representations dynamically during the diffusion process. Crucially, it enables the use of the **Discrete NELBO**, which is mathematically proven to be a "tighter" bound for text than Gaussian MSE, leading to superior perplexity.

Theoretical Core: The Diffusion Duality Proof

The Breakthrough:

The authors provide a formal proof that the discrete marginal distributions of a USDm are exactly equivalent to the argmax of a continuous Gaussian diffusion process.

The Bridge:

The **Diffusion Transformation Operator** T mathematically synchronizes the continuous signal strength at with the discrete state transition rate at.

Practical Impact:

Because the two worlds are dual, we can train the model in a continuous space while optimizing for discrete token accuracy, effectively "smoothing" the learning landscape.

Optimization: High-Efficiency Training

Loss Function:

$$L_{\text{train}} = \mathbb{E}_{\mathbf{x},t \sim U[\beta,\gamma],\tilde{q}_t} \sum_{\ell\in[L]} f_{\text{Duo}}\left(\mathbf{z}_t^{\ell} := \arg\max(\mathbf{w}_t^{\ell}), \mathbf{x}_\theta\left([\text{softmax}(\mathbf{w}_t^{\ell}/\tau)]_{\ell'=1}^{L}, t\right), \alpha_t := \mathcal{T}(\tilde{\alpha}_t); \mathbf{x}^{\ell}\right).$$

1

Rao-Blackwellized NELBO:

To reduce GPU memory and training time, Duo uses an improved loss function fduo that analytically computes noise expectations, significantly reducing gradient variance.

2

Curriculum Learning:

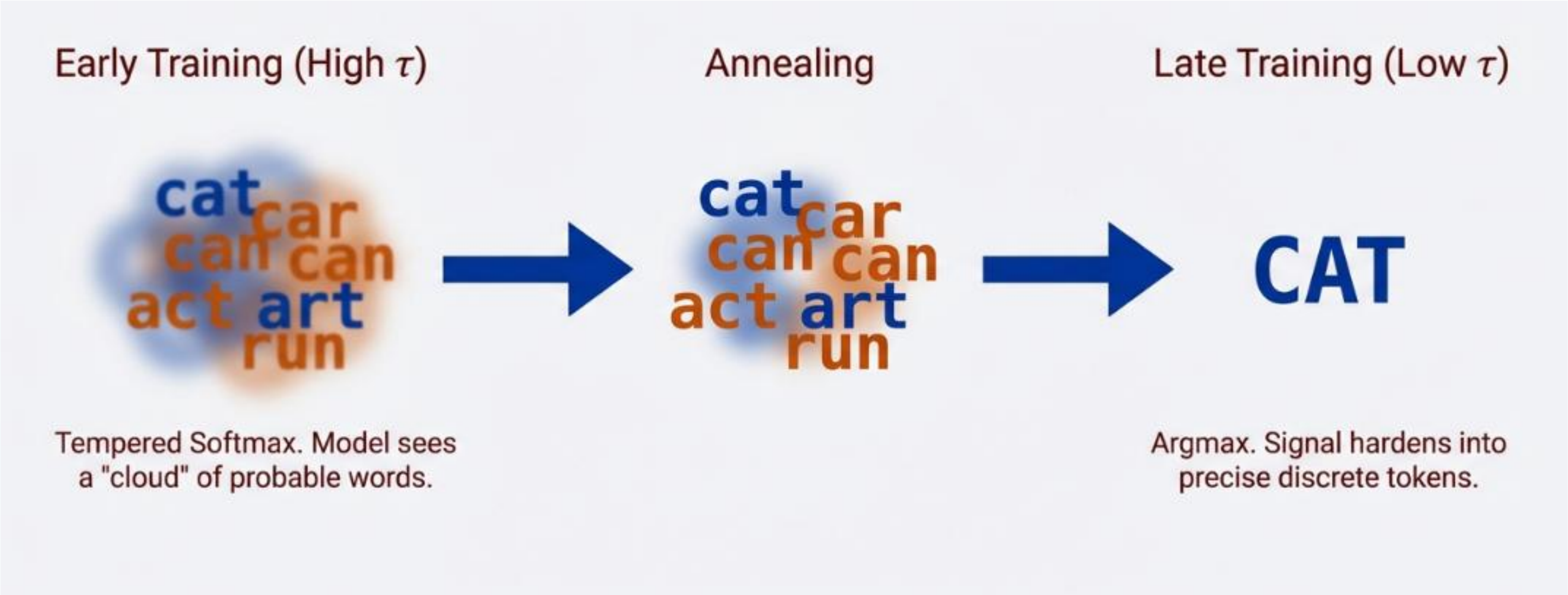
The training begins with a **tempered softmax** relaxation. Early on, a high temperature τ allows the model to see a "soft" blend of all words, preventing the signal loss that occurs with a "hard" argmax in large vocabularies.

3

Preserving the Signal:

This relaxation is critical because even 15% noise in a 30,000-word vocabulary can cause a "hard" argmax to flip randomly. The softmax allows the model to still "see" the correct word as a high-scoring runner-up.

Curiculum Training Visualization



Generation: From Markov Jumps to ODE Flow

The Speed Problem:

Traditional discrete diffusion requires hundreds of Markov steps to generate a sentence.

Probability Flow ODE:

The Duality proof allows us to view the reverse process as a deterministic ODE flow.

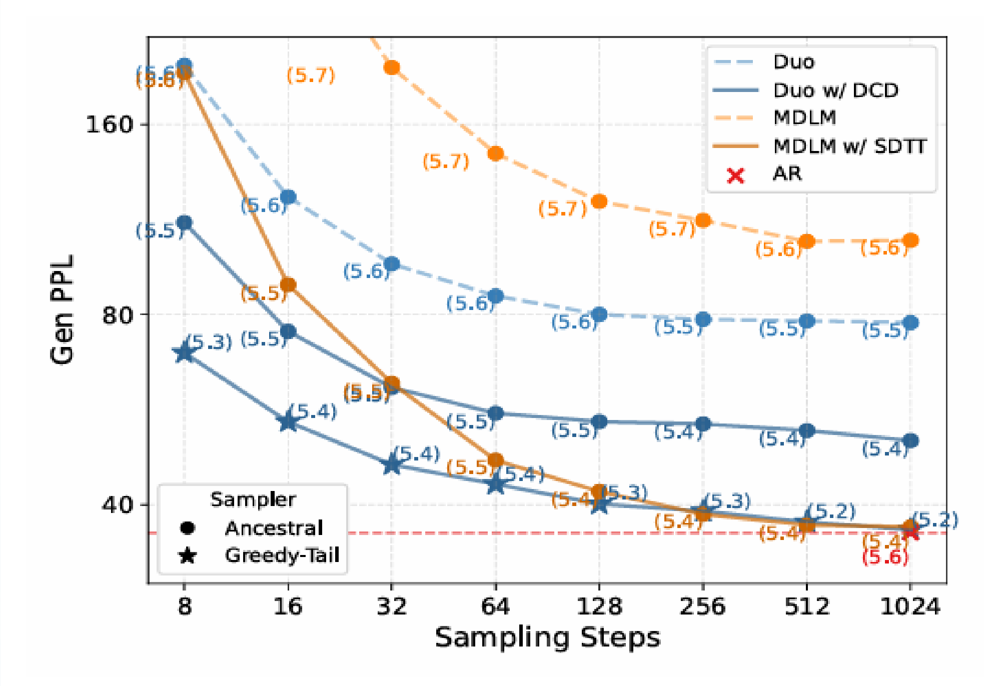
Consistency Distillation:

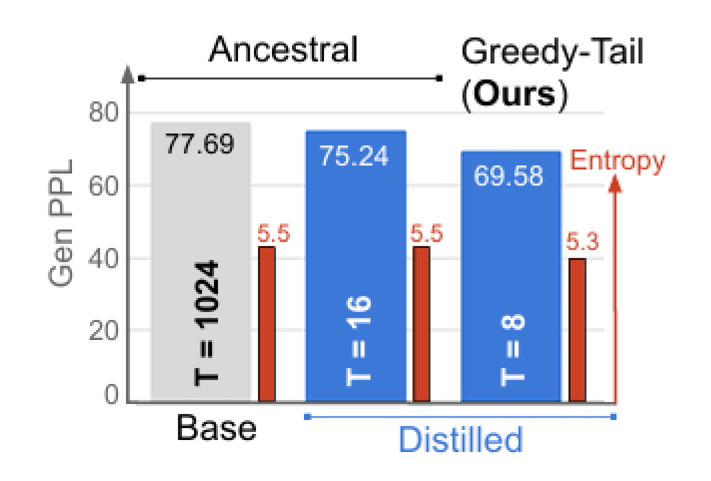
Duo applies Consistency Distillation to "shortcut" the path. This allows the model to generate high-quality text in only **1 to 8 steps**, achieving 100x faster sampling with minimal impact on quality.

$$\mathrm{L}_{DCD}(\theta, \theta^-) = \sum_{\ell \in [L]} D_{KL}\left(\mathbf{x}_\theta^\ell(\mathbf{z}_t^\ell, t), \mathbf{x}_{\theta^-}^\ell(\mathbf{z}_s^\ell, s)\right)$$

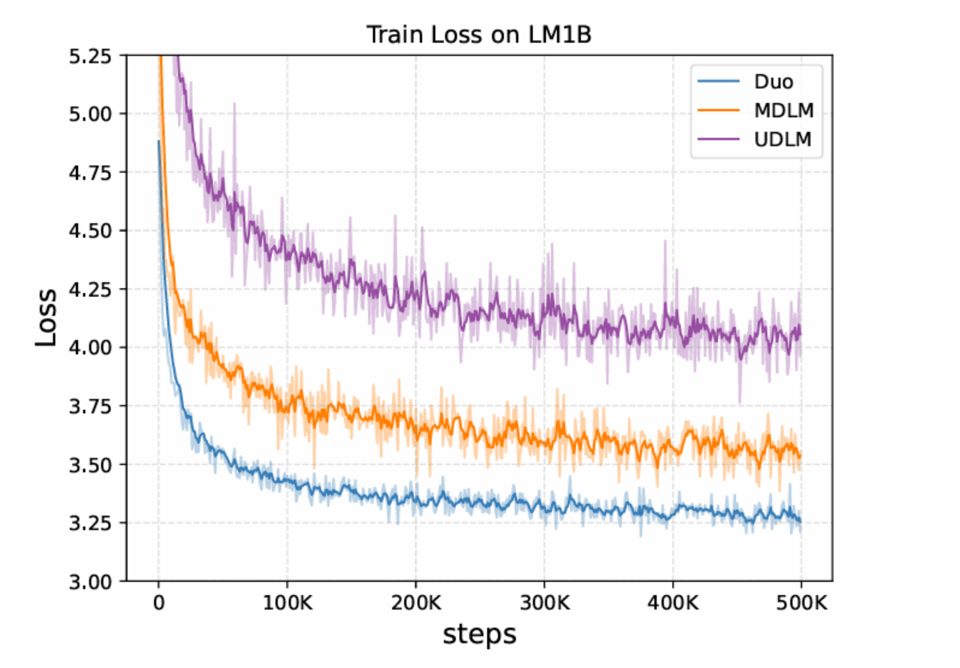
Performance: Benchmarks and Beyond

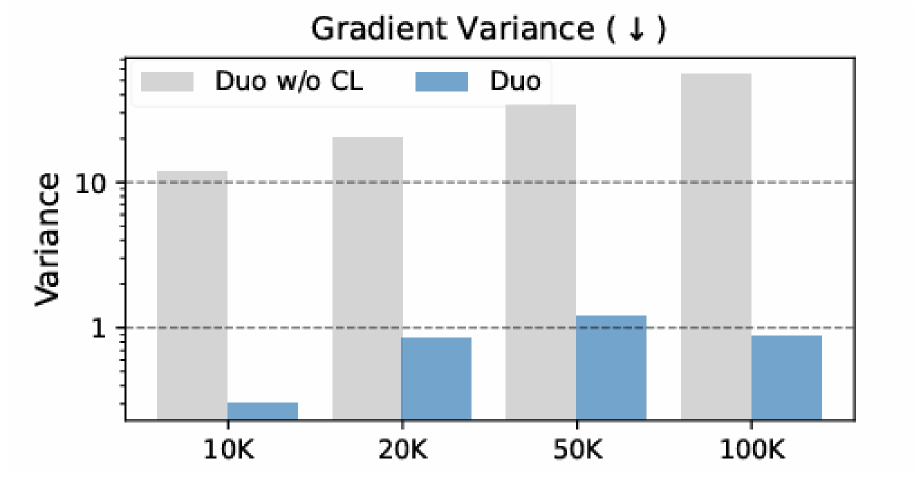
Sampling results:





Curiculum Training Impact:





Technical Details

Model Architecture (DUO)

- Transformer-based diffusion model (DiT)
- 12 layers, hidden size 768, 12 attention heads
- 170M parameters
- 128-dimensional time embedding
- Rotary positional encoding
- Adaptive LayerNorm conditioned on diffusion time
- No weight tying between input and output embeddings

Training Configurations

- Hardware: 8× NVIDIA H100 GPUs
- Precision: bfloat16 forward passes
- Optimizer: AdamW
- Batch size: 512
- Learning rate: 3×10^{-4}. 2,500-step warmup, then constant
- Dropout: 0.1
- Trained on OWT and LM1B for 1M steps

Implementation Plan

1

Paper Understanding & Method Analysis

- Study the theoretical formulation of the diffusion objective
- Understand the discrete Gaussian diffusion duality
- Map equations to implementation components
- Identify essential vs optional components of the method
- Clarify evaluation metrics and baselines

2

Dataset & Computational Resources

- Dataset: A subset of OpenWebText (OWT) dataset
- Resources: NVIDIA T4 GPU from Kaggle's and Colab's free versions

3

Model Training

- Implement a reduced-scale (tiny) Diffusion Transformer
- Train using limited batch size and fewer steps

Implementation Plan

4

Evaluation & Result Visualization

- Monitor validation metrics (NLL, BPD, PPL)
- Compare performance against commonly used models
- Compare results with paper’s results
- Visualize validation perplexity vs steps

5

Methodology Extensions & Improvements

- Test different sequence lengths and tokenization strategies
- Study papers and look for potential techniques to incorporate to this framework

Our Work so far

Model's Training Parameters	
Model Architecture	Tiny Diffusion Transformer (32M Parameters)
Sequence Length	512 Tokens
Training Algorithm	DUO
Training Batch Size	8
Evaluation Batch Size	8
Optimizer	AdamW
Learning Rate	5×10^{-4}

DUO vs AR			
Training Algorithm	BPD	NLL	PPL
DUO	7.5654	5.2439	189.422
AR	8.3271	5.7719	321.162

Tasks Distribution

Name	Task
Spyridon Agathos	Theoretical Comprehension and Paper Research
Konstantinos Leivadas	
Stefanos Rompos	Paper Reproduction and Technical Implementations
Kris Koutsi	

Thank you!

Q & A