Final Report
MambaDiff: Revolutionizing Seq2Seq Models with Diffusion Model and Mamba Architectures

Guannan He, Xiangyu Zhang, Xinyu Chang
Johns Hopkins University
ghe10@jh.edu xzhan344@jh.edu xchang23@jh.edu

Abstract

In this work, we introduce MambaDiff, a novel architecture that integrates the strengths of diffusion models with the Mamba architecture to address the challenges faced in sequence-to-sequence (Seq2Seq) modeling tasks. Traditional Seq2Seq models, while powerful, often struggle with maintaining contextual relevance and computational efficiency, especially over longer sequences. By leveraging the gradual refinement capabilities of diffusion models and the scalable, efficient processing of the Mamba architecture, MambaDiff aims to enhance the quality of generated text across extensive contexts significantly. We demonstrate the potential of our approach through diverse NLP tasks such as open-domain dialogue, question generation, text simplification, and paraphrasing. This paper will detail the motivation behind this integration, the theoretical and practical benefits, and preliminary results that highlight its effectiveness compared to existing models.

1 Motivation

In the field of natural language processing (NLP), generating contextually relevant and fluent text, especially over long sequences, poses significant challenges [4]. Existing models often struggle with maintaining coherence and computational efficiency simultaneously. To tackle these issues, our project proposes an integration of the DiffuSeq diffusion model with the Mamba architecture [4]. This combination aims to overcome the limitations of current NLP models by ensuring high-quality text generation across extended contexts. The motivation for this project is to improve the capabilities of text generation systems, making them more efficient and effective at understanding and generating human-like text, and to apply the improved models on seq2seq tasks.

2 Related Work & Literature Review

DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models [3]

This paper presents a new approach to language modeling using diffusion models. It adds noise to the input data in the forward pass and employs a denoising process in backpropagation, resulting in coherent and contextually relevant text. The iterative refinement technique used improves the fluency and usefulness of the text generated.

Mamba: Linear-Time Sequence Modeling with Selective State Space [4]

This paper proposes a unique architecture combining selective state space models, including the H3 model and a Gated Multi-Layer Perceptron (MLP) framework, with a selective mechanism for

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

optimized input filtering. It introduces a hardware-aware parallel algorithm for improved handling of long-context inputs in recurrent mode. The architecture outperforms traditional models like Transformers, RetNet, and H3++, demonstrating superior efficiency in processing complex input sequences.

3 Methods & Models

We plan to merge DiffuSeq models with the Mamba architecture. This method combines the gradual improvement process of diffusion models with the fast and adjustable structure of Mamba. Our goal is to make this combined model better at learning and creating high-quality outputs quickly. We will adjust the diffusion process to fit Mamba's way of working, focusing on how to add and reduce noise effectively within this new framework. This approach aims to improve performance in generating text and images, among other tasks. Our tests will check if this new model can outdo current methods in terms of quality, speed, and flexibility with different data types.

3.1 Diffusion Process

We employ the diffusion process for paraphrase generation due to its ability to produce diverse outputs. Figure 1 illustrates the diffusion process applied to the Quora Question Pairs dataset. For the input question pairs (original sentence and paraphrase reference), we first embed both parts using an embedding map and concatenate them. Subsequently, we add noise ,through cosine calculation, to the embedding of original sentence while keeping the paraphrase reference fixed as guidance for the reverse process. The forward process establishes the transformation relationship from the original data to noisy data, serving as the foundation for the reverse denoising process. The objective of the backward denoising process is to recover the original data from pure noise by learning the reverse transformation. We train the Mamba model to approximate the back-diffusion process, enabling it to generate high-quality samples from noise.

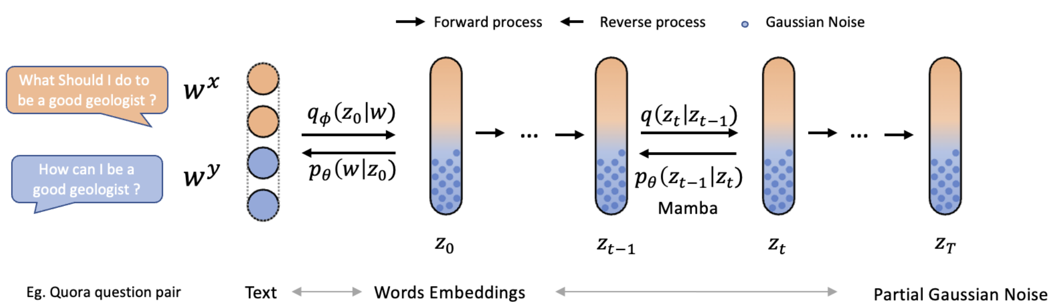


Figure 1: Diffusion Process on QQP dataset

3.2 Inputs & Outputs

We plan to use four distinct datasets for our experiments (See Datasets Section), each serving a different task but uniformly featuring paragraphs or sentences as inputs and generating words or extended sentences as outputs.

4 Hypothesis

We hypothesize that by combining the contextually aware generation capabilities of DiffuSeq with the long-sequence efficiency of the Mamba architecture, we can achieve a marked improvement in generating coherent and contextually relevant text over extended sequences compared to existing models. Specifically, we anticipate that the iterative refinement process of DiffuSeq will enhance the textual relevance and fluency, while Mamba's selective state space utilization will enable efficient memory usage and scalability to longer contexts without performance degradation.

5 Experiments

5.1 Baselines

We plan to use DiffuSeq model, which is the diffusion model plus transformer, as our baseline model to compare results. If we got enough time, we will compare our results also to the baseline models used in the paper of DiffuSeq model[3], such as GRU with attention and transformer [14], GPT2 [13], GPVAE [2], and LevT [7].

5.2 Metrics

We use quality and diversity to evaluate our model. Standard metric BLEU [12] and ROUGE [10] score are applied to evaluate quality. For intra-diversity, we use distinct unigram (dist-1). For sentence-level diversity evaluation, we plan to implement sentence-level self-BLEU [16].

5.3 Datasets

Seq2Seq contains many tasks, we plan to use four popular tasks to test the performance of our model. **Open domain dialogue** necessitates the creation of insightful replies based on the context of the dialogue. The objective of **Question Generation (QG)** is to produce questions when provided with a context. For acquiring ample training examples, **text simplification** involves rephrasing complex text into sequences that are easier to understand, using simpler grammar and vocabulary. The **paraphrasing** task involves crafting an alternative expression in the same language that conveys the same meaning.

Task	Datasets	Training Samples
Open-domain Dialogue	Commonsense Conversation[15]	3382k
Question Generation	Quasar-T[11]	117k
Text Simplification	Wiki-alignment[9]	677k
Paraphrase	QQP[1]	144k

Table 1: Four Datasets

6 Midway Progress

The whole project plan is to merge diffusion models with the Mamba architecture. This method combines the gradual improvement process of diffusion models with the fast and adjustable structure of Mamba. Our goal is to make this combined model better at learning and creating high-quality outputs quickly. We will adjust the diffusion process to fit Mamba's way of working, focusing on how to add and reduce noise effectively within this new framework. And test our combined model on sequence-to-sequence (SEQ2SEQ) text generation tasks: open domain dialogue, question Generation, text simplification, and paraphrasing.

By now, we have reproduced the experiment results of the original diffusion sequence model[4] on Quora Question Pairs(QQP) Dataset [1], and are replacing the Transformer backbone with Mamba blocks.

6.1 Difference between Mamba and Transformer

Transformer[14], equipped with attention layers, has been widely used in the field of sequence modeling due to its efficiency in processing dense information within the context window, and therefore model complex data [4]. However, its ability is limited by a finite context window and exhibits quadratic computational scaling with increasing window length [4].

Conversely, structured state-space sequence models (SSMs) [5][6], traditionally applied in control theory to represent a dynamic system via state variables, have recently demonstrated their utility in sequence modeling. Essentially, SSMs integrate features of Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). However, SSMs generally underperform in modeling highly discrete and dense data, such as text [4]. To address these limitations, Mamba combines the strengths of Transformer and SSMs, thereby retaining the modeling efficacy of Transformers while achieving linear scalability with respect to sequence length.

6.2 Mamba Architecture

6.2.1 Initialization

Mamba employs Kaiming uniform initialization [8] to initialize the model's weights, facilitating the training of exceedingly deep rectified models directly from scratch and enabling the exploration of more extensive and deeper network architectures. Subsequently, the weights are then divided by the square root of the product of the number of residuals per layer and the number of layers to implement the initialization scheme of OpenAI GPT-2.

This modified initialization which accounts for the accumulation on the residual path with model depth. Scale the weights of residual layers at initialization by a factor of $\frac{1}{\sqrt{N}}$ where N is the number of residual layers. The Mamba architecture can be seen in Fig 2. In Mamba, structured SSMs independently map each channel (e.g. $D = 5$) of an input x to output y through a higher dimensional latent state h (e.g. $N = 3$). (Δ, A, B, C) parameters are constant across time (Time-invariant). After discretization, the parameters become time-variant, allowing more efficient computations.

6.2.2 Selective SSMs

The main limitation of SSMs is their computational efficiency, particularly due to the usage of global convolutions which are not selective. Also, SSMs have a tradeoff between expressivity and speed. Larger hidden state dimensions lead to more expressive but slower models. Mamba employs three techniques to overcome these limitations:

Combining Blocks: The architecture simplifies by combining linear attention and MLP blocks into one, inspired by Gated Attention Units (GAUs). This reduces complexity while maintaining performance.

Expanding Model Dimension: The architecture expands the model dimension by a factor to manage the number of parameters efficiently.

Repeating Blocks (Mixer Model Block): The Mixer Model Block, combined with standard normalization and residual connections, forms the basis of the Mamba architecture, which is a stack matching the parameter count of a Transformer's Multi-Head Attention (MHA) and MLP blocks.

6.2.3 Mixer Model Block

In the Mixer Block Model, after embedding the inputs, if the model employs fused addition and normalization, it first integrates the residual with the hidden states. Depending on the type of normalization function utilized, the model then selects the appropriate fusion function—either Root Mean Square Normalization or Layer Normalization—to execute the fused addition and normalization operations.

It is important to notice that Mixer Model Block deviates from the conventional residual and layer normalization sequence, which typically follows the pattern of Layer Normalization -> Attention/MLP -> Add. Instead, the Mixer Model Block adopts a sequence of Add -> Layer Normalization -> Attention/MLP/Mixer. This reordering facilitates the simultaneous return of outputs from both the residual branch (output of Add) and the main branch (output of MLP / Mixer), enhancing the model's capability to integrate and process information effectively. The main reason for this change is performance considerations: addition and layer normalization can be fused together for calculation, thus improving computational efficiency.

6.3 Preliminary results

Our goal is to compare our MambaDiff model with the original DiffuSeq with Transformer block on four tasks. So far, we first choose the widely used QQP dataset, which comes from the community

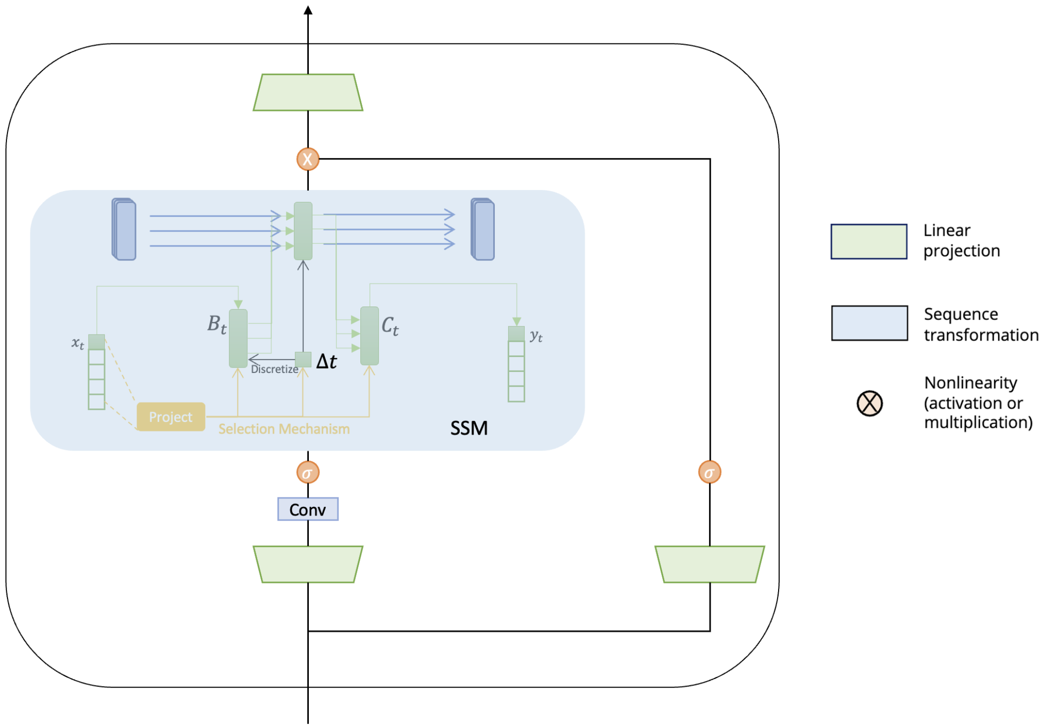


Figure 2: Mamba Architecture

question and answer forum Quora and has 147K positive pairs. We successfully ran through the DiffuSeq model on the QQP data set. We measured gradient normalization, also loss, mse, and negative log likelihood (nll) for training and evaluation sets. We observed that at around 500 steps, the training plateaus and at 1000 step, the evaluation plateaus. The result at step 11000 can be seen in Table 2

	grad norm	loss	mse	nll	eval loss	eval mse	eval nll
DiffuSeq	0.3809	0.0242	0.0242	0.2821	0.0248	0.0248	0.3148

Table 2: Result of DifuSeq on QQP dataset

7 After Midway

7.1 Dataset

Due to the limitation of time, we pick the **Paraphrase** task to test the performance of our model. This task involves crafting an alternative expression in the same language that conveys the same meaning. Quora Question Pair (QQP) dataset has 400k question pairs. Of the 400k pairs, 144k have true labels which have been chosen for training.

Original Sentence: 'How can I be a good geologist?'
Paraphrase Reference: 'What should I do to be a great geologist?'

Table 3: Example of QQP

Referencing Table 3 and Table 3, given the original sentence (as "source"), the output of our model (as "recover") will be used to compare with the paraphrase reference (as "reference") for evaluation.

recover: ’what is the purpose of life?’
reference: ’what’s are the meaning of life? ’
source: ’what’s are the meaning of life? ’

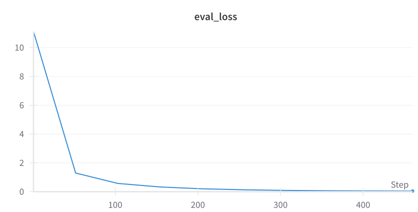
Table 4: Example of Model Output

7.2 Conclusions

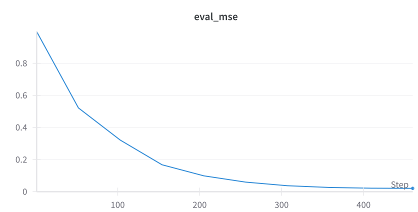
We leverage 32 Mamba blocks as the primary backbone, replacing the conventional transformer structure. These Mamba blocks, characterized by their efficient processing and scalable capabilities, are integral in handling complex patterns and high-dimensional data efficiently. There are around 7 billion parameters used during our training on MambaDiff.

7.2.1 Selected Plots for Paraphrase Task

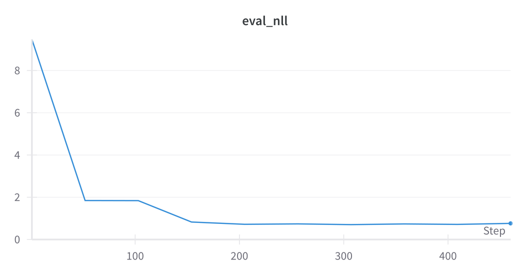
Fig 3 shows the loss, mean square error, and negative log likelihood of MambaDiff on evaluation dataset with 2000 diffusion steps and 97027072 parameters, which is comparable with Transformer block.



(a) Loss in Evaluation Datasets



(b) Mean Squared Error on Evaluation Dataset



(c) Negative Log Likelihood on Evaluation Dataset

Figure 3: Evaluation metrics across different datasets

7.2.2 Results

As recorded in Table 5, the results of 80,000-steps MambaDiff are the best among the three MambaDiff models we have trained. However, there are still gaps when we compare MambaDiff to the base models: DiffuSeq, GRU-attention and GPT2-base FT. However, comparing the results of two models at same step (10k step), we found that our model performs better than DiffuSeq at initial stage, which shows the great potential of our model. Also, from the table it is evident that MambaDiff exhibits an increasing trend with additional steps.

Comparing the results of DiffuSeq with our MambaDiff, it is evident that MambaDiff achieves competitive results while utilizing only one-tenth of the parameters. Additionally, MambaDiff demonstrates improved efficiency, with a runtime of 32 hours compared to DiffuSeq's 108 hours.

Methods	BLEU	R-L	Score	dist-1	Len
Paraphrase					
GRU-attention	0.1894	0.5129	0.7763	0.9423	8.31
GPT2-base FT	0.1980	0.5212	0.8246	0.9798	9.67
DiffuSeq (90 million para.) - 10k steps	0.0004	0.0022	0.2711	0.3056	62.08
DiffuSeq (90 million para.) - 50k steps	0.1921	0.5405	0.8042	0.9717	11.11
MambaDiff (7 million para.) - 10k steps	0.0069	0.0414	0.2937	0.9760	93.03
MambaDiff (7 million para.) - 60k steps	0.0774	0.3119	0.5755	0.9165	10.84
MambaDiff (7 million para.) - 70k steps	0.0857	0.3329	0.5944	0.9154	10.88
MambaDiff (7 million para.) - 80k steps	0.0895	0.3439	0.6032	0.9165	10.98

Table 5: Comparative results of various models on QQP dataset.

8 Future Works

The results of MambaDiff exhibit an increasing trend with additional steps and with the use of more parameters. This suggests that, given sufficient time, further runs will likely improve MambaDiff's performance, potentially surpassing DiffuSeq and other baseline models.

References

[1] DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. Quora question pairs. Kaggle, 2017. URL `https://kaggle.com/competitions/quora-question-pairs`.

[2] Wanyu Du, Jianqiao Zhao, Liwei Wang, and Yangfeng Ji. Diverse text generation via variational encoder-decoder models with gaussian process priors. *arXiv preprint arXiv:2204.01227*, 2022.

[3] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

[4] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[5] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[6] Albert Gu, Isaiah Johnson, Karan Goel, Khalil Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In *Advances in neural information processing systems*, volume 34, pages 572–585, 2021.

[7] Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. *Advances in neural information processing systems*, 32, 2019.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[9] Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. Neural crf model for sentence alignment in text simplification. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2020.

[10] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[11] Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745, 2018.

[12] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

[15] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018.

[16] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100, 2018.