Abstract

Modern language models can generate high-quality short texts. However, they often meander or are incoherent when generating longer texts. These issues arise from the next-token-only language modeling objective. Recent work in self-supervised learning suggests that models can learn good latent representations via contrastive learning, which can be effective for discriminative tasks. Our work analyzes the application of contrastive representations for generative tasks, like long text generation. We propose one approach for leveraging contrastive representations, which we call Time Control (TC). TC first learns a contrastive representation of the target text domain, then generates text by decoding from these representations. Compared to domain-specific methods and fine-tuning GPT2 across a variety of text domains, TC performs competitively to methods specific for learning sentence representations on discourse coherence. On long text generation settings, TC preserves the text structure both in terms of ordering (up to +15% better) and text length consistency (up to +90% better).

1 Introduction

Large language models (LLM) such as GPT-2 have been extremely successful in text generation (Radford et al., 2019; Brown et al., 2020). However, LLMs are known to generate incoherent long texts. One reason is that they are unable to plan ahead or represent long-range text dynamics (Kiddon et al., 2016; Fan et al., 2019; Hua & Wang, 2020; Duboue & McKeown, 2001; Stent et al., 2004; Tamkin et al., 2020). As a result, they oftentimes produce wandering content with poor discourse structure and low relevance (Hua & Wang, 2020; Zhao et al., 2017; Xu et al., 2020); the text reads as if the model has no anchored goal when generating. These problems with coherence are further exacerbated when forcing autoregressive models to generate longer texts as the model struggles to extrapolate beyond its expected text end point. These problems suggest that LLMs currently fail to properly capture how documents evolve from beginning to end. Doing so is critical for succeeding in goal-oriented tasks such as story, dialog or recipe generation.

Prior work has explored the use of planning-based methods for generating globally coherent text (Kiddon et al., 2016; Fan et al., 2019; Hua & Wang, 2020; Duboue & McKeown, 2001; Stent et al., 2004). However, these methods rely on manually defining text dynamics for specific domains. Other work has attempted to use sentence representations for modeling text, such as with variational auto-encoders (Bowman et al., 2016) or contrastive learning (Gao et al., 2021; Devlin et al., 2019). However, most of these works focus on discriminative tasks like classifying the order of

Correction note

This is a revised version of the original ICLR 2022 paper. During post-publication code review, we discovered that the original version of the code did not leverage goal-directedness during decoding. While we still find that contrastive representations lead to gains in our evaluations, this error affects other claims made in the paper on goal-directed decoding. To correct this and help improve our understanding of goal-directed decoding, this updated version of the manuscript contains results on both goal-directed and non goal-directed baselines. We detail the difference between the original work and this updated work in Appendix [H]. The original version of this work can be found here: [https://arxiv.org/abs/2203.11370v1](https://arxiv.org/abs/2203.11370v1).
sentences. Methods including van den Oord et al. (2019) have tried decoding from latent contrastive representations, however they fail to generate coherent long sequences.

We propose one way of leveraging contrastive learning representations for generation, which we call Time Control (TC). We begin by assuming that meandering text generated without a goal can be represented as Brownian motion in latent space; this motion enforces the embeddings of neighboring sentences to be similar to each other, whereas those of distant sentences to be dissimilar. Goal-directed behavior can be incorporated into this model by conditioning on a fixed start and end point. In this case, the Brownian motion becomes a Brownian bridge and the resulting latent trajectories abide by simple, closed-form dynamics.

We derive a novel contrastive objective for learning a latent space with Brownian bridge dynamics. We can then use this latent space to generate text that retains local coherence and has improved global coherence. To perform text generation, Time Control first generates a sequence of latent embeddings sampled from the Brownian bridge process pinned at a start and end point. It then conditionally generates sentences using this latent sequence. In our work, we decode the latent sequences by fine-tuning GPT2 to generate text conditioned on Time Control’s latent sequence.

In summary, our work’s contributions are the following:

- We derive a novel contrastive learning objective based on Brownian bridge stochastic processes.
- We explore how contrastive representations can be used in both the discriminative and generative setting with our model Time Control which decodes from these learned representations.
- Across a range of text domains, we show that Time Control generates more or equally coherent text on tasks including forced long text generation, compared to task-specific methods.
- We validate that our latent representations capture text dynamics competitively by evaluating discourse coherence.
- We ablate our method to understand the importance of the contrastive objective, enforcing Brownian bridge dynamics, explicitly modeling latent dynamics, and decoding from a latent plan.

2 RELATED WORKS

Generating long, coherent text is conceptually difficult for autoregressive models because they lack the ability to model text structure and dynamics (Lin et al., 2021). This means that they struggle to plan and look ahead which leads to generating globally incoherent text. Forcing autoregressive models to generate longer texts exacerbates this incoherence because the models struggle to extrapolate beyond their expected text end point. Prior work has tried to address the problem of generating globally coherent text with planning-based approaches (Puduppully et al., 2019; Moryossef et al., 2019; Fan et al., 2019; Kiddon et al., 2016). However, planning-based approaches rely on domain-specific heuristics for capturing text structure and dynamics.

Our work uses a contrastive objective to learn latent dynamics in text without domain-specific heuristics. Contrastive objectives have been applied to several domains, including language (Devlin et al., 2019; Iyer et al., 2020; Liu & Liu, 2021), vision (Chen et al., 2020), and general time series data (Hyvarinen & Morioka, 2016; Hyvarinen et al., 2019). In particular for language, contrastive objectives have been applied to the next-sentence prediction task for improving BERT embeddings (Devlin et al., 2019) and to the discourse coherence setting (Nie et al., 2019; Chen et al., 2019b) for evaluating how coherent pairs of sentences are. However, they are not used for generation and are limited to classification tasks like discourse coherence. Prior work has also tried fitting latent variable models (Bowman et al., 2016), however these generally result in poor language generation (He et al., 2018) or are domain-specific (Weber et al., 2020; Arora et al., 2016).

Our work is closely related to Contrastive Predictive Coding (CPC) from van den Oord et al. (2019). The key difference is CPC implicitly learns unconditioned latent dynamics, whereas our contrastive learning objective imposes goal-conditioned dynamics on our latent space. Additionally, our method builds off of recent findings that contrastive objectives can be used to approximate local
Hello, I’d like to buy tickets for tomorrow.

Could you confirm my tickets just in case?

What movie theater do you prefer?

Hi, I’m looking to purchase tickets for my family.

Consider a set of triplet observations, \((x_0, x_t, x_T)\) that are part of the same conversation. The encoder maps positive triplets to a smooth Brownian bridge trajectory. It embeds \(z_t\) close to the expected embedding \(\mu_t\) pinned by \(x_0, x_T\). The green oval area illustrates the uncertainty over \(z_t\) as a function of how close \(t\) is to 0 and \(T\). In contrast, a negative random sentence \(x'\) from a different conversation is not coherent with \(x_0\) and \(x_T\); thus, it is embedded far from \(\mu_t\). This is captured by our contrastive loss, \(\mathcal{L}\).

\[
\mathcal{L} = -\log \frac{\exp(d(z_t, \mu_t))}{\exp(d(z_t, \mu_t)) + \exp(d(x', \mu_t))}
\]

Figure 1: Latent space for a positive triplet of sentences \((x_0, x_t, x_T)\) that are part of the same conversation. The encoder maps positive triplets to a smooth Brownian bridge trajectory. It embeds \(z_t\) close to the expected embedding \(\mu_t\) pinned by \(x_0, x_T\). The green oval area illustrates the uncertainty over \(z_t\) as a function of how close \(t\) is to 0 and \(T\). In contrast, a negative random sentence \(x'\) from a different conversation is not coherent with \(x_0\) and \(x_T\); thus, it is embedded far from \(\mu_t\). This is captured by our contrastive loss, \(\mathcal{L}\).

3 Methods

This section details our novel contrastive learning objective based on Brownian bridge dynamics. First, we discuss training an encoder via our contrastive learning objective to map sentences to a Brownian bridge latent space. Next, we discuss how a decoder learns to reconstruct sentences from this latent space. Finally, we discuss how to generate text at inference time.

3.1 Training an Encoder with Brownian Bridge Dynamics

Our encoder is a nonlinear mapping from raw input space to latent space, \(f_\theta : \mathcal{X} \rightarrow \mathcal{Z}\). The objective for the encoder is to map high-dimensional sequential data into low-dimensional latents which follow a stochastic process of interest—in this paper, it is the Brownian bridge process. The density of a Brownian bridge process between an arbitrary start point \(z_0\) at \(t = 0\) and end point \(z_T\) at \(t = T\) is,

\[
p(z_t|z_0, z_T) = \mathcal{N} \left( \left(1 - \frac{t}{T}\right)z_0 + \frac{t}{T}z_T, \frac{t(T-t)}{T} \right).
\]

This density is intuitive to understand: It acts like a noisy linear interpolation between the start and end point of the trajectory, where \(z_t\) should be more like \(z_0\) at the start and more like \(z_T\) at the end of the trajectory. Uncertainty is highest in the middle region, and low near the end points (rf. Figure[1]).

Consider a set of triplet observations, \((x_1, x_2, x_3)\). The goal of our work is to ensure that \(f_\theta(x_1), f_\theta(x_2), f_\theta(x_3)\) follow the Brownian bridge transition density in Equation[1]. We ensure this using a contrastive objective. Formally, given multiple sequences of data points, \(X = \{x_1, ..., x_N\}\), we draw batches consisting of randomly sampled positive triplets \(x_0, x_t, x_T\) where \(0 < t < T\): \(B = \{(x_0, x_t, x_T)\}\). Our encoder is optimized by,

\[\text{L2We use indices 0, t, T to denote the start, middle and end point of a Brownian bridge, but these do not correspond to strictly sampling the first, middle and last sentence of a document. x_0, x_t, x_T can be any sentence in a document as long as x_0 comes before x_t and x_t before x_T in the document.}\]
\[ \mathcal{L}_N = \mathbb{E}_X \left[ -\log \frac{\exp(d(x_0, x_t, x_T; f_\theta))}{\sum_{(x_0, x_t, x_T) \in \mathcal{B}} \exp(d(x_0, x_t, x_T; f_\theta))} \right], \text{where} \]

\[ d(x_0, x_t, x_T; f_\theta) = -\frac{1}{2\sigma^2} \left\| \frac{f_\theta(x_t)}{z_t} - \frac{(1 - \frac{t}{T}) f_\theta(x_0) - \frac{t}{T} f_\theta(x_T)}{\text{mean in Equation}} \right\|_2^2 \]

\( \sigma^2 \) is the variance in Equation\[ (\mathcal{T} - t)/T. \] Note that Equation\[ (3) \] sums over negative middle contrasts, \( x_{t'} \). This objective can be viewed as maximizing the extent to which true triplets from the data follow the Brownian bridge process while minimizing the extent to which an alternative mid-point sampled from another sequence does so. \[ ^3 \]

Figure \[ (1) \] illustrates how the objective translates into the language setting for training the encoder. The objective samples triplet sentences from a document. Sentences drawn from the same document make up a smooth latent trajectory; they should be close to each other and follow the conditional density in latent space. Sentences drawn from different documents should not make up a smooth latent trajectory and should less likely follow bridge dynamics.

**Connection to mutual information estimation and triplet classification** We draw connections between our contrastive loss and the mutual information estimation setup from van den Oord et al. (2019); Poole et al. (2019) (as \( |\mathcal{B}| \to \infty \)) and the classification setup from Liu et al. (2021) (\( |\mathcal{B}| = 1 \)).

Following van den Oord et al. (2019); Poole et al. (2019), this objective can be seen as a lower bound on the mutual information between the two end points and the middle point: \( I(X_t, \{X_0, X_T\}) \geq \log(N) - \mathcal{L}_N \). Hence, by minimizing the contrastive loss, we are maximizing the amount of information between the trajectory and the linear interpolation of its end points.

Assuming \( |\mathcal{B}| = 1 \), we can draw a connection to the classification setup studied in Liu et al. (2021). They train a classifier to distinguish in- vs. out-of-order input pairs and show that the Bayes optimal logits for pair-wise classification can be written as a function of the stochastic process transition kernel. This is equivalent to the our loss on a single triplet \( i \): \( l_i = -\log \frac{\exp(d(x_0, x_t, x_T; f_\theta))}{\exp(d(x_0, x_t, x_T; f_\theta)) + \exp(d(x_0, x_t, x_T; f_\theta))} \). Liu et al. (2021) consider pairs whereas our work considers triplets; we show in Appendix \[ (A) \] the pairwise and triplet setups are equivalent.

### 3.2 Training a Decoder with Latent Plans

Here we discuss how to train a language model to decode latent sequences for generation. We first map all the sentences in the training dataset to our learned latent space using the pretrained encoder \( f_\theta \). This gives us a Brownian bridge trajectory of sentence-level latent codes \( (z_0, \ldots, z_t, \ldots, z_T) \) for a document in the dataset. Then, rather than learning a decoder from scratch, we fine-tune GPT2 (Radford et al., 2019) to generate text conditioned on past context and the latent plan.

We fine-tune in the following manner. Let \( x_1 \ldots x_W \) be a document with \( W \) tokens and \( T \) sentences used to train the decoder. Using the encoder \( f_\theta \), we can obtain embeddings \( z_1 \ldots z_T \) for each sentence. The decoder is a standard auto-regressive language model that is modified in the following way: at time \( t \), the decoder must predict \( x_t \) using all tokens in the past \( x_{<t} \), as well as the sentence embedding \( z_{s_t} \), where the index \( s_t \in [T] \) is a map which takes each token to its corresponding sentence. This is a form of a reconstruction objective, as the identity of \( x_t \) is encoded in \( z_{s_t} \).

### 3.3 Generating Text with Latent Plans at Inference Time

Figure \[ (2) \] illustrates how the trained decoder generates text at inference time. Given two end points \( z_0, z_T \), we sample a trajectory from a latent Brownian bridge, and then generate from the decoder conditioned on this bridge. In many situations, we may not know the endpoints of the Brownian bridge explicitly. In this case, we encode a set of sentences corresponding to start and end points (e.g. the first and last sentences of our training set), and fit a Gaussian to these points to form a density

\[ ^3 \text{Empirically, we found Brownian bridge dynamics easier to recover with triplets rather than pairs of contrasts. Appendix} \ [ (C.2)] \text{discusses some of the pair-wise contrast results.} \]
Figure 2: Time Control generates text conditioned on a latent plan. A latent plan is first generated by running Brownian bridge dynamics pinned between a sampled start $z_0$ and goal latent variable $z_T$ forward. A decoder then conditionally generates from this latent plan on a sentence-level.

estimate. Generating in this case involves first sampling from the Gaussian, and then generating as before from the bridge. More details on training and generation can be found in Appendix C.

4 EXPERIMENTS

We now evaluate the ability of a model trained with a contrastive objective to capture text dynamics. Specifically, we aim to answer the following research questions (RQ):

RQ1: Can the contrastive learning objective model local text dynamics? Section 4.1 investigates this question using a sentence ordering prediction task: given two sentences from the same document, we evaluate whether different models can predict their original order.

RQ2: Can the contrastive learning objective model global text dynamics? Section 4.2 investigates this question on text generation for Wikipedia city articles by examining the length of generated sections.

RQ3: Can the contrastive learning objective help generate long coherent documents? Section 4.3 investigates this question on forced long text generation: we evaluate how well models preserve global text statistics (such as typical section orders and lengths) when forced to extrapolate during generation.

We run our setup with different latent dimensions ($d = 8, 16, 32$). Our encoder architecture is a frozen, pretrained GPT2 model from Huggingface (Radford et al., 2019; Wolf et al., 2020) and a trainable MLP network. We extract GPT2’s last layer hidden state that corresponds to the end-of-sentence (EOS) token and train the 4-layer MLP on top of the hidden state. The MLP network has intermediate ReLU activations and is trained with stochastic gradient descent with a learning rate of 1e-4 and with momentum 0.9. For more details on the machines we used for our experiments, please refer to Appendix B. This version of the manuscript contains results after correcting for random seeding and GPU nondeterminism. Please refer to Appendix F for details on the version differences.

Encoder ablations We perform three ablations on our encoder model. Recall that Time Control encoder (A) explicitly models latent structure with (B) Brownian bridge dynamics using a (C) contrastive loss. (A) replaces explicit dynamics with Implicit Dynamics (ID) where future latents are directly predicted with an autoregressive model (van den Oord et al., 2019). (B) replaces Brownian bridge dynamics with Brownian motion (BM); latents follow the transition density $z_t | z_s \sim N(z_s, t - s)$ in Equation 3. Note $z_t$ is centered at $z_s$ and is not conditioned on a goal endpoint. (C) replaces the contrastive loss with a Variational Auto-Encoder (VAE) and centers the priors over $z_0$ to 0 and $z_T$ to 1, as done in our setup. Appendix E includes more detail on the ablations.

Decoder instantiations We compare two methods for decoding from contrastive representations. The first method decodes from the path $[z_0, z_1, \ldots, z_T]$ as detailed in Section 3.3, and the second method decodes from the latent plan $z_0$, $z_1$, $z_T$. We find that decoding from the latent plan generally performs better as it captures more of the text’s local and global dynamics.
The latter 4 methods are designed as sentence embedding, and SimCSE (Gao et al., 2021). We evaluate how well the Brownian bridge contrastive learning objective models local text dynamics, and Taskmaster-2. Recipes, each with a title, ingredients and set of directions. A recipe is constructed as “[TITLE] [METHOD] [STEP] 1 text [STEP] 2 text ...” Recipe NLG (Bien et al., 2020) contains recipes, each with a title, ingredients and set of directions. A recipe is constructed as “[TITLE] [INGREDIENTS] [DIRECTIONS] text”. Taskmaster-2 (TM-2) (Byrne et al., 2019) contains conversations on finding restaurants between an assistant and a user. The assistant’s turn is marked with an “[ASSISTANT]” tag, and the user’s turn is marked with a “[USER]” tag. TicketTalk (Byrne et al., 2021) contains conversations on booking movie tickets between an assistant and a user. The assistant’s and user’s turns are similarly marked as in TM-2.

4.1 Modeling Local Text Dynamics

We evaluate how well the Brownian bridge contrastive learning objective models local text dynamics (RQ1) on the discourse coherence setting (Jurafsky 2000). Discourse coherence is often measured by how well representations capture discourse structure by testing for whether a linear classifier can detect in-order and vs. out-of-order sentence pairs (Chen et al., 2019a). We compare Time Control’s encoder against GPT2’s last layer’s hidden state corresponding to the EOS token (Radford et al., 2019), BERT (Devlin et al., 2019), ALBERT (Lan et al., 2019), Sentence BERT (Reimers et al., 2019), and SimCSE (Gao et al., 2021). The latter 4 methods are designed as sentence embedding models. We also compare to our ablations.

The setup is the following: The encoder takes in two sentences \(x_t, x_{t+k}\) to produce their latents \(z_t, z_{t+k} \in \mathbb{R}^d\). At random, the latents are fed either in- or out-of-order. A linear classifier is trained

| Method      | Wikisection | TM-2 | TicketTalk |
|-------------|-------------|------|------------|
|             | \(k = 5\)   | \(k = 10\) | \(k = 5\) | \(k = 10\) | \(k = 5\) | \(k = 10\) |
| BERT        | \(72.7 \pm 0.1\) | 80.4 \pm 0.2 | 75.6 \pm 0.0 | 86.7 \pm 0.0 | 73.1 \pm 0.2 | 86.4 \pm 0.0 |
| ALBERT      | 67.7 \pm 15.1 | \(82.3 \pm 0.3\) | \(88.9 \pm 1.7\) | \(92.2 \pm 0.9\) | 74.5 \pm 9.8 | \(93.3 \pm 0.6\) |
| S-BERT      | 71.8 \pm 0.1 | 79.0 \pm 0.1 | 75.6 \pm 0.0 | 86.7 \pm 0.0 | 76.2 \pm 0.1 | 88.8 \pm 0.0 |
| SimCSE      | 72.6 \pm 0.1 | 79.5 \pm 0.1 | 77.8 \pm 0.0 | 88.3 \pm 0.0 | \(78.0 \pm 0.0\) | 90.1 \pm 0.1 |

Table 1: Discourse coherence accuracy measured by the test accuracy of the trained linear classifier, reporting \(\mu \pm\) standard error over 3 runs. Random accuracy is 50%. The highest mean score for the non-GPT2-based methods (rows above the double lines) are marked in gray cells. The highest mean score for the GPT2-based methods (rows below the double lines) are underlined. When applicable, the methods are run with varying latent dimensions marked in parentheses (dimension).
on 100 epochs with stochastic gradient descent with a learning rate of 1e-4 and with momentum 0.9. We varied the sentence distance \( k \in \{1, 5, 10\} \) and found that on some domains and for \( k = 1 \), all methods scored near random accuracy; we have omitted those results in the main paper. Otherwise, the results are summarized in Table 1 where we report the mean and standard error accuracy on a held-out discourse dataset of 3000 examples on 3 runs. Our method is able to compete with sentence embedding specific methods, like ALBERT, BERT, SimCSE. Additionally, though GPT2 is used as a base encoder for Time Control, we observe that Time Control greatly improves upon GPT2 with significant gains on TM-2, TicketTalk, and Wikisection (up to +20%). Out of all the GPT2-based methods, it performs the best.

Neither VAE nor BM perform better than random accuracy on most of the domains. This suggests that the variational lower bound does not recover the latent embeddings as well as contrastive objectives and the choice of stochastic processes matter for learning informative structural embeddings.

Overall, our model performs competitively on both task-oriented conversations like TM-2 and TicketTalk and enumerative, factual domains like Wikisection. This answers RQ1 in the positive: Time Control can model local text dynamics, like in conversations and articles.

### 4.2 Modeling Global Text Dynamics

| Method       | MM % (↓)       |
|--------------|----------------|
| GPT2         | 9.9 ± 3.6      |
| SD           | 28.3 ± 7.9     |
| SS           | 223.9 ± 18.5   |
| VAE (8)      | 8.1 ± 3.0      |
| VAE (16)     | 14.1 ± 7.8     |
| VAE (32)     | 18.5 ± 4.5     |
| STATIC VAE (8) | 8.9 ± 2.2   |
| STATIC VAE (16) | 13.1 ± 5.8 |
| STATIC VAE (32) | **6.7 ± 1.9** |
| ID (8)       | 13.3 ± 3.8     |
| ID (16)      | 61.8 ± 5.5     |
| ID (32)      | 84.6 ± 11.0    |
| STATIC ID (8) | 12.3 ± 3.4    |
| STATIC ID (16) | 77.1 ± 4.4   |
| STATIC ID (32) | 79.6 ± 8.0   |
| BM (8)       | 16.2 ± 11.4    |
| BM (16)      | 21.1 ± 11.8    |
| BM (32)      | 51.6 ± 14.2    |
| STATIC BM (8) | 15.4 ± 9.2    |
| STATIC BM (16) | 11.8 ± 5.4   |
| STATIC BM (32) | 13.0 ± 5.0   |
| TC (8)       | 13.1 ± 4.4     |
| TC (16)      | 15.4 ± 10.1    |
| TC (32)      | 28.1 ± 12.2    |
| STATIC TC (8) | 12.8 ± 3.2    |
| STATIC TC (16) | 13.0 ± 7.5   |
| STATIC TC (32) | 13.0 ± 7.8   |

Table 2: Percentage of length mismatch (MM) during generation. These results are run over 3 random GPU machines, each with 3 seeds. The method with the highest mean is marked in bold, and methods with overlapping confidence intervals are underlined.

We evaluate how well the Brownian bridge contrastive learning objective models global text dynamics (RQ2) by assessing whether the methods mimic document structure on Wikisection. We check whether the generated section lengths match with the average lengths in the dataset. We focus on Wikisection because it is the only dataset with long, well-defined sections (rf. Appendix on dataset statistics). Each document contains an abstract, history, geography, and demographics section on a city.

Time Control plans a latent trajectory by running the bridge process between the start and end latent, \( z_0 \sim p(z_0) \) and \( z_T \sim p(z_T) \). We compare against fine-tuned GPT2. We also include two oracle methods that are fine-tuned GPT2 models with additional section-embedding supervision.
One is “sec-dense GPT2” (SD) where each token’s embedding contains the current section identity; section $i$’s embedding is added onto the token’s positional token embedding. The other is “sec-sparse GPT2” (SS) where the token embedding contains an indicator that is 1 if the token is the start of a new section, and 0 otherwise. For VAE and ID, we calculate the density estimate $p(z_0)$ and $p(z_T)$ and run a linear interpolation between the start and end latents. For BM, we calculate the density estimate $p(z_0)$ and run Brownian motion. Additionally, we run Static for Time Control, VAE, ID, BM which samples $z_0 \sim p(z_0)$ and repeatedly decodes from this latent representation. Note that the difference between these Static models is that the decoders are finetuned with their respective objectives (Section 4).

Table 2 reports in percentage how much the generated section lengths deviate from the average section lengths. VAE best matches the section lengths out of all the methods, despite performing at near random accuracy on discourse coherence. We hypothesize that TC does not match the text lengths as well as the VAE method, as the VAE is more explicitly trained to match the text distribution. Nonetheless, TC and Static TC perform competitively with other top-performing baselines like VAE and Static VAE. Another surprising observation was that fine-tuned GPT2, SD and SS mismatch the section lengths by 10-224%. Upon further investigation, we noticed the models overshoot short sections and undershoot long sections; this causes the section length deviations. ID and Static ID performs extremely poorly in contrast to these methods; this highlights the challenge of interpolating on learned dynamics models which is exacerbated in the long text generation settings. The results affirm that contrastive objectives can model global text dynamics, such as in matching document structure, however probabilistic objectives like VAE might be a better alternative learning objective.

### 4.3 Generating globally coherent text

| Method    | WH      | TM-2     | TT      |
|-----------|---------|----------|---------|
| GPT2      | 10.6 ± 4.3 | 81.7 ± 10.7 | 108.5 ± 8.6 |
| VAE (8)   | 10.8 ± 6.3 | 120.4 ± 36.3 | 80.9 ± 7.1  |
| VAE (16)  | 13.2 ± 6.3 | 109.4 ± 51.2 | 75.0 ± 4.5  |
| VAE (32)  | 11.0 ± 9.9 | 66.9 ± 10.8 | 77.5 ± 25.0 |
| Static VAE (8) | 8.6 ± 4.4 | 123.8 ± 35.3 | 81.4 ± 5.7  |
| Static VAE (16) | 12.9 ± 5.3 | 115.5 ± 60.6 | 75.2 ± 3.2  |
| Static VAE (32) | 12.8 ± 5.3 | 73.9 ± 10.9 | 80.2 ± 23.9 |
| ID (8)    | 14.9 ± 6.5 | 200.3 ± 27.7 | 108.8 ± 9.5 |
| ID (16)   | 37.7 ± 4.0 | 198.2 ± 19.2 | 68.7 ± 20.3 |
| ID (32)   | 35.9 ± 8.5 | 231.5 ± 97.1 | 78.6 ± 22.5 |
| Static ID (8) | 13.5 ± 5.6 | 120.2 ± 37.4 | 64.1 ± 6.0  |
| Static ID (16) | 37.2 ± 2.7 | 93.9 ± 32.3 | 66.2 ± 20.5 |
| Static ID (32) | 25.8 ± 4.4 | 124.4 ± 50.4 | 58.9 ± 16.9 |
| BM (8)    | 11.8 ± 5.1 | 93.4 ± 20.4 | 68.2 ± 28.9 |
| BM (16)   | 20.8 ± 11.5 | 106.8 ± 17.6 | 111.3 ± 45.6 |
| BM (32)   | 30.2 ± 13.4 | 118.6 ± 17.7 | 170.5 ± 36.2 |
| Static BM (8) | 7.8 ± 3.2 | 102.5 ± 14.6 | 66.3 ± 26.2 |
| Static BM (16) | 10.7 ± 6.6 | 93.7 ± 11.7 | 66.2 ± 34.7 |
| Static BM (32) | 11.1 ± 9.7 | 111.5 ± 17.2 | 82.7 ± 25.9 |
| TC (8)    | 13.5 ± 6.1 | 41.9 ± 9.5 | 69.8 ± 6.2  |
| TC (16)   | 8.5 ± 4.4 | 21.5 ± 4.7 | 60.8 ± 16.6 |
| TC (32)   | 13.3 ± 3.6 | 33.2 ± 3.3 | 53.5 ± 27.4 |
| Static TC (8) | 13.4 ± 5.5 | 15.6 ± 2.7 | 36.4 ± 11.4 |
| Static TC (16) | 10.8 ± 5.4 | 13.7 ± 3.4 | 37.4 ± 12.9 |
| Static TC (32) | 10.3 ± 7.1 | 8.6 ± 5.3 | 30.2 ± 17.7 |

Table 3: Section lengths deviating from expected length in forced long text generation reported in % ($\%$). These results are run over 3 random GPU machines, each with 3 seeds. The method with the highest mean is marked in bold, and methods with overlapping confidence intervals are marked in yellow cells.
Table 4: Ordering in forced long text generation. ROC Stories and TM-2 omitted because they are not applicable. These results are run over 3 random GPU machines, each with 3 seeds. The method with the highest mean is marked in bold, and methods with overlapping confidence intervals are underlined.

We evaluate how well the Brownian bridge contrastive learning objective helps generate globally coherent text (RQ3) where the EOS token is omitted. We refer to this as the forced long text generation setup because the model must extrapolate beyond its natural end point in generation. Appendix D.3 details how the latent plan is generated. For reference, 1000 tokens is about 50% longer than the average WikiSection document (the longest text domain). This setting is intrinsically difficult for auto-regressive models which do not have the ability to “look ahead” during generation (Lin et al., 2021).

We evaluate model extrapolation on two metrics. First is ordering: how well do the models maintain the structure of the document (e.g., turn-ordering in TicketTalk)? Second is length consistency. Length consistency captures common failure modes such as a model which stops modeling a conversation between two agents, and instead outputs a long monologue from one agent.

The results are summarized in Table 3 for length consistency and Table 4 for ordering. According to the length consistency metric in Table 3, Static TC maintains text flow the best. Surprisingly, Static TC performs better than TC, which decodes from an actual Brownian bridge sequence. We hypothesize that training the decoder with the contrastive encoder regularizes the model to generate better forced long texts. VAE, BM, ID, and their Static variants do particularly worse than Static TC on length consistency on task-oriented and turn-taking domains like TM-2 and TT, suggesting that the Brownian Bridge objective can model these domains more faithfully. This suggests the importance of good latent dynamics for long text generation. Here, a common failure mode of the nominal GPT2 model is that it produces nonsensical text where it would naturally end generation; this is particularly noticeable on TM-2 and TicketTalk. Figure 5 shows an example of our model’s behavior versus GPT2’s behavior on generating long texts in TicketTalk. The example illustrates how Time Control continues the dialog whereas GPT2 utter nonsensical text.

According to the ordering metric in Table 4, none of the methods surpass the other consistently. These results suggest that that even though Static TC is better at matching text lengths on the
forced long text setting, decoding from latent variables to preserve semantically meaningful metrics like section ordering still remains a challenge.

5 Discussion on Static and Non-Static Methods

In an earlier version of this paper, we evaluated the Static methods. Surprisingly, the Static methods perform well despite not leveraging the dynamics at test time. This raises an interesting question of whether there are better ways to leverage contrastive representations at decoding time.

For more details on the differences between the original manuscript and the current manuscript, please refer to Appendix H.

6 Conclusion

Our work explores using representations learned via contrastive learning for generative and discriminative tasks. We propose one way of learning effective language representations which follow Brownian bridge dynamics.

Empirically, we demonstrate this leads to benefits such as simultaneously yielding good representations for discriminative tasks like discourse coherence and generating long texts that preserve section lengths. Future work includes understanding why decoding from a static latent sequence is just as effective—if not more effective—as a dynamic sequence. Additionally, decoding from a latent sequence while preserving other semantically meaningful measures beyond text length still remains a challenge, as suggested by the ordering metrics in the forced long text generation setting. Although our work focuses on benefits of learning stochastic process latents for language generation, our approach can be extended to other domains with sequential data like videos or audio such as in [Shin & Wang] or extended to handle arbitrary bridge processes without known fixed start and end points.

Table 5: Example of forced long text generation on TicketTalk with Time Control vs. fine-tuned GPT2. Both models are forced to extrapolate when generating long texts. They start coherently, but only Time Control extrapolates coherently. For space reasons, some of the text has been removed, marked with [...].
7 REPRODUCIBILITY STATEMENT

In the supplemental, we include a zip file containing our code and processed datasets. We’ve also included in the appendix how we processed the datasets. Our results on showing equivalence between the triplet classification setting and pairwise classification setting are also included in the appendix.

ACKNOWLEDGMENTS

REW is supported by the National Science Foundation Graduate Research Fellowship. The authors would give special thanks to CoColab members Mike Wu, Gabriel Poesia, Ali Malik and Alex Tamkin for their support and incredibly helpful discussions. The authors would like to thank the rest of CoCoLab and Rishi Bommasani for reviewing the paper draft. The authors also thank Chris Donahue for helpful pointers on using ILM.

REFERENCES

Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. Sector: A neural model for coherent topic segmentation and classification. Transactions of the Association for Computational Linguistics, 7:169–184, 2019. doi: 10.1162/tacl_a_00261.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. Transactions of the Association for Computational Linguistics, 4:385–399, 2016.

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wisniewski, and Agnieszka Lawrynowicz. RecipeNLG: A cooking recipes dataset for semi-structured text generation. In Proceedings of the 13th International Conference on Natural Language Generation, pp. 22–28, Dublin, Ireland, December 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.inlg-1.4.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp. 10–21, 2016.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cetinkaya. Taskmaster-1: Toward a realistic and diverse dialog dataset. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4516–4525, 2019.

Bill Byrne, Karthik Krishnamoorthi, Saravanan Ganesh, and Mihir Kale. TicketTalk: Toward human-level performance with end-to-end, transaction-based dialog systems. 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), pp. 671–680, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.55. URL https://aclanthology.org/2021.acl-long.55.

Mingda Chen, Zewei Chu, and Kevin Gimpel. Evaluation Benchmarks and Learning Criteria for Discourse-Aware Sentence Representations. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 649–662, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1060. URL https://www.aclweb.org/anthology/D19-1060.

Mingda Chen, Zewei Chu, and Kevin Gimpel. Evaluation benchmarks and learning criteria for discourse-aware sentence representations. In Proc. of EMNLP, 2019b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pp. 1597–1607. PMLR, 2020.
Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. URL [http://arxiv.org/abs/1810.04805](http://arxiv.org/abs/1810.04805).

Chris Donahue, Mina Lee, and Percy Liang. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Pablo A. Duboue and Kathleen R. McKeown. Empirically estimating order constraints for content planning in generation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL ’01, pp. 172–179, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1073012.1073035. URL [https://doi.org/10.3115/1073012.1073035](https://doi.org/10.3115/1073012.1073035).

Angela Fan, Mike Lewis, and Yann Dauphin. Strategies for Structuring Story Generation. *arXiv:1902.01109 [cs]*, June 2019. URL [http://arxiv.org/abs/1902.01109](http://arxiv.org/abs/1902.01109).

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2018.

Xinyu Hua and Lu Wang. 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)*, pp. 781–793, 2020.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA. *arXiv:1605.06336 [cs, stat]*, May 2016. URL [http://arxiv.org/abs/1605.06336](http://arxiv.org/abs/1605.06336).

Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 859–868. PMLR, 2019.

Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. Pretraining with Contrastive Sentence Objectives Improves Discourse Performance of Language Models. *arXiv:2005.10389 [cs]*, May 2020. URL [http://arxiv.org/abs/2005.10389](http://arxiv.org/abs/2005.10389).

Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 329–339, 2016.

Mahnaz Koupaei and William Yang Wang. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*, 2018.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL [https://aclanthology.org/W04-1013](https://aclanthology.org/W04-1013).

Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. 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*, pp. 5147–5173, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.405. URL [https://aclanthology.org/2021.naacl-main.405](https://aclanthology.org/2021.naacl-main.405).
A S Howing the Equivalence of the Pairwise and Triplet Classification Task

In this section, we focus on the classification setup where the goal is to train a classifier to detect in- vs. out-of-order groups of inputs. Specifically, we want to show that the classification setup from Liu et al. (2021) where the inputs come in pairs is equivalent to the setup where the inputs come in triplets. The triplet inputs are what we assume for Time Control.

Notation We denote latent space as $Z$; we typically do not observe this space directly. We denote the observation space as $X$; this is the space we observe directly.

We define the Brownian bridge for $t \in [0, T]$ as

$$B_t = B(t) = W(t) - \frac{t}{T} W(T), \quad (4)$$

where $W(t)$ is a standard Wiener process, $\mathcal{N}(0, t)$. We state the assumptions:
Lemma 1 of their work states the following: The optimum of the contrastive learning objective $L_z$.

Published as a conference paper at ICLR 2022

Our setting however assumes that the algorithm receives triplets of data points, the transition kernel. In particular, we’re interested in transitions of a Brownian bridge pinned at.

Let’s say we have two positive triplet samples, $z_{1}^0, z_{2}^0, z_{3}^0$, and $z_{1}^3, z_{2}^3, z_{3}^3$, where $t_1 < t_2 < t_3$. Following Liu et al. (2021), minimizing $L$ yields the following on each triplet:

$$
\log p^{Δt}(z_{1}^i, z_{2}^i, z_{3}^i) = \log q(z') + \log h^*(z_{1}^i, z_{2}^i, z_{3}^i) - \log(1 - h^*(z_{1}^i, z_{2}^i, z_{3}^i)) \\
\log p^{Δt}(z_{1}^i, z_{2}^i, z_{3}^i) = \log q(z') + \log h^*(z_{1}^i, z_{2}^i, z_{3}^i) - \log(1 - h^*(z_{1}^i, z_{2}^i, z_{3}^i)).
$$

Taking the difference in log probabilities between Equations (10, 11) results in

$$
\log p^{Δt}(z_{1}^0, z_{2}^0, z_{3}^0) - \log p^{Δt}(z_{1}^3, z_{2}^3, z_{3}^3) = \left[\log h^*(z_{1}^0, z_{2}^0, z_{3}^0) - \log(1 - h^*(z_{1}^0, z_{2}^0, z_{3}^0))\right] \\
- \left[h^*(z_{1}^3, z_{2}^3, z_{3}^3) - \log(1 - h^*(z_{1}^3, z_{2}^3, z_{3}^3))\right].
$$

Similar to the pair-wise classification setting, we’ve shown that minimizing $L$ in the triplet classification setting results in approximating the transition kernel of the Brownian bridge process.

### B Machines Used for Training

We found that our decoding results are sensitive to the GPU machine type. For example, in the length mismatch results in Table 2, TC (8) had a deviation of 7.7 ± 1.2 in one set of three seeds and a deviation of 17.3 ± 5.2 in another set of three seeds. In light of this, we randomized over the choice of machines. These machines include: RTX A6000, RTX A5000, GeForce RTX 3090, TITAN RTX, TITAN Xp, TITAN V, and GeForce GTX TITAN X machines.
C  TRAINING DETAILS

Here we describe the training procedure for the encoder and the fine-tuning procedure for decoding.

**Brownian bridge encoder**  The encoder architecture is a frozen, pretrained GPT2 model from Huggingface [Radford et al., 2019] [Wolf et al., 2020] and a trainable MLP network. We extract the GPT2’s last layer hidden state that corresponds to the end-of-sentence (EOS) token and train the 4-layer MLP on top of the hidden state. The MLP network has intermediate ReLU activations and is trained with stochastic gradient descent with a learning rate of 1e-4 and with momentum 0.9. We train the encoder for 100 epochs on each of the datasets.

The text fed into GPT2 are fed in on a sentence level. This means that the input $x_t$ refers to the $t$th sentence of a document. The sentences are separated from each other in the main text as “ ” which is added to the tokenizer as a separate token for indexing convenience.

**Fine-tuning GPT2 with latent embeddings**  After training the encoder, we run it on the training dataset to collect an accompanying latent trajectory for each text. The encoder is run on the dataset at a sentence level: we separate the text by sentences and pass the sentences through the encoder.

The sentence latent embeddings are aligned with the tokens of that sentence and offset by one token before the start of that sentence token. Let’s illustrate by an example. We denote [SOS] as the start of the document token, [s1] as sentence 1 tokens and [s2] as sentence 2 tokens. [ . ] is the period token which we’ve added into the tokenizer. $z_i$ denote the latent variable corresponding to the $i$th sentence.

Let’s say that the sequence fed into GPT2 is “[SOS] [s1] [s1] [s1] [ . ] [s2] [s2] [s2]”. Then the corresponding latent trajectory is “$z_1, z_1, z_1, z_1, z_2, z_2, z_2$”. The latent variables are added onto the positional embeddings. We then fine-tune GPT2 as normal.

D  GENERATION WITH EMBEDDINGS

D.1  NORMAL TEXT GENERATION

We first sample a start and end latent, $z_0 \sim p(z_0), z_T \sim p(z_T)$ where $p(z_0), p(z_T)$ are calculated as the density estimates over the training dataset. We pin our trajectory to the start and end latent, and run the Brownian bridge using a first-order approximation. For normal long text generation, we set $T$ to be the average number of sentences in each of the dataset. For forced long text generation, we set $T$ to be proportional to the number of sentences needed in order to generate 1000 tokens. By the end, we have a trajectory $z_0, z_1, ..., z_T$.

Generation starts with feeding the SOS token and the first latent $z_0$. Once GPT2 emits a [ . ] token and terminates sentence $t$, we transition to the next latent $z_{t+1}$. This process continues until GPT2 is finished with generation. If GPT2 generates more sentences than there are latents in the trajectory, the last latent $z_T$ is used until the end of generation.

D.2  FORCED LONG TEXT GENERATION

Let $S_{\text{avg}}(\mathcal{D})$ denote the average number of sentences in a document and $T_{\text{avg}}(\mathcal{D})$ denote the average number of tokens in a document. Rather than planning a trajectory of length $S_{\text{avg}}(\mathcal{D})$ (the average number of sentences in a document) which is what is done in Section 4.3 for normal text generation, we scale the trajectory length to $c \cdot S_{\text{avg}}(\mathcal{D})$. $c$ is determined by how many more tokens we need in order to fill up to GPT-2 maximum context length of 1024: $c = \frac{1024 - T_{\text{avg}}(\mathcal{D})}{T_{\text{avg}}(\mathcal{D})}$.

E  ABLATIONS

The following methods ablate the encoder model in Time Control; in other words, the ablations dissect the assumptions we make in the first step of Time Control described in Section 3.1. Recall that Time Control (A) explicitly models latent structure with (B) Brownian bridge dynamics using a (C) contrastive loss. (A) replaces explicit dynamics with Implicit Dynamics (ID) where future latents are directly predicted with an autoregressive model [van den Oord et al., 2019]. (B) replaces Brownian bridge dynamics with Brownian motion (BM) which doesn’t rely on pinning trajectories. Latents follow the transition density $z_t | z_s \sim N(z_s, t - s)$. (C) replaces the contrastive loss with the Variational Auto-Encoder (VAE). Below we detail how these ablations are implemented.
E.1 Implicit Dynamics (ID)

The Implicit Dynamics ablation is where we compare our explicit dynamics objective to van den Oord et al. (2019) which suggests an implicit latent dynamics objective. This ablation thus changes the learned latent dynamics. In van den Oord et al. (2019), they train two models. One is a non-linear encoder \( g_{\text{enc}}(x_t) = z_t \) which takes observation \( x_t \) (eg. a sentence) and maps it to a latent representation \( z_t \). Two is an autoregressive context model \( g_{\text{ar}}(z_{\leq t}) = c_t \) which summarizes a sequence of latent variables \( z_{\leq t} \) into a context latent representation \( c_t \). Rather than directly predicting a future observation \( x_{t+k} \) (\( k \) steps from the current timestep \( t \)), they model the density ratio that preserves the mutual information between \( x_{t+k} \) and the context variable \( c_t \):

\[
f_k(x_{t+k}) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})} \tag{13}
\]

They model this with a log-bilinear model \( f_k(x_{t+k}) = \exp(z_t^T W_k c_t) \) which applies a linear transformation \( W_k \) for modelling latents \( k \)-steps away from timestep \( t \). This way, they avoid directly learning the generative model \( p(x_{t+k}|c_t) \).

They train both models jointly via a contrastive InfoNCE loss. Given a set \( X = \{x_1, \ldots, x_N\} \) of \( N \) random samples (eg. sentences from different documents) containing one positive sample from \( p(x_{t+k}|c_t) \) and \( N-1 \) negative samples from a proposal distribution \( p(x_{t+k}) \), they optimize:

\[
\mathcal{L} = -\frac{1}{N} \sum_{x_j \in X} \log \left( \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right). \tag{14}
\]

The encoder \( g_{\text{enc}} \) for Implicit Dynamics has the same architecture as Time Control. The context encoder \( g_{\text{ar}} \) using a 2400-hidden-unit GRU, as done in van den Oord et al. (2019). We then train both encoders using the InfoNCE loss, as done in prior work. Since \( g_{\text{enc}} \) and \( g_{\text{ar}} \) are trained to align latents up to a linear rotation, we use \( g_{\text{enc}} \) for extracting the sentence embeddings.

E.2 Brownian Motion (BM)

The Brownian motion ablation is where we remove goal-conditioning in the dynamics. The main change is in distance function in Equation 3. BM instead optimizes the following equation:

\[
d(x_t, x_{t'}; f_\theta) = -\frac{1}{2\sigma^2} \left\| \frac{f_\theta(x_{t'})}{s_{t'}} - \frac{f_\theta(x_t)}{s_t} \right\|^2_2 \tag{15}
\]

where \( \sigma^2 \) is the variance in of the Wiener process, \( \sigma^2 = t' - t \).

E.3 Variational Auto-Encoder (VAE)

The Variational Auto-Encoder ablation is where we compare our contrastive objective with a variational one. This ablation changes the encoder objective from Section 3.1 from Equation 2 to the ELBO objective. Below we derive the ELBO objective. Similar to the contrastive objective, we’re deriving the ELBO over the triplet dynamics in the Brownian bridge.
The length mismatch in order as they are listed above. The ordering metric used in Table 4 is

The test dataset contains 431 articles. The section lengths have a similar breakdown:

- Abstract: 75.8 ± 1.4
- History: 191.5 ± 3.7
- Geography: 83.9 ± 1.5
- Demographics: 342.6 ± 4.6

The test dataset contains 431 articles. The section lengths have a similar breakdown:

- Abstract: 73.5 ± 2.6
- History: 180.2 ± 6.2
- Geography: 85.2 ± 2.7
- Demographics: 332.5 ± 8.6

The ordering metric used in Table 4 is 1 if all four section ids occur exactly once and come in the order as they are listed above.

The length mismatch in % used in Table 4 is calculated with respect to the training set lengths.
Wikihow (Koupaee & Wang, 2018) contains how-to articles organized by a title, method, and steps. Each article includes multiple methods for completing a multi-step procedural task such as “How to Register to Vote”. We scraped all the available English articles covering a wide range of topics following Koupaee & Wang (2018). We mark each with its own section id tokens: Each article is represented as “[TITLE] text [METHOD] text [STEP] 1 text [STEP] 2 text ...”

The training dataset contains 1566 articles. The section lengths are,

- Title: 10.4 ± 2.2
- Method: 8.7 ± 2.2
- Steps (total step length): 480.2 ± 231.5

The test dataset contains 243 articles. The section lengths are,

- Title: 10.7 ± 2.2
- Method: 8.6 ± 2.1
- Steps (total step length): 480.1 ± 224.0

The ordering metric used in Table 4 is 1 if all the section ids appear in order, the TITLE and METHOD section ids are not repeated, and the step numbers come in order. It’s 0 otherwise.

The deviation in section length measured in Table 3 is calculated with respect to the training set lengths. We check for whether the models are able to maintain the section lengths when it has to extrapolate. The most common failure mode is that the model generates incoherent text and allocates this text to the last section of what it’s generated thus far, resulting in the deviation in section lengths.

Recipe NLG (Bien et al., 2020) contains recipes, each with a title, ingredients and set of instructions. A recipe is constructed as “[TITLE] text [INGREDIENTS] text [DIRECTIONS] text”.

The training dataset contains 4000 recipes. The section lengths are,

- Title: 9.7 ± 3.4
- Ingredients: 23.8 ± 4.5
- Directions (total step length): 62.0 ± 14.5

The test dataset contains 1000 recipes. The section lengths are,

- Title: 9.4 ± 3.0
- Ingredients: 24.1 ± 4.5
- Directions (total step length): 63.3 ± 13.7

The ordering metric used in Table 4 is 1 if all the section ids appear exactly once and in order. It’s 0 otherwise.

Taskmaster-2 (TM-2) (Byrne et al., 2019) which contains conversations on finding restaurants between an assistant and a user. The assistant’s turn is marked with an “[ASSISTANT]” tag, and the user’s turn is marked with a “[USER]” tag.

The training dataset contains 2000 conversations. The section lengths are,

- User: 11.8 ± 5.6
- Assistant: 18.0 ± 3.7

The test dataset contains 1276 conversations. The section lengths are,

- User: 11.7 ± 5.6
- Assistant: 19.5 ± 3.0

The ordering metric used in Table 4 does not apply here because the user and assistant don’t take turns in the dialog.
Table 6: Perplexity after fine-tuning.

| Method      | Wikisection | Wikihow | TM-2 | TicketTalk | Recipe |
|-------------|-------------|---------|------|------------|--------|
| GPT2        | 5.9         | 15.3    | 4.5  | 4.4        | 7.5    |
| sec-dense   | 5.9         | –       | –    | –          | –      |
| sec-sparse  | 5.9         | –       | –    | –          | –      |
| VAE (8)     | 5.5         | 15.3    | 4.5  | 4.1        | 7.3    |
| VAE (16)    | 5.5         | 15.3    | 4.5  | 4.1        | 7.3    |
| VAE (32)    | 5.5         | 15.3    | 4.5  | 4.1        | 7.3    |
| ID (8)      | 5.5         | 14.9    | 4.5  | 4.0        | 7.3    |
| ID (16)     | 5.4         | 14.9    | 4.3  | 3.9        | 7.0    |
| ID (32)     | 5.3         | 14.9    | 4.2  | 3.9        | 6.7    |
| BM (8)      | 5.5         | 15.2    | 4.5  | 4.1        | 7.3    |
| BM (16)     | 5.5         | 15.3    | 4.5  | 4.1        | 7.3    |
| BM (32)     | 5.5         | 15.2    | 4.5  | 4.1        | 7.3    |
| TC (8)      | 5.5         | 15.2    | 4.3  | 4.0        | 6.9    |
| TC (16)     | 5.5         | 15.3    | 4.3  | 4.0        | 6.9    |
| TC (32)     | 5.5         | 15.2    | 4.3  | 4.0        | 7.0    |

The deviation in section length measured in Table 3 is calculated with respect to the training set lengths. We check for whether the models are able to maintain the utterance lengths between the user and assistant when it has to extrapolate. The most common failure mode is that the model generates incoherent text and allocates this text to the last section of what it’s generated thus far, resulting in the deviation in section lengths.

**TicketTalk** [Byrne et al., 2021] which contains conversations on booking movie tickets between an assistant and a user. The assistant’s and user’s turned are similarly marked as in TM-2.

The training dataset contains 2000 conversations. The section lengths are,

- User: 11.8 ± 5.6
- Assistant: 18.0 ± 3.7

The test dataset contains 1276 conversations. The section lengths are,

- User: 11.7 ± 5.6
- Assistant: 19.5 ± 3.0

The deviation in section length measured in Table 3 is calculated similarly to TM-2.

The ordering metric used in Table 4 applies because the user and assistant take turns in the dialog. The ordering metric is 1 if the user and assistant take turns in the conversation. It’s 0 otherwise.

**ROC Stories** [Mostafazadeh et al., 2016] is a short stories dataset. Each story contains 5 sentences. No additional tokens are added in this dataset. The training dataset contains 2000 stories, and the test dataset contains 1000 stories. This dataset was used in the original text infilling experiments we had. Those previous results are in Table 21 Table 11 and Table 12.

G **PERPLEXITY AFTER FINE-TUNING**

Table 6 reports the final perplexity scores after fine-tuning GPT2 on the different domains with the methods. We fine-tune for 10 epochs and checkpoint the models every 1000 steps; we keep the model checkpoint that scores the lowest PPL on a held-out validation set.

H **DIFFERENCES IN THE CORRECTED MANUSCRIPT**

This section discusses updates and differences from an earlier version of this paper due to correcting bugs during a post-publication code review and re-running experiments. Sources of
randomness, such as the behavior of the torch dataloader, GPU nondeterminism, and Weights and Biases logging have caused some differences in the experimental outcomes even in cases where there were no bugs. To make the differences clear, we highlight both original and current results, as well as a short explanation of any differences. The numbers reported in the main text correspond to the current version of the codebase available at [https://github.com/rosewang2008/language_modeling_via_stochastic_processes](https://github.com/rosewang2008/language_modeling_via_stochastic_processes) with version hash: 95ded1e82f904e1c2dbce5701f54ab880a1a1e62.

H.1 DISCOURSE COHERENCE

For reference, Table 7 shows both the original and new results. The original results are marked with ORIGINAL.

Differences In the previous version, ALBERT, SimCSE and TC were competitive with each other. In the updated version, BERT is competitive with these methods as well. TC is the most competitive out of all the GPT2-based models.

Cause This discrepancy is due to a wandb step logging issue where wandb previously only reported on early training epochs. Therefore, the previous discourse coherence results generally under-report the actual performance—this particularly impacts Wikisection. In general, all methods improve and do better than random accuracy after resolving this issue.

H.2 LENGTH MISMATCH

For reference, Table 8 shows both the original and new results. The original results are marked with ORIGINAL.

Differences In the previous version, TC performed the best. In the updated version, STATIC VAE performs the best, followed by STATIC TC.

Causes The variability is caused by the dataset loader being seeded separately from usual seeding sources, GPU nondeterminism combined with sensitivity of the evaluation to this nondeterminism. The experiments reported in the paper are all seeded. If the experiments are run on the same machine, they replicate exactly. We observe different experimental results when run on different machines. For example, in our experiments, TC (8) on the history section mismatches the original section by about ~ 45 tokens on one machine, and ~ 21 tokens on another machine.

H.3 SECTION LENGTH DEVIATION

For reference, Table 9 shows both the original and new results. The original results are marked with ORIGINAL.

Differences In the previous version, TC performed the best in not deviating from the expected section lengths (Table 4). In the updated version, this is still the case with STATIC TC. Surprisingly, we found that STATIC BM and VAE performs competitively in one of the domains as well.

Causes Besides the variability causes mentioned in Table 3 section, there was an improper implementation of forced generation related to how we specify the min and max generation length. This affected all methods equally by cutting decoding short.

H.4 ORDERING

For reference, Table 10 shows both the original and new results. The original results are marked with ORIGINAL.

Differences In the previous version, TC performed the best along with the VAE baseline in correct ordering (Table 5). In the updated version, there’s a lot more overlap between methods.

Causes The causes are the same as for Table 4.

I ORIGINAL TEXT-INFILLING AND HUMAN EVALUATION RESULTS

Due to the difficulty of re-running the human evaluation results, we have decided not to include an updated version of the human evaluation results. This impacts our text-infilling setting and long text generation setting. For transparency, we have included the text and results from the original version below.
Table 7: Discourse coherence accuracy measured by the test accuracy of the trained linear classifier, reporting \(\mu \pm \text{standard error over 3 runs. Random accuracy is 50\%. When applicable, the methods are run with varying latent dimensions marked in parentheses (dim). The label ORIGINAL are the experiments from the original camera-ready version.}

Original text for text infilling experiment  We also report human evaluations on how coherent the generated sentence is as a fill-in sentence. Participants were asked to rank the generated fill-in sentence from ILM, LM, and Time Control on a scale of 1-5 (not reasonable to very reasonable). Appendix [K] includes more details on the human experiment setup.

We evaluate the text coherence with the BLEU score ([Papineni et al. 2002]), ROUGE ([Lin 2004]), BLEURT ([Sellam et al. 2020]) and BERTScore ([Zhang et al. 2019]) between the generated and ground truth infill sentence. Due to space constraints, the last three metrics are in the Appendix, Table [21]. We also report human evaluations on how coherent the generated sentence is as a fill-in sentence. Participants were asked to rank the generated fill-in sentence from ILM, LM, and Time Control on a scale of 1-5 (not reasonable to very reasonable). Appendix [K] includes more details on the human experiment setup.

The BLEU scores are summarized in Table[11]. Time Control generates fill-in sentences that much more closely overlap with the ground truth than ILM and LM. The ablated methods perform worse to varying degrees. On average, VAE performs worse than LM and ID. This suggests that the
Table 8: Percentage of length mismatch (MM) during generation. The label ORIGINAL are the experiments from the original camera-ready version.

| Method            | MM % (↓) |
|-------------------|----------|
| ORIGINAL GPT2     | 17.5 ± 0.1 |
| ORIGINAL SD       | 10.0 ± 0.1 |
| ORIGINAL SS       | 10.6 ± 0.1 |
| GPT2              | 9.9 ± 3.6  |
| SD                | 28.3 ± 7.9 |
| SS                | 223.9 ± 18.5 |
| ORIGINAL VAE (8)  | 10.8 ± 0.1 |
| ORIGINAL VAE (16) | 9.6 ± 0.1  |
| ORIGINAL VAE (32) | 8.7 ± 0.1  |
| VAE (8)           | 8.1 ± 3.0  |
| VAE (16)          | 14.1 ± 7.8 |
| VAE (32)          | 18.5 ± 4.5 |
| Static VAE (8)    | 8.9 ± 2.2  |
| Static VAE (16)   | 13.1 ± 5.8 |
| Static VAE (32)   | 6.7 ± 1.9  |
| ORIGINAL ID (8)   | 10.8 ± 0.1 |
| ORIGINAL ID (16)  | 154.8 ± 0.1|
| ORIGINAL ID (32)  | 138.6 ± 0.1|
| ID (8)            | 13.3 ± 3.8 |
| ID (16)           | 61.8 ± 5.5 |
| ID (32)           | 84.6 ± 11.0|
| Static ID (8)     | 12.3 ± 3.4 |
| Static ID (16)    | 77.1 ± 4.4 |
| Static ID (32)    | 79.6 ± 8.0 |
| ORIGINAL BM (8)   | 9.2 ± 0.1  |
| ORIGINAL BM (16)  | 17.8 ± 0.1 |
| ORIGINAL BM (32)  | 10.8 ± 0.1 |
| BM (8)            | 16.2 ± 11.4|
| BM (16)           | 21.1 ± 11.8|
| BM (32)           | 51.6 ± 14.2|
| Static BM (8)     | 15.4 ± 9.2 |
| Static BM (16)    | 11.8 ± 5.4 |
| Static BM (32)    | 13.0 ± 5.0 |
| ORIGINAL TC (8)   | 16.8 ± 0.2 |
| ORIGINAL TC (16)  | 7.9 ± 0.1  |
| ORIGINAL TC (32)  | 9.3 ± 0.1  |
| TC (8)            | 13.1 ± 4.4 |
| TC (16)           | 15.4 ± 10.1|
| TC (32)           | 28.1 ± 12.2|
| Static TC (8)     | 12.8 ± 3.2 |
| Static TC (16)    | 13.0 ± 7.5 |
| Static TC (32)    | 13.0 ± 7.8 |
interpolating embeddings learned via the variational objective yields embeddings which hurt the autoregressive model’s ability to generate locally coherent text.

**Original text for forced long text human evaluations** The third metric is human evaluations, which measures the long-tail text quality. We examine the long-tail behavior as the complete 1000-token generation is long.

[..] These results are additionally confirmed by our human evaluation experiments: human evaluators rank Time Control’s extrapolation ability better on all latent dimension settings than that of GPT2 (Table 13).
Table 10: Ordering in forced long text generation. ROC Stories and TM-2 omitted because they are not applicable. The label ORIGINAL are the experiments from the original camera-ready version.

| Method                      | Wikisection | Wikihow | TicketTalk | Recipe |
|-----------------------------|-------------|---------|------------|--------|
| ORIGINAL GPT2               | 50.4 ± 1.0  | 61.5 ± 3.5 | 75.8 ± 1.6 | 36.8 ± 3.7 |
| GPT2                        | 47.4 ± 5.4  | 61.3 ± 9.1 | 19.7 ± 2.3 | 71.9 ± 2.5 |
| ORIGINAL VAE (8)            | 57.1 ± 3.5  | 66.3 ± 1.7 | 66.1 ± 4.1 | 71.4 ± 0.6 |
| ORIGINAL VAE (16)           | 47.3 ± 3.3  | 60.5 ± 2.9 | 0.8 ± 0.0  | 45.6 ± 0.5 |
| ORIGINAL VAE (32)           | 58.3 ± 0.3  | 60.9 ± 5.6 | 38.9 ± 1.7 | 87.5 ± 0.2 |
| VAE (8)                     | 49.7 ± 4.3  | 56.8 ± 10.3| 44.0 ± 10.5| 61.8 ± 16.8|
| VAE (16)                    | 49.9 ± 5.3  | 67.7 ± 11.2| 50.9 ± 17.4| 54.0 ± 35.7|
| VAE (32)                    | 41.3 ± 5.6  | 62.6 ± 5.9 | 23.1 ± 15.5| 85.2 ± 3.7 |
| STATIC VAE (8)              | 51.0 ± 4.3  | 53.9 ± 11.6| 45.5 ± 11.3| 70.6 ± 13.9|
| STATIC VAE (16)             | 49.8 ± 6.1  | 67.9 ± 8.4 | 52.0 ± 18.6| 64.8 ± 14.7|
| STATIC VAE (32)             | 45.9 ± 6.1  | 61.1 ± 6.0 | 21.9 ± 14.8| 78.8 ± 3.9 |
| ORIGINAL ID (8)             | 34.6 ± 0.0  | 59.3 ± 2.9 | 30.8 ± 1.6 | 68.7 ± 2.3 |
| ORIGINAL ID (16)            | 35.7 ± 0.0  | 31.7 ± 0.9 | 63.9 ± 0.4 | 78.1 ± 3.1 |
| ORIGINAL ID (32)            | 47.9 ± 2.0  | 16.9 ± 0.6 | 45.6 ± 0.6 | 85.6 ± 2.3 |
| ID (8)                      | 50.8 ± 3.8  | 60.5 ± 11.9| 50.0 ± 3.4 | 65.6 ± 15.0|
| ID (16)                     | 62.3 ± 6.7  | 27.8 ± 20.1| 55.4 ± 19.7| 55.7 ± 8.5 |
| ID (32)                     | 64.0 ± 3.2  | 44.1 ± 7.9 | 64.8 ± 6.2 | 81.5 ± 5.2 |
| STATIC ID (8)               | 51.0 ± 5.0  | 68.5 ± 10.4| 49.2 ± 13.4| 75.9 ± 7.1 |
| STATIC ID (16)              | 21.2 ± 4.1  | 28.3 ± 20.9| 54.3 ± 21.3| 56.6 ± 15.7|
| STATIC ID (32)              | 35.4 ± 7.0  | 48.5 ± 8.9 | 67.8 ± 10.5| 77.0 ± 6.4 |
| ORIGINAL BM (8)             | 54.7 ± 1.9  | 56.0 ± 3.5 | 77.0 ± 4.7 | 82.7 ± 0.4 |
| ORIGINAL BM (16)            | 61.4 ± 1.0  | 63.8 ± 4.7 | 43.3 ± 2.4 | 34.9 ± 1.5 |
| ORIGINAL BM (32)            | 44.3 ± 1.4  | 50.2 ± 1.2 | 14.8 ± 1.0 | 87.1 ± 0.1 |
| BM (8)                      | 50.1 ± 3.4  | 61.8 ± 8.9 | 53.8 ± 17.3| 71.0 ± 6.9 |
| BM (16)                     | 40.7 ± 6.1  | 51.3 ± 20.5| 37.7 ± 15.4| 37.3 ± 16.6|
| BM (32)                     | 43.4 ± 7.4  | 41.9 ± 15.1| 49.2 ± 3.7 | 53.2 ± 7.6 |
| STATIC BM (8)               | 51.5 ± 6.6  | 65.2 ± 10.1| 54.2 ± 19.3| 54.9 ± 17.9|
| STATIC BM (16)              | 42.8 ± 5.3  | 66.5 ± 15.1| 22.8 ± 19.1| 45.0 ± 14.0|
| STATIC BM (32)              | 46.3 ± 4.2  | 66.7 ± 10.3| 23.4 ± 12.6| 30.1 ± 12.4|
| ORIGINAL TC (8)             | 52.3 ± 2.5  | 76.7 ± 7.5 | 81.8 ± 1.0 | 41.7 ± 1.0 |
| ORIGINAL TC (16)            | 57.9 ± 1.0  | 63.1 ± 6.2 | 88.0 ± 1.3 | 64.1 ± 1.0 |
| ORIGINAL TC (32)            | 36.5 ± 2.8  | 59.1 ± 5.5 | 83.6 ± 1.3 | 76.4 ± 1.0 |
| TC (8)                      | 49.2 ± 10.5 | 64.3 ± 14.7| 48.3 ± 23.8| 56.0 ± 10.4|
| TC (16)                     | 42.5 ± 7.0  | 66.2 ± 17.7| 58.8 ± 11.7| 66.6 ± 10.0|
| TC (32)                     | 36.9 ± 7.7  | 69.7 ± 12.3| 35.4 ± 16.5| 77.0 ± 3.5 |
| STATIC TC (8)               | 50.5 ± 11.2 | 69.7 ± 12.3| 58.1 ± 19.6| 55.7 ± 11.3|
| STATIC TC (16)              | 48.5 ± 5.5  | 67.9 ± 19.0| 67.3 ± 8.9 | 70.3 ± 7.5 |
| STATIC TC (32)              | 52.1 ± 6.4  | 71.5 ± 10.5| 49.1 ± 18.0| 82.4 ± 3.3 |

Table 11: BLEU on ground truth infill and generated between 1 and 5. Higher is better.

| Method | BLEU (†) |
|--------|----------|
| LM     | 1.54 ± 0.02 |
| ILM    | 3.03 ± 0.11 |
| VAE (8) | 0.75 ± 0.17 |
| VAE (16)| 0.62 ± 0.07 |
| VAE (32)| 0.03 ± 0.0 |
| ID (8)  | 2.9 ± 0.3 |
| ID (16) | 0.9 ± 0.0 |
| ID (32) | 1.0 ± 0.1 |
| TC (8)  | 3.80 ± 0.06 |
| TC (16) | 4.30 ± 0.02 |
| TC (32) | 5.4 ± 0.11 |

Table 12: Human evaluations on text infilling. Scores were ranked between 1 and 5. Higher is better.

| Method | Human |
|--------|-------|
| LM     | 2.4 ± 0.06 |
| ILM    | 3.77 ± 0.07 |
| TC (8) | 3.64 ± 0.07 |

Table 13: Human evaluations on tail end quality in forced long text generation. Scores were ranked better.
J ORIGINAL GENERATION EXAMPLES

The following generation examples were generated from the original paper manuscript.

We provide example outputs from the different models on text infilling. We denote the prefix sentences as “LEFT CONTEXT”, the suffix sentences as “RIGHT SENTENCES”, the generated output from the models as “GENERATED” and the ground truth infill sentence as “GT”.

J.1 TEXT INFILLING

LM  See Table [14]

Table 14: Example of text infilling with Donahue et al. (2020)’s LM model on ROCStories.

| LEFT CONTEXT: John couldn’t wash his clothes because his washing machine was broken. | GENERATED: He had an idea. |
| GT: John didn’t know how to fix washing machines. | RIGHT CONTEXT: He bought a book to tell him how to repair it. He changed the fan belt. When he plugged it back in, it worked. |

| LEFT CONTEXT: I found a stray cat yesterday. I gave it some milk. It came back again today. I was going to give it some milk but didn’t. | GENERATED: I guess I need to get something. |
| GT: I had run out of milk. | RIGHT CONTEXT: |

| LEFT CONTEXT: Toni took a trip to the local fair with her family. She immediately laid eyes on a pretty balloon. The balloon was red and star shaped. Unfortunately it slipped out of her hands. |
| GENERATED: Her swollen fingers had horrible rashes. |
| GT: She was sad to see it go. | RIGHT CONTEXT: |

| LEFT CONTEXT: Miles was a roofer. He loved his job and was very good at it. |
| GENERATED: He dreamed of finding a good title. |
| GT: However, one day he was not paying attention and fell off the roof. |
| RIGHT CONTEXT: Luckily, he only broke his leg. Miles decided that he needed to be more careful. |

| LEFT CONTEXT: Lindsey was feeling especially antsy at school. |
| GENERATED: The school wasn’t paying much attention to her. |
| GT: She did not want to go to Lacrosse practice after school. |
| RIGHT CONTEXT: She tried to fake a sickness, but nobody believed her. Lindsey ended up skipping practice without an excuse. She hoped she wouldn’t get in trouble. |

| LEFT CONTEXT: Last night Jack went on a blind date. When he arrived at the restaurant a he heard someone yell his name. He spotted a blonde waving from across the room. |
| GENERATED: She was a waiter. |
| GT: She was nothing like the photo she sent. |
| RIGHT CONTEXT: But he figured he would try to make the most of it. |

| LEFT CONTEXT: Rico was working really hard on his essay for English. After working on it all night, he decided to take a short nap. He slept in and was late for for school. When he got to school he noticed that his essay was gone. |
| GENERATED: teacher found the essay in his bag and he passed it. |
| GT: Rico’s teacher let him turn in the essay the next day. |
| RIGHT CONTEXT: |

ILM  See Table [15]
Table 15: Example of text infilling with Donahue et al. (2020)’s ILM model on ROCStories.

| LEFT CONTEXT | GENERATED | GT | RIGHT CONTEXT |
|--------------|-----------|----|---------------|
| My 98 year old friend and I played blackjack yesterday. | Last week’s family game night was intense. | We were playing Monopoly and nobody made any headway for hours. Everyone was trying their hardest to win and the game kept going. It wasn’t until we finally decided to check the rules that we knew why. There were many different pieces missing. |
| Tom was jealous of his brother. His brother was older and stronger. His brother went wherever he wanted. | Tom decided to use steroids. | One day his brother was grounded for staying out too late. |
| His first time at the opera. He only went after his girlfriend begged. He sat for three hours in sheer boredom. Finally it was over. | He turned on the tv and heard her thanking him. | This made Tom really happy. |
| My best friend played a video game online. One day, she met a nice boy there. They talked every day and fell in love. They finally met in person and hit it off. | The two became a very loving couple. |
| The family decided to go to the park for Veteran’s Day. | They asked permission from the park manager. | We had a barbecue outside. |
| Jennifer always wanted to have the birthday of her dreams. Her parents surprised her with a big birthday party at the park. All of her friends and family came to celebrate her special day. She was so happy as she blew out the candles on her cake. | Her parents had made a big surprise to her by turning the cake into a carnival. |
| Terry’s daughter was in extreme pain in her mouth. | Terry and her husband decided to stop the baby from teething. | After an evaluation, Terry realized that her daughter was teething. |

Time Control See Table 16
Table 16: Example of text infilling with Time Control (d=8) on ROCStories.

| LEFT CONTEXT: | GENERATED: | GT: | RIGHT CONTEXT: |
|---------------|------------|-----|-----------------|
| My day on Sunday this week went as expected. My family and I attended Church from 8 to 11 in the morning. | We went to a local church in the afternoon. | Afterwards, we went to eat dinner at a restaurant. | The rest of my day was spent doing homework assignments. I then went to sleep feeling happy about my life. |
| His first time at the opera. He only went after his girlfriend begged. He sat for three hours in sheer boredom. Finally it was over. | He was so happy he didn’t want to leave. | He learned he didn’t like the opera. | |
| My best friend played a video game online. One day, she met a nice boy there. They talked every day and fell in love. They finally met in person and hit it off. | Until that day, they were married. | The two became a very loving couple. | |
| The other day at the clinic I had to help put a dog down. He seemed really sad and lonely. Like he knew what was going to happen. | As we laid it down and it took its final breaths it stared at me. | I stayed calm, but cried after we were finished. | |
| Tom was jealous of his brother. His brother was older and stronger. His brother went wherever he wanted. | Tom was jealous of his brother. | One day his brother was grounded for staying out too late. | |
| Jays habit of buying expensive beer was catching up to him. He was spending more money on beer than food. He needed to find another source of income to support this habit. A friend recommended he try out Amazon MTurk. | He found the site and bought a few beers. | Jay become slightly less poor from Amazon Mturk. | |
| John couldn’t wash his clothes because his washing machine was broken. | John went to the store to buy a new one. | John didn’t know how to fix washing machines. | He bought a book to tell him how to repair it. He changed the fan belt. When he plugged it back in, it worked. |
Table 17: Example of text infilling with **Time Control** (d=16) on ROCStories.

| LEFT CONTEXT | GENERATED | GT | RIGHT CONTEXT |
|--------------|-----------|----|---------------|
| Tom was jealous of his brother. His brother was older and stronger. His brother went wherever he wanted. | Tom’s brother was very jealous of his brother. | One day his brother was grounded for staying out too late. | This made Tom really happy. |
| Jackie was 11 and had to get braces. She was worried about what her friends would think. She tried to hide them when she first got them. | But she was too embarrassed to tell them. | Eventually her friends saw them and she was embarrassed. | Her friends noticed she was embarrassed and decided to comfort her. |
| Sally was going to surprise the office with a cake. Sally felt that a cake would be a good way to make them smile. She went to the supermarket to pick up the cake. | She bought the cake and was very happy with it. | At the office she gathered the employees around the conference table. | She then brought out a cake and they all felt better. |
| Lars was playing XBOX. His controller stopped working during a game. Lars didn’t have a car so he had to walk all the way to the store. The store was being remodeled when he got there so he went to another. | Video games were all over the floor. | Lars wasn’t able to find a controller. | |
| Emma had been working as a dishwasher. Her hands cracked and bled from the hot soapy water. Then her mom noticed and concocted a special salve for her. Emma used the salve every night before bed. | Soon she was able to keep her hands clean and happy. | Her hands got better. | |
| Jerome dribbled the ball quickly. Sam tried to grab it from him, but wasn’t fast enough. He chased Jerome down the court. | He scored a point with a long shot. | Jerome pushed Sam backwards and threw the ball. | Jerome scored points for his team. |
| Class started in 10 minutes and I had a math assignment due. My older friend volunteered to do it for me. Unfortunately, my teacher found out about the copying. | She was very upset with me for doing the assignment. | She crumbled my paper and threw it in the trash. | My teacher gave me an F for the assignment. |
Table 18: Example of text infilling with **Time Control** (d=32) on ROCStories.

| LEFT CONTEXT | GENERATED | GT | RIGHT CONTEXT |
|--------------|-----------|----|---------------|
| I went to the park to play frisbee with my dog. | We played all day. | I tossed the frisbee to my dog and he would catch it in his mouth. But my dog went and got it. |
| I was tired of working at Walmart. The hours were bad. The store music was awful. | I was very bored. | I handed my two weeks in to the manager. I then found another job and was happy. |
| Sam bought a new SUV. It was all wheel drive. He figured he would take it off road. He hit a few hard bumps and broke his suspension. | Unfortunately he had to pay a lot of money for it. Sheepishly, he brought it to the dealership for repair. |
| Missy got drunk and went to get a tattoo. She decided to get a tattoo on her forehead. The next day, Missy was horrified at what she had done. | Missy scraped up her money to pay for a tattoo removal procedure. After much wasted money, the tattoo was gone. |
| Jake was going on a road trip to see his family. He Got in the car and drove. | Jake’s family was driving down the road. The car’s tires exploded due to too much air. Jake hitchhiked for 30 miles. When Jake got to his family he was happy his trip was over. |
| Bob decided to start a business. | However, he did not know the market at all. He opened up a grocery store and was doing very well. After a year, his profits dropped and he had to declare bankruptcy. Bob worked hard and reopened his business. |
| I hadn’t seen my girlfriend in a while. She got a new job so it’s hard to talk. The job takes up all of her time. | I had to ask her to go out with me. Finally she called me to hang out. Liver | I was really happy to see her and we made plans. |

**Variational Auto-Encoder** See Table 19.
Table 19: Example of text infilling with Variational Auto-Encoder on ROCStories.

LEFT CONTEXT: I went to the park to play frisbee with my dog.
GENERATED: served served served [...] 
GT: I tossed the frisbee to my dog and he would catch it in his mouth.
RIGHT CONTEXT: I accidentally threw the frisbee too far. The frisbee landed into the pond. But my dog went and got it.

LEFT CONTEXT: Tim had a dentist appointment today. He was going to get his wisdom teeth pulled.
GENERATED: [...] 
GT: His dentist numbed his gums.
RIGHT CONTEXT: His wisdom teeth popped out instantly. It was less hassle than he thought it would be.

LEFT CONTEXT: Tom was studying for the big test. He then fell asleep due to boredom.
GENERATED: The The The The The The The The The [...] 
GT: He slept for five hours.
RIGHT CONTEXT: He woke up shocked. Tom hurried to study as much as possible before the test.

Implicit Dynamics  See Table 20

Table 20: Example of text infilling with Implicit Dynamics on ROCStories.

LEFT CONTEXT: Tom was jealous of his brother. His brother was older and stronger. His brother went wherever he wanted.
GENERATED: he wasn’t like him. he wasn’t like him........................................... isn’t... doesn’t... doesn’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don’t... don't. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t. don’t.
GT: One day his brother was grounded for staying out too late.
RIGHT CONTEXT: This made Tom really happy.

LEFT CONTEXT: The family decided to go to the park for Veteran’s Day.
GENERATED: I didn’t I knew why why why why [...] 
GT: We had a barbecue outside.
RIGHT CONTEXT: In addition, everyone decided to take a swim in the water. Before we left, we had a huge cake. We then swam for an hour at the lake.

J.2 FORCED LONG TEXT GENERATION
We include examples of GPT2 long text generation in Figures 3 and 4.

K HUMAN EXPERIMENTAL SETUP
We use Mechanical Turk to collect human evaluation for long text generation and text infilling models. Examples from all the systems are selected randomly and each example is evaluated by 10 annotators.

K.1 LONG TEXT GENERATION EVALUATION
The annotators are provided with a dialogue between a movie assistant and a user generated by GPT and Time Control, where the goal of the assistant is to help the user to book a movie ticket. We show the last few generated interactions between the assistant and the user to the annotators, and ask them to assess whether the provided snippet could be a reasonable ending for such a conversation. We use a 5-point Likert scale where 1 corresponds to a “Very Unreasonable Ending” and 5 corresponds to a “Very Reasonable Ending”.

K.2 TEXT INFILLING EVALUATION
Note: These results were generated from the original ICLR camera-ready (v1).

Given the preceding and following context, we ask annotators to evaluate whether the generated fill-in sentence is a reasonable completion. We evaluate generated fill-in sentence by ILM, LM and Time Control on a scale of 1-5.
EXAMPLE TRAJECTORIES OF THE LEARNED LATENT SPACE

See Figures 7-10 for example latent trajectories over coherent vs. incoherent (randomly scrambled) held-out Wikisection documents. These trajectories show the recovered latent structure by Time Control and the three ablations from our work. Figure 7 are latent trajectories from Time Control. Figure 8 are latent trajectories from Implicit Dynamics. Figure 9 are latent trajectories from Brownian motion. Figure 10 are latent trajectories from Variational Auto-Encoder.

WHY DOES VARIATIONAL AUTO-ENCODER PERFORM WELL ON THE RECIPE DOMAIN?

In Table 4, VAE does much better than TC. Here we investigate potential reasons why.

We looked into comparing the latent structure recovered by the VAE baseline and TC; see Figure 11. What we found is the Time Control is best at extracting time over the course of the document (i.e. its latents recover a Bridge process correlated with time), whereas the VAE doesn’t elicit strong temporal structure. We hypothesize that temporal structure is not all you need to succeed in the Recipe domain.
Figure 4: Example 2 of GPT2 forced long text generation.

N  ADDITIONAL EXPERIMENTS

On discourse coherence, we ran ALBERT on $k = 1$ since ALBERT is trained on this setting. We found that it got slightly above random chance, unlike the most of the baselines we compare against. The discourse accuracy is $50.1 \pm 9.0$ on Wikisection, $55.8 \pm 4.0$ on TicketTalk and $60.8 \pm 2.8$ on TM-2.

O  PREVIOUS EXPLORATORY THREADS

O.1  SUCCESS AND FAILURE MODES IN FINE-TUNING GPT2

We found that fine-tuned GPT2 was able to replicate certain aspects of a document corpus, such as section header ordering (92% accurate). However, when going from document-level to section-level statistics, we noticed that GPT-2 seemed to be either undershooting or overshooting the section lengths. These results are reported in Table 2.

O.2  INVESTIGATING THE IMPORTANCE OF DISTANCES BETWEEN SAMPLED SENTENCES

In earlier iterations of the work, we did explore learning with pairwise contrasts with fixed t-distances, e.g. distances of length 1, 5, and 10 sentences. We observed that the resulting latent trajectories elicited different fits to Brownian bridge dynamics, and the quality in fit varied in t across domains; we’ve include some examples in Figure 12 on Wikisection.
Figure 5: Example 1 of Time Control forced long text generation.
[ USER ] Hello. I am thinking of going to see a movie today. [ ASSISTANT ] OK. And where will you be seeing the movie? [ USER ] Maybe in Creek's End, Oregon? [ ASSISTANT ] Creek's End, Oregon. Got it. Is there a particular movie you have in mind? [ USER ] Just any James Bond movie. [ ASSISTANT ] OK. I think you are talking about No Time To Die. Does that sound right? [ USER ] Yes it is, No Time To Die is the name. [ ASSISTANT ] I think action movies are always a great choice. In Creek's End, Oregon, I see that No Time To Die is showing in two theaters: AMC Mountain 16 and AMC Mercado 24. [ USER ] I want the AMC Mercado 24 its what I used to call it. [ ASSISTANT ] It looks like tonight's remaining showtimes for No Time To Die at AMC Mercado 24 are 5:00pm, 7:10pm, and 9:45pm. Which is best for you? [ USER ] Can I change the theater from the one I picked because the other one has other showings around it? [ ASSISTANT ] OK, No Time To Die it is. [ USER ] Can I have showtimes for two instead of one? [ ASSISTANT ] Sure, you chose AMC Mercado 24. [ USER ] Can I change the time, I want the earliest showtime instead? [ ASSISTANT ] Let me check showtimes at AMC Mercado 24. [ USER ] Can I actually change the showtime to the showtime you said? [ ASSISTANT ] Sure, you chose the showtime at 3:00pm. [ USER ] Okay that works better, can I change the time again. [ ASSISTANT ] Remaining showtimes for No Time To Die at AMC Mercado 24 are 4:30pm, 6:40pm and 9:45pm. [ USER ] Can I purchase two tickets to the showing of No Time To Die at AMC Mercado 24? [ ASSISTANT ] Sure, you would like two tickets for No Time To Die at AMC Mercado 24 at 3:00pm. Correct? [ USER ] Yes, please. [ ASSISTANT ] Okay. So to confirm you would like two tickets for No Time To Die at AMC Mercado 24 at 3:00pm tonight at the AMC Mercado 24 theater? [ USER ] That's right. [ ASSISTANT ] Is it OK to go ahead and purchase these tickets? [ USER ] Sure, let's do it. [ ASSISTANT ] OK. Your tickets are purchased and details for how to proceed have been sent to your email address. Can I help with anything else? [ USER ] Actually, let me think before I purchase the tickets since it will be a rough night at work. [ ASSISTANT ] I understand. Could you please detail how I got these tickets? [ USER ] Can I just send them to my email? [ ASSISTANT ] Sure. Let me see what I can find. [ ASSISTANT ] Unfortunately I cannot find any movie title that matches your description. [ USER ] Can I get the dealer's list of movie names from the AMC Mercado 24 website? [ ASSISTANT ] Showtimes for No Time To Die at AMC Mercado 24 are 4:30pm, 7:10pm, and 9:45pm. [ USER ] Ok, let me see what I can find. [ ASSISTANT ] Unfortunately I cannot find any movie title that matches your description. [ USER ] No I think we have to choose between No Time To Die and King Johnny please. [ ASSISTANT ] Got it. Let me see what I can find. [ ASSISTANT ] Unfortunately I cannot find any movie title that matches your description. [ USER ] Ok, well, I don't want to buy any tickets. [ ASSISTANT ] Showtimes for No Time To Die at AMC Mercado 24 are 3:00pm, 5:10pm, 7:30pm and 9:45pm. [ USER ] Can you fix something else? [ ASSISTANT ] That's it for now. [ USER ] Thanks. Could you actually order the tickets? [ ASSISTANT ] No need, I think we are done with this. [ USER ] Can you just order these? [ ASSISTANT ] Sure. I will be buying these tickets for you at AMC Mercado 24 at 3:00pm today. Is there anything else you would like to know? [ USER ] Can I make sure my debit card got charged again? [ ASSISTANT ] Sure. Let me see what I can find. [ ASSISTANT ] Unfortunately I cannot find any movie title that matches your description. [ USER ] Ok! No problem. [ ASSISTANT ] Showtimes for No Time To Die at AMC Mercado 24 are 4:30pm, 7:10pm, and 9:45pm. [ USER ] Ok great, I would like to make this change. [ ASSISTANT ] Is there anything I can help with now? [ USER ] Please, could you send the tickets to my email? [ ASSISTANT ] Sure. I am glad to help. Let me know what you need help with.

Figure 6: Example 2 of Time Control forced long text generation.
| Method      | Latent dim | BERTScore | ROUGE | BLEURT |
|-------------|------------|-----------|-------|--------|
|             |            | Precision | Recall | F1 | 1-F1 | 2-F1 | L-F1 |       |
| LM          | -          | 0.45      | 0.50  | 0.47 | 22.6 | 1.8  | 14.0 | 0.31  |
| ILM         | -          | 0.50      | 0.51  | 0.50 | 21.6 | 1.8  | 20.7 | 0.33  |
| VAE         | 8          | 0.21      | 0.26  | 0.21 | 1.1  | 0.9  | 1.1  | 0.27  |
| VAE         | 16         | 0.17      | 0.25  | 0.18 | 1.4  | 0.1  | 1.5  | 0.26  |
| VAE         | 32         | 0.10      | 0.10  | 0.12 | 0.2  | 0.6  | 0.2  | 0.26  |
| InfoNCE     | 8          | 0.22      | 0.29  | 0.23 | 7.7  | 1.3  | 7.4  | 0.18  |
| InfoNCE     | 16         | 0.18      | 0.28  | 0.20 | 1.9  | 0.0  | 1.9  | 0.18  |
| InfoNCE     | 32         | 0.20      | 0.28  | 0.21 | 3.0  | 1.0  | 5.8  | 0.18  |
| TC (Ours)   | 8          | 0.51      | 0.51  | 0.51 | 17.5 | 1.6  | 16.1 | 0.30  |
| TC (Ours)   | 16         | 0.47      | 0.49  | 0.49 | 15.9 | 1.5  | 14.3 | 0.34  |
| TC (Ours)   | 32         | 0.50      | 0.50  | 0.50 | 18.4 | 1.4  | 17.1 | 0.32  |

Table 21: BERTScore (Zhang et al., 2019), ROUGE, and BLEURT (Sellam et al., 2020) on ground truth infilled sentence and the generated sentence.
Figure 7: Time Control’s latent trajectories over coherent vs. incoherent (randomly scrambled) held-out Wikisection documents. The encoder learns Brownian bridge-like latent trajectories over the coherent documents. The incoherent documents map to noisy trajectories that don’t evolve over time.
Figure 8: Implicit Dynamics’s latent trajectories over coherent vs. incoherent (randomly scrambled) held-out Wikisection documents.
Figure 9: Brownian motion’s latent trajectories over coherent vs. incoherent (randomly scrambled) held-out Wikisection documents.
Figure 10: Variational Auto-Encoder’s latent trajectories over coherent vs. incoherent (randomly scrambled) held-out Wikisection documents.
Figure 11: Comparing the latent structure recovered by the VAE baseline and TC on held-out Recipe documents. Noticeably, TC learns temporally relevant activations whereas VAE latent does not correlate with time.
Figure 12: Recovered latent trajectories on held-out Wikisection documents where we used pair-wise contrasts with varying time distances; this was on TC, $d = 8$. Notice how the recovered latent structure varies depending on $k$, the distance between sampled sentences.