Speeding Discrete diffusion models Sampling

Final Project – Generative models

Lecture: Dr. Eliya Nachmani

Students: Yarden Choen, Oriya Sheetrit, Noam Klainer, Ohad Kiperman

Ben-Gurion University of the Negev

Date: 10.12.24

Contents

1 Abstract

Diffusion models have demonstrated remarkable success in modeling continuous data types such as images, audio, and video. However, their application in discrete domains like natural language has remained limited. Recent advancements in discrete diffusion models have enabled progress in handling complex datasets such as natural language and DNA sequences. Nevertheless, these approaches often require hundreds or thousands of denoising steps to achieve sample quality comparable to their continuous counterparts, posing significant challenges to efficiency.

This project builds on recent breakthroughs to address a critical limitation in discrete diffusion models: inference inefficiency. Existing models face challenges in capturing dependencies between output variables during each denoising step, resulting in a high number of inference iterations to achieve optimal performance. Furthermore, the vast search space at each step (reflecting the size of the language) exacerbates these inefficiencies. To overcome these challenges, this study proposes leveraging the Byte Latent Transformer (BLT) framework to create compact latent representations of sequences. These representations will then be utilized within the Simple and Effective Masked Diffusion Language Models (MDLM) model to optimize computational efficiency. The latent representations will be decoded to reconstruct the original sequence.

The proposed sampling procedure focuses on predicting the initial bytes of each sequence patch, enabling parallelized generation of the remaining bytes due to their conditional independence. By integrating these methods, we aim to reduce computational demands while maintaining high generative quality. This research seeks to enhance the scalability of diffusion models, making them faster and more practical for natural language processing tasks.

Relevant Papers

1. **Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution**

   This paper introduces a novel score entropy loss to improve discrete diffusion models. The method achieves significant performance improvements in language modeling tasks, surpassing GPT-2 in zero-shot perplexity and demonstrating better trade-offs between compute and quality. Its emphasis on parameterizing reverse discrete diffusion using data distribution ratios aligns with the goals of efficient sampling for discrete models.

   Reference: Lou, A., Meng, C., & Ermon, S. (2024). Proceedings of the 41st International Conference on Machine Learning.

2. **Latent Diffusion for Language Generation**
   The authors propose using latent space representations for discrete data diffusion. By leveraging pre-trained encoder-decoder models, they achieve superior results in text generation tasks compared to previous diffusion-based approaches. The integration of latent space allows for significant reductions in computational costs while maintaining high-quality generation.
   Reference: Lovelace, J., Kishore, V., Wan, C., & Shekhtman, E. (2023). 37th NeurIPS Conference.

3. **Byte Latent Transformer: Patches Scale Better Than Tokens**
   This paper introduces the Byte Latent Transformer (BLT), a tokenizer-free architecture that matches token-based LLM performance at scale with significant improvements in efficiency. By dynamically segmenting data into patches based on byte entropy, BLT achieves superior scaling trends compared to traditional tokenization methods. The method demonstrates improved robustness and character-level understanding, making it a strong candidate for discrete data modeling.
   Reference: Pagnoni, A., Pasunuru, R., Rodriguez, P., et al. (2024). Meta AI Research.

4. **Scaling Diffusion Language Models via Adaptation from Autoregressive Models**
   This work explores adapting autoregressive language models to diffusion models, providing a scalable approach to bridge the gap between these paradigms. By using attention mask annealing and leveraging pre-trained AR models, it achieves competitive results with reduced training costs. This method's adaptability makes it relevant for extending discrete diffusion models.
   Reference: Gong, S., Agarwal, S., Zhang, Y., et al. (2024). ArXiv Preprint.

5. **Simple and Effective Masked Diffusion Language Models**
   This paper introduces a simplified framework for masked diffusion language models (MDLM), demonstrating that simple masked discrete diffusion models can achieve state-of-the-art performance in language modeling tasks. The authors propose an effective training recipe and derive a simplified, Rao-Blackwellized objective that significantly improves performance. The MDLM framework achieves strong results on language modeling benchmarks, approaching the performance of autoregressive models and showcasing the potential of masked diffusion methods for discrete data.
   Reference: Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., & Kuleshov, V. (2024). NeurIPS Proceedings.

2 Problem Description

Masked Diffusion Language Models (MDLM) have emerged as a powerful framework for generative modeling of discrete data. These models transform data into a noisy state and then reconstruct it step-by-step during inference.

2.1 Noising Process

The noising process gradually transitions data into a "mask state" (m) using a Markov process. At each step, the data either remains in its current state with probability $1 - \beta_i$ or transitions to the mask state with probability β_i. Over time, as defined by a masking schedule $\beta(t)$, the probability of the data being fully masked approaches 1. This process can also be extended to a continuous-time framework, enabling smoother transitions between states.

2.2 Denoising Process

The denoising process reverses the noising operation to reconstruct the original data. A neural network predicts conditional probabilities $p_\theta(x_s|x_t)$, progressively unmasking the data step-by-step. Each unmasking step depends on the outcomes of previous steps, ensuring sequential dependencies. Ancestral sampling is employed to gradually restore the original data from the mask state, maintaining consistency throughout the reconstruction process.

2.3 Challenges in Accelerating Inference

Discrete diffusion models face significant challenges in reducing inference time due to their inherent characteristics. Unlike continuous diffusion, where noise evolves smoothly, discrete diffusion involves abrupt transitions to a "mask state," making intermediate states poorly defined and unsuitable for approximations like step skipping. This categorical nature also causes a substantial loss of information about the original data during noising, necessitating precise step-by-step inference to reconstruct the data accurately. Techniques such as teacher-student distillation, effective in continuous models, struggle in discrete diffusion due to the absence of smooth mappings between noisy and clean states. Similarly, consistency models, which directly map noisy states to clean data, are incompatible with the abrupt, discontinuous transitions of discrete diffusion. Moreover, training and sampling in discrete diffusion require modeling complex state transitions across steps, making the process computationally intensive and error-prone.
These challenges—sequential dependency, abrupt transitions, and masking-induced information loss—highlight the need for novel strategies tailored to the unique dynamics

of discrete diffusion models to accelerate inference while maintaining accuracy.

3 Chosen Method for Solving

To address the inefficiencies of discrete diffusion models, this project proposes a solution that combines the Byte Latent Transformer (BLT) for latent representation and latent discrete diffusion for faster inference.

3.1 Preprocessing

- **Entropy-Based Patching**: The Byte Latent Transformer introduces an entropy-based segmentation mechanism to dynamically create variable-sized patches of bytes. This preprocessing step allocates computational resources to high-entropy regions of data, optimizing the model's ability to focus on complex areas while simplifying predictable regions.

3.2 Architecture

- **Byte Latent Transformer (BLT)**:
  - BLT is a tokenizer-free architecture designed to process raw byte data directly, bypassing tokenization overheads.
  - It uses lightweight local transformers to encode raw bytes into patches, which are then processed by a larger latent transformer for context-aware reasoning.
  - This dynamic patching scheme allows the model to scale efficiently, simultaneously increasing patch and model size without increasing inference costs.
- **Latent Discrete Diffusion**:
  - Inspired by latent diffusion for continuous data, this approach operates in a compressed latent space rather than directly modeling discrete variables.
  - A compression network transforms variable-length encoder outputs into fixed-length latent representations, reducing the dimensionality and computational requirements for subsequent diffusion steps.
  - A reconstruction network maps the latent outputs back to a higher-dimensional space for decoding into the final data format.

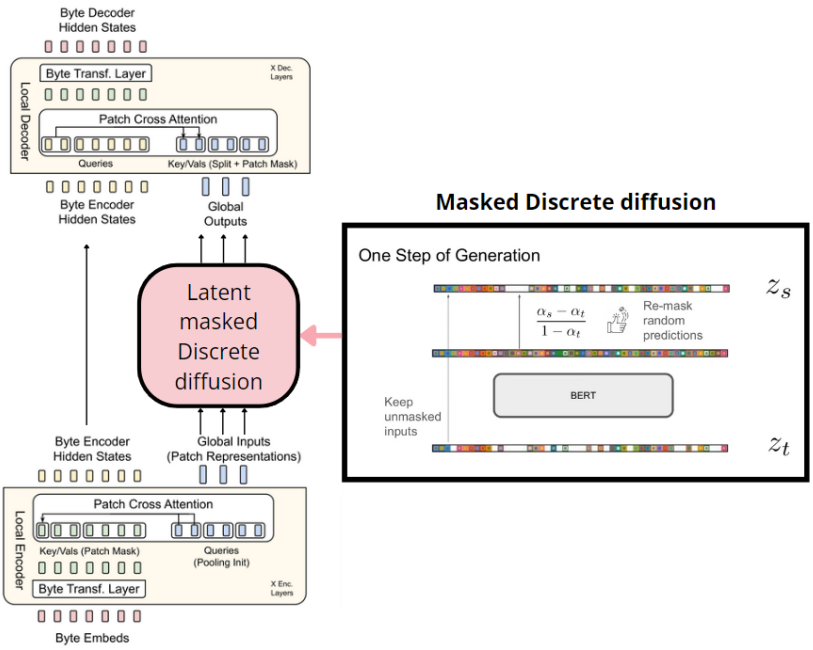

Figure 1: In our proposed model, we replace the Latent Transformer with a Latent Masked Discrete Diffusion module.

3.3 Loss Function

Our approach integrates concepts from both the *Byte Latent Transformer (BLT)* and the *Simple and Effective Masked Diffusion Language Models (MDLM)*.

- **Byte Latent Transformer (BLT):** The BLT framework introduces an entropy-based segmentation strategy, dynamically dividing data into variable-sized patches based on byte entropy. This patch-based architecture enables efficient processing and improved scalability. The model optimizes the following loss function, focusing on byte-level predictions within each patch:

$$\mathcal{L}_{BLT} = -\sum_{i=1}^{N} w_i \cdot \log p_\theta(x_i | C_i)$$

  where:

  - N is the number of bytes in the sequence,
  - w_i represents entropy-based weights that prioritize high-entropy regions,
  - $p_\theta(x_i|C_i)$ is the probability of byte x_i given the context C_i modeled by the BLT.

  This loss formulation ensures that the model allocates more focus on complex, information-rich regions, enhancing both efficiency and performance.

- **Masked Diffusion Language Models (MDLM):** MDLM employs a continuous-time Evidence Lower Bound (ELBO) as its core objective, designed to approximate the log-likelihood of the data. The ELBO can be expressed as:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q(x_{0:T})}\left[\sum_{t=1}^{T} \log p_\theta(x_{t-1}|x_t) - \log q(x_{t-1}|x_t, x_0)\right]$$

  where:

  - $q(x_{0:T})$ is the forward noising process,
  - $p_\theta(x_{t-1}|x_t)$ represents the model's prediction for the denoised token at time $t-1$,
  - T is the total number of diffusion steps.

  To simplify training, this ELBO is transformed into a weighted cross-entropy loss:

$$\mathcal{L}_{MDLM} = \sum_{t=1}^{T} \lambda_t \cdot \mathrm{CE}(x_0, \hat{x}_t)$$

  where:

  - λ_t are time-dependent weights emphasizing critical diffusion steps,
  - CE denotes the cross-entropy between the original data x_0 and the model's prediction $\hat{x}_t$.

  This Rao-Blackwellized objective reduces variance during training and enhances model stability, allowing for more efficient learning in discrete settings.

3.4 Inference

Our inference framework leverages the synergistic combination of BLT's patching mechanism and latent diffusion to achieve efficient sampling while maintaining generation quality.

3.4.1 Parallel Generation

The model implements a two-tier parallelization strategy:

- **Patch-Level Processing:** Multiple entropy-based patches are processed concurrently through the latent transformer
- **Intra-Patch Generation:** Conditionally independent bytes within patches are generated simultaneously

3.4.2 Adaptive Sampling

We aim to employ an entropy-guided sampling schedule that:

$$T_{steps}(p) = \max\{\lceil \alpha \cdot H(p) \rceil, T_{min}\} \tag{1}$$

where $T_{steps}(p)$ is the number of denoising steps for patch p, $H(p)$ represents patch entropy, α is a scaling factor, and T_{min} is the minimum step count.

3.4.3 Latent Space Operations

The inference process operates primarily in a compressed latent space $\mathcal{Z}$, where:

$$z_t = E(x_t) \in \mathcal{Z}, \quad x_t = D(z_t) \tag{2}$$

with E and D representing the encoder and decoder networks respectively. This approach reduces computational complexity while maintaining generation fidelity.

4 Novelty of Method with Respect to the Literature

Previous research has introduced the concept of leveraging latent spaces within discrete diffusion models, highlighting their potential for handling discrete data effectively. Additionally, recent advancements have explored the use of the Byte Latent Transformer (BLT), a tokenizer-free architecture, for processing raw byte-level data.
Our project seeks to integrate these two fields into a unified and novel framework. Specifically, the proposed model incorporates BLT for efficient processing of raw byte data but diverges from prior approaches by employing a discrete diffusion network, rather than a Latent Transformer, for next-patch prediction. This combination represents an innovative approach, merging the strengths of discrete diffusion and tokenizer-free architectures to address the challenges of discrete data modeling.

5 Datasets

For this project, we will use the **WebText** and **OpenWebText** datasets.

- **One Billion Words Dataset**: This dataset consists of a large collection of news articles, providing diverse, high-quality text data suitable for language modeling tasks. It is widely used for benchmarking language models due to its extensive vocabulary and varied content.

- **OpenWebText**: As an open-source alternative to WebText, OpenWebText replicates the methodology of the original dataset and makes it accessible to the community. This dataset ensures a broad coverage of text content, maintaining a similar level of quality to WebText.

Both datasets represent state-of-the-art benchmarks for natural language processing tasks and are particularly well-suited for training and evaluating discrete diffusion models. Their inclusion in this project aligns with the goal of improving the inference time and the quality of model outputs, as they provide rich textual data for fine-tuning and evaluation.

6 Definition of Success and Training Goals

The success of the proposed model will be evaluated by comparing its performance to state-of-the-art diffusion and autoregressive (AR) models, focusing on key quantitative and qualitative benchmarks. The criteria emphasize relative improvements in efficiency, generative quality, and scalability.

6.1 Quantitative Metrics

- **Perplexity**: Achieve lower perplexity score than leading AR and other discrete diffusion models, such as GPT-2, and Simplified Masked Diffusion (SMD) on standard benchmarks like WikiText and OpenWebText.
- **Inference Efficiency**:
  - Reduce the number of denoising steps required for high-quality outputs compared to existing discrete diffusion models.
  - Achieve inference speeds comparable to or better than other models while maintaining generative quality.
  - Optimized computational efficiency by implementing flop-controlled training, allocating fewer floating-point operations (FLOPs) to simpler patches while dedicating more resources to complex ones.

6.2 Specific Achievements

- **Entropy-Based Patching**: Successfully integrate adaptive computation techniques, such as entropy-based patching inspired by Byte Latent Transformer (BLT).
- **Scalable Applicability**: Demonstrate practical scalability by outperforming baseline diffusion and AR models on large-scale tasks, including:

- **Summarization**: Generate concise, coherent, and factually accurate summaries.
- **Storytelling**: Produce diverse and contextually rich narratives, with fewer repeated patterns and higher creativity metrics than baseline approaches.
- **Conditional Generation**: Generate text conditioned on input prompts with higher coherence and relevance compared to both AR and diffusion counterparts.

7 Concerns

This project acknowledges several potential challenges, limitations, and ethical considerations:

- **Technical Challenges**:
  - The integration of Byte Latent Transformer (BLT) with discrete diffusion frameworks may require extensive hyperparameter tuning and optimization to ensure compatibility and performance.
  - Latent diffusion for discrete data is a relatively new approach, and potential instabilities during training and inference may arise.
  - Balancing computational efficiency with generative quality may pose difficulties, especially for tasks with strict performance benchmarks.
- **Limitations**:
  - The reliance on entropy-based patching may lead to suboptimal performance on datasets with uniform complexity or low entropy variance.
  - Scalability to extremely large datasets or domains outside natural language processing may require further modifications to the proposed architecture.

Addressing these concerns will be crucial to ensure the integrity of the proposed solution.

8 Additional remarks

8.1 Complex Data Distribution

Discrete diffusion models are designed to manage discrete data distributions, such as those found in language and DNA sequences.However, the diffusion process on such data can result in substantial changes to the tokens due to the inherent lack of smoothness in the

process. This characteristic makes training and inference more challenging, particularly when using a small number of steps.

8.2 Predict Tokens In Parallel

To minimize the number of steps during the de-noising process, we aim to predict multiple tokens simultaneously at each step. However, in many tasks, it is crucial to account for the dependencies between tokens. This requirement often necessitates additional steps to generate a coherent and clear sequence. Moreover, in tasks such as sentence completion, these dependencies further complicate the process.

8.3 Skipping Steps Strategies

Continuous diffusion models benefit from advanced numerical solvers that approximate the reverse process effectively with reduced steps. These solvers enable skipping strategies while preserving sample quality. Discrete diffusion models lack such solvers, as their step-wise framework does not directly support integration or interpolation.

9 Inputs/Output of the final training model

Input/Output of the Final Trained Model

The final trained model, operates as follows:

- **Input:** The model accepts a partially masked or noisy sequence of categorical variables (e.g., text tokens). These sequences can represent natural language or other discrete data types (e.g., amino acid sequences). The input includes a context sequence for conditional generation or a completely unconditioned sequence for unconditional generation. Each token in the sequence may be masked, allowing the model to predict missing or noisy tokens step-by-step.
- **Output:** The model generates a fully reconstructed sequence where the masked or noisy tokens are replaced with high-quality predictions. For natural language tasks, the output consists of coherent and contextually appropriate text sequences.

10 Experiments

The first step we took was to run the model presented in the paper "Simple and Effective Masked Diffusion Language Models." The results we obtained are illustrated in Figures 2 and 3. Figure 2 shows the training loss over time. As observed, the loss starts at a high value of around 10 and decreases sharply during the initial training iterations. After approximately 5,000 steps, the loss stabilizes and continues to decline gradually, reaching a value close to 4. This pattern indicates that the model effectively learns from the data, reducing the error rate as training progresses. Figure 3 presents the validation metrics: bits per dimension (bpd), perplexity (ppl), and negative log-likelihood (nll). All three metrics exhibit a consistent downward trend, reflecting improved model performance during validation. Figure 4 presents an example of text generated by the MDLM trained model. Overall, these results indicate effective learning dynamics, with both training and validation metrics showing clear improvement as the model converges.

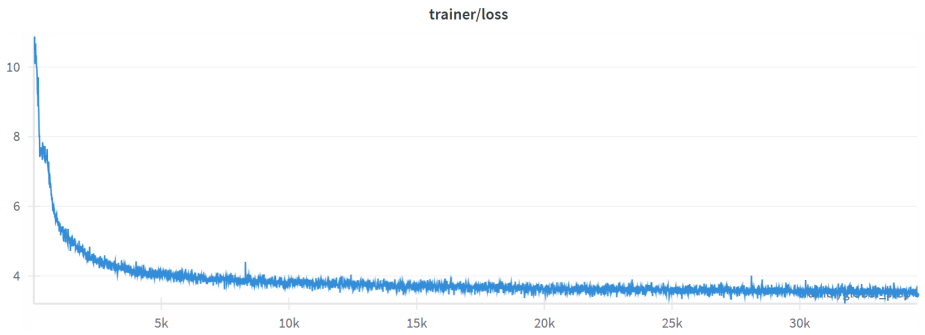

Figure 2: mdlm training

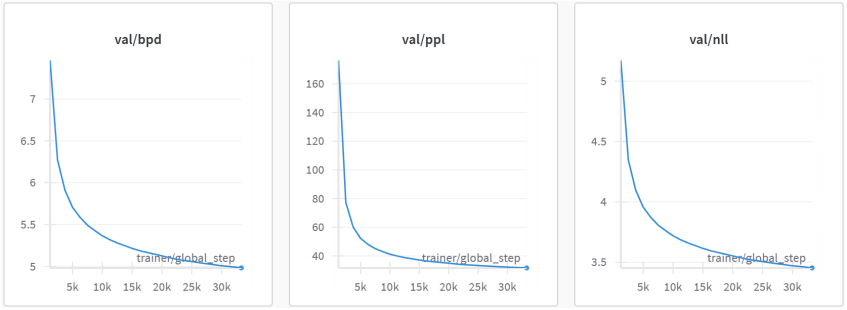

Figure 3: mdlm validation

```
When I was four, the family had worked hard during its childhood. We
were one too much of the loved ones, but my youngest son survived. And
I started to realize there were some issues with my writing. It was the
first time I started really writing. John was a lawyer, and my father
was fine. Those were the few things that I needed beforehand, what
prevented me from writing books. But what liked the idea of writing a
novel?

Well, at least I was a writer, because I liked to make them work. But
yeah, I wrote the other two books, and my wife and I were in between at
a different time in life. The reason about it is that they didn't
really decided. I didn't think any other writer would make a major
decision or decision. John felt that it was the fault of my father,
```

Figure 4: Generated text using MDLM model

We attempted to reproduce the model from *Byte Latent Transformer: Patches Scale Better Than Tokens*, but faced significant challenges due to missing critical code components and a complex, hard-to-interpret codebase. Additionally, the lack of access to the large datasets and pre-trained models mentioned in the paper made replication difficult. These issues, combined with the extensive computational resources required for training, created major technical and data-related obstacles. **Therefore, we decided to explore a different approach.**

11 Change of Direction

Our initial investigation into accelerating discrete diffusion inference revealed several fundamental challenges:

- **Categorical Nature** – Discrete diffusion's abrupt state transitions prevent effective step-skipping and intermediate state approximation
- **Sequential Constraints** – Step-wise reconstruction requirements impede parallelization efforts
- **Optimization Barriers** – Traditional acceleration techniques (e.g., teacher-student distillation, consistency models) prove ineffective due to discontinuous state transitions

These limitations prompted us to explore an alternative approach leveraging multimodal conditioning for efficient sampling.
We propose accelerating discrete sequence generation through multimodal conditioning. Our approach introduces a secondary modality to provide structural guidance, enabling parallel token prediction while maintaining sequence coherence.

11.1 Multi-Flow Architecture

The framework builds upon Multi-Flow models (Campbell et al. 2024), integrating:

- **Factorized Flows** – Separate but coupled continuous and discrete flows for different modalities
- **Joint Generation** – Simultaneous handling of structural and sequential information while preserving cross-modal dependencies

With Multi-Flow, we hypothesize that incorporating structural information as an additional conditioning signal will enable more parallelized token prediction in sequence generation. To validate this approach, we will first test it on protein sequence generation, where structure serves as the guiding modality. Specifically:

1. Develop a student-teacher framework for sequence prediction with reduced inference steps
2. Evaluate sampling efficiency and generation quality through perplexity and structural metrics
3. Extend methodology to broader applications (e.g., text generation with image conditioning)

By moving away from purely discrete diffusion methods and embracing multimodal flow-based learning, our project aims to develop a generalizable framework for fast and efficient sequence generation across different modalities.

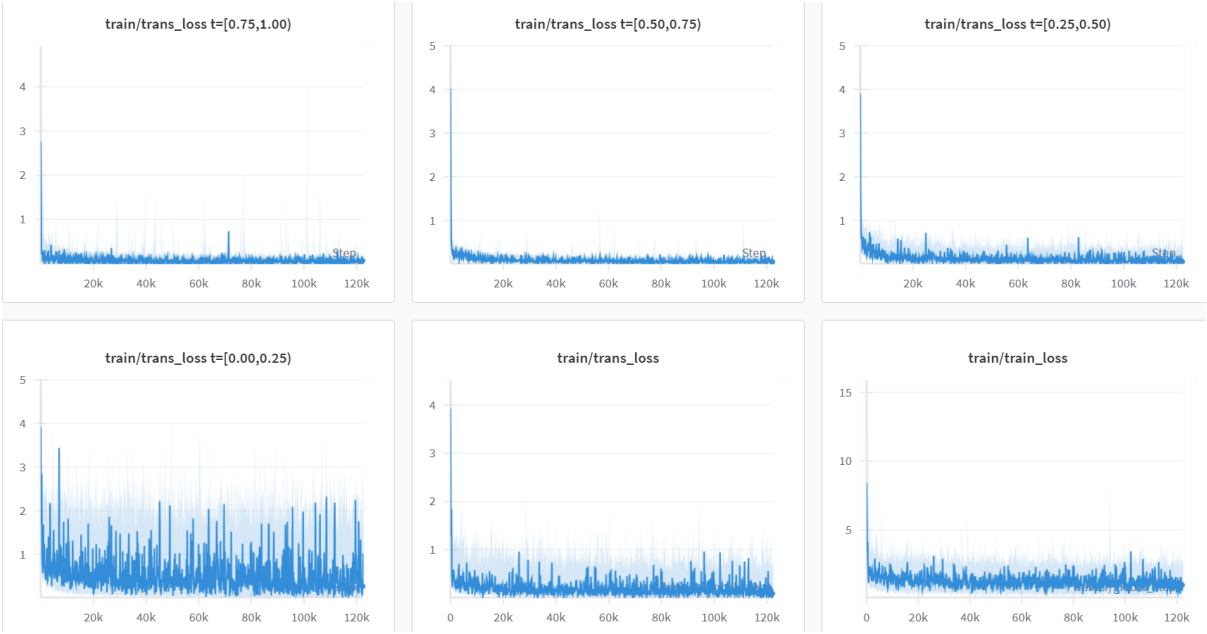

Figure 5: Multi-Flow training

The graphs in Figure 5 illustrate the training loss curves for different time intervals (t) within the Multi-Flow model.

While we successfully managed to run the basic model, we faced challenges in implementing our modifications. Unfortunately, due to the extensive time spent troubleshooting the previous idea, we did not have enough time within the course timeline to fix the issues in the new model and achieve final results.

11.2 Conclusion

In this project, we delved into the innovative and rapidly evolving field of discrete diffusion models. We identified key limitations in the current landscape of discrete diffusion models, particularly regarding inference inefficiency and computational constraints. To address these challenges, we initially proposed an innovative approach based on the *Byte Latent Transformer* framework. However, this idea proved problematic due to significant computational issues, including missing critical code components and the absence of accessible pre-trained models and datasets, which limited our progress.

Recognizing these obstacles, we shifted our focus to a new and promising direction—leveraging a multimodal acceleration framework through the Multi-Flow architecture. This approach introduces structural conditioning to enhance sequence generation efficiency. While we successfully managed to run the basic model, we encountered difficulties in implementing our modifications. Unfortunately, the extensive time spent troubleshooting the initial idea limited our ability to fully resolve these issues and achieve conclusive results within the course timeline.

Despite these challenges, we believe that our final approach shows great potential. Further research is needed to fully explore its advantages and limitations, particularly in optimizing performance and scalability.