

Segment-Level Diffusion: A Framework for Controllable Long-Form Generation with Diffusion Language Models

Anonymous ACL submission

Abstract

Diffusion models have shown promise in text generation, but often struggle with generating long, coherent, and contextually accurate text. Token-level diffusion doesn't model word-order dependencies explicitly and operates on short, fixed output windows, while passage-level diffusion struggles with learning robust representations for long-form text. To address these challenges, we propose Segment-Level Diffusion (SLD), a framework that enhances diffusion-based text generation through text segmentation, robust representation training with adversarial and contrastive learning, and improved latent-space guidance. By segmenting long-form outputs into multiple latent representations and decoding them with an autoregressive decoder, SLD simplifies diffusion predictions and improves scalability. Experiments on four datasets demonstrate that, when compared to other diffusion and autoregressive baselines SLD achieves competitive or superior fluency, coherence, and contextual compatibility in automatic and human evaluations.

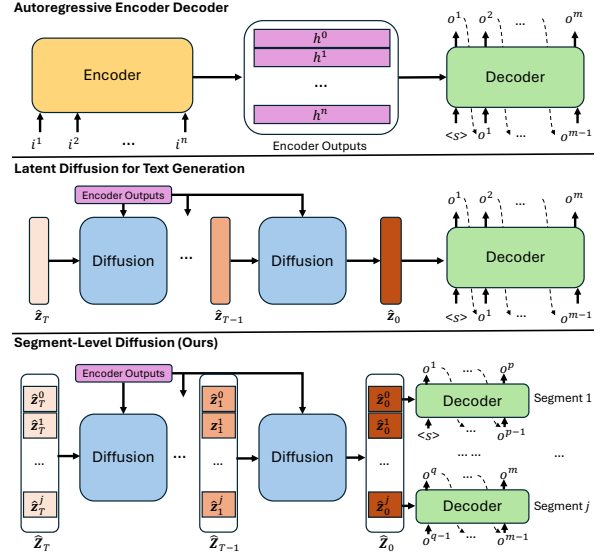


Figure 1: Comparison of AR models (top), latent diffusion (middle), and our segment-level diffusion (bottom). Unlike latent diffusion, which de-noises a single latent representation, our method splits outputs and representation into segments as the cross-attention target for conditional generation with parallel autoregressive decoding, improving text quality and controllability.

1 Introduction

Transformer-based autoregressive (AR) language models have become the prevailing standard in natural language generation (Vaswani et al., 2017; Zhao et al., 2023). However, the nature of next-token prediction inherently makes them prone to error propagation and incorrect handling of long-term dependencies, while also complicating controllable generation (He et al., 2021; Wu et al., 2018).

Diffusion models, which are non-autoregressive (NAR) generative models widely successful in image and video generation, have also shown promise in text generation (Ho et al., 2020; Radford et al., 2021; Singer et al., 2023). Li et al. (2022) pioneered the application of diffusion models to discrete text generation by predicting continuous word embeddings. Building on this work, Lin et al. (2023) introduced GENIE, a pre-trained diffusion

language model that enhances semantic understanding through continuous paragraph-level de-noising. These approaches fall under token-level diffusion, as they directly generate word embeddings. In contrast, Lovelace et al. (2023) proposed latent diffusion for text generation (LD4LG), encoding text into latent representations, applying diffusion to high-level semantic structures, and decoding them into text using an AR decoder. Chen and Yang (2023) leveraged diffusion models for controllable dialogue generation, operating on high-level discourse representations to enable precise control over the semantics of the dialogues generated.

However, existing diffusion language models face challenges in generating longer texts. In such cases token-level diffusion becomes computationally expensive, as it either requires pre-training with larger output windows, or relies on iterative

diffusion sampling (Tang et al., 2023; Yi et al., 2024). The fixed output window is also wasteful when the generated sequence is shorter. Additionally, unlike AR methods, they do not model word-order dependencies explicitly, often resulting in ungrammatical or incoherent output. Generating latent representations for passages with multiple sentences is harder, since they are highly sensitive to noise which can lead to abrupt changes of meaning in the decoded text, and learning a smooth latent distribution is challenging (Vahdat et al., 2021; Zhang et al., 2023).

To address these limitations, we propose a novel approach for diffusion-based text generation, Segment-Level Diffusion (SLD), illustrated in Figure 1. Inspired by the concept of image patches (Ding et al., 2023), we use a diffusion model to perform high-level semantics and structural planning, generating a latent representation for each segment (*e.g.*, sentences in paragraphs, utterances in dialogues), instead of handling long texts with a single latent representation. Then, an AR decoder decodes predicted representations to texts. To improve text generation, we integrate adversarial training (Miyato et al., 2017) and contrastive learning (Gao et al., 2021) to smoothen the latent representation distribution and optimize the AR decoder with respect to the diffusion process. Additionally, we incorporate extra loss signals from both decoding and latent space reconstruction to further strengthen control, improving coherence and fidelity in text generation.

We compare our SLD model against three diffusion models, GENIE (Lin et al., 2023), LD4LG (Lovelace et al., 2023), Diffuse-CG (Chen and Yang, 2023), and an autoregressive baseline, Flan-T5 (Chung et al., 2024). The evaluation includes summarization (XSum, Narayan et al. 2018), title-to-story generation (ROCStories, Mostafazadeh et al. 2016), summary-to-dialogue generation (DialogSum, Chen et al. 2021), and multiparty decision-making dialogue generation (DeliData, Karadzhov et al. 2023). Evaluation by both automatic and human metrics shows that SLD generates text that is more coherent and fluent, better aligned with the provided input, and matches ground-truth references more closely.

2 Related Work

Token-Level Diffusion Li et al. (2022) adapted diffusion model for discrete text generation by oper-

ating in the continuous space of word embeddings jointly learned by the model. The architecture iteratively de-noises sampled Gaussian noise into a sequence of word vectors. A rounding method is then applied to project the embeddings predicted into the nearest embeddings. Extending this work, Gong et al. (2023a) applied token-level diffusion to sequence-to-sequence generation tasks. Lin et al. (2023) advanced this approach by incorporating pre-training, which enhanced semantic and syntactic coherence by training diffusion decoders to reconstruct clean paragraphs from corrupted embeddings. These models achieve sequence-to-sequence generation using encoded text as classifier-free guidance (Ho and Salimans, 2022). Zhou et al. (2023) unifies discrete text generation and continuous representations by using BART (Lewis, 2019) with self-prompting to recover masked tokens.

However, token-level diffusion has notable limitations. Unlike AR decoding methods that always condition on previously decoded tokens, NAR generation does not model word-order dependencies explicitly, often resulting in text that lacks grammatical correctness and fluency. Furthermore, the fixed output window restricts the length of the generated text. It is computationally expensive to retrain the entire token-level diffusion model with larger output windows, even more so for architectures without a pre-trained language model backbone (Gulrajani and Hashimoto, 2024; Lou et al., 2023; Austin et al., 2021). Even though existing literature has accelerated diffusion sampling (Gong et al., 2023b; Tang et al., 2023), token-level diffusion remains inefficient if the generated sequence is shorter than the output window, as NAR decoding always generates the full output.

Passage-Level Diffusion Lovelace et al. (2023) built on the concept of latent space diffusion (Rombach et al., 2022) by compressing and predicting texts using high-level semantic representations, rather than directly predicting fine-grained token representations. Such compression is beneficial for both performance and efficiency, as it provides a length-independent representation and removes information not needed for diffusion prediction, in contrast to representations from traditional language encoders. A separate AR decoder is employed to ensure the fluency of the generated text.

However, this approach primarily focuses on short text generation, as learning robust latent representations for long passages remains challeng-

ing, and it is crucial to ensure the smoothness of the learned distribution for high-quality generation (Vahdat et al., 2021). Without proper regularization, the learned distribution may be susceptible to abrupt semantic changes due to small perturbations, increasing the difficulty of the task for the diffusion model. Although Zhang et al. (2023) proposed techniques to improve the distributional smoothness of latent representations, the correspondence between latent representations and specific components of the generated text remains unclear. This ambiguity complicates fine-grained guidance and limits control over the generation process.

These limitations result in existing token- and passage-level diffusion models struggling to generate long and coherent text. Despite their ability to generate outputs up to 64 tokens in length, they were primarily evaluated on tasks involving short text generation (e.g., QQP paraphrasing, XSum summarization) with outputs typically around 30 tokens or less (Gong et al., 2023a; Sharma et al., 2019; Yi et al., 2024; Li et al., 2023).

3 Segment-Level Diffusion

To address the challenges faced by diffusion language models in controllable long-form generation, we propose Segment-Level Diffusion (SLD). In this section, we first provide an overview of the language generation process using diffusion models in latent space, as illustrated in Figure 1. We then introduce our improvements, offering an overview of the three training stages of SLD, as illustrated in Figure 2: output segmentation, representation learning, and training diffusion processing for semantic planning. Detailed training algorithm of the our model is outlined in Appendix as Algorithm 1.

3.1 Formulation

Given an input text sequence $\mathbf{i} = \{i^1, i^2, \dots, i^n\}$ consisting of n tokens and an output sequence $\mathbf{o} = \{o^1, o^2, \dots, o^m\}$ consisting of m tokens, we model the conditional probability $p(\mathbf{o}|\mathbf{i})$ using a learnable diffusion model $R(\cdot; \theta_R)$. We follow Lovelace et al. (2024) by introducing additional encoding and decoding components to convert texts into continuous latent representations $\mathbf{z} \in \mathbb{R}^{k \times h_{rep}}$. The diffusion model operates on continuous latent variables $\mathbf{z}$ across T time steps, modelled as a Markov chain (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song and Ermon, 2020), and consists of two processes: a backward process for inference

and a forward process for training.

Inference The backward process generates the latent representation of the predicted output text $\hat{\mathbf{o}}$ by iteratively removing noise from an initial noisy sample. Starting with a variable $\hat{\mathbf{z}}_T \sim \mathcal{N}(\hat{\mathbf{z}}_T; \mathbf{0}, \mathbf{I})$, the diffusion model with parameters θ_R predicts the de-noised variable $\hat{\mathbf{z}}_{t-1}$ at each time step t as follows:

$$p(\hat{\mathbf{z}}_{t-1}|\hat{\mathbf{z}}_t; \theta_R) = \mathcal{N}(\hat{\mathbf{z}}_{t-1}; \boldsymbol{\mu}_{\theta_R}^{t-1}, \boldsymbol{\sigma}_{\theta_R}^{t-1}) \quad (1)$$

where $\boldsymbol{\mu}_{\theta_R}$ and $\boldsymbol{\sigma}_{\theta_R}$ are the predicted mean and variance at each time step. The diffusion model, $R(\cdot; \theta_R)$, estimates $\hat{\mathbf{z}}_{t-1}$. It conditions on the input sequence $\mathbf{i}$, using the encoder outputs from a pre-trained text encoder $Enc_{ctx}(\cdot; \theta_{ctx})$ as below.:

$$\hat{\mathbf{z}}_{t-1} = R(\hat{\mathbf{z}}_t, t, Enc_{ctx}(\mathbf{i}; \theta_{ctx}); \theta_R) \quad (2)$$

The model keeps refining the noisy sample $\hat{\mathbf{z}}_T$ with respect to the input sequence $\mathbf{i}$ to recover $\hat{\mathbf{z}}_0$ which will be converted to text. The predicted latent representation $\hat{\mathbf{z}}_0$ is passed to a function parameterised by θ_g , which reconstructs it to match the input dimensions of an AR decoder with parameters θ_{dec} for decoding:

$$g(\hat{\mathbf{z}}_0; \theta_g) \in \mathbb{R}^{k \times h_{lm}}, \hat{\mathbf{o}} = Dec(g(\hat{\mathbf{z}}_0; \theta_g); \theta_{dec}). \quad (3)$$

Training The diffusion model $R(\cdot; \theta_R)$ is trained by minimizing a regression loss to predict the noise added during the forward process. In the forward process, an original representation $\mathbf{z}_0$ of encoded $\mathbf{o}$ from a training instance $(\mathbf{i}, \mathbf{o}) \sim \mathcal{D}$ is gradually corrupted into Gaussian noise over T time steps. The encoding process consists of an encoder with parameters θ_{enc} that encodes the output texts:

$$Enc(\mathbf{o}; \theta_{enc}) \in \mathbb{R}^{m \times h_{lm}} \quad (4)$$

and a compression function with parameters θ_f that projects encoder outputs into a length-independent latent space using:

$$\mathbf{z} = f(Enc(\mathbf{o}; \theta_{enc}); \theta_f) \in \mathbb{R}^{k \times h_{rep}} \quad (5)$$

Here, we reduce the dimension of encoder outputs to a fixed-length representation with $k \leq m$ and $h_{rep} \ll h_{lm}$. The corruption is modelled as:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1 - \beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}) \quad (6)$$

where β_t controls the variance of the added noise at each step. The objective is to minimize the distance

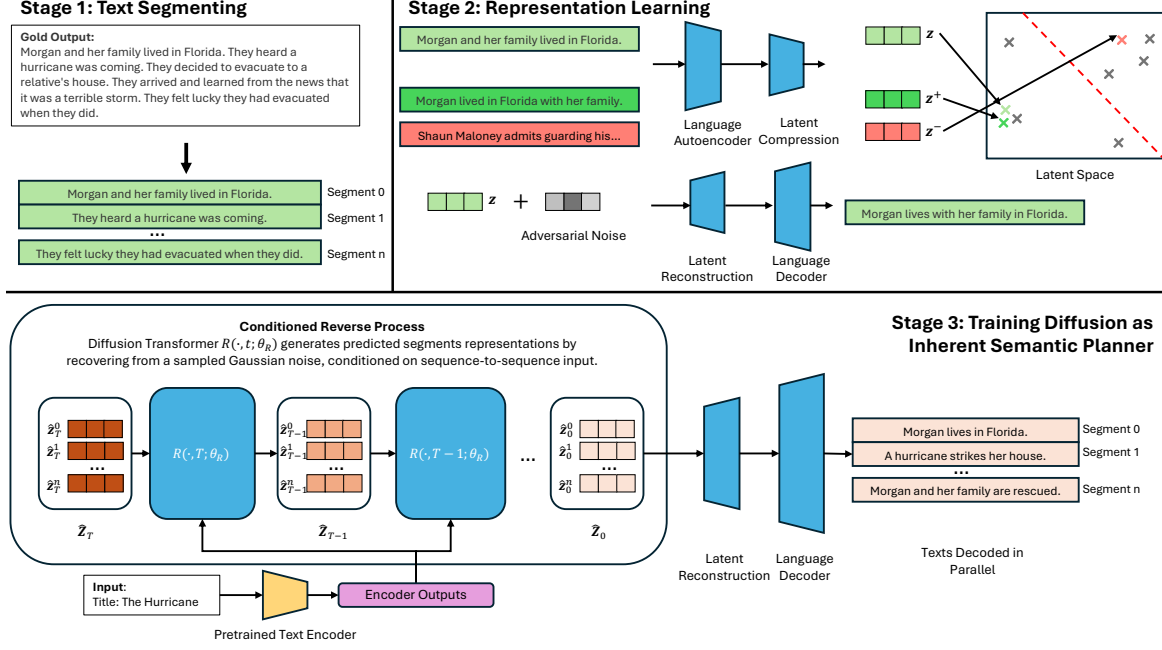


Figure 2: Overview of the training pipeline of SLD. In the first stage, gold output is divided into segments. In the second stage, we use contrastive and adversarial learning to ensure latent representations are robust to drastic semantic changes. Finally, we train a diffusion model as an inherent semantic planner conditioned on given inputs.

between the predicted representation $\hat{\mathbf{z}}_t$ and the true posterior $\mathbf{z}_t$ which is computed in closed form by sampling from the forward process.

$$\mathcal{L}(\theta_R) = \sum_{t=1}^T \mathbb{E}_{q(\mathbf{z}_t|\mathbf{z}_0)} \|\hat{\mathbf{z}}_t - \mathbf{z}_t\|_2^2 \quad (7)$$

The loss above trains the model to iteratively reverse the corruption applied during the forward process, enabling high-quality data generation in the backward process.

3.2 Segmented Text Generation

Inspired by the concept of image patches (Ding et al., 2023), in the first stage, we segment long outputs into smaller segments, such as sentences or dialogue utterances, rather than projecting the entire output into a single latent space representation. This segmentation effectively reduces the size and complexity of each latent representation, simplifying diffusion predictions and enabling greater flexibility for scaling, allowing the model to handle long-form text more efficiently. Formally, we construct $\mathbf{P} = \{\mathbf{p}^1, \dots, \mathbf{p}^j\}$, where each $\mathbf{p}^j$ corresponds to a non-overlapping contiguous segment of tokens in $\mathbf{o}$. This process yields a set of latent representations $\mathbf{Z} = \{\mathbf{z}^1, \dots, \mathbf{z}^j\}$, after encoding, establishing a one-to-one correspondence between each segment and its respective latent representation. As shown in Figure 2, a story is divided into n

segments (sentences in this case), and the diffusion model will predict the same number of representations and decode them into n segments in parallel.

3.3 Learning Latent Representations for Robust Decoding

As mentioned earlier, performing diffusion in latent space for text generation requires training the language autoencoder $Enc(\cdot; \theta_{enc})$, latent compression $f(\cdot; \theta_f)$, reconstruction $g(\cdot; \theta_g)$, and decoder $Dec(\cdot; \theta_{dec})$. A straightforward approach is to use the loss incurred during decoding $\hat{\mathbf{p}} = Dec(g(\mathbf{z}; \theta_g); \theta_{dec})$, where $\mathbf{z} = f(Enc(\mathbf{p}; \theta_{enc}); \theta_f)$, for a patch of text $\mathbf{p} = \{o_1, \dots, o_p\}$. We denote the parameters collectively as $\theta_{in} = \{\theta_{enc}, \theta_f\}$ for the encoding and compression, $\theta_{out} = \{\theta_g, \theta_{dec}\}$ for reconstruction and decoding, and $\theta_{rep} = \theta_{in} \cup \theta_{out}$ for latent representation parameters. The parameters are trained using the standard cross-entropy loss as below.

$$\mathcal{L}_{cnv}(\theta_{rep}) = - \sum_{l=1}^p \log p(o_l | o_{<l}, \mathbf{p}; \theta_{rep}) \quad (8)$$

However, learning a robust latent representation is non-trivial. Zhang et al. (2023) highlighted that a good latent representation should ensure low latent-to-text conversion error and smoothness in the latent distribution. Small perturbations to the latent representation should not significantly affect de-

coding, while textual segments with similar meanings should appear relatively close together in the latent space. To achieve this, we augment the cross-entropy loss with contrastive learning and adversarial training as regularization techniques.

Contrastive Representation Learning In our framework, we operate in latent spaces for segments of text $\mathbf{p}$, which are relatively short compared to the paragraphs in Zhang et al. (2023). This allows us to easily obtain meaningful positive examples (e.g., paraphrases) $\mathbf{p}^+$ and negative examples (e.g., out-of-domain text) $\mathbf{p}^-$ for contrastive learning (Gao et al., 2021). During training, we derive online representations $\mathbf{z}$, $\mathbf{z}^+$, and $\mathbf{z}^-$ for these segments and employ the following loss, where τ is the temperature parameter and $\text{sim}()$ is a similarity function (e.g., cosine similarity):

$$\mathcal{L}_{\text{cst}}(\theta_{in}) = -\log \frac{e^{\text{sim}(\mathbf{z}, \mathbf{z}^+)/\tau}}{e^{\text{sim}(\mathbf{z}, \mathbf{z}^+)/\tau} + e^{\text{sim}(\mathbf{z}, \mathbf{z}^-)/\tau}} \quad (9)$$

Noise for Decoding Robustness To make the decoding process less sensitive to noise in latent representations, and optimise the decoder with respect to the outputs of the diffusion model instead of the encoding process, we introduce noise into both the input text and latent representations during training. Specifically, we apply the following two noise injection strategies with small probabilities.

First, inspired by Zhang et al. (2023), we substitute a small portion of the original input text $\tilde{\mathbf{p}} = \text{sub}(\mathbf{p})$ with uniformly randomly sampled tokens from the model’s vocabulary, requiring the model to exhibit a certain level of error tolerance.

Second, we add adversarial noise to the latent representations $\mathbf{z}$. Extending Miyato et al. (2017)’s adversarial noise approach, we define ϵ_{adv} as the noise norm. The adversarial noise $\mathbf{r}_{adv}$ is computed as the normalized negative gradient of the loss:

$$\mathbf{r}_{adv} = -\frac{\epsilon_{adv} \mathbf{g}}{\|\mathbf{g}\|_2}, \text{ where} \quad (10)$$

$$\mathbf{g} = \nabla_{\mathbf{p}} \sum_{l=1}^p \log p(o_l | o_{<l}; \hat{\theta}_{out}) \quad (11)$$

and $p(\cdot; \hat{\theta}_{out})$ is modelled with $\hat{\theta}_{out}$ as a frozen copy of the parameters. This approach simulates the “worst-case” noise scenario, training the model’s reconstruction and decoding network to recover sequences under adversarial conditions with the

following loss:

$$\mathcal{L}_{adv}(\theta_{out}) = -\sum_{l=1}^p \log p(o_l | o_{<l}, \mathbf{z} + \mathbf{r}_{adv}; \theta_{out}). \quad (12)$$

Combining all three losses, the loss function for representation learning is defined as:

$$\mathcal{L}_{rep}(\theta_{rep}) = \frac{1}{N} \sum_{\mathbf{p}} (\mathcal{L}_{cnv}(\theta_{rep}) + \lambda_1 \mathcal{L}_{cst}(\theta_{in}) + \lambda_2 \mathcal{L}_{adv}(\theta_{out})) \quad (13)$$

where N is the size of the mini-batch, and λ_1, λ_2 are hyperparameters controlling the contribution of contrastive and adversarial losses.

3.4 Diffusion for Semantic Planning

After the aforementioned components are trained to convergence, we describe the final stage of training diffusion model as a segment-level semantic planner. The diffusion model leverages learned segment representations to plan and generate meaningful passages consisting of a sequence of segments.

Given a context $\mathbf{i}$ and collated output texts segments $\mathbf{P}$, we derive the corrupted latent representation of patches $\mathbf{Z}_t \in \mathbb{R}^{n \times k \times h_{rep}}$ at time t . Absolute positional embedding is applied to the flattened representation with respect to $n \times k$ (Vaswani et al., 2017). A transformer-based model, which is typically a diffusion transformer (DiT) (Peebles and Xie, 2023), is used for de-noising, defined as:

$$\hat{\mathbf{Z}}_{t-1} = R(\mathbf{Z}_t, t, \text{Enc}_{ctx}(\mathbf{i}; \theta_{ctx}); \theta_R), \quad (14)$$

where $\text{Enc}_{ctx}(\cdot; \theta_{ctx})$ is a pre-trained language encoder with frozen parameters. The encoded outputs serves as the cross-attention target for the diffusion transformer enabling conditional generation. We define the diffusion de-noising loss as:

$$\mathcal{L}_{noise}(\theta_R) = \mathbb{E}_{\mathbf{P}, \mathbf{i}, t} \left\| \hat{\mathbf{Z}}_{t-1} - \mathbf{Z}_{t-1} \right\|_2^2. \quad (15)$$

To strengthen the guidance and ensure the fluency of the decoded text, we add a post-diffusion training loss, which incorporates loss signals from the reconstruction and decoding processes. This strategy effectively teaches the diffusion model how to use $g(\cdot; \theta_g)$ and $\text{Dec}(\cdot; \theta_{dec})$, further enhancing the quality of the generated text. Similar to Zhang et al. (2024)’s pixel level guidance, we freeze the reconstruction and decoding parameters

and define the additional objectives for θ_R as follows :

$$\mathcal{L}_{\text{rec}}(\theta_R) = \mathbb{E}_{\mathbf{P}, \mathbf{i}, t} \left\| g(\hat{\mathbf{Z}}_t; \theta_g) - g(\mathbf{Z}_t; \theta_g) \right\|_2^2, \quad (16)$$

$$\mathcal{L}_{\text{dec}}(\theta_R) = \mathbb{E}_{\mathbf{P}, \mathbf{i}, t} \left[- \sum_{l=1}^p \log p(o_l | o_{<l}, \hat{\mathbf{Z}}; \theta_{\text{dec}}) \right]. \quad (17)$$

Combining these three losses above with hyperparameters λ_3 and λ_4 as weighting factors, we define the diffusion loss function as below.

$$\mathcal{L}_{\text{diff}}(\theta_R) = \mathcal{L}_{\text{noise}}(\theta_R) + \lambda_3 \mathcal{L}_{\text{rec}}(\theta_R) + \lambda_4 \mathcal{L}_{\text{dec}}(\theta_R) \quad (18)$$

4 Evaluation

4.1 Datasets and Baselines

We evaluate our implementation on datasets with an increasing number of utterances to assess its performance across various tasks. We start with the XSum dataset (Narayan et al., 2018), consisting of BBC news articles paired with concise, one-sentence summaries, to compare our model against baseline short-form diffusion models. We then scale up to longer outputs using the ROCStories dataset (Mostafazadeh et al., 2016) for title-to-story generation, and the DialogSum dataset (Chen et al., 2021) for summary-to-dialogue generation. These datasets allow us to evaluate the model’s capability for long-form generation. Additionally, we test our model on dialogue generation with DeliData (Karadzhov et al., 2023). This dataset consists of multi-party problem-solving dialogues during which participants propose solutions that are scored for their correctness. By comparing the predicted user score trajectories against the ground truth and identifying hallucinations, we analyse the effectiveness of applying control to the model’s generation.

We compare our model against a range of baselines. Specifically, we use LD4LG (Lovelace et al., 2023) as the diffusion baseline and Flan-T5 Large (Chung et al., 2024) as the autoregressive baseline. For the XSum dataset, we also compare against the token-level diffusion model GENIE (Lin et al., 2023). For the DialogSum dataset, we include comparisons with the dialogue-level diffusion model Diffuse-CG (Chen and Yang, 2023).

4.2 Evaluation Metrics

We use ROUGE as the primary evaluation metric to assess the quality and similarity of generated

text with respect to the gold output. While ROUGE provides a baseline for lexical overlap, we acknowledge its limitations in capturing semantic fidelity, coherence, and conversational nuances, particularly in controlled long-form generation. To address this, we extend human evaluation guidelines from Clark et al. (2023), assessing repetition, fluency, coherence, compatibility (ROCStories/DialogSum), and hallucination (DeliData):

- **Repetition:** Check for repetitive tokens or utterances that affect meaning.
- **Fluency/Grammar:** Assess grammatical correctness and fluency.
- **Coherence:** Evaluate logical flow and naturalness of interactions.
- **Compatibility:** Ensure alignment with the story title/dialogue summary.
- **Hallucination:** Detect impossible choices or non-existent participants.

Human scores range from 0 to 3, with higher scores indicating better performance. Details and examples are in Appendix C. Following the evaluation metrics in the literature, we also evaluate the perplexity of generated text using GPT-2 Large (Radford et al., 2019) as teacher model, and record the average length of generated texts.

4.3 Implementation Details

We build upon the design of Latent Diffusion for Language Generation (LD4LG) proposed by Lovelace et al. (2023), using Flan-T5 Base (Chung et al., 2024) as the backbone to initialize our encoder and decoder. We incorporate the Perceiver Resampler (Alayrac et al., 2022) as the compression and reconstruction unit and employ a pre-LayerNorm transformer as the de-noising model (Vaswani et al., 2017). For contrastive learning targets, we use Llama-3-8B-Instruct (Llama Team, 2024) to generate paraphrases for each text segment. For XSum, we sampled out-of-domain (OOD) texts from Movie-Dic (Banchs, 2012) dataset as hard negative targets. For other datasets, we sample from CNN/Daily Mail (See et al., 2017).

5 Results

We present the results of our model against the baselines on XSum, ROCStories, DialogSum and DeliData in Table 1. For short-form generation task, XSum (~ 30 tokens), our model demonstrates on-par performance compared with other baselines. For long-form generation (≥ 50 tokens) with nat-

XSum							
Models	ROUGE-1	ROUGE-2	ROUGE-L	Length	PPL	Fluency*	
Gold	N/A	N/A	N/A	21.25	22.93	2.98	
Flan-T5	44.31	20.44	35.76	19.25	18.02	2.85	
GENIE	36.14	12.63	28.37	20.39	145.37	1.78	
LD4LG	32.96	11.70	26.74	20.43	44.33	2.83	
Ours	34.24	12.38	27.77	21.27	46.09	2.78	

ROCStories							
Models	ROUGE-L	Length	PPL	Repetition (mean, var)	Fluency (mean, var)	Coherence (mean, var)	Compatibility (mean, var)
Gold	N/A	42.53	20.99	2.93, 0.08	2.87, 0.16	2.86, 0.17	2.74, 0.34
Flan-T5	16.46	32.40	9.14	2.41, 0.82	2.62, 0.59	2.33, 0.78	2.07, 0.83
LD4LG	16.57	36.56	65.32	2.14, 1.26	1.78, 1.02	1.54, 0.92	1.79, 1.17
Ours	16.13	40.70	43.67	2.45, 0.89	2.41, 0.83	2.10, 0.88	2.42, 0.88

DialogSum							
Models	ROUGE-L	Length	PPL	Repetition (mean, var)	Fluency (mean, var)	Coherence (mean, var)	Compatibility (mean, var)
Gold	N/A	117.32	9.68	2.95, 0.07	2.86, 0.14	2.91, 0.11	2.92, 0.11
Flan-T5	26.34	131.92	3.78	1.83, 0.82	2.60, 0.55	2.07, 0.64	2.27, 0.60
LD4LG	20.90	81.28	43.82	1.93, 1.32	1.43, 0.84	1.39, 0.90	1.61, 1.16
Ours	27.97	113.92	16.39	2.80, 0.23	2.83, 0.17	2.40, 0.45	2.57, 0.37
Diffuse-CG**	27.57	84.23	68.45	N/A	N/A	N/A	N/A

DeliData							
Models	ROUGE-L	Length	PPL	Repetition (mean, var)	Fluency (mean, var)	Coherence (mean, var)	Hallucination (mean, var)
Gold	N/A	53.04	13.42	2.60, 0.46	2.66, 0.42	2.55, 0.47	2.85, 0.29
Flan-T5	25.83	234.35	9.79	2.27, 0.50	2.59, 0.39	2.30, 0.43	2.48, 0.52
LD4LG	21.14	68.08	51.10	2.52, 0.89	2.06, 0.80	1.78, 0.71	1.63, 0.87
Ours	30.51	71.61	13.41	2.50, 0.54	2.60, 0.34	2.30, 0.47	2.48, 0.55

Table 1: Comparison of results on the XSum, ROCStories, DialogSum, and DeliData datasets. Fluency* for XSum is calculated using the oracle model; details are provided in Appendix C. Results for Diffusion-CG** are directly taken from Chen and Yang (2023).

urally occurring segmentation (sentences in ROCStories, utterances in DialogSum and DeliData), our model shows better overall performance, especially in repetition, fluency and compatibility.

Short-form Although Flan-T5 achieves the highest ROUGE scores on XSum, SLD achieves on-par performance compared to other diffusion methods while maintaining a length closer to the reference. Importantly, the results indicate the importance of an AR decoder. For Flan-T5, LD4LG and SLD that models $p(o_l | o_{<l}, i)$, they have substantially higher fluency scores than GENIE which uses diffusion models to decode token level embeddings directly, modelling $p(o | i)$. In addition, autoregressive models can terminate generation early, while GENIE always has to predict up to its 64 token with unnecessary paddings due to its NAR nature.

Long-form Our method consistently achieves the highest ROUGE-L scores for summary-to-dialogue

and multi-party dialogue generation tasks, outperforming all baselines in datasets such as DialogSum and DeliData. In terms of human evaluation metrics, SLD shows clear advantages in reducing repetition and enhancing compatibility, producing logically consistent and contextually accurate outputs across tasks like ROCStories and DialogSum. This demonstrates diffusion guidance for controlled generation makes our outputs adhere to the given topic better. Furthermore, SLD maintains competitive fluency and coherence, matching autoregressive models like Flan-T5, while LD4LG’s performance drops as output sequence gets longer. SLD also demonstrates robustness in handling complex dialogue structures, aligning well with user score trajectory distributions in multi-party settings, as detailed in Appendix D. Additionally, SLD’s outputs closely match the desired length and structure of gold references, showcasing its adaptability across diverse text generation tasks.

Overall, SLD improves across datasets and metrics, particularly in long-form, contextually accurate text generation. While Flan-T5 excels in metrics like perplexity, it suffers from catastrophic repetition after fine-tuning. LD4LG struggles with coherence and fluency in longer texts due to abrupt semantic shifts in latent representations, as reflected by high variance in human evaluations. In contrast, SLD achieves better human scores, particularly in fluency, coherence, and compatibility.

6 Analysis

Representation to Text To assess representation learning, we tested various autoencoder-decoder configurations using BLEU (Papineni et al., 2002) to compare input text with recovered text after encoding. Using DialogSum utterances as segments, we evaluated LD4LG, ML-Planner, and SLD with and without contrastive learning. For consistency, segments were limited to 64 tokens, represented in a latent space of 32×64 . An LD4LG baseline with longer dialogues (up to 512 tokens) and latent dimensions of 256×64 was also included.

Figure 3 shows that ML-Planner failed to converge within five epochs, while LD4LG achieved a BLEU score of 1.00, indicating perfect recovery. However, LD4LG’s performance degraded when scaled to longer texts, highlighting limitations in generalization. Without contrastive learning, our model occasionally corrupted words, altering meaning. Incorporating contrastive learning enabled meaningful paraphrases instead of semantic corruption, as demonstrated in Table 2, emphasizing its role in enhancing representation quality.

Decoding after De-noising We further investigated how latent representations behave under perturbations to evaluate their robustness during the de-noising process. We randomly selected 100 sentences from the ROC dataset, along with their paraphrases and OOD sentences sampled from the CNN/Daily Mail dataset. We visualized a 2D PCA projection of learned representations, detailed in Appendix B. Without contrastive learning, the representations of original sentences and OOD sentences showed significant overlap, increasing the risk of abrupt semantic changes during decoding. In contrast, representations learned with contrastive training were better clustered and distinct, providing improved robustness for diffusion predictions.

To investigate further, we sampled the de-noising trajectory of a test sentence and analysed the de-

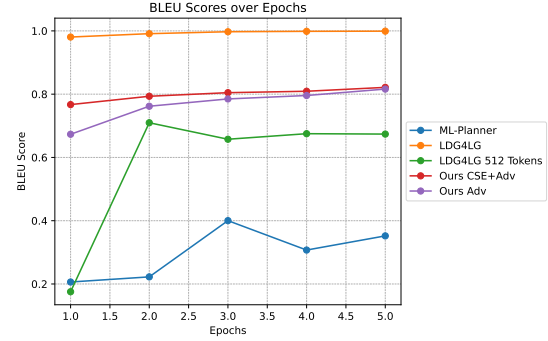


Figure 3: BLEU score of different auto-encoder/decoder models for text conversion on DialogSum dataset of a single utterance.

Model	Generated Text
Gold	#Person1#: What made you decide on this type of occupation?
Planner	#Person1#: I'm afraid I can't.
LD4LG	#Person1#: What made you decide on this type of occupation?
Adv	#Person1#: What's your decide on this type of occupation?
Adv+CSE	#Person1#: What made you decide on this type of job?

Table 2: Text quality comparison of different models for text generation. Red indicates wrong conversion, orange indicates corrupted token and green indicates admissible paraphrase.

coded text along the trajectory. Representations trained with both contrastive learning and adversarial training produced text that was more robust to noise and less prone to abrupt semantic shifts. This robustness facilitates smoother and more reliable predictions during the diffusion process.

7 Conclusion

We propose Segment-Level Diffusion (SLD) for controllable long-form text generation using latent space diffusion. Key innovations include text segmentation, robust representation learning via adversarial and contrastive training, and improved latent-space guidance. SLD addresses challenges in generating fluent, coherent, and contextually accurate long-form text, bridging the gap between latent diffusion models and practical long-form generation. It offers a scalable framework for applications like story and dialogue generation. Our results highlight SLD’s potential as a new paradigm for controllable text generation and provide insights for future diffusion-based language models.

Limitations

This work focuses exclusively on text generation in English, leaving the model’s potential for multilingual tasks unexplored. Furthermore, our experiments and evaluations did not involve real-world use cases, limiting insights into practical applicability. Future research could extend our approach to multilingual and application-oriented scenarios such as outline-controlled generation (Li et al., 2024; Lee et al., 2024). Additionally, we did not explicitly examine the relationship between the reduced dimensionality of the length-independent latent representations and the original dimensionality of encoded text segments of varying lengths. Due to limited computational capacity, we did not run hyperparameter search on loss ratios but chosen them empirically based on model’s performance on the development set. These hyperparameters were chosen empirically, without a systematic exploration of their impact. Future work could leverage principles from information theory (Tishby and Zaslavsky, 2015) to quantify the information capacity of these representations and to balance compression and utility more effectively. Developing a generalized framework to streamline hyperparameter selection across diverse datasets and pre-training tasks would also enhance the scalability of our method. Finally, our modular training approach, where individual components are optimized separately, may introduce suboptimal performance during inference due to error propagation and misalignment between training and inference objectives. Future work could explore end-to-end training strategies to jointly optimize all components, reducing such discrepancies and improving overall performance. Due to inconsistent sampling strategies and output windows of token-level diffusion and other baseline models, we did not perform inference speed comparison.

Ethics Statement

No personally identifiable information (PII) was collected or processed during the human evaluation, and all data handling adhered to the General Data Protection Regulation (GDPR) and the University’s research guidelines¹. Participants were recruited from within the University on a voluntary basis and were assigned anonymized random IDs to ensure their privacy during the evaluation

¹<https://www.research-integrity.admin.cam.ac.uk/academic-research-involving-personal-data>

process. Additionally, all data points presented to annotators were manually reviewed to ensure fairness and accuracy in assessing our methods and to minimize potential bias or harm to participants. This approach reflects our commitment to ethical research practices and to safeguarding the well-being and integrity of participants throughout the evaluation process.

References

- Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. *Flamingo: a visual language model for few-shot learning*. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993.
- Rafael E. Banchs. 2012. *Movie-DiC: a movie dialogue corpus for research and development*. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 203–207, Jeju Island, Korea. Association for Computational Linguistics.
- Jiaao Chen and Diyi Yang. 2023. *Controllable conversation generation with conversation structures via diffusion models*. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7238–7251, Toronto, Canada. Association for Computational Linguistics.
- Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. *DialogSum: A real-life scenario dialogue summarization dataset*. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

694	Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. <i>Journal of Machine Learning Research</i> , 25(70):1–53.	750
695		751
696		752
697		753
698		
699	Elizabeth Clark, Shruti Rijhwani, Sebastian Gehrmann, Joshua Maynez, Roei Aharoni, Vitaly Nikolaev, Thibault Sellam, Aditya Siddhant, Dipanjan Das, and Ankur Parikh. 2023. SEAHORSE: A multilingual, multifaceted dataset for summarization evaluation . In <i>Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing</i> , pages 9397–9413, Singapore. Association for Computational Linguistics.	754
700		755
701		756
702		757
703		
704		758
705		759
706		760
707		761
708		
709	Zheng Ding, Mengqi Zhang, Jiajun Wu, and Zhuowen Tu. 2023. Patched denoising diffusion models for high-resolution image synthesis. In <i>The Twelfth International Conference on Learning Representations</i> .	762
710		763
711		764
712		765
713		766
714		767
715		768
716		
717		769
718		770
719		771
720		772
721		773
722		774
723		
724		775
725		776
726		777
727		778
728		779
729		780
730		781
731		782
732		
733		783
734		784
735		785
736		786
737		787
738		788
739		789
740		790
741		791
742		
743		792
744		793
745		
746		794
747		795
748		796
749		
		797
		798
		799
		800
		801
		802
		803
		804
		805

806	in <i>Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023</i> .	863
807		864
808		
809		
810	Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification . In <i>5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings</i> . OpenReview.net.	865
811		866
812		867
813		868
814		869
815		870
816	Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories . In <i>Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 839–849, San Diego, California. Association for Computational Linguistics.	871
817		
818		
819		
820		
821		
822		
823		
824		
825		
826	Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization . In <i>Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing</i> , pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.	872
827		873
828		874
829		
830		
831		
832		
833	Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation . In <i>Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics</i> , pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.	875
834		876
835		877
836		878
837		879
838		880
839		881
840	William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers . In <i>IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023</i> , pages 4172–4182. IEEE.	882
841		
842		
843		
844		
845	Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision . In <i>Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event</i> , volume 139 of <i>Proceedings of Machine Learning Research</i> , pages 8748–8763. PMLR.	883
846		884
847		885
848		886
849		887
850		888
851		889
852		890
853		891
854		892
855	Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. <i>OpenAI blog</i> , 1(8):9.	893
856		894
857		895
858		896
859	Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In <i>Proceedings of the IEEE/CVF conference</i>	897
860		898
861		899
862		900
	on computer vision and pattern recognition, pages 10684–10695.	901
	Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks . In <i>Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.	902
		903
		904
		905
		906
		907
	Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset .	908
		909
		910
		911
	Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2023. Make-a-video: Text-to-video generation without text-video data . In <i>The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023</i> . OpenReview.net.	912
		913
		914
		915
		916
		917
	Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics . In <i>Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015</i> , volume 37 of <i>JMLR Workshop and Conference Proceedings</i> , pages 2256–2265. JMLR.org.	
	Yang Song and Stefano Ermon. 2020. Improved techniques for training score-based generative models . In <i>Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual</i> .	
	Zecheng Tang, Pinzheng Wang, Keyan Zhou, Juntao Li, Ziqiang Cao, and Min Zhang. 2023. Can diffusion model achieve better performance in text generation? bridging the gap between training and inference! <i>arXiv preprint arXiv:2305.04465</i> .	
	Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context . <i>ArXiv preprint</i> , abs/2403.05530.	
	Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In <i>2015 IEEE information theory workshop (itw)</i> , pages 1–5. IEEE.	
	Arash Vahdat, Karsten Kreis, and Jan Kautz. 2021. Score-based generative modeling in latent space . In <i>Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual</i> , pages 11287–11302.	

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Lijun Wu, Xu Tan, Di He, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. [Beyond error propagation in neural machine translation: Characteristics of language also matter](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3602–3611, Brussels, Belgium. Association for Computational Linguistics.
- Qihua Yi, Xiangfan Chen, Chenwei Zhang, Zehai Zhou, Linan Zhu, and Xiangjie Kong. 2024. Diffusion models in text generation: a survey. *PeerJ Computer Science*, 10:e1905.
- Christina Zhang, Simran Motwani, Matthew Yu, Ji Hou, Felix Juefei-Xu, Sam Tsai, Peter Vajda, Zijian He, and Jialiang Wang. 2024. [Pixel-space post-training of latent diffusion models](#). *ArXiv preprint*, abs/2409.17565.
- Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua M. Susskind, and Navdeep Jaitly. 2023. [PLANNER: generating diversified paragraph via latent language diffusion model](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. [A survey of large language models](#). *ArXiv preprint*, abs/2303.18223.
- Kun Zhou, Yifan Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Diffusion-nat: Self-prompting discrete diffusion for non-autoregressive text generation. *arXiv preprint arXiv:2305.04044*.

Appendix

A Training

All experiments were conducted on a single NVIDIA A100-SXM-80GB GPU. Training for latent representation learning took approximately 8–10 hours per dataset, while diffusion training required 36–60 hours per dataset. Including inference and other experiments, the total GPU usage amounted to around 500 hours. The detailed training algorithm is presented in Algorithm 1.

Loss Ratio In representation learning phase, contrastive loss with hyperparameter λ_1 in Equation 13 helps distinguish between nuanced representations, which is crucial in domains with subtle differences in meaning between text segments. For domains like DeliData, where dialogue actions are highly structured and limited (e.g., proposing choices, picking cards, discussing, revising, and submitting), contrastive loss can be reduced or even set to zero (e.g., $\lambda_1 = 0.1$ or 0). Adversarial loss in this same equation controls the model’s tolerance to noise, ensuring robust latent representations even in challenging cases. A large λ_2 can hinder the learning of meaningful representations, but our experiments suggest that moderate values (e.g., $\lambda_2 \approx 0.2$) are crucial for domains with longer text segments, such as DialogSum, where segments (e.g., multi-sentence utterances) require higher noise tolerance to maintain generation quality. Tasks with shorter segments or clearer structure might allow for lower λ_2 values. In Equation 18 for diffusion training, λ_3 controls the loss ratio of reconstruction back to the language model’s hidden state dimensions and λ_4 corresponds to the reconstruction back to the original text. We provide the hyperparameters we’ve chosen for each dataset in the table below.

Dataset	λ_1	λ_2	λ_3	λ_4
XSum	0.5	0.2	0.5	0.5
ROCStories	0.2	0.2	0.5	0.5
DialogSum	0.5	0.2	0.5	0.5
Delibot	0	0.2	0.5	0.5

Table 3: Hyperparameter choices of loss ratio.

Inference Sampling We use the sampling algorithm as Lovelace et al. (2023), a DDPM pytorch implementation². For training we use 2000 steps in forward process as target. For inference, we use spaced diffusion with only 250 steps.

²<https://github.com/lucidrains/denoising-diffusion-pytorch>

B Ablation Study: Latent Representation

We conducted an ablation study to evaluate how different representation learning methods affect the noised latent representations during the diffusion process. The "Vanilla" configuration corresponds to the original LD4LG implementation. As shown in Figure 6 and Table 7, our method achieves a smoother latent distribution and more robust representations, resulting in improved diffusion predictions and decoding.

C Human Evaluation

For ROCStories, DialogSum, and DeliData, we recruited 5 participants to evaluate 25 dialogues generated by LD4LG, Flan-T5, and SLD for each dataset. Annotators also rated the gold dialogue output as a reference. In total, we collected $3 \times 25 \times 4 \times 5 = 1,500$ data points across 4 evaluation criteria. Before presenting the dialogues, annotators were provided with instructions (example shown in Table 5).

The evaluation process begins by assessing readability. If a dialogue is deemed unreadable, all criteria are automatically scored as 0. Otherwise, the dialogue is rated with a minimum score of 1. An example of the web interface is provided in Figure 5.

Additionally, we employed Gemini-1.5 (Team et al., 2024) as an LLM-based evaluator. Using a carefully designed prompt (Table 6), Gemini-1.5 provided fluency ratings for predictions from the XSum dataset, achieving high alignment with human annotators.

D Case Study: DeliData

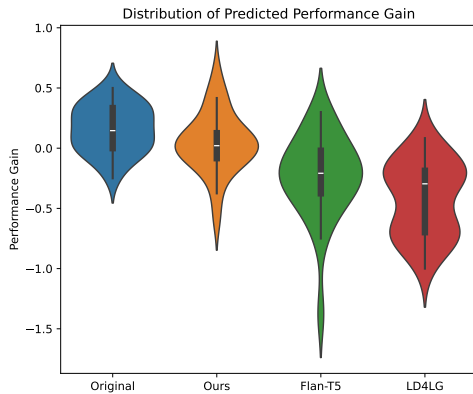


Figure 4: Violin plot of performance gain distribution of the DeliData dialogue continuations. Our model demonstrate a closer distribution with respect to the gold distribution, demonstrating the output is better controlled.

To evaluate the controlled generation capabilities of our model, we performed a fine-grained score trajectory analysis on DeliData. This involved comparing the distribution of performance gains at the end of group discussion dialogues, based on users' choices before and after revisions. Following the guidelines of Karadzhov et al. (2023), we observed that dialogue continuations generated by our model produced a performance gain distribution closer to the ground truth, as shown in Figure 4.

E Case Study: ECQA

We additionally evaluate how diffusion training affects the knowledge and reasoning abilities of the language model backbone. We test our model against baselines on Flan-T5's pre-training task, ECQA (Aggarwal et al., 2021), a knowledge QA dataset with Chain-of-Thought (CoT) reasoning. We measure model performance by prediction accuracy and ensure fair comparisons by using the same one-shot CoT exemplar setup for Flan-T5 Base and Large (Chung et al., 2024). We train LD4LG and our SLD model with the same hyperparameters as ROCStories, as shown in Table 3.

Model	Acc	MAUVE	ROUGE-L
Flan-T5 Base	0.75	0.13	33.88
Flan-T5 Large	0.88	0.21	35.86
LD4LG	0.63	0.74	38.66
Ours	0.723	0.75	40.14

Table 4: Performance comparison across models on ECQA.

The results indicate that SLD preserves the knowledge in its Flan-T5 Base backbone effectively. The diffusion model learns to predict latent representations of unseen concepts within the backbone's original knowledge almost perfectly. However, due to the backbone's limitations, all models fail to answer questions about novel concepts, as shown in Table 11. LD4LG frequently struggles to select the correct answer from the provided choices, as illustrated in Table 12, highlighting the importance of our text segmentation. While both LD4LG and SLD achieve high MAUVE and ROUGE-L scores due to direct training on the dataset, SLD outperforms LD4LG, demonstrating the advantage of our post-diffusion control.

F Sampled Generations

We provide sample output comparisons in Tables 8, 9, and 10.

Example of Human Evaluation Instructions

5. Compatibility

- Evaluate whether the dialogue accurately reflects the facts provided in the summary. Rate from 1 to 3:
- **1:** The dialogue significantly deviates from the summary, either by altering key facts or diverging from the main topic.
- **3:** The dialogue faithfully reflects the summary, covering all key ideas accurately.

Examples:

- **Summary:** Anna asks Bob about his vacation, and Bob talks about visiting a tropical island.

- Rating 1:

- #Anna#: "How was your weekend?"
- #Bob#: "I stayed home and watched TV."
- **Rating:** 1 (Completely unrelated to the summary).

- Rating 2:

- #Anna#: "How was your vacation?"
- #Bob#: "It was great. I visited some nice places."
- **Rating:** 2 (Partially reflects the summary but lacks details about the tropical island).

- Rating 3:

- #Anna#: "How was your vacation?"
- #Bob#: "It was amazing! I visited a beautiful tropical island and spent my days snorkeling and relaxing on the beach."
- **Rating:** 3 (Accurately reflects the summary and includes key details).

Table 5: An example of human evaluation instructions for evaluating compatibility for DialogSum.

LLM-as-a-judge Prompt

• Rate the grammatical quality of the summary on a scale of 1 to 3:

- **1:** Excessive grammatical and clerical errors, making the summary unnatural.
- **3:** Completely fluent with no grammatical errors.

Examples

Rating 1:

- *Summary:* Blaenau hub to is set up to tech of finger printer security.
- **Rating:** 1 (Frequent grammatical errors make the summary unnatural).

Rating 2:

- *Summary:* A hub for fingerprints tech secure property will be settet in Blaenau Gwent.
- **Rating:** 2 (Some grammatical errors, but it's understandable).

Rating 3:

- *Summary:* A hub for developing fingerprint security technology is being set up in Blaenau Gwent.
- **Rating:** 3 (Completely fluent and grammatically correct).

Instructions:

You are a skilled text quality evaluator specializing in assessing fluency.

Your task is to evaluate the fluency of the following text on a scale from 1 to 3 with respect to the examples above:

{text}

- 1: Poor fluency (e.g., disjointed, unclear, or grammatically incorrect).
- 2: Moderate fluency (e.g., some grammatical errors or awkward phrasing but mostly understandable).
- 3: High fluency (e.g., clear, smooth, and grammatically correct).

Only return the numerical score, enclosed by dollar signs (\$\$), without any additional commentary or explanation.

Table 6: Prompt for Gemini-1.5 Pro to evaluate XSum fluency.

Please read through the generated texts below and rate them, after you finish, please click on the "Submit Evaluation" button and hit "Jump to Dialogue" to proceed to the next datapoint.

Dialogue Summary

Dialogue Summary: #Person1# didn't go to school today. #Person2# wants to skip class tomorrow to go to the movies.

Previous

Next

Dialogue 1 of 4

Model 1

#Person1#: Did you go to school today?
#Person2#: I didn't feel like it.
#Person1#: Why didn't you?
#Person2#: I wanted to go to the movies.
#Person1#: So what are you going to do tomorrow?
#Person2#: I don't know.
#Person1#: What are you going to do?
#Person2#: I don't know.
#Person1#: Did you go to school today?
#Person2#: No, I didn't go.
#Person1#: Did you go to school today?

Is this dialogue comprehensible?

☒ Yes
☐ No

Repetition (1: Excessive, 3: Minimal)

1

1

Fluency/Grammar (1: Poor, 3: Excellent)

1

1

Coherency (1: Illogical, 3: Natural)

1

1

Compatibility (1: Unrelated, 3: Faithful)

1

1

Model 2

#Person1#: Hi, how are you doing today?
#Person2#: I'm doing well, thanks.
#Person1#: So, did you go to school today?
#Person2#: No, I didn't go to school today.
#Person1#: Why not?
#Person2#: I was thinking of skipping class tomorrow to go to the movies.
#Person1#: That sounds like a good idea. I was thinking of doing the same.
#Person1#: Do you think we could talk about it for a moment?
#Person2#: Of course, come on in. It doesn't matter.

Is this dialogue comprehensible?

☒ Yes
☐ No

Repetition (1: Excessive, 3: Minimal)

1

3

Fluency/Grammar (1: Poor, 3: Excellent)

1

3

Coherency (1: Illogical, 3: Natural)

1

3

Compatibility (1: Unrelated, 3: Faithful)

1

3

Model 3

#PEerson2#: I didn't. Didn't you? #1#: Today, so I was going to be so busy today. #2#: 10:00 today, So I was supposed to be doing it.
#PEPerson3#: I was thinking of why I was having it done.
#PePerson4#: That wasn't naughty.##: You didn't plan on having any classes for tomorrow.
#Person2 #3: You were going to skip class tomorrow?
#Pen3#? I won't miss class tomorrow. [
#Pend4#] I don't have any idea of mind.
#PPPerson6#: Do you want to just go to go to the movies?
#PPPerson3] I'd like to miss my class today.
#PBPperson8#: Just I want to skip school today.

Is this dialogue comprehensible?

☒ Yes
☐ No

Repetition (1: Excessive, 3: Minimal)

1

1

Fluency/Grammar (1: Poor, 3: Excellent)

1

1

Coherency (1: Illogical, 3: Natural)

1

1

Compatibility (1: Unrelated, 3: Faithful)

1

1

Model 4

#Person1#: Did you go to school today?
#Person2#: Of course. Did you?
#Person1#: I didn't want to, so I didn't.
#Person2#: That's sad, but have you gone to the movies recently?
#Person1#: That's a switch.
#Person2#: I'm serious, have you?
#Person1#: No, I haven't. Why?
#Person2#: I really want to go to the movies this weekend.
#Person1#: So go then.
#Person2#: I really don't want to go by myself.
#Person1#: Well anyway, do you plan on going to school tomorrow?

Is this dialogue comprehensible?

☒ Yes
☐ No

Repetition (1: Excessive, 3: Minimal)

1

3

Fluency/Grammar (1: Poor, 3: Excellent)

1

3

Coherency (1: Illogical, 3: Natural)

1

3

Compatibility (1: Unrelated, 3: Faithful)

1

3

Additional Comments (optional)

Submit Evaluation

[Jump to Dialogue](#)

Figure 5: An example of human evaluation interface of DialogSum. Each session starts with a comprehensive instructions with examples, followed by model outputs and questions.

Algorithm 1: Training Algorithm for Segment-Level Diffusion (SLD)

Input: Input-output text pairs $(\mathbf{i}, \mathbf{o})$, frozen context encoder with parameter θ_{ctx} , model parameters $\theta_{in} = \{\theta_{enc}, \theta_f\}$ for the encoding and compression, $\theta_{out} = \{\theta_g, \theta_{dec}\}$ for reconstruction and decoding, where $\theta_{enc}, \theta_{dec}$ are initialised from pre-trained language encoder decoder, and $\theta_{rep} = \theta_{in} \cup \theta_{out}$, hyperparameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, temperature τ , batch size N , maximum diffusion steps T .
Output: Trained latent space diffusion model.

1 Training: Latent Representation Learning
2 for each mini-batch of text pairs $(\mathbf{i}, \mathbf{o})$ do
3 1. Output Utterance Segmentation: Segment $\mathbf{o}$ into patches $\mathbf{P} = \{\mathbf{p}^1, \dots, \mathbf{p}^j\}$.

4 2. Latent Representation Encoding: Encode patches into latent space $\mathbf{Z} = \{\mathbf{z}^1, \dots, \mathbf{z}^j\}$.

5 3. Compute Conversion Loss:

$$\mathcal{L}_{cnv}(\theta_{rep}) = - \sum_{l=1}^p \log p(o_l | o_{<l}, \mathbf{p}; \theta_{rep})$$

4. Contrastive Representation Learning: Sample positive and negative examples $(\mathbf{p}^+, \mathbf{p}^-)$ and compute:

$$\mathcal{L}_{cst}(\theta_{in}) = - \log \frac{e^{(\text{sim}(\mathbf{z}, \mathbf{z}^+)/\tau)}}{e^{(\text{sim}(\mathbf{z}, \mathbf{z}^+)/\tau)} + e^{(\text{sim}(\mathbf{z}, \mathbf{z}^-)/\tau)}}$$

5. Noise Robustness Training: Add random noise to patches and latent space $\mathbf{z}$:

$$\mathcal{L}_{adv}(\theta_{out}) = - \sum_{l=1}^p \log p(o_l | o_{<l}, \mathbf{z} + \mathbf{r}_{adv}; \theta_{out}),$$

 where $\mathbf{r}_{adv}$ is adversarial noise computed using a frozen offline copy of θ_{out} .

6 6. Update Parameters: Combine losses and update θ_{rep} :

$$\mathcal{L}_{rep}(\theta_{rep}) = \frac{1}{N} \sum_{\mathbf{p}} (\mathcal{L}_{cnv}(\theta_{rep}) + \lambda_1 \mathcal{L}_{cst}(\theta_{in}) + \lambda_2 \mathcal{L}_{adv}(\theta_{out})).$$

7 end
8 Training: Diffusion for Utterance Planning
9 Freeze θ_{rep} and train θ_R using latent representations.

10 for each mini-batch of context $\mathbf{i}$ and segmented patches $\mathbf{P}$ do
11 1. Generate Corrupted Latent Representations:
12 Sample $\mathbf{Z}_t$ from $q(\mathbf{Z}_t | \mathbf{Z})$.
13 2. Compute De-noising Loss:

$$\mathcal{L}_{noise}(\theta_R) = \mathbb{E}_{\mathbf{P}, \mathbf{i}, t} \|R(\mathbf{Z}_t, t, Enc_{ctx}(\mathbf{i}; \theta_{ctx})) - \mathbf{Z}_{t-1}\|_2^2.$$

3. Strengthen Guidance with Post-training: Freeze θ_g and θ_{dec} and compute:

$$\hat{\mathbf{Z}}_t = R(\mathbf{Z}_{t+1}, t+1, Enc_{ctx}(\mathbf{i}; \theta_{ctx}))$$

$$\mathcal{L}_{rec}(\theta_R) = \mathbb{E}_{\mathbf{P}, \mathbf{i}, t} \|g(\hat{\mathbf{Z}}_t; \theta_g) - g(\mathbf{Z}_t; \theta_g)\|_2^2,$$

$$\mathcal{L}_{dec}(\theta_R) = \mathbb{E}_{\mathbf{P}, \mathbf{i}, t} \left[- \sum_{l=1}^p \log p(o_l | o_{<l}, \hat{\mathbf{Z}}_t; \theta_{dec}) \right].$$

4. Update Parameters: Combine losses and update θ_R :

$$\mathcal{L}_{diff}(\theta_R) = \mathcal{L}_{s2s}(\theta_R) + \lambda_3 \mathcal{L}_{rec}(\theta_R) + \lambda_4 \mathcal{L}_{dec}(\theta_R)$$

14 end
15 Return: Trained model parameters θ_{rep} and θ_R .

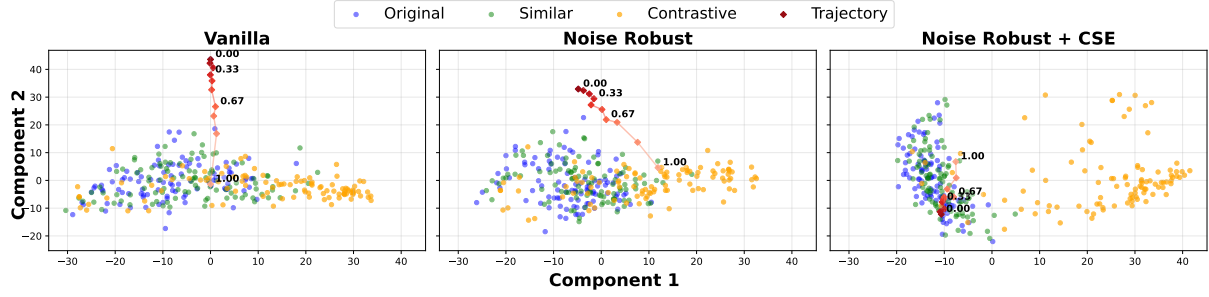


Figure 6: Comparison of PCA 2D projections of latent representations for sampled segmented sentences from ROCStories (Blue), their paraphrases (Green), and out-of-domain (OOD) sentences sampled from CNN/Daily Mail (Orange) under three training paradigms: Vanilla training, Noise Robust training, and Noise Robust + Contrastive learning. The red trajectory illustrates the denoising path of the sentence 'David noticed he had put on a lot of weight recently.' The trajectory is annotated with noise ratios, where 1.0 (Lighter Red) represents pure Gaussian noise and 0.0 (Darker Red) indicates no noise.

Noise Ratio	LD4LG	Adversarial	CSE + Adversarial
Original: David noticed he had put on a lot of weight recently.			
0.00	David noticed he had put on a lot of weight recently.	David noticed he had put on a lot of weight recently.	David noticed he had put on a lot of weight recently.
0.11	David noticed he had put on a lot of weight recently.	David noticed he had put on a lot of weight recently.	David noticed he had put on a lot of weight recently.
0.22	Amanda noticed he had put on a lot of weight recently.	David noticed he had put on a lot of weight recently.	David noticed he had put on a lot of weight recently.
0.33	David he had to give up a thousand of cigarettes.	David noticed he had put on a lot of weight.	David noticed he had put on a lot of weight recently.
0.44	he had put did on a lot of weight recently.	David a had put on a lot of weight.	David noticed he had put on a lot of weight recently.
0.56	he noticed he had put on a lot of weight lately.	David – he had to put on a lot of weight.	David realized he had put on a lot of weight recently.
0.67	He expected he gotten ban on a lot of weight recently.	I had to give a dose of weight.	In mentioned he had put on, a lot of weight recently.
0.78	It would smile a pa heard she had a sign for a little on	you have ad on a lot of food?	lization marche, 15 had jumped on to a lot of weight of a
0.89	Some weeks contacted he's got, ration.	You are the threw in a defo.	The fat, hearing the of rounded.
1.00	The sitter of Edward that she defended, and said.	, he, and the...	I thought, 10thed he would constantly res, sanctwing of

Table 7: Comparison of decoded text segment for varying noise ratios across three models: Vanilla, Noise Only, and CSE + Noise. The original sentence is listed separately for reference.

Title	Lilo's Walk
Model	Generated Text
Gold	Sarah has a dog named Lilo. Sarah took Lilo for a walk. Lilo rolled around in dirt during his walk. Sarah had to take Lilo to the groomer. When Lilo came home from the groomer, he looked beautiful and clean.
LD4LG	Lilo was about to take a walk to school. She saw no people on the grass. Lilio was walking by the grass, she ran every grass. Llo knew she was bored and walked to class.
Flan-T5	Lilo wanted to go for a walk. Lilo's mom took her to the park. Lilo and her mom walked to the park. Lilo had a great time walking to the park. Lilo and her mom walked home together.
Ours	Lilo was walking down the sidewalk for her dog. Lilo has not liked to walk across the woods. Lilo took him to walk in the park. Lilo's mom would walk with her every day. Lilo continued to walk again.

Table 8: Text quality comparison of generated outputs for the story "Lilo's Walk" in ROCStories. The table compares the outputs of LD4LG, Flan-T5, and our model against the gold standard. All model generates admissible story, however, LD4LG has minor word corruptions (Orange).

Title	Dialogue Summary: #Person1# invites Jimmy to go workout and persuades him into working out on arms and stomach.
Model	Generated Text
Gold	#Person1#: Hey Jimmy. Let's go workout later today. #Person2#: Sure. What time do you want to go? #Person1#: How about at 3:30? #Person2#: That sounds good. Today we work on Legs and forearm. #Person1#: Hey. I just played basketball earlier, so my legs are a little sore. Let's work out on arms and stomach today. #Person2#: I'm on a weekly schedule. You're messing everything up. #Person1#: C'mon. We're only switching two days. You can do legs on Friday. #Person2#: Alright. I'll meet you at the gym at 3:30 then.
Flan-T5	#Person1#: Hey, Jimmy, why don't you come work out with me? #Person2#: What are you talking about? #Person1#: I'm talking about working out on your arms and your stomach. #Person2#: What are you talking about? #Person1#: I mean, you lift weights on your arms and you lift weights on your stomach. #Person2#: That's not working. #Person1#: But you lift weights on your arms and you lift weights on your stomach. #Person2#: That's not working. #Person1#: But you lift weights on your arms and you lift weights on your stomach. #Person2#: That's not working. #Person1#: But you lift weights on your arms and you lift weights on your stomach. #Person2#: That's not working. (repeated 7 times more)
LD4LG	#Person1#: Hi, Jimmy. We're going to do some exercises and arm exercises on this weekend. Would you like to join us? #PEerson2#: Yeah, that sounds good for us. #PEPerson3#: Yeah, working on arms with arms, but I've thought I'm going to start on my stomach. #EPerson4#: Working??. I don't think I just have to hold up my weight. #EPerson3#: Why not go on your arms and arms and then put arms on your stomach. Then you can work for for any time.
Ours	#Person1#: Hey Jimmy, how's it going? #Person2#: I'm still not sure, physically. #Person1#: So, what's going on? #Person2#: I'm just not feeling up to it. #Person1#: Come on, let's get moving. #Person2#: What do you mean? #Person1#: Well, we could work on your arms and stomach. #Person2#: Oh, no thank you. I don't know it as well. #Person1#: It's not that hard, I'll show you. #Person2#: Okay, I'll try. #Person1#: That's the spirit!

Table 9: Text quality comparison of generated dialogues for the summary: "#Person1# invites Jimmy to go workout and persuades him into working out on arms and stomach" in DialogSum. The table compares the outputs of LD4LG, Flan-T5, and our model against the gold standard. Flan-T5 generates repetitive text (Brown), while LD4LG sometimes corrupts usernames (Orange).

Meta-Info	Users: [USER_SYS] [USER_0] [USER_1] Cards: 4 9 V U
Previous Turns	[USER_0]HI! I went with flipping the card with the letter U
Model	Generated Text
Gold	[USER_1]I selected U and 9 [USER_1]You need to select 9 also, the other side might be a vowel [USER_0]I completely overlooked that you could select 2 cards. I agree both should be selected [USER_0] [PICK] U 9 [USER_1]the other two cards are irrelevant [USER_0]yes i agree [USER_1]Are you ready to submit the HIT? [USER_0]yes, i submitted my final answer [USER_1] [PICK] U 9 [USER_SYS][FINISHED]
LD4LG	[USER_0] Hello [USER_0] I picked the letter "U" and the number "9". I chose u and 9 because I think those are the ones you need to check to see if the rule is correct? I didn't think about it that way. I was thinking that it should be 2 cards with a vowel and a number on the other side. I also did not see any reason why I chose the same card. [USER_1] I picked U, 9 and [PICK] 4 U. Why did you pick the letter 8 and u? I thought it was 2 cards that would verify the rule. So I picked 2 cards and ur correct answer
Flan-T5	[USER_0] Hi, it was nothing like this. [PICK] U 9 [USER_0] I have chosen the cards to flip over that one. it does say an even number would have to have a vowel and a consonant. [USER_0] You think that would work well to prove either of the 2 cards? [USER_0] I did what you said [USER_1] you should flip them to see if there's a vowel or consonant on the side (apart from the number) [USER_0] hmmm i could imagine choosing them all because the rule needs to be followed. [USER_1] maybe we should keep to u and 9 then? [USER_1] alright I'm gonna submit those two I guess [USER_1] I don't think we need to do all four as its not a vowel so I don't think we need to see it because the rule doesn't state that consonants can't have an even number or vowel [USER_0] Oh really? lol [USER_1] so u and 9? [USER_1] yea it is a vowel and a consonant [USER_1] we're testing the rule so turning them all would test the rule [USER_0] well guess we'll see what happens when we're done [USER_1] it says to ONLY flip the cards that have letters. i think to check the rule that cards with vowels have an even number so that's the only one you need to flip [USER_0] [PICK] U [USER_0] how do we know that the rule doesn't actually apply to all evens [USER_1] we're testing the rule to see if there is a vowel on the other side [USER_1] but it doesn't specify if there's a non vowel on the other side [USER_0] the task asks if there is a number in each of the 3 cards [USER_0] oh yeah so just the card with the vowel would test it [USER_0] yeah :) [USER_0] but then we have to test the other two cards [USER_0][USER_1] hmmm lmao! should we flip them all then? [USER_0] well yeah, mine seems reasonable. [USER_0] tiger sorry im confused lol lol [USER_0] u and 9 maybe too [USER_2] ok so maybe not all 3? [USER_0] yeah u and 9? [USER_0] thats what im going with emuerk? [USER_0] yea I see where you are coming from lol [USER_1] we're trying to test the rule, but its not necessarily involving vowels or consonants [USER_1] so just u and 9? [USER_1] yea tiger, we have to participate anyways [USER_0] yeah yea [USER_0] so u and 9? [USER_0] yea [USER_2] if u is a vowel, it explains the rule about vowels. if v is an odd number it doesn't mean the rule is wrong [USER_0] u and 9? [USER_2] yea u and 9? [USER_0] yeah both would show the rule [USER_2] [PICK] U 9 [USER_2] yeah u and 9 is my thinking too. [USER_0] same [USER_1] so what do we do now [USER_0] submit u and 9? [USER_1] [PICK] U 9 [USER_1] now im thinking u and 9? [USER_2] [PICK] U 9 [USER_2] [PICK] U 9 [USER_SYS] [FINISHED]
Ours	[USER_1] I have the same opinion. [USER_0] What do you think? [USER_0] And I would also need to flip the 9 to make sure there's not a vowel on the back [USER_0] I have the same number [USER_0] why 4? [USER_0] I think we need to flip the 4 as well to see if there is a vowel [USER_0] Oh, that makes sense [USER_0] and flip the U, to see if there is an even number [USER_0] I kind of like this good idea [USER_1] [PICK] U 4 [USER_0] [PICK] U 4 [USER_SYS][FINISHED]

Table 10: Text quality comparison of generated continuations for a DeliData example. The table compares the outputs of LD4LG, Flan-T5, and our model against the gold standard. LD4LG hallucinated a card choice (Red). Flan-T5 generates unnatural repetitive outputs (Brown) and hallucinate an extra user (Red).

Input	Answer the following question by reasoning step by step: antarctic ocean is named after a continent. What’s another ocean that’s named after a landmass? Choices: antarctic, atlantic, pacific, indian ocean, pacific.
Model	Generated Text
Gold	India is a name of a country which is a land mass. So Indian ocean is named after a landmass. Antarctic is always given in the question and the other options are not names of land masses. The answer is indian ocean.
Flan-T5 Base	atlantic ocean is named after a continent. atlantic ocean is named after a landmass. Therefore, the final answer is atlantic.
LD4LG	The atlantic ocean is named for antarcam. So atlantic is named an antarcha ocean. All the other options are not named another ocean named for landmass. The answer is atlantic.
Ours	The antarctic ocean is named to pacific because it is named another an landmass. So the antio is named by atlantic ocean. All the other options are not named by which ant ocean is called another after landmasses. The answer is atlantic.
Flan-T5 Large	Pacific Ocean is named after pacific. Pacific Ocean is named after pacific. Therefore, the final answer is pacific.

Table 11: Models’ Predictions on ECQA: All diffusion models gives a wrong answer that is consistent with the knowledge of their Flan-T5 Base backbone.

Input	Answer the following question by reasoning step by step: If someone brings candy to a group of children what will happen? Choices: listen to music, spring into action, wave goodbye, play chess, experience fear.
Model	Generated Text
Gold	Springing into action means to become active unlike playing chess and experiencing fear. Candies make children spring into action and not listen to music or wave goodbye. The answer is spring into action.
Flan-T5 Base	Spring into action is the act of bringing something to a group of children. If someone brings candy to a group of children, spring into action will happen. Therefore, the final answer is spring into action.
LD4LG	Springing into action leads to spring into action and not play chess or wave fear. Springing onto action which leads to Spring into music unlike listening to music and experiencing fear. The answer is spring into fear.
Ours	Spring into action is an activity unlike listening to music, play chess. Candy makes a group of children springing into action. The answer is spring into action.
Flan-T5 Large	Spring into action is to get excited. Candy is a good thing. Children get excited when someone brings candy to them. Therefore, the final answer is spring into action.

Table 12: Models’ Predictions on ECQA: LD4LG initially mentions the correct answer, but predicts a non-existent answer by blending choices “spring into action” and “experience fear” together.