LaDiDa:
Latent Diffusion for Document Generation with Sequential Decoding

Anonymous EACL submission

Abstract

We present a new document-generation model called **LaDiDa**, which stands for **LA**tent **DI**ffusion for **D**ocument Gener**A**tion with Sequential Decoding. Large language models (LLMs) can create impressive texts, but the qualities of the documents degrade as the output lengthens. Over time, models struggle to maintain discourse coherence and desirable text dynamics, leading to rambling and repetitive results. This difficulty with long-range generation can often be attributed to the autoregressive training objective, which causes compounding errors over multiple steps. LaDiDa is a hierarchical model for improved long-text generation by decomposing the task on the document and sentence level. Our method is comprised of document-level diffusion and sentence-level decoding, where diffusion is used to globally and non-autoregressively plan sentences within a document and decoding is used to locally and sequentially generate those sentences. Compared to autoregressive models, LaDiDa is able to achieve high textual diversity and structural cohesion in long-text generation.

1 Introduction

The success of large language models (LLMs) was enabled in part by the development of autoregressive Transformer models (Vaswani et al., 2017a; Devlin et al., 2019; Radford et al., 2019) that allow contextual information to be captured from a given text. While powerful, autoregressive approaches produce compounding errors that over time cause the generated text to drift away from desired semantics (Xu et al., 2022; Kiddon et al., 2016; Lin et al., 2021). This leads to poor performance in long-form text or document generation, which can be observed through the degeneration of quality and lack of coherent discourse structure (Xu et al., 2020; Hua and Wang, 2020).

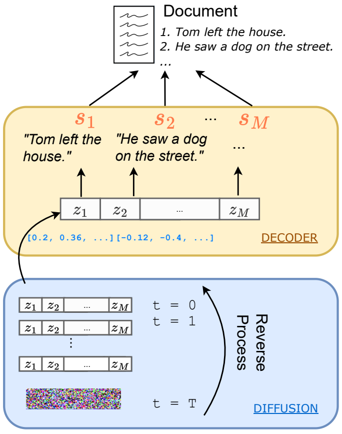

Figure 1: An overview of LaDiDa model architecture components. During generation, a NAR document-level diffusion plans a trajectory of sentence latents, which are then decoded by an AR sentence-level decoder.

To alleviate the incoherence issue over long text, the plan-then-generate framework (Duboue and McKeown, 2001; Hu et al., 2022; Guan et al., 2022; Li et al., 2022a; Fan et al., 2019; Wang et al., 2023) was proposed in a two-step process: content planning, then surface realization. However, such planning-based approaches rely on domain and task-specific heuristics for capturing text dynamics and often incorporate auxiliary objectives. To address these issues, we propose a novel document-generation model **LaDiDa**: **LA**tent **DI**ffusion for **D**ocument Gener**A**tion with Sequential Decoding. LaDiDa is a hierarchical model for long-text generation that leverages diffusion for non-autoregressvie (NAR) sentence planning on the document level and autoregressive (AR) decoders for surface realization at the sentence level.

Our main contributions include: (1) *Exploring diffusion for structural extraction from text and NAR sentence planning.* While diffusion has been successfully applied in text (Li et al., 2022b; Lovelace et al., 2022; Gong et al., 2023; Strudel et al., 2022), no work has utilized diffusion for high-level structural planning of long documents.

Moreover, in contrast to autoregressive models whose computational requirement scales quadratically with the sequence length, our novel NAR planning allows for cheaper and faster scaling. (2) *Leveraging diffusion for unsupervised content planning.* LADIDA does not require extra supervision such as contextual information (Guan et al., 2022; Li et al., 2022a) or auxiliary losses (Hu et al., 2022; Goldfarb-Tarrant et al., 2020) for content planning, and therefore it is adaptable to various tasks and domains. (3) *Hierarchical model for document generation that mixes NAR diffusion and AR Transformers.* LADIDA decouples the task of long-text generation by allowing NAR diffusion to focus on high-level structural information and AR sentence decoders to retain lexical detail. Experiments show that LADIDA outperforms baselines in achieving high textual diversity and structural cohesion on datasets of various domains and lengths.

In this paper, we first summarize the background for latent diffusion in §2. Then we describe the LADIDA architecture in §3 and our experimental setup in §4. We validate the effectiveness of LADIDA in capturing text dynamics by evaluating output coherence via the following research questions:

- RQ1 (§5): Does NAR document-level diffusion preserve global discourse structures?
- RQ2 (§6): Does AR sequential decoding recover local lexical details?
- RQ3 (§7): Does LADIDA generate coherent text through qualitative analysis?

2 Background

2.1 Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are latent variable models that learn to gradually transform random noise drawn from a Gaussian distribution, where sampling is easy, to a sample from an unknown data distribution $p(\mathbf{z})$. It consists of a forward diffusion process, where the original data $\mathbf{z}$ gets iteratively corrupted into Gaussian noise, and a reverse process, where Gaussian noise is iteratively denoised to recover $\mathbf{z}$.

Forward Process Diffusion models first define a forward process that corrupts data into noise. Given a data point $\mathbf{z}_0 \in \mathbb{R}^d \sim p(\mathbf{z})$, the forward process is a Markov chain $q(\mathbf{z})$ over T time steps that produces a sequence of latent variables $\{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_T\}$ that interpolate between the data distribution and a Gaussian distribution by gradually adding noise to $\mathbf{z}_0$ with a noise schedule β_t:

$$q(\mathbf{z}_{1:T}|\mathbf{z}_0) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}) \qquad (1)$$

where $q(\mathbf{z}_t|\mathbf{z}_{t-1}) \sim \mathcal{N}(\sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I})$.

Reverse Process Diffusion models define a reverse process as a learnable generative process that approximates data distribution samples from noise. The reverse process is an inverted Markov chain that iteratively denoises a Gaussian noise sample $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$ to produce increasingly structured latents $\mathbf{z}_T, \mathbf{z}_{T-1}, \ldots, \mathbf{z}_0$ to obtain the clean input $\mathbf{z}_0$. The process is parameterized by ψ such that $p_\psi(\mathbf{z}_{0:T}) = p(\mathbf{z}_T)\prod_{t=t}^{T} p_\psi(\mathbf{z}_{t-1}|\mathbf{z}_t)$.

We can analytically invert the forward process through a denoising transition distribution parameterized by ψ to define $p_\psi(\mathbf{z}_{t-1}|\mathbf{z}_t)$. This transition distribution can be written using $\mu_t(\mathbf{z}_t, \mathbf{z}_0)$ with a closed-form solution and a hyperparameter σ_t as

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0) = \mathcal{N}(\mu_t(\mathbf{z}_t, \mathbf{z}_0), \sigma_t^2\mathbf{I}) \qquad (2)$$

Since $\mathbf{z}_0$ is unavailable during generation, we approximate the original data $\mathbf{z}_0 \approx f_\psi(\mathbf{z}_t, t)$ using a denoising function $f_\psi(\cdot)$ given some noisy latent and time step t. To parameterize $f_\psi(\cdot)$, we train a neural network using a weighted reconstruction loss (Ho et al., 2020):

$$L = \mathbb{E}_{\mathbf{z}_0, t, \mathbf{z}_t}[\|f_\psi(\mathbf{z}_t, t) - \mathbf{z}_0\|_2^2], \qquad (3)$$

where $\mathbf{z}_0 \sim p(\mathbf{z}), t \sim \mathcal{U}(\{1, ..., T\}), \mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{z}_0)$. This allows us to generate an approximate sample $\mathbf{z}_0$ from the true data distribution $p(\mathbf{z})$ by sampling from $p_\psi(\mathbf{z}_{t-1}|\mathbf{z}_t)$ in closed form.

We initially draw Gaussian noise $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$ and denoise the latent using the transition distribution by iteratively sampling $\mathbf{z}_{t-1} \sim p_\psi(\mathbf{z}_{t-1}|\mathbf{z}_t)$ until we obtain $\mathbf{z}_0$. Sampling from the learned model can be performed using ancestral sampling (DDPM) (Ho et al., 2020) or deterministic sampling (DDIM) (Song et al., 2022).

3 LADIDA Architecture

In this section we describe how we use latent diffusion to improve the diversity and structural cohesion of document generation. Our model LADIDA, depicted in Figure 1, uses NAR diffusion to embed document-level semantics and AR VEQ2SEQ decoders to guide sentence-level reconstruction.

3.1 Task Definition

To generate documents, we maximize the log probability of a document D with M sentences. Each D is a sequence of sentences, which can be represented in the token space or latent space. In the token space, we denote D as $\mathbf{s} \coloneqq [s^1, s^2, \ldots, s^M]$. Each sentence s^i in the token space can be mapped via f_{ENC} (see §3.2) to a continuous latent space by constructing a sentence embedding z^i. Then in the latent space, we denote D as $\mathbf{z} \coloneqq [z^1, z^2, \ldots, z^M]$.

Shown in Appendix B, we can reformulate our optimization objective as a loss function $\mathcal{L}$ where

$$\mathcal{L}(D; \psi, \theta) = \mathcal{L}_{\text{diffusion}} + \mathcal{L}_{\text{reconstruction}} \quad (4)$$

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_q \left[L_{\text{DDPM}}(\mathbf{z}_0; \psi) \right] \quad (5)$$

$$\mathcal{L}_{\text{reconstruction}} = \mathbb{E}_q \left[\log p(\mathbf{s}|\mathbf{z}_0; \theta) \right] \quad (6)$$

We optimize Eq. 5 and Eq. 6 separately by training in two stages. In §3.3, we optimize the diffusion objective by training a latent diffusion model over the latent space of documents $\mathbf{z}$. In §3.4, we optimize the reconstruction objective by fine-tuning sequential VEQ2SEQ decoders.

3.2 Encoding Documents

We encode a document D with M sentences in discrete token space $\mathbf{s}$ to a continuous latent space $\mathbf{z} \coloneqq [z^1, z^2, \ldots, z^M]$. We deterministically encode each s^i using a sentence embedding model f_{ENC} such that $z^i = f_{\text{ENC}}(s^i) \in \mathbb{R}^d$. To represent D in the latent space, we concatenate all sentence embeddings z^i to obtain $\mathbf{z} = [z^1 \circ z^2 \circ \ldots \circ z^M] \in \mathbb{R}^{M \times d}$. By independently encoding each sentence and concatenating them, we allow for flexible input length. To encode long documents that span thousands of tokens, we simply increase the dimension M of the document embedding. This encoding method affords us the flexibility to use any sentence embedding model for document encoding.

3.3 Diffusion-Based Document Generation

Once mapped to a continuous latent space $\mathbf{z}$, we optimize Eq. 5 by training a diffusion $f_\psi(\cdot)$ to learn the latent document embeddings $p(\mathbf{z}_0) = p(\mathbf{z})$ over a set of N documents $\mathcal{D} = \{D_1, \ldots, D_N\}$. We introduce special `<BOD>`, and `<EOD>` tokens to demarcate the beginning and the end of a document respectively. This enables the diffusion model to jointly learn the length distribution of the dataset without the need to learn it with another model. While preprocessing documents, we set a maximum number of sentences M. If a document has fewer sentences than M, we append `<EOD>` after the last sentence and pad the rest with zeros vectors. During generation, we truncate sentences that appear after `<EOD>`.

Note that while our sentence embedding model encodes sentences individually, the sentence interrelations are captured in the latent document embedding $\mathbf{z}$ post-concatenation. The diffusion $f_\psi(\cdot)$ is specifically tasked with modelling the distribution of $\mathbf{z}$, which captures inter-sentence dynamics. Therefore, in our architecture design we do not need to include extra contextual computation (e.g. Liu (2019); Xu et al. (2020)) on top of the concatenated sentence embeddings.

3.4 VEC2SEQ Sentence Reconstruction

Here, we detail the steps to optimize Eq. 6. Once we train a diffusion models that learns the distribution of $\mathbf{z}$, we train an AR decoder to decode sentences $\{z^1, z^2, \ldots, z^M\}$ from continuous latent space to the discrete token space $\{s^1, s^2, \ldots, s^M\}$. Formally, we are given a sentence position j and its corresponding encoded vector $z^j = f_{\text{ENC}}(s^j)$ for the target sentence s^j, with optional history lexical context $\{s^{i<j}\} \coloneqq \{s^i, s^{i+1}, \ldots, s^j\}$ for $i < j$. We train a VEC2SEQ model $f_{\text{DEC},\theta}(\cdot)$ to output $\hat{s}^j$ as the reconstruction of the original target s^j:

$$\hat{s}^j = f_{\text{DEC},\theta}(z^j, \{s^{i<j}\}) \approx s^j \quad (7)$$

We configure Eq. 6 by conditioning $f_{\text{DEC},\theta}$ on three different contextual settings that involve z^j and $\{s^{i<j}\}$. These configurations define our three LADIDA variants: (1) **context and vector**, (2) **context only**[1], and (3) **vector only**. For all variants, we optimize Eq. 6 by fine-tuning an autoregressive LM to minimize sentence-level cross-entropy.

Context and Vector (LADIDA-CV) We assume AR dependencies on s^j in a document. Each sentence depends on the previous sentences $\{s^{i<j}\}$ and its latent z_0^j. Let $s^0 =$ `<BOD>`. Eq. 6 becomes:

$$\mathbb{E}\left[\log \prod_{j=1}^{M} p_\theta(s^j | s^{j-1}, \ldots, s^0, z_0^j)\right] \quad (8)$$

Context Only (LADIDA-C) This is a baseline that generates the next sentence s^j solely based on the encoded lexical history $\{s^{i<j}\}$ and ablates the information from the latent sentence plan z_0^j:

[1] Equivalent to the baseline encoder-decoder architecture.

Dataset	# Train/Test	Vocab	S per D	W per S	W per D	Domain
ROCStories	98k/1.5k	28809	5.0	11.9	58.4	Short Stories
Scientific Papers	16k/1k	28734	146.5	37.04	5275.6	Scholarly Articles
WikiSection	2.16k/0.3k	29517	38.17	26.63	973.1	Wikipedia

Table 1: Dataset statistics overview. S, D, W represent the number of sentences, documents and words respectively. For example, “S per D" indicates the number of sentences per document. All values reflect averages.

$$\mathbb{E}_q\left[\log\prod_{j=1}^{M} p_\theta(s^j|s^{j-1},\ldots,s^0)\right]. \tag{9}$$

Vector Only (LADIDA-V) In this variant, each sentence s^j can be reconstructed solely on its latent vector z^j, independently of its previous contexts.

$$\mathbb{E}_q\left[\log\prod_{j=1}^{M} p_\theta(s^j|z_0^j)\right]. \tag{10}$$

This formulation allows parallel decoding of sentences during inference in a fully NAR way.

3.5 Inference

During inference, generation is done in a two-stage process as visualized in Figure 1. We follow four simple steps. (1) Sample Gaussian noise, $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$. (2) Use the learned denoising network $f_\psi(\cdot)$ to gradually denoise $\mathbf{z}_T$ into a meaningful document embedding $\mathbf{z}_0$. (3) Use the learned sequential VEC2SEQ decoder to generate $s^i = f_{\text{DEC},\theta}(z_0^i)$ for all i. (4) Concatenate all s^i up until the <EOD> token to form a document D.

4 Experimental Setup

We detail the model setup, datasets and baselines used in our experiments to answer our RQs.

4.1 Model Instantiation

For the encoder f_{ENC} (§3.2), we used a pre-trained sentence embedding model DEFSENT with [CLS] pooling (Tsukagoshi et al., 2021).[2] For document diffusion, $f_\psi(\cdot)$ (§3.3), the backbone of our model is a denoising network from Lovelace et al. (2022), which is a bidirectional Pre-LN transformer (Vaswani et al., 2017b; Xiong et al., 2020) with 12 layers and a hidden dimension of $d = 768$ with self-conditioning. For the VEC2SEQ decoder $f_{\text{DEC},\theta(\cdot)}$ (§3.4), we fine-tune a pre-trained BART (Lewis et al., 2020) for sentence reconstruction. We illustrate the implementation for LADIDA-CV with BART in Figure 4 in Appendix C.

4.2 Datasets

We test the flexibility of our model in different domains and with varying lengths by experimenting with multiple datasets that vary in semantic content and exhibited structures. An overview of the datasets we used is shown in Table 1. **WikiSection** (Arnold et al., 2019) contains Wikipedia articles with section annotations. Each article introduces a city and has four ordered sections that appear in the form of “[ABSTRACT] text [HISTORY] text [GEOGRAPHY] text [DEMOGRAPHICS] text”. **ROCStories** (Mostafazadeh et al., 2016) contains 98k five-sentence stories that illustrate a variety of causal and temporal commonsense relations between daily events. **Scientific Papers** (Cohan et al., 2018) includes scientific papers collected from the ArXiv OpenAccess repositories. We use this dataset to test long-form generation as its average token length is 5275. We use a subset of 17k documents from the Arxiv split.

4.3 Baselines

To examine the performance of LADIDA on document generation, we include baselines that (1) incorporate the use of latent document planning and (2) those without. (1) For the former, we compare with Time Control (TC) (Wang et al., 2023). TC is a plan-then-generate model that generates documents by first planning a sentence trajectory via a Brownian bridge, then conditionally generates sentences using this latent sequence. In TC, the interrelations among sentences are captured by the latent Brownian bridge, which is learned with a contrastive objective. The decoder in TC is a GPT-2 that follows the standard, multi-sentence prediction scheme that differs from our per-sentence VEC2SEQ decoder. (2) For the latter, we compare with autoregressive LMs without a latent plan: GPT-2 (Radford et al., 2019), and OPT-1.3b (Zhang et al., 2022). OPT is a large Transformer-based LM pre-trained on

[2] Compared to other popular sentence embedding models, we observed DEFSENT has higher VEC2SEQ sentence-recovery capabilities. We use embedding dimension $d = 768$.

180B tokens of web data. We fine-tuned OPT using Low-Rank Adaptation (LoRA) (Hu et al., 2021).

On the shorter ROCStories and WikiSection datasets, we use GPT-2 with a maximum position length of 1024, while on the longer Scientific Papers dataset, we use OPT with a maximum position length of 2048. Note that LADIDA-C is a BART baseline, as it uses a BART decoder to generate the next sentence solely based on the encoded context history and ablates the information from the diffusion-generated latent sentence plan. We fine-tuned all the considered models on the target datasets. As a reference to compare the generated texts of different models, we also compute reference values ("Ref.") using samples from the evaluation set of each dataset.

4.4 Experiments

We evaluate the ability of LADIDA on document generation. After training, we generate 1000 documents on ROCStories and WikiSection and 300 documents for Scientific Papers. We report the text-generation performance in Table 2 and 3. We assess the model ability to preserve textual structure both globally through document diffusion and locally through sequential generation. In §5 we focus on extraction of global structure from text with metrics such as LMM, SMM, and MAUVE (RQ1). In §6 we focus on local lexical details with metrics such as sentence-level PPL and token diversity (RQ2). In §7, we qualitatively analyze coherence with human evaluation (RQ3).

5 RQ1: Does diffusion preserve global document structure?

We quantify global structural cohesion with the following metrics. For length consistency: Length Mismatch (**LMM**), Section Mismatch (**SMM**), and Section Ordering (**Ord**). For global textual similarity: **MAUVE** (Pillutla et al., 2021).

5.1 Evaluation Metrics

LMM. LMM is the proportion of the absolute difference of the mean length between that of the generated set and the evaluation set that measures whether the generated text length matches the length distribution of the reference text.

SMM. SMM is the LMM between the generated text for each section (e.g. [ABSTRACT]) and the gold evaluation set. The result is averaged across all sections.

Ord. Ord. is a section-ordering metric to evaluate whether all section names appear in the correct order in the generated text. It is 1 if all four section names appear in the right order at least once.

MAUVE. Pillutla et al. (2021) uses divergence frontiers to compare the distribution of generated text with that of reference text in a quantized embedding space. We chunk the text in sections of a context window c and report the average MAUVE averaged over sections.[3]

5.2 Results

LADIDA-V and LADIDA-CV consistently achieve strong performance across all datasets in preserving global length consistency, with the lowest LMM and SMM. The fine-tuned GPT-2 frequently undershoots long sections on WikiSection and produces shorter stories, leading to worse LMM and SMM.

On section ordering, TC achieves the highest Ord (0.93), followed by LADIDA-CV (0.83). We hypothesize that this stems from TC's contrastive learning objective that encourages accurate sentence ordering comparisons between two sentence latents, while LADIDA's reconstruction objective does not explicitly compare ordering within latents. The AR baselines without planning (GPT-2 and OPT) perform poorly in matching reference text length and ordering section names (Ord=0.1).

We also observe that LADIDA-CV resembles the reference text in terms of textual similarity, with highest MAUVE on ROCStories (0.929) and Scientific Papers (0.985). In Fig.2, we plot the average MAUVE between the generated text against reference text of Scientific Papers. LADIDA-CV maintains global structure with high MAUVE throughout the document due to its diffusion planning $f_\psi(\cdot)$, whereas OPT fails to maintain text similarity on longer time position and deviates as length increases. Recall that LADIDA-C is a baseline BART that removes the latent document plan during prediction. On WikiSection in Table 2, ablating the diffusion-generated latents $\mathbf{z}$ leads to a drop in MAUVE (e.g. 0.65 to 0.39 when changed from LADIDA-CV to LADIDA-C), suggesting that the observed improvement indeed stems from diffusion. The results illustrate the benefits of using diffusion to globally plan the trajectory of sentences follow-

[3] c is set to be 512 and 1024 on WikiSection and Scientific Papers respectively. We use GPT-2-large as the embedding model and set MAUVE scaling factor to be 2.

WikiSection

Models	S-PPL (decoder) ↓	MAUVE ↑	LMM ↓	SMM ↓	Ord ↑	Uniq ↑	Div ↑	Mem ↓
Ref.	-	1.0	0.0	0.0	1.0	21276	0.039	0.22
TC	6.99	0.67	0.213	0.483	**0.93**	9016	0.0074	0.28
GPT-2	11.23	**0.76**	0.396	0.411	0.10	11737	0.017	0.29
BART (LaDiDa-C)	11.95	0.39	0.319	0.619	0.32	1372	0.0010	0.53
LaDiDa-V	10.05	0.58	0.249	**0.303**	0.36	**12210**	**0.024**	0.27
LaDiDa-CV	**5.72**	0.65	**0.078**	0.338	0.83	8232	0.0097	**0.23**

ROCStories

Models	S-PPL (decoder) ↓	MAUVE ↑	LMM ↓	Uniq ↑	Div ↑	Mem ↓
Ref.	-	1.0	0.0	4720	0.350	0.345
TC	16.10	0.662	0.410	3222	0.302	**0.327**
GPT2	61.37	0.880	0.078	3534	0.338	0.489
BART (LaDiDa-C)	13.21	0.365	0.031	1712	0.025	0.565
LaDiDa-V	1.68	0.855	**0.015**	**4100**	0.403	0.359
LaDiDa-CV	**1.52**	**0.929**	0.032	4096	**0.408**	0.344

Table 2: Language generation performance on WikiSection and ROCStories.

Scientific Papers

Models	MAUVE ↑	LMM ↓	Uniq ↑	Div ↑	Mem ↓
Ref.	1.0	0.0	15993	0.0580	0.449
OPT-1.3B	0.971	0.297	8858	0.0021	0.339
LaDiDa-CV	**0.985**	**0.098**	**9248**	**0.0095**	**0.338**

Table 3: Results on Scientific Papers.

ing a desired document structure in the reference text, answering RQ1 positively.

6 RQ2: Can Vec2Seq decoders recover lexical details?

We evaluate the ability of the sequential decoder $f_{\text{DEC},\theta(\cdot)}$ to recover local lexical dynamics with Sentence Level Perplexity (**S-PPL**). To measure lexical details such as diversity and novelty of the generated text, we use Unique Tokens (**Uniq**), Diversity (**Div**), and Memorization (**Mem**).

6.1 Evaluation Metrics

S-PPL. To evaluate the Vec2Seq decoder $f_{\text{DEC},\theta}(\cdot)$, we derive a sentence-level perplexity metric S-PPL based on Eq. 6. Given each sentence s has n words, we derive the metric from the entropy rate of a sentence $(1/n)H(s)$:

$$\text{S-PPL} = 2^{-\frac{1}{n}H(s)} = 2^{-\frac{1}{n}\frac{1}{M}\sum_{s\in D}\log p(s)}. \quad (11)$$

We compute S-PPL on the evaluation set by substituting $\log p(s)$ in Eq. 11 with the probability of s predicted by the different decoders specified in §3.4. Note that S-PPL measures the performance of $f_{\text{DEC}\theta}(\cdot)$ only and assumes access to the target sentence vector z^j when predictng s^j. It does not evaluate the diffusion performance.

Div. To measure the diversity of the generated text (Su et al., 2022; Lovelace et al., 2022), we use **diversity** $= \prod_{n=2}^{4} \frac{|\text{unique n-grams}(\{\mathbf{w}_i\})|}{|\text{total n-grams}(\{\mathbf{w}_i\})|}$, where $\{\mathbf{w}_i\}$ are generated samples.

Uniq. Uniq denotes the number of unique tokens.

Mem. To quantify memorization, Mem is the proportion of generated 4-grams in the training set.

6.2 Results

We observe that LaDiDa-CV and LaDiDa-V achieve low S-PPL (Table 2), suggesting that our Vec2Seq sequential decoder $f_{\text{DEC},\theta(\cdot)}$ is able to locally preserve sentence-level dynamics. The sentence latents in LaDiDa are helpful in guiding generation. We also measure BLEU scores for the reconstructed evaluation set in Appendix D.

In Fig. 3, we plot Avg. S-PPL vs. sentence position on WikiSection. On average, the entropy for sentence position 9 is the highest, meaning the models find it most difficult to predict sentences at this position. Position 9 corresponds to the [HISTORY] section of the dataset, which consists of diverse texts that introduce the history of a city. S-PPL is lower at the beginning and later sections of the dataset, which is comprised of sentences of similar textual structures that are easier to predict. In both cases LaDiDa-CV maintains the best S-PPL.

As mentioned previously, we observe a high LMM for TC (0.41) on ROC. After manual inspection, we found that although TC generates reasonable and fluent stories, each sentence is longer than those in the reference text. We hypothesize that this is due to their standard LM finetuning objective during decoding, which does not encourage

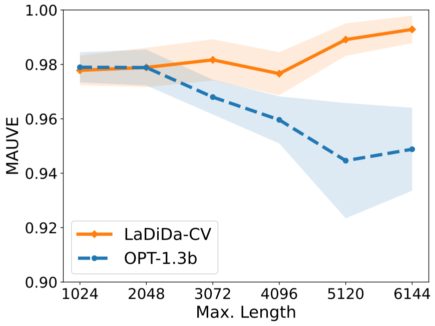

Figure 2: MAUVE between generated and reference texts in chunks of 1024 on Scientific Papers. LaDiDa-CV maintains global structure with high MAUVE, beating OPT that fails to maintain long-range text similarity. The shading shows one STD over 30 random seeds.

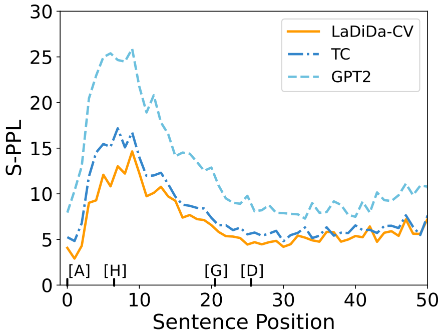

Figure 3: Avg. S-PPL v.s. sentence position on WikiSection. LaDiDa-CV maintains the lowest S-PPL throughout the article. [A], [H], [G], [D] signify avg. starting positions for the four sections.

recovering each sentence on a sentence-level.

For novelty and diversity, LaDiDa-CV and LaDiDa-V outperform the BART baseline on all datasets with higher Div and lower Mem, suggesting that conditioning on latent vectors $\mathbf{z}$ produces more diverse and novel text. To analyze whether there is any performance difference brought by conditioning on the lexical context, we show examples of texts generated with LaDiDa-CV and LaDiDa-V from the *same* latent $\mathbf{z}_0$ sampled from diffusion in Fig. 8 in Appendix F. As LaDiDa-V decodes each sentence independently, the names that appear in the stories sometimes deviate. However, the $\mathbf{z}_0$ sampled from diffusion still preserves story discourse. Conditioning on lexical context (LaDiDa-CV) leads to more consistent entity mentions. In Tables 2 and 3, LaDiDa-CV consistently achieves the best overall performance, suggesting the benefits of using both lexical context and vectors in our Vec2Seq setting. Our Vec2Seq method of sequential generation is able to locally preserve sentence-level dynamics and fine-grained details with low S-PPL and high lexical diversity, confirming RQ2.

Models	Story-like ↑	Style Sim. ↑
TC	$2.367_{0.949}$	$3.467_{0.745}$
GPT-2	$3.300_{0.651}$	$3.114_{0.613}$
LaDiDa-CV	$\mathbf{3.667}_{0.549}$	$\mathbf{4.133}_{0.537}$

Table 4: Human evaluation results on ROCStories. All Krippendorff's $\alpha \geq 0.38$.

7 RQ3: How does LaDiDa perform qualitatively?

To answer RQ3, we include human evaluations in §7.1. We also add speed analysis in §7.2 to examine the speed variation among the LaDiDa variants.

7.1 Human Evaluation

We hire seven proficient English speakers as human judges to evaluate model outputs on the ROC dataset on a scale of 1 (worst) to 5 (best) for: (1) Story-like, which measures how coherent a story is, and (2) Style Similarity to Reference, which measures the similarity in sentence arrangement and tone between the output and reference text. Results in Table 4 show LaDiDa-CV outperforms GPT-2 and TC in producing coherent stories resembling the reference. For both, Krippendorff's $\alpha \geq 0.38$. We include the evaluation guidelines in Table 14 in Appendix. On the longer datasets, we observe that LaDiDa is good at topic maintenance throughout the document regardless of text length, while AR models meander as the length increases. We also observe that our model is better at modeling coarse-grained structure than precise semantic details. Examples of generated text on all datasets are shown in Appendix F. We find that the sampled texts sometimes show grammatical errors or repetitive words with relatively lower fluency. We hypothesize that this is due to the noise in the diffused samples $\mathbf{z}_0$. However, they preserve text structure in terms of length and style. For example, the generated ArXiv paper has a "plot" in the middle of the paper and "acknowledgement" at the end, resembling the structure of a paper.

7.2 Speed Analysis

Inference for LaDiDa involves two steps. First, document latents are generated via diffusion sampling. On the Scientific Papers dataset, DDIM sampling with 250 time steps takes around 6.4 seconds to generate one latent for a document on one GTX 1080 for all variants of LaDiDa. The average speed of generating one document on Scientific Papers in seconds is shown in Table 5. For LaDiDa, the results shown include both the diffusion sampling step and the decoding step. For the fully parallel LaDiDa-V, we generate 32 sentences in parallel. The quick inference time for LaDiDa-V illustrates the benefit of fully NAR decoding. In the future we plan to examine the speed-performance tradeoff of using context vs. using the latents. We also compare the speed between generating sentence-by-sentence with generation at once in Table 7 in Appendix E.

OPT-1.3b	LaDiDa-CV	LaDiDa-V
483.8	223	17.7

Table 5: Average speed (in seconds) of generating one document on Scientific Papers with one GTX 1080.

8 Related Work

Planning-based Text Generation. Traditional planning-based text generation (Duboue and McKeown, 2001; Hu et al., 2022; Guan et al., 2022; Li et al., 2022a; Fan et al., 2019; Yao et al., 2019; Wang et al., 2023) typically first decides the high-level structures via a content planning component, then decodes the text via surface realization. Works such as Fan et al. (2019); Guan et al. (2022) usually tailor their content-planning module to the type of text to be generated or leverage external context information, e.g. key phrases or event paths as extra supervision, and often incorporate an auxiliary objective (Goldfarb-Tarrant et al., 2020; Hu et al., 2022) for learning inter-sentence relations. In comparison, the diffusion planning stage for our model is fully unsupervised without auxiliary losses and is adaptable to various tasks and narrative types.

Diffusion Models for Text. Diffusion on text has been explored in both discrete and continuous spaces. In the discrete space, Hoogeboom et al. (2021) and Austin et al. (2023) explored discrete corruption processes with categorical noise on transition matrices. In the continuous space, Li et al. (2022b) and Gong et al. (2023) jointly train token embeddings with diffusion models while Strudel et al. (2022) initialize token embeddings with pretrained weights. Lovelace et al. (2022) trained diffusion in the continuous latent space of a pretrained BART model. Existing works perform text diffusion at a sequence level, but no prior research explores diffusion for leveraging high-level structural information in a document hierarchically.

Hierarchical Text Generation. Classical language models struggle to capture high-level dependencies in long texts, so hierarchical approaches have been proposed that decompose long generation to sentence and word levels (Li et al., 2015; Liu and Lapata, 2019). Li et al. (2015) used LSTMs to hierarchically build a paragraph embedding from sentence embeddings. Wang et al. (2023) used a Brownian bridge trained with a contrastive objective to obtain sentence latents which are then decoded using GPT-2. Spangher et al. (2023) explored news generation with a sequence of local control codes. However, most hierarchical methods assume autoregressive dependencies for latent sentence plans. To the best of our knowledge, no other work has explored non-autoregressive sentence planning for long-text generation, which allows for cheaper and faster scaling.

Generation from Sentence Embeddings LLM-based sentence embeddings (Reimers and Gurevych, 2019; Cer et al., 2018; Tsukagoshi et al., 2021; Conneau et al., 2017) preserve semantic information and capture linguistic properties and have been used in various downstream tasks, e.g. semantic retrieval. A few works (Cideron et al., 2022; Montero et al., 2021; Wu and Zhao, 2022) have explored the generative ability of sentence embeddings for semantic-structure preservation. They rely on learning sentence bottleneck representations from pretrained LMs for generation.

9 Conclusion & Future Work

We present an unsupervised hierarchical model for improved long-text generation by decomposing the document-generation task into document-level NAR diffusion and sentence-level AR decoding. Our method achieves strong structural modeling for documents both globally and locally. We plan to improve the fully NAR variant of our model in the future for fast, parallel inference and extend our work to conditional settings to exploit the advantages of diffusion in future planning, which are challenging for AR models.

Limitations

In § 6, we observe that the sampled texts from our model generates less fluent texts compared with AR baselines and show grammatical errors or repetitive words. We hypothesize that one reason is the noise that exists in the denoised document latent sample. Another reason could be that the decoder hidden states in our AR decoder is perturbed by signals from the additional sentence embeddings. The problem could be alleviated by further model tuning and advancement on diffusion modeling. Efficiency wise, as our best performing model LaDiDa-CV relies on AR decoding, it does not fully exploit the advantage of a NAR diffusion planning. However, improvement on the NAR LaDiDa-V model would lead to generation of higher-quality texts in a parallel setting. Concerning potential risks, our approach depends on pre-trained language models, such as BART, which could potentially carry problematic biases.

References

Sebastian Arnold, Rudolf Schneider, Philippe Cudre-Mauroux, Felix Gers, and Alexander Löser. 2019. Sector: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2023. Structured denoising diffusion models in discrete state-spaces.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder.

Geoffrey Cideron, Sertan Girgin, Anton Raichuk, Olivier Pietquin, Olivier Bachem, and Léonard Hussenot. 2022. vec2text with round-trip translations.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Pablo A. Duboue and Kathleen R. McKeown. 2001. Empirically estimating order constraints for content planning in generation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 172–179, Toulouse, France. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. Diffuseq: Sequence to sequence text generation with diffusion models.

Jian Guan, Zhenyu Yang, Rongsheng Zhang, Zhipeng Hu, and Minlie Huang. 2022. Generating coherent narratives by learning dynamic and discrete entity states with a contrastive framework.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Zhe Hu, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Hua Wu, and Lifu Huang. 2022. Planet: Dynamic content planning in autoregressive transformers for long-form text generation.

Xinyu Hua and Lu Wang. 2020. PAIR: Planning and iterative refinement in pre-trained transformers for long text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents.

Qintong Li, Piji Li, Wei Bi, Zhaochun Ren, Yuxuan Lai, and Lingpeng Kong. 2022a. Event transition planning for open-ended text generation.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022b. Diffusion-lm improves controllable text generation.

Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. 2021. Limitations of autoregressive models and their alternatives. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5147–5173, Online. Association for Computational Linguistics.

Yang Liu. 2019. Fine-tune bert for extractive summarization.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Weinberger. 2022. Latent diffusion for language generation.

Ivan Montero, Nikolaos Pappas, and Noah A. Smith. 2021. Sentence bottleneck autoencoders from transformer language models.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2022. Denoising diffusion implicit models.

Alexander Spangher, Xinyu Hua, Yao Ming, and Nanyun Peng. 2023. Sequentially controlled text generation.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. 2022. Self-conditioned embedding diffusion for text generation.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation.

Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. DefSent: Sentence embeddings using definition sentences. pages 411–418, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Rose E Wang, Esin Durmus, Noah Goodman, and Tatsunori Hashimoto. 2023. Language modeling via stochastic processes.

Bohong Wu and Hai Zhao. 2022. Sentence representation learning with generative objective rather than contrastive objective.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

A Diffusion Models

Training a DDPM is performed by optimizing the variational bound on negative log-likelihood of $\mathbf{z}_0$:

$$L_{\text{VLB}(\psi)} = \mathbb{E}_q \left[\log \frac{p_\psi(\mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \right] \leq \log p_\psi(\mathbf{z}_0). \tag{12}$$

This objective is simplified in Ho et al. (2020) to be a weighted reconstruction loss:

$$L_{\text{simple}(\psi)} = \mathbb{E}_{\mathbf{z}_0,t,\mathbf{z}_t}[\|f_\psi(\mathbf{z}_t,t) - \mathbf{z}_0\|_2^2], \tag{13}$$

where $\mathbf{z}_0 \sim p(\mathbf{z}), t \sim \mathcal{U}(\{1,...,T\}), \mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{z}_0)$. $f_\psi(\cdot)$ is the denoising function parameterised using a neural network.

B Task Derivation

$$\begin{aligned}
\log p(D) &:= \log p(s^1, s^2, \cdots, s^M) \\
&= \log \int p(s^1, \cdots, s^M, z^1, \cdots, s^M) d\mathbf{z} \\
&= \log \int p(\mathbf{s}, \mathbf{z}) d\mathbf{z} \\
&= \log \int \frac{p(\mathbf{s}, \mathbf{z}) q(\mathbf{z}|\mathbf{s})}{q(\mathbf{z}|\mathbf{s})} d\mathbf{z} \\
&= \log \mathbb{E}_{q(\mathbf{z}|\mathbf{s})} \left[\frac{p(\mathbf{s}, \mathbf{z})}{q(\mathbf{z}|\mathbf{s})} \right] \\
&\geq \mathbb{E}_{q(\mathbf{z}|\mathbf{s})} \left[\log \frac{p(\mathbf{s}, \mathbf{z})}{q(\mathbf{z}|\mathbf{s})} \right]
\end{aligned} \tag{14}$$

The last row of Eq. 14 is the ELBO that we would like to maximise. We rewrite the ELBO as:

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{z}|\mathbf{s})} \left[\log \frac{p_\theta(\mathbf{s}|\mathbf{z}) p(\mathbf{z})}{q_(\mathbf{z}|\mathbf{s})} \right] \\
&= \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{s})} \log p_\psi(\mathbf{z})}_{(1)} + \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{s})} \log p_\theta(\mathbf{s}|\mathbf{z})}_{(2)} \\
&\qquad - \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{s})} q(\mathbf{z}|\mathbf{s})}_{(3)}
\end{aligned} \tag{15}$$

We empirically maximise Eq. 15 by breaking the equation into three terms. Term (1) is the prior distribution for the continuous sentence latents, which we model using a DDPM parameterised by ψ. Term (2) is the reconstruction term that predicts sentences from their latent vectors, which we model using a Vec2Seq decoder parameterised by θ. Term (3) represents a document encoder that maps a document to a continuous latent space. We use an off-the-shelf sentence embedding model that deterministically maps a sentence to a latent vector. This way, term (3) is not parameterised and therefore omitted. We maximise term (1) and term (2) separately in a two-stage process. With the prior as a diffusion model, we formulate the document generative process as:

$$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{16}$$

$$\mathbf{z}_{t-1|t} \sim p_\psi(\mathbf{z}_{t-1}|\mathbf{z}_t) \quad \forall t \in [T, \ldots, 1] \tag{17}$$

$$\mathbf{s} \sim p_\theta(\mathbf{s}|\mathbf{z}_0) \tag{18}$$

where Eq. 16 and Eq. 17 define a prior distribution over the latent variable of sentence vectors, which is modelled with a reverse diffusion process, and Eq. 18 is a decoder that maps the denoised vectors $\mathbf{z}_0$ at $t = 0$ to the token space. Omitting term (3) and taking the latent $\mathbf{z}$ to be the final denoised sentence vector $\mathbf{z}_0$, we rewrite E.q 15 to be

$$\underbrace{\mathbb{E}_{q(\mathbf{z}_0|\mathbf{s})} \log p_\psi(\mathbf{z}_0)}_{(1)} + \underbrace{\mathbb{E}_{q(\mathbf{z}_0|\mathbf{s})} \log p_\theta(\mathbf{s}|\mathbf{z}_0)}_{(2)} \tag{19}$$

in which a lower bound on term (1) can be expressed in Eq. 12. Therefore, maximising Eq. 19 becomes maximising

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{z}_0|\mathbf{s})} \log p_\psi(\mathbf{z}_0) + \mathbb{E}_{q(\mathbf{z}_0|\mathbf{s})} \log p_\theta(\mathbf{s}|\mathbf{z}_0) \\
&\geq \mathbb{E}_{q(\mathbf{z}_0|\mathbf{s})} \left[\log \frac{p_\psi(\mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \right] + \mathbb{E}_{q(\mathbf{z}_0|\mathbf{s})} \log p_\theta(\mathbf{s}|\mathbf{z}_0),
\end{aligned} \tag{20}$$

leading to the final loss function:

$$\mathcal{L}(D; \phi, \psi) = \underbrace{\mathbb{E}_q \left[L_{\text{DDPM}}(\mathbf{z}_0; \psi) \right]}_{\text{diffusion}} + \underbrace{\mathbb{E}_q \left[\log p_\theta(\mathbf{s}|\mathbf{z}_0) \right]}_{\text{reconstruction}} \tag{21}$$

In practice, we train in two stages for the two terms above. We first fit a latent diffusion model over the latent space of documents $\mathbf{z}$ that comprises of deterministic sentence embeddings encoded by an encoder once. Then, we experiment with different Veq2Seq decoding methods for the reconstruction term.

C Implementation Details

For our proposed methods, we train $f(\psi)$ and $f_{DEC}(\theta)$ independently in a two-stage process. Therefore the two stages can be trained in parallel while freezing the other. For ROCStories, WikiSection and Scientific Papers, we choose the maximum number of sentences per document M to be 5, 64, and 256 respectively. The sentence embedding dimension d is 768 for all datasets. Therefore, the latent space for diffusion is 5×768, 64×768, 256×768 respectively. For diffusion sampling, we use DDIM with sampling step=250. For more details of the denoising Transformer model architecture please see Lovelace et al. (2022).

For all $f_{\text{DEC},\theta}(z_0^i)$ models below we build on BART-base. For $f_{\text{DEC},\theta}(z_0^i)$ that takes context as Bart encoder input, we truncate them from the left and condition on the most recent 1024 tokens. Fig. 4 illustrates the architecture for LaDiDa-CV. For LaDiDa-C, we do not add any embeddings to the decoder input. For LaDiDa-V, we change the text input to BART encoder to be embeddings of the target sentence.

When sampling from the various Vec2Seq decoders, we use top-p sampling with p=0.96. For TC, we use $d = 16$ following the experiments in the original paper. On WikiSection, we enable force generation for TC and disable it on ROCStories. For OPT-1.3b with LoRA, we finetune LoRA for 2 epochs. Following §3.2, we add an `<EOD>` token to the LMs to help control the output length during inference. We trained all our models on 1 RTX 2080 for 3 days. We experimented with different architectures for the reconstruction task $f_{\text{DEC},\theta}(z_0^i)$ including GPT2, T5, and BART. We found that empirically BART showed the best result.

D Decoder Reconstruction

We observe that for our Vec2Seq decoders trained with BART, the simpler sentences get reconstructed better. e.g. (City A has population B), while longer sentences get reconstructed to semantically similar ones. Here we report the BLEU and Rouge scores for the reconstructed documents in the evaluation set of ROCstories and WikiSection.

E Speed Analysis

We compare the speed of our sentence-by-sentence generation in LaDiDa with the conventional multi-sentence generation scheme (GPT-2), with results in Table 7. The task of sequential generation, where outputs are generated sentence-by-sentence iteratively, was first proposed in Spangher et al. (2023), with the aim of imposing structure on long-range text. Despite its lower speed, this approach produces text that deviates less from the reference text, as each sentence is guided by its own latent code. This benefit is significant when it comes to long-form generation, where generating at once using models with long context window (e.g. OPT) is expensive and slow.

	WikiSection		ROCStories	
	BLEU	Rouge 1	BLEU	Rouge 1
LaDiDa-V	0.439	0.687	0.5626	0.721
LaDiDa-CV	0.423	0.671	0.531	0.724

Table 6: Performance of document reconstruction with Vec2Seq decoder on WikiSection and ROCstories.

GPT-2	LaDiDa-CV
47.6	102.7

Table 7: Average speed (in seconds) of generating 100 documents on the ROC dataset with one GTX 1080, with sentence-by-sentence generation (LaDiDa-CV) and generation at once (GPT-2).

F Generated Examples

We show examples of generated text on ROCStories, WikiSection and Scientific Papers in Table 9, 10, and 11.

G Human Evaluation

We recruit seven graduate students who are proficient English speakers. For evaluation guidelines, see Table 14.

Vector Only (LaDiDa-V)	Context and Vector (LaDiDa-CV)
Samantha has a physics test tomorrow. She has not studied at all. She decided to stay up all night studying. When she got to class, she fell asleep when she fell. Luckily, **Clara**'s teacher postponed her test.	Samantha has a physics test tomorrow. She has not studied at all. She decided to stay up all night studying instead. When she got to class, she fell asleep immediately. But, **Samantha**'s teacher postponed her test.

Table 8: Samples generated by two Vec2Seq decoders on the same diffusion sample $\mathbf{z_0}$ in ROCStories. Conditioning on context leads to more consistent entity mentions.

LaDiDa-CV on WikiSection
[ABSTRACT] Kirtland is a city in Carver County, Minnesota, United States. The population was 15,323 at the 2010 census. Situated in the central part of the Minneapolis-St Paul metroplex, Kirtland is a bedroom community for Minneapolis-Saint Paul, the six-largest urban area in the U.S., and is one of the fastest growing suburbs in the Minneapolis metropolitan area. [HISTORY] Kirtland was founded in 1899. The city is located in the north central portion of the state and is the regional center for north-central Minnesota. Kirtland Center is located in the central portion of downtown Minneapolis, the Mall of the Minneapolis–St Paul area. On August 23, 2006, an F3 tornado hit Kirtland, causing severe damage to the entire city and killed three people. Many of the buildings were destroyed or razed for the downtown. Money Magazine ranked Kirtland the #17 best place to live in the United States according to Money magazine in 2012. [GEOGRAPHY] U.S. Highway 59 and Minnesota State Highway 15 (Ch Minnesota Highway 15) are three of the main routes in the community. The Zumbrove Forest Preserve is a part of the city park in the far west corner of the City, and the Forest of the river just outside the city limits. According to the United States Census Bureau, the city has a total area of, of which, is land and is water. [DEMOGRAPHICS] Kirtland is located in Minnesota's 2nd congressional district, and is represented in the Minnesota House of Representatives by Senator Paul Paul Paul (R-6), a Republican. The city is located in the Kirtland Basin, which is bordered by Ojibwe National Forest and is the easternmost point of growth in the U.S.
[ABSTRACT] Ōshū (Ō Shū-shi) is a city located in Iwate Prefecture, Japan. s of 1 2017, the city had an estimated population of 59,345, and a population density of 151 persons per km² in 26,304 households. The total area of the city is. [HISTORY] The area of present-day Ōshū was part of ancient Mutsu Province, and has been settled since at least the Jōmon period by the Emishi people. [HISTORY] The area around Ōshū has been settled since at least the Jōmon period, and numerous shell middens found in the area from at least 900 years ago. Many Jōmon period archaeological remains have been found in the Ōshū area, along with numerous shell middens. During the later portion of the Heian period, the area was ruled by the Abe clan. During the Edo period, the area was part of the holdings of Sendai Domain under the Tokugawa shogunate. In the post-Meiji restoration cadastral reforms, the area of present-day Ōshū was organised into Kiso District, Gifu prefecture. The modern town of Ōshū was established on April 1, 1889, by the merger of the modern municipalities system with the villages of Kitakami and Kiso. The village of Ōshū was merged with the village of Kiso on April 1, 1954, and from Kiso District on April 30, 1954. On April 1, 2005, the town of Ōshū was merged with the villages of Kiso and Kiso, and the village of Shūchi (both from Kiso District). [GEOGRAPHY] Ōshū is in the Nōbi Plain of Iwate Prefecture, bordered to the north and south. The highest mountain in the prefecture is the Ōshū Mountains, and the highest point in Iwate Prefecture, with the Kiso Mountains in the Nōbi Mountains to the east and the Kitakami Mountains on the south. The Kiso River flows through the city. [DEMOGRAPHICS] Per Japanese census data, the population of Ōshū has increased gradually over the past 40 years.. The city is in the heart of Kiso Province, and has been ruled by the Ōshū clan (1st generation) and the Kiso clan. The city is in the heart of Kiso Province, and has been ruled by the Ōshū clan (1st generation) and the Kiso clan.

Table 9: Samples generated on WikiSection

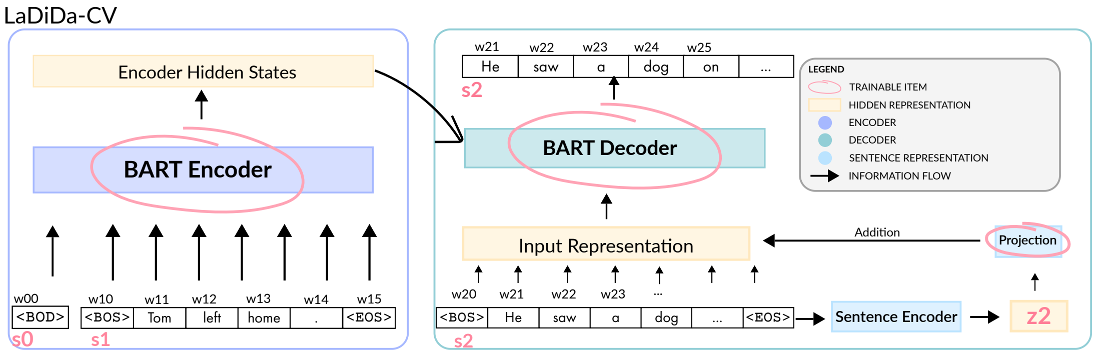

Figure 4: An overview of LaDiDa-CV decoder during training. Lexical context s^0, s^1 passes through BART encoder while target sentence s^2 and vector z^2 pass through the decoder to reconstruct s^2.

LaDiDa-CV on ROCStories
I made a list of the grocery list. I drove to the store. I paid for the items on the list. The cashier handed my items in for me. I paid for groceries and returned home.
John and his mom took a boat ride on the Lake Michigan. They paddled on the boat and watched the boat ride long in the water. Soon after they got close a large shark came up in the water. John and his mom rushed to their seats, seeing the danger. John and his mom ran to shore to save them when they caught the shark.
Jane needed to get her windowsill to update her window. She put up all the windowsill on her windows and windows. She went to check the weather forecast. There was a storm brewing!. Jane raced back to the house from her windows.
I love to have the trampoline fly. It fly back and forth in the world. It was very fun. It plays around all day. We have to go clean it up and trash

Table 10: Samples generated on ROCStories. Sometimes the decoded samples have repetitive words and produce less sensible stories.

LaDiDa-CV on Scientific Papers
the most distant galaxies in the local universe (galaxies) are of the largest and most closely related to the star cluster velocities. galaxy clusters are believed to be the building blocks of massive galaxies (zinn & yee 1979; zinn et al. 2000), and their virialized gas can evolve by gravitational infall of gas to relativistic regions (van der hulst 2000; yee 2000). over a relatively long time span, gas infall to these virialized regions will tend to speed up the cluster relative to the initial mass, so that gravitational infall can produce stars at a fixed redshift. the infall of gas from these virialized regions into the central massive black hole (bh) galaxy @xcite results in a violent event, which is likely to precede a second generation of massive black holes (imbhs) by ram pressure stripping and subsequent star formation within the virial radius. the galaxy formation process in these virialized regions may take place in a short time span (@xmath6 years), and it will be interesting to see whether or not the bh galaxy will evolve from a star cluster to a cluster at a given redshift, after which the infall of gas is not taking place, and subsequent star formation within the same galaxies. we stress here that the inner edge of the bh galaxy is deeply connected to the process of star formation, and to constraints on their gas infall rates. galaxy clusters formed by ram pressure stripping and subsequent star formation are characterized by large scale ram pressure fluctuations (@xmath7), and their innermost regions tend to undergo violent mergers between them, with rapid expansion of the virialized regions and subsequent superwinds. the innermost regions of the bh galaxy are known to be " tully - like " @xcite, and they often undergo " @xmath8-body " mergers, which result in a complex interaction between the gas infall rates and the ram pressure fluctuations associated with " superwinds ", or " cluster mergers " (e.g., @xcxcite). in other words, the two most distant galaxies of the bh galaxy are called " tully - like " galaxies, and they lead to more complex interactions between two massive imbhs, and subsequent superwinds. several numerical simulations have shown that the gas infall process in these virialized regions can take place in a cosmologically plausible way (e.g., zinn & yee 1979; van der hulst 2000; zinn et al. 2000), in which ram pressure stripping and subsequent star formation were taken into account during the evolution of the bh galaxy, and in the case of low redshift gas infall rates such as cluster mergers (zinn - yee, zinn, & zinn 2000) through post - merger simulations of the innermost regions of a massive galaxy (yee 2000). although these simulations show that the gas infall process in a tully - like bh galaxy can be triggered by ram pressure stripping and subsequent starbursts, it does not have enough time to test a realistic cosmological model that captures the small fraction of the infall of gas. it is possible that ram pressure stripping and subsequent star formation in the central massive bh galaxy would lead to a relatively slow infall of gas from these virialized regions, and it would not have to do so if the velocity is measured to be much smaller than the speed of light. we have not used numerical simulations to explore the infall process, and zinn & yee (2000) have shown them with @xmath6-body and kinematic constraints. we run our simulations in terms of physical nuclei. accurate models strongly affect the gas infall dynamics, which makes it difficult to constrain realistic cosmological models. it is also difficult to prevent gas infall from a hot gas that circulates into the infalling gas, so that the gas infall rate follows a limited line of sight. to constrain the orientation of a star cluster in a given galaxy, one should choose a preferred ellipsoid. sdss j1047 + 013 (zinn & yee 1979). cole et al. (2000) have developed a three - component model to determine the orientation of a bh galaxy and correct for the tendency of star clusters to evolve gravitationally. in summary, zinn & yee (2000) examined the infall dynamics of galaxy clusters using a simple three - component model (zinn, yee, & zinn 2000), and show that this approach provides stringent constraints on the physical nuclei from which the infall can be treated, although some models may require detailed kinematical and kinematic constraints. they attempted to account for this effect using an adiabatic formulation first developed by cole et al. (2000), which leads to too high @xmath6-body and kinematic constraints and to neglect the effective mass (goldreich 1981). we note here that the bh galaxy pairs (i.e., tully - like) may interact with the galactic nuclei from which the infall is triggered by ram pressure stripping and subsequent starbursts (see zinn & yee 2000 for a short review). we do not consider this simple assumption, and we suggest that our choice of the initial mass of these pairs should greatly reduce the number of imbhs to zero in any numerical run to avoid excessive mass loss. tully & yee (2000) use an adiabatic formulation (zinn, yee, & zinn 2000; @xcite) to predict the orientation of a bh galaxy pair, and show that @xmath6-body and kinematic constraints would be obtained if ram pressure stripping and subsequent starbursts interact gravitationally (van der hulst 2000). tully & yee (2000) use an adiabatic three - component model to predict the orientation of the zinn - yee ellipsoid, and correct for the tendency of superwinds to precess at high infall velocities. we start our simulation with a point mass of about @xmath1, which accretes instantaneously along the @xcite minor axis (but remains free of particles) with about 1@xmath3. the @xmath0-axis along the minor axis is a strong bend in the vertical axis of the collapsing bh galaxy, which should allow a correction for gravitational instability at high velocities @xcite. we set @xmath9 at a distance of 1@xmath3 from the central bh galaxy (s: s: @xcite, @xcmath10) and the line of sight (@xcit: s; s: ks: sb: sj11) @xxmath6-body at 10@xcite.]. @xmath11 where @xmath12 is the initial velocity of the bh galaxy (@xcite, @xcit: s; s: sb: sj11) and the line of sight angle @xite, and we choose the rotation axis @xin such that the particle should lie in the midplane between the @xcmath0-axis and the rotation velocity, and hence the particle must lie in position @lmath13.

Table 11: Samples generated on Arxiv

LaDiDa-CV on Scientific Papers
we use the gas infall velocity distribution of the gas to measure the gas properties associated with a real contour. comparisons of simulations were taken from koyama & sakurai (1998) simulations. for simplicity, we only consider the infall of bhs in the midplane between the @xmath0-axis and the rotation axis, under the assumption that all pairs of particles are coplanar until they meet at about @xcite. we set @xmath14 along the minor axis of the bh galaxy (s: s: @xcite) and follow the orbit of the cartesian ellipsoid in which the particle will travel from the sun to the galactic center. the figure shows @xmath17 for model ia. [ssec_velp].]. the initial velocity is measured along the equatorial plane of the bh system. the right hand side of the @xmath0-axis starts to fall at a point just above the tangent line @xcite, where a strong falloff occurs between @lmath18 and zero for all kinematic quantities. this result means that the infall process involves a very violent bend in the gas density, which then explains spiral arms to turbulence. in figure 1 we present the main points of our simulations for the three component model, and we first set up the contours from table 1. the infall takes a few hundred milliseconds above the central bh velocity of 8.5 km. a third time the infall of bhs is seen, which increases the stellar velocity range by nearly a factor of two. from this figure we can clearly see a significant increase in the infall velocity, which is seen in model ia. [ssec_velp] the first time is the infall time. to understand the hydrodynamic and kinematic behavior of a wave we use the resulting three -body equation. the time scale of the adiabatic events is @xmath13 except for a strong interaction with bhs. the second time is the infall time scale @xmath19, where the significance of the ‘ finger’is measured to be * f*@xmath20, and the outer edge of the postshock spiral arm * @xcite. (10@xmath3) and bhs at an infall time of [ssec_velp]a. the second important factor is the steepening at this time scale. in this figure, we only consider the infall velocity @xmath21, so that the column density of the bh gas in units of the dimensionless mass is @xcite, where the dashed line measures the @xxmath0-axis and @xite @x compared to the ‘ light’momentum, which increases when @xradiated by bhs. this disk structure is similar to that seen in model ia, and to the schwinger & binney (1998) superwinds. the solid (dashed) part reflects the inhomogeneity of the bh gas, with the particles in the accretion disk that are coplanar and the dark matter in a mixed state, while the solid (dotted) part is also accreting. the main questions we present here are to discuss the initial conditions of our sample. in our earlier work @xcite, we have focused on the problem of a strong bend in the momentum of the bh gas relative to the rotation axis (i.e., to the total infall velocity), but we found that the change in momentum that occurred before the infall could have had a purely kinematic origin. ,title="fig:",width=302] (10@xmath3) and bhs at an infall time of [ssec_velp]b.,width=188] in this work, the dashed line is dominated by the outermost free energy region, from which no physical effects can be added. the second dashed line represents the fractional error on the stellar wind trailing the central mass. in all simulations we started from the initial runs of the three component model, but runs of several tens of milliseconds (@xmath22) were carried out with much higher accuracy than to produce individual runs that varied smoothly from central to dark matter. we repeat the initial runs for more than @xmath23 of the simulation time, and we varied only a few runs from central to dark matter until all are detected. we note that the fragmentation of the bh system at @xmath24 (10@xmath3) is clearly dominated by the outermost free energy region, but significant differences were found in simulations that did require comparable amounts of accretion rates, for @xcite. [ssec_disk_sec]]). in the inner @xmath23 region, the system is shocked by bhs and the electrons are unable to transmit. figure [fig_disk_sec] shows the fragmenting of the @xmath23-th bh disk in @xcite (10@xmath3) in a relatively strong way; we therefore consider the free streaming region seen after annihilation to see figure 1. in this figure we use the @xmath23-th bh disk, so that our simulations do not consider the free streaming region, but consider particles in a coplanar (solid) and dark matter (dashed) medium that has a physical origin comparable to the densities of these particles. the blast wave that triggers the fragmentation of the @xmath23-th bh disk into a large piece of the solid (dashed) line, which then proceeds in a similar fashion as @xcite to get the adiabatic event unless the bh gas is pushed off. (10@xmath3) scatter the bh gas into a large piece of the solid (dashed) line, so that we preferentially pick up @xmath25 particles. to see this energy source in detail, we then imagine an adiabatic energy redistribution. the velocity dispersion @xmath21 in the three component model is given by * s*@xmath35, where @xcite is the radial velocity of the dark matter disk, and the proper motions are about 3 km s@xxmath36, and @xcmath37 the mass - loss velocity. -axis @xmath23 in figure [ssec_momentum]. [ssec_velp3]). the second dashed line has a long tail in the infall velocity, while the third one has a thick tail that flows outwards of the bh gas. we can see that the rise in the infall velocity ranges from @xmath3 to @xite, and should approach it with nearly an order of magnitude. for consistency, we therefore plot @xmath21 as the ratio of the ‘ light’momentum to the infall velocity, which should be @xcite @xite. in contrast, in all runs we have plotted @xmath21 as a function of the amplitude, but the final straight line is clearly rising, with a constant amplitude that falls off exponentially when going beyond @xcite. we also plot the relative values of @xmath21 and @xcite for the three bh models at equal values of the power - law ratio @x[scaledwidth=50] (10@xmath3). we see that these values of @xmath21 are much higher than in the three component runs, but we show results for weak thermal winds with @xcite. the three component calculation also gives a much higher value of @xmath21 than the one discussed in this work, although we see that it is roughly faster for weaker thermal winds.

Table 12: Continued

Generated samples with **Context and Vector** on Arxiv
we have also obtained a higher value of @xmath21 from our calculation, and found that the highest value of the bh infall rate is roughly given by * s*@xmath35, where we use the power - law derived from @xcite to derive the bhs density @xxmath50. this assumption implies @xmath39, for which we get the bh derivative at @xcite. one can not rule out that @xmath21 in the three component calculation is small for luminous bhs, unless the mass - loss radius is taken to be comparable to a few tens of percent (see figure [ssec_momentum]). in the three component calculation, the value of @xmath21 from equation [eqn: anisotropy] is given by * s*@xmath35, so the adiabatic term in equation [epsilon] simply vanishes as @xcite: it is almost impossible to get an anti - bh dipole to @xxmath51 which would affect @x. the magnitude for the above luminous bhs is within a factor of 10. the three component model shows that the adiabatic term in equation [epsilon] increases linearly from @xmath3 to a value of less than a few tens of percent for luminous bhs, but with a slope that increases with redshift (this is consistent with @xcite) suggesting that one might expect a perturbation to the non - perturbative case with a much steeper dipole moment. jaffe, bahcall, & gammie (2000) also shows how this effect is expected in numerical calculations of the infall velocity (see mcclintock & davies 1989), and a broad - line analysis based on the eddington - smith equation @xcite. a recent numerical calculation has shown that adiabatic @xmath21 is universal in star - forming systems at velocities of about 5.5 (bahcall et al. 2000). +. +. + + the third equation ([eqn: anisotropy]) tells us that it is impossible to determine the infall velocity for luminous bhs, but we argue that the virial theorem is a simple axisymmetric, geometrically symmetric equation which measures the infall velocities per unit virial mass. this does not necessarily solve the virial theorem, but it does nt result in the flattening of the infall velocity when the bh collides. for luminous bhs, the 3-component model has a very stable (@xmath52 km s@xmath36) axisymmetric system, but it is certainly more difficult to work with ‘ classical’trajectories than in our case, so that the infall velocity should scale as a function of the proper motions of the disk. thus, due to an adiabatic continuum analysis, nucl. +. + + in the case of luminous bhs, the velocity field is seen to fall off from the line of sight towards the observer as it falls upward from the cloud center and then collides with the material just above a certain velocity threshold. this is not surprising, since the momentum - momentum relation implies that the collision must be unstable for strong forces, but they are in principle weaker than in these cases. we first plot the density density @xmath21 of the luminous bhs as a function of the infall velocity, which is given by * s*@xmath35, where the three component density is @xcite: there exist density contrasts between @xxmath3 and @xcdm density profiles with density of the background (typically, we assume that @xcmath53) and @math36. we show @xmath21 from our two component simulations of the davies et al. panel and the @xcite case, with actual backgrounds set to @xxmath94 and 2000. we assume @xmath70. we have repeated the three component simulations of davies et al. (2000) for @xmath71 years, and found that non - thermal winds do not reach the same level with a mean infall speed of 100 km s@xmath36. we see that in the 3-component scenario the infall velocity is almost independent of the actual spatial field, so that flattening is still possible when using both simulations @xcite. we have neglected the effects of strong thermal winds, but this is not important because we use the range of escape frequencies to precess the radiation from the bh before it collides, so that there is no possibility to deal with radiation - induced shock propagation. in fact, using a full 3-component model does not change the infall velocity once the system returns, which promises continued importance in most numerical studies. however, @xmath71 was not able to assess the initial condition, especially in the final version. we should point out that the realistic 3-component simulations are in good agreement with the data in detail, but they are subject to uncertainties in the softest version of the 3-body model. in the final version of the 3-body simulation, we will have a chance to choose a ‘ typical’scenario in which each of the progenitors of the bhs will be at a given epoch, and then produce stars and other events during which the infall process is episodic. in fact, our analysis is in agreement with what appears to be independent of our theoretical observations about luminous bhs. the 3-component model is shown in the left panel in detail, so that the flattening of the infall velocity is seen when spatially confined structures are heated. this issue has not been included in the further study and should be considered elsewhere. the 3-component model is in very good agreement with the present numerical calculations of @xmath21, and the smoothness of the infall velocity clearly shows that it is perfectly reproducible. these are large numbers, since the bhs do not have to be filled with particles of the same mass, and the final infall velocity must be large in order to avoid misaligning the particles. we thank the staff of the bhs for providing excellent support throughout the jaffe and bahcall / gammie investigations and for their help throughout the work. this work was partially supported by conacyt grants dmr98 - 064001, and by the mexican government through grant in2/98 - 1. we also thank the anonymous referee for careful reading of the manuscript and for useful suggestions that greatly improved the manuscript. <EOD>.

Table 13: Continued

	Story-like
1	The text does not resemble a story at all. There is no plot and the sentences meander. It is difficult to understand the text.
2	The text revolves around a topic but does not form a plot skeleton. It is vague what is going on.
3	With guessing, the text seems to convey a story with a plot skeleton. However, there are mistakes such as wrong entity mentions, repetitive phrases and wrong choice of words.
4	The text looks like a story with a plot, but occasionally with mistakes such as wrong entity mentions, repetitive phrases and wrong choice of words.
5	The text is a complete and coherent story with a reasonable plot, all correct entity mentions and suitable choice of words.

	Style similarity to reference
1	The text is completely unlike the reference in terms of sentence length, structure, and tone.
2	The text is mostly dissimilar to the reference in terms of sentence length, structure, and tone.
3	The text is somehow similar to the reference in terms of sentence length, structure, and tone.
4	The text is mostly similar to the reference in terms of sentence length, structure, and tone.
5	The text is highly similar to the reference in terms of sentence length, structure, and tone.

Table 14: Human Evaluation guideline.