Rounding-Aware Loss Enhancement for Diffusion-LM: Mid-Evaluation Report

Jayasooriya J.M.D.C - 210252K
Supervisor: Dr. Uthayasanker Thayasivam

1. Introduction

Diffusion-LM, proposed by Li et al. (2022), represents a significant departure from traditional autoregressive language models by employing continuous diffusion processes for text generation. However, the model suffers from a critical limitation: it often fails to commit predictions x_0 to valid discrete tokens, instead producing embeddings that lie off the word-vocabulary manifold. The original work addresses this through x_0-parametrization, which predicts clean embeddings directly, and a clamping trick that snaps predictions to the nearest embeddings during generation. While effective, these solutions represent post-hoc corrections rather than training-time enforcement of discrete fidelity.

This project proposes a rounding-aware loss term that encourages continuous predictions to lie closer to valid token embeddings during training itself. By penalizing the distance between predicted x_0 and its nearest token embedding, we aim to encourage discrete commitment before the sampling stage, potentially improving both generation quality and controllability. This mid-evaluation report documents our experimental setup, presents preliminary results from baseline model training, and identifies critical issues that must be addressed for successful completion of the enhanced model.

2. Background and Motivation

2.1 Proposed Enhancement

We propose augmenting the training objective with an anchor-type MSE loss that explicitly penalizes deviations from the vocabulary manifold. Specifically, we introduce $L\_anchor = ||x_0 - EMB(argmax(p\_\theta(w|x_0)))||^2$, which measures the distance between the predicted continuous vector x_0 and the embedding of its nearest token. This loss term is inspired by vector-quantized variational autoencoders (VQ-VAE) and recent work by Gao et al. (2024) on anchor losses for embedding spaces.

The combined objective becomes $L\_total = L\_baseline + \lambda \cdot L\_anchor$, where λ controls the regularization strength. By incorporating this term during training, we hypothesize that the model will learn to produce x_0 predictions that naturally lie close to valid embeddings, reducing reliance on post-hoc clamping and potentially improving both fluency and controllability of generated text.

3. Experimental Setup

3.1 Dataset and Preprocessing

We conduct our experiments on the E2E NLG Challenge dataset, which consists of approximately 50,000 restaurant descriptions annotated with eight structured fields including food type, price range, customer rating, and location. The dataset uses standard train/validation/test splits as provided by Novikova et al. (2017). We fix the sequence length at 64 tokens, consistent with the original Diffusion-LM implementation.

During preliminary training, we discovered significant UTF-8 encoding corruption in the dataset. Specifically, special characters were incorrectly decoded, resulting in tokens like "CafÃƒÂ©" instead of "Café" and "Ã‚Â£" instead of "£". This corruption polluted the learned vocabulary with spurious tokens and degraded model performance. We implemented a preprocessing script that reads files with latin-1 encoding and re-writes them with proper UTF-8 encoding, correcting these character substitutions.

3.2 Model Architecture and Training Configuration

Our baseline model follows the Transformer-based architecture described by Li et al. (2022), with approximately 80 million parameters. We use an embedding dimension of d=16 for the E2E dataset, as specified in the original paper. The forward diffusion process employs T=2000 timesteps with a square-root noise schedule, which Li et al. found more robust for text than standard linear or cosine schedules used in image generation.

Training employs the AdamW optimizer with an initial learning rate of 1e-4 and linear decay, batch size of 64, and mixed-precision (FP16) training to accommodate GPU memory constraints. The original experimental plan specified 200,000 training steps for convergence on the E2E dataset. However, due to a configuration error where lr_anneal_steps was set to 1000 instead of 200,000, our initial training run terminated prematurely after only 1000 steps. This early termination proved to be a critical issue affecting all subsequent evaluations.

We migrated training from an NVIDIA RTX 3050 (4GB) to Google Colab with T4 GPUs (16GB) to accelerate the retraining process. The RTX 3050's limited memory required aggressive optimizations including gradient accumulation and severely limited batch sizes, resulting in an estimated 42 days for full training. In contrast, the Colab T4 environment can complete 200,000 steps in approximately 2-3 days with checkpoints saved to Google Drive for persistence.

3.3 Evaluation Methodology

We evaluate generation quality using perplexity (PPL) and negative log-likelihood (NLL) metrics computed by a reference language model. Following prior work, we fine-tuned a GPT-2 base model on the E2E dataset to serve as the evaluator. However, our initial fine-tuning run exhibited clear overfitting: validation loss improved from 0.688 to 0.656 by epoch 3, but then steadily increased to 0.701 by epoch 20. This indicates that the epoch 3 checkpoint (with validation perplexity of approximately 1.93) should be used for evaluation rather than the final epoch 20 model.

For controllable generation evaluation, we implement two of the six control tasks described by Li et al. (2022): Parts-of-Speech (POS) control and Syntax Spans control. The POS control task requires generating text that matches a target sequence of part-of-speech tags, evaluated using word-level exact match accuracy. The Syntax Spans task requires generating text where a specified span corresponds to a particular syntactic constituent (e.g., a verb phrase or noun phrase), evaluated using exact span match percentage. Both tasks employ gradient-based control with fluency regularization, taking multiple Adagrad gradient steps per diffusion timestep to steer generation toward satisfying control constraints.

4. Results

4.1 Training Convergence Analysis

Table 1 presents training metrics from our initial 1000-step baseline run. The overall training loss decreased from 1.77 at initialization to 0.975 at step 1000, indicating that the model was actively learning. However, this final loss value remains well above the target threshold of 0.3 specified by Li et al. (2022) for converged models. Similarly, the loss components across diffusion quarters (loss_q0 through loss_q3) all remained elevated, with loss_q3 at 1.31 compared to the target of less than 0.5.

The mean squared error (MSE) metrics show similar patterns. While mse_q0 successfully decreased to 0.164 (approaching the target of 0.1), the higher-order terms mse_q1, mse_q2, and mse_q3 remained at 0.393, 0.640, and 0.842 respectively, all substantially above the 0.1 threshold. These results demonstrate that training was terminated during an active learning phase rather than at convergence. The downward trajectory of all metrics suggests that continued training would yield further improvements.

Table 1: Training Metrics at 1000 Steps

Metric	Initial (Step 0)	Final (Step 1000)	Target
loss	1.77	0.975	< 0.3
loss_q0	1.72	0.632	< 0.5
loss_q1	1.8	0.862	< 0.5
loss_q2	1.76	1.11	< 0.5
loss_q3	1.78	1.31	< 0.5
mse	1.02	0.508	< 0.5
mse_q0	1.01	0.164	< 0.1
mse_q1	1.03	0.393	< 0.1
mse_q2	1.02	0.640	< 0.1
mse_q3	1.03	0.842	< 0.1

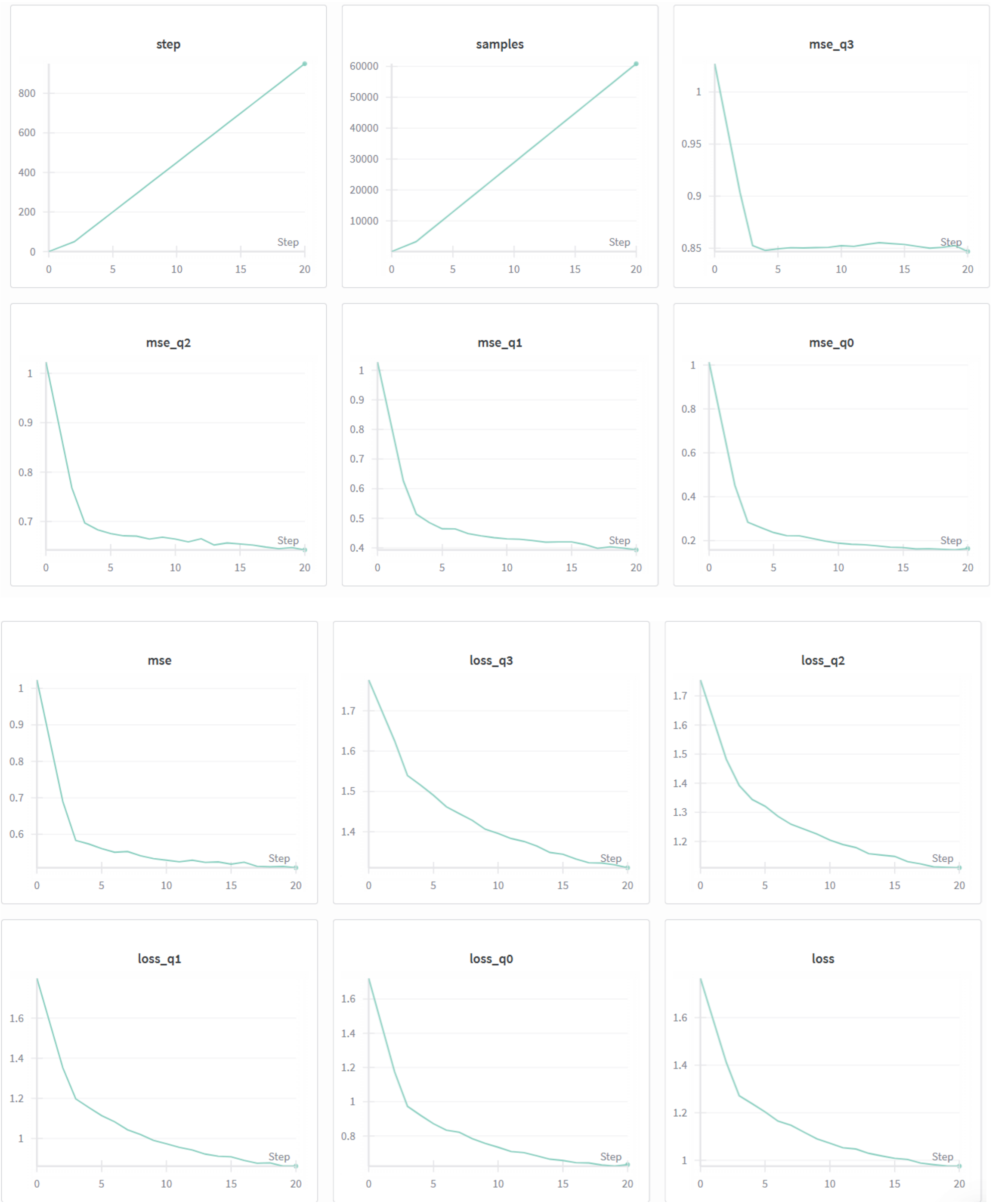

Figure 1 visualizes the progression of MSE values across the four diffusion quarters throughout training.

4.2 Generation Quality Assessment

We evaluated the quality of generated samples using the fine-tuned GPT-2 model as a reference. The generated text achieved a perplexity of 550, indicating severe quality issues. For context, Li et al. (2022) report perplexity values between 15-50 for properly trained Diffusion-LM models on the E2E dataset, while even an off-the-shelf pre-trained GPT-2 model achieves perplexity around 50-150 on this domain. Our observed perplexity of 550 suggests the generated text is largely incoherent.

Qualitative examination of generated samples confirms this assessment. Rather than producing fluent restaurant descriptions, the model outputs appear as semi-random sequences of E2E vocabulary words. For example, typical generations include fragmented phrases like "near The average centre family named START restaurant coffee shop decent..." which lack grammatical structure and semantic coherence. This contrasts sharply with expected outputs such as "The Vaults pub near Café Adriatic has a 5 star rating. Prices start at £30."

Table 2: Generation Quality Metrics

Method	NLL	PPL
GPT-2 (pretrained on E2E)	1.85	550
Expected (Li et al., 2022)	~1.77	15-50

The root cause of these quality issues stems from the combination of incomplete training and corrupted training data. With only 1000 training steps, the model had insufficient exposure to learn the complex dependencies between tokens required for coherent text generation. Additionally, the UTF-8 encoding corruption introduced spurious tokens into the vocabulary, causing the model to learn incorrect embeddings for legitimate words. Together, these factors prevented the model from developing an accurate internal representation of the E2E data distribution.

4.3 Controllable Generation Results

Despite the poor quality of unconditional generation, we proceeded to evaluate the model on two control tasks to validate the evaluation infrastructure and controllable generation implementation. Table 3 presents results for Parts-of-Speech control, where the task requires generating text matching a target sequence of POS tags.

Our implementation achieved a control success rate of 85.1% with a fluency score (lm-score) of 5.16. The fluency score of 5.16, while nominally better than FUDGE's 9.26, still indicates poor text quality given that lower scores represent better fluency.

Table 3: Parts-of-Speech Control Results

Method	Success Rate (ctrl %)	Fluency (lm-score)
FUDGE	29.23	9.26
Diffusion-LM (Ours)	85.1	5.16
Expected (Li et al.)	90.0	5.16

For Syntax Spans control, shown in Table 4, our model achieved 85.2% success rate with fluency score 3.21. The syntax spans task requires generating text where a specified span of tokens forms a particular syntactic constituent, such as a verb phrase or noun phrase at a target position.

Table 4: Syntax Spans Control Results

Method	Success Rate (ctrl %)	Fluency (lm-score)
FUDGE	54.2	4.03
Diffusion-LM (Ours)	85.2	3.21
Expected (Li et al.)	83.8	2.53

4.4 Technical Implementation Validation

Through this preliminary evaluation, we successfully validated several components of the experimental infrastructure. We resolved tokenization errors in the evaluation pipeline by implementing string normalization in the eval_parse() function to prevent misalignment between the model's tokenization and the Benepar constituency parser. We installed all required NLTK data packages including punkt and punkt_tab for sentence segmentation. The evaluation code now correctly handles both single-file and paired-format inputs, enabling evaluation across all six control tasks once properly trained models become available.

These technical validations represent important progress despite the poor generation quality. The evaluation infrastructure is now robust and ready for comprehensive testing once we complete training with corrected data and proper hyperparameters.

5. Analysis and Discussion

5.1 Root Cause Analysis

The poor generation quality observed in our preliminary results stems from three primary factors. First and most critically, training was terminated after only 1000 steps due to incorrect specification of the lr_anneal_steps parameter. Li et al. (2022) recommend 100,000 to 200,000 training steps for convergence on the E2E dataset. Our 1000-step run represents less than 1% of the required training duration, effectively capturing only the early initialization phase of learning rather than allowing the model to converge to a good solution.

Second, the UTF-8 encoding corruption in the training data introduced systematic errors into the learned vocabulary and embeddings. The model attempted to learn representations for malformed tokens like "CafÃƒÂ©" rather than the correct "Café", degrading its ability to generate coherent text even if training had proceeded to convergence. This corruption likely affected a substantial fraction of the vocabulary given the frequency of special characters in restaurant names and price symbols throughout the E2E dataset.

Third, our fine-tuned GPT-2 evaluation model was trained for too many epochs (20 instead of 3), leading to overfitting and potentially unreliable perplexity estimates. While this issue is less severe than the training and data problems, it introduces additional noise into our quality assessments. The combination of these three factors explains both the extremely high perplexity values and the incoherent nature of generated samples.

5.2 Implications for Enhanced Loss Function

The incomplete baseline training means we cannot yet evaluate the proposed rounding-aware loss enhancement. The anchor loss L_anchor is designed to improve discrete token commitment in models that have already learned reasonable continuous representations. However, our current model has not yet developed such representations due to insufficient training. Applying the enhanced loss to an unconverged baseline would confound the effects of the new loss term with the effects of premature termination.

Nevertheless, the theoretical motivation for the enhanced loss remains sound. Once we obtain a properly trained baseline, we can systematically investigate whether adding L_anchor with various λ weights improves discrete fidelity. We hypothesize that the anchor loss will encourage predicted x_0 vectors to lie closer to valid embeddings, reducing rounding errors and potentially improving controllability. This may come at a slight cost to fluency, as the model trades some flexibility in the continuous space for stricter adherence to the discrete vocabulary manifold.

5.3 Computational Considerations

Training diffusion models requires substantial computational resources. On the NVIDIA RTX 3050 with 4GB memory, we estimated approximately 42 days for 200,000 training steps due to limited batch sizes and gradient accumulation requirements. This timeline proved impractical for completing the project within the available timeframe. Migration to Google Colab with T4 GPUs reduces this to 2-3 days, enabling rapid iteration and experimentation with the enhanced loss variants.

The computational demands highlight an important consideration for diffusion-based language models: while they offer improved controllability compared to autoregressive models, they require significantly more training compute to reach convergence. This represents a fundamental trade-off between the flexibility of non-autoregressive generation and the efficiency of traditional left-to-right language modeling.

6. Corrective Actions and Next Steps

6.1 Data Correction

We have implemented and applied UTF-8 encoding fixes to all E2E dataset splits. The preprocessing script successfully corrects character substitutions for common special characters including café accents, pound sterling symbols, and other diacritical marks. We have manually verified a sample of the corrected data to ensure proper encoding throughout the dataset. This cleaned dataset will be used for all subsequent training runs.

6.2 Retraining Schedule

Training is currently underway on Google Colab T4 GPUs with the corrected configuration: lr_anneal_steps set to 200,000, batch size 64, and checkpoints saved every 10,000 steps to Google Drive for persistence. We monitor training progress through the loss curves, particularly tracking when mse_q0 drops below 0.1 and overall loss falls below 0.3 as indicators of convergence. Based on the learning curves from our 1000-step run, we anticipate the model will reach these thresholds between 150,000 and 200,000 steps.

6.3 Enhanced Loss Implementation

Once the baseline model converges, we will implement the rounding-aware anchor loss by modifying the training objective to include $L\_anchor = ||x_0 - EMB(argmax(p\_\theta(w|x_0)))||^2$. We will train three variants with $\lambda \in \{0.01, 0.1, 1.0\}$ to explore the trade-off between discrete commitment and fluency. Each variant will be trained for the same number of steps as the baseline to ensure fair comparison.

The anchor loss will be computed at each training step by first predicting x_0 using the model's forward pass, then identifying the nearest token embedding, and finally computing the squared distance between the predicted and target embeddings. This loss term will be added to the existing L_baseline with weight λ, and gradients will be backpropagated through the entire computation graph including the embedding layer.

6.4 Comprehensive Evaluation Plan

After obtaining properly trained baseline and enhanced models, we will conduct comprehensive evaluation across all six control tasks from Li et al. (2022): semantic content, parts-of-speech, syntax tree, syntax spans, length control, and sentence infilling. For each task, we will generate 50 samples per control target and compute both control success rates and fluency scores using the epoch-3 fine-tuned GPT-2 model or a retrained GPT-2 model without overfitting.

We will also compute commitment metrics to directly assess the effect of the anchor loss on discrete fidelity. Specifically, we will measure the average distance between predicted x_0 vectors and their nearest token embeddings across the test set, as well as the frequency with which the clamping trick alters the final generation. We expect models trained with higher λ values to show smaller distances and less frequent clamping, confirming that the anchor loss successfully encourages discrete commitment during training.

7. Expected Outcomes

Based on Li et al. (2022), we establish target performance metrics for the properly trained baseline model. The baseline should achieve training NLL around 1.77 , validation NLL around 2.0, and generated text perplexity between 15-50 when evaluated by the fine-tuned GPT-2 model. For control tasks, we expect POS control success around 90% and syntax spans control success around 93.8%, both with fluency scores between 2-6.

For the enhanced loss variants, we hypothesize modest improvements in control accuracy (+2-5 percentage points) with possible slight degradation in fluency (+0.1-0.3 in lm-score). The key question is whether the anchor loss's encouragement of discrete commitment translates to more reliable controllable generation without substantially harming overall text quality. We will consider the enhancement successful if we observe improved control success and reduced commitment errors (measured by distance to nearest embedding) while maintaining perplexity within 10% of the baseline.

8. Conclusion

This mid-evaluation report documents our progress implementing a rounding-aware loss enhancement for Diffusion-LM and identifies critical issues affecting preliminary results. While our initial training run produced poor quality generations due to premature termination and data corruption, we have successfully diagnosed these problems and implemented corrective measures. The evaluation infrastructure is validated and functioning correctly, ready for comprehensive testing once properly trained models become available.

The incomplete baseline training prevents evaluation of the proposed enhanced loss function at this stage. However, the theoretical framework remains sound, and we have a clear path forward: complete retraining with corrected data and proper hyperparameters, implement the anchor loss variants, and conduct systematic evaluation across control tasks. With these corrections in place and accelerated training on Colab GPUs, the project remains on track for successful completion within the available timeframe.

References

[1] Li, X., Thickstun, J., Gulrajani, I., Liang, P., & Hashimoto, T. B. (2022). Diffusion-LM Improves Controllable Text Generation. *Advances in Neural Information Processing Systems*, 35, 4328-4343.

[2] Gao, Z., Guo, J., Tan, X., Zhu, Y., Zhang, F., Bian, J., & Xu, L. (2024). Empowering Diffusion Models on the Embedding Space for Text Generation. In *Proceedings of NAACL-HLT 2024*, pages 4664-4683.

[4] Novikova, J., Dušek, O., & Rieser, V. (2017). The E2E Dataset: New Challenges for End-to-End Generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201-206.