BBScoreV2: Learning Time-Evolution and Latent Alignment from Stochastic Representation

Tianhao Zhang[1]*, **Zhecheng Sheng**[1]*, **Zhexiao Lin**[2]*, **Chen Jiang**[1]*, **Dongyeop Kang**[1]
[1]University of Minnesota, Twin Cities
[2]University of California, Berkeley
{zhan7594, sheng136, jian0649, dongyeop}@umn.edu
zhexiaolin@berkeley.edu

Abstract

Autoregressive generative models play a key role in various language tasks, especially for modeling and evaluating long text sequences. While recent methods leverage stochastic representations to better capture sequence dynamics, encoding both temporal and structural dependencies and utilizing such information for evaluation remains challenging. In this work, we observe that fitting transformer-based model embeddings into a stochastic process yields ordered latent representations from originally unordered model outputs. Building on this insight and prior work, we theoretically introduce a novel likelihood-based evaluation metric BBScoreV2. Empirically, we demonstrate that the stochastic latent space induces a "clustered-to-temporal ordered" mapping of language model representations in high-dimensional space, offering both intuitive and quantitative support for the effectiveness of BBScoreV2. Furthermore, this structure aligns with intrinsic properties of natural language and enhances performance on tasks such as temporal consistency evaluation (e.g., Shuffle tasks) and AI-generated content detection.

1 Introduction

Generative models are rapidly gaining traction in NLP (Zou et al., 2023; Yang et al., 2023; Yi et al., 2024), particularly for the complex task of modeling and generating long text sequences—a challenge central to downstream applications such as text generation and machine translation. Recently, stochastic representations of latent spaces have emerged as a promising approach, showing considerable success in areas including time-series analysis (Liu et al., 2021), dynamical flow modeling (Albergo et al., 2023; Albergo and Vanden-Eijnden, 2023), and video generation (Zhang et al., 2023). In the context of text generation, Wang et al. (2022) introduced a method that models long sequences as stochastic dynamical flows, yielding strong results in producing coherent long texts. However, accurately learning the time-dependent probability density functions inherent in text data remains an open problem. Furthermore, effectively leveraging the information encoded in stochastic representations continues to be a significant challenge that has not yet been fully addressed.

Brownian bridge (BB) process helps to learn time-evolution in the stochastic representation While the temporal evolution captured in articles offers insights into linguistic properties like coherence and theme (Sheng et al., 2024), effectively encoding this temporal information into latent representations remains difficult. Drawing inspiration from the Time-control model (Wang et al., 2022) and Stochastic Interpolation (Albergo and Vanden-Eijnden, 2023; Albergo et al., 2023), we propose using the "bridge process" from stochastic process theory (Øksendal and Øksendal, 2003) to **encode and evaluate** sentence-level temporal information within latent representations. Furthermore, by leveraging the raw embeddings from frozen language models, we can also incorporate sentence-level structural information. In this work, we utilize the BB, the simplest bridge process characterized by fixed start and end points (Øksendal and Øksendal, 2003) and widely applied across various domains. We believe that more complex bridge processes, such as the Schrödinger bridge (Albergo and Vanden-Eijnden, 2023; Albergo et al., 2023), could offer richer encoding capabilities, representing a promising avenue for future research.

To evaluate such encoded time-evolution information, we introduce BBScoreV2, a novel likelihood-based evaluation metric for long-text assessment. BBScoreV2 evaluates the time evolution within a stochastic representation by considering both its temporal and structural dependencies, as detailed in Section 3.1. This metric is particularly

*Equal contribution.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 3048–3062
November 4-9, 2025

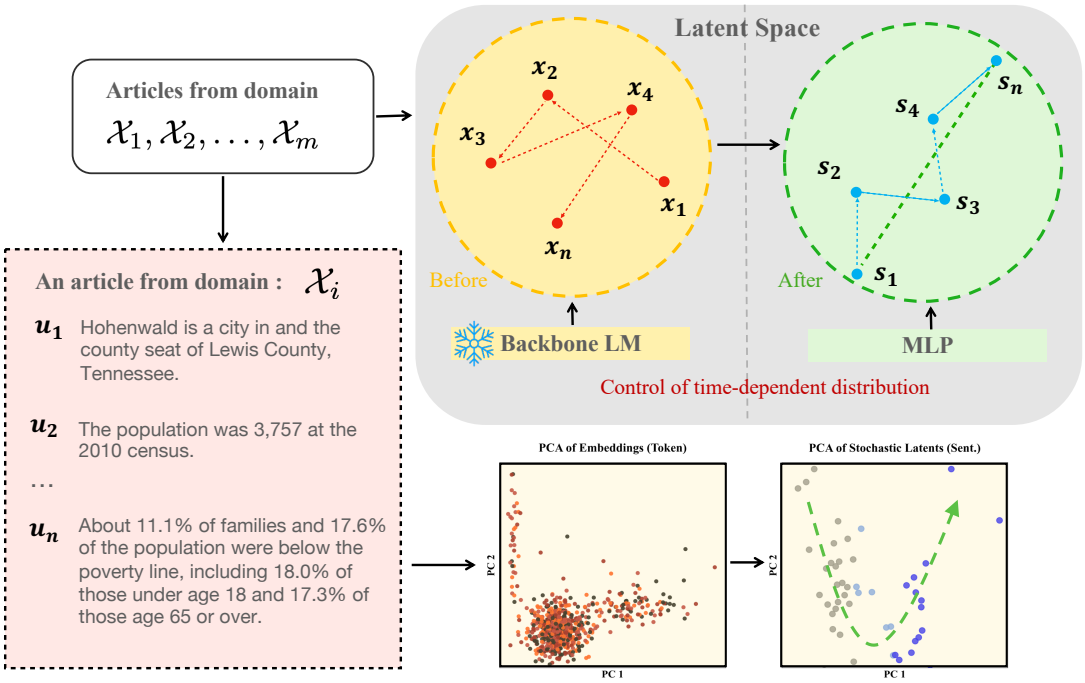

Figure 1: Schematic diagram of the Stochastic Representation in the latent space. An article from domain $\mathcal{X}_i$, segmented into sentences $(u_1, u_2, \cdots u_n)$, is processed by the encoder which consists of a pre-trained language model (LM) and a multi-layer perceptron (MLP). The encoder maps each sentence into latent space and after optimizing for the stochastic objective, the latent trajectory becomes time dependent.

useful for article coherence evaluation, exemplified by the shuffle task, which disrupts the natural temporal order. Existing methods for this task (Lai and Tetreault, 2018; Jeon and Strube, 2022) often depend heavily on the training domain, are limited by their training paradigms, can only assess pairwise data, and are restricted to articles of the same length. In contrast, BBScoreV2 assesses general temporal order, offering greater flexibility while maintaining comparable performance. To demonstrate this, we generalize the standard Shuffle test (Barzilay and Lapata, 2005; Joty et al., 2018; Moon et al., 2019) into a more robust Mixed Shuffle test. This new test compares shuffled and unshuffled versions both within and between different articles, allowing for evaluation of the metric's robustness independent of individual article characteristics like length. Furthermore, BBScoreV2 proves valuable in downstream applications such as Human-AI discrimination and exhibits strong performance in out-of-domain (O.O.D.) scenarios, likely due to its ability to capture the general structural and temporal information in human writing and preserve it in the stochastic representation.

The main contributions of our work can be summarized as follows:

- We demonstrate that clustered language model embeddings can be effectively structured into temporal ordered stochastic representations via a simple multi-layer architecture.

- We propose a novel likelihood-based metric (BBScoreV2) to evaluate temporal and structural dependencies within the stochastic representation with solid theoretical foundation.

- We hypothesized and validated that temporal and structural information encoded in the stochastic representation, as measured by the BBScoreV2, can potentially serve as an effective and flexible metric for multiple downstream tasks such as coherence evaluate and AI-generated text detection.

2 Related work

Stochastic processes have demonstrated robust capabilities in modeling complex tasks across various fields, including biology (Horne et al., 2007) and finance (Øksendal and Øksendal, 2003). Recently, the use of stochastic representations to model latent spaces has shown considerable promise in diverse applications such as time-series analysis (Liu et al., 2021) and dynamical flow modeling (Albergo et al., 2023; Albergo and Vanden-Eijnden,

2023). Notably, such methods also excel in generation tasks, including video generation (Zhang et al., 2023), and long text generation (Wang et al., 2022). A critical aspect of these tasks is to incorporate time-evolution into the latent representation, which requires capturing the time-dependent probability density functions embedded within real-world data. Generally, there are two approaches to tackle this challenge. One method is the likelihood-free training paradigm (Durkan et al., 2020), exemplified by contrastive learning techniques, which have demonstrated significant effectiveness in handling high-dimensional data (van den Oord et al., 2018; Wang et al., 2022; Zhang et al., 2023). This approach enables the learning of predictive density indirectly, rather than through direct reconstruction (Mathieu et al., 2021). The alternative method is the traditional likelihood-based approach, such as stochastic interpolants (Albergo et al., 2023; Albergo and Vanden-Eijnden, 2023), which requires the pre-definition of specific target stochastic processes. Both methods exhibit substantial potential in their respective tasks.

Coherence of articles, as defined by (Reinhart, 1980), referring to the logical flow and connection of ideas in a text, is one of the most complex temporal dynamic encoded in the articles. Studies have shown that transformers, while effective in generating tasks, often struggle with capturing coherence (Deng et al., 2022). To improve how language models learn long-text dynamics, methods using latent spaces have been developed (Bowman et al., 2016; Gao et al., 2021), focusing on sentence embeddings by considering neighboring utterances. However, these methods often produce static representations and neglect the text's dynamic nature. A recent approach using stochastic representations, such as the BB, incorporates "temporal dynamics" to improve long-range text dependencies (Wang et al., 2022). This method shows promise in generating coherent long texts through capturing structural and temporal information.

In addition to generative tasks, evaluating coherence in a given text also remains a challenge (Sheng et al., 2024; Maimon and Tsarfaty, 2023). Building on stochastic concepts, (Sheng et al., 2024) developed a heuristic metric for coherence assessment, grounded in the unsupervised learning approach proposed by Wang et al. (2022). This score demonstrated considerable performance on artificial shuffle tasks. However, their method relies on a heuristic understanding of the BB and fails to adequately establish a theoretical foundation for the metric setup, which limit the effectiveness and flexibility of their score, particularly its sensitivity to article length.

3 Method

3.1 Brownian bridge process

In this section, we introduce a stochastic representation of the encoded sequences by modeling them using BBs. We begin by defining a standard BB $\{B(t) : t \in [0, T]\}$ with $B(0) = 0$ and $B(T) = 0$. For any $t \in [0, T]$, the process $B(t)$ follows a normal distribution $B(t) \sim N(0, t(T - t)/T)$. Additionally, for $s, t \in [0, T]$ with $s < t$, the covariance between $B(s)$ and $B(t)$ is given by $\mathrm{Cov}(B(s), B(t)) = s(T - t)/T$. A more general BB start from a and end at b can then be constructed as $a + (t/T)(b - a) + \sigma B(t)$, where a and b are fixed start and end points, respectively, and σ is the standard deviation of the process.

3.2 Contrastive learning encoder

The encoder architecture consists of two components: a frozen, pre-trained language model and a trainable multilayer perceptron (MLP) network. We extract the hidden state corresponding to the end-of-sentence (EOS) token from the last layer of the language model. This hidden state serves as an input to a four-layer MLP, which is trained to map the input to the latent space. The purpose of the encoder is to learn a non-linear mapping from the raw input space to the latent space, denoted as $f_\theta : \mathcal{X} \to \mathcal{S}$. We train the encoder using contrastive learning (CL) loss (L_{CL}), which enhances its ability to differentiate between positive and negative samples, following the approach of (van den Oord et al., 2018; Wang et al., 2022).

We adopt the CL encoder framework as presented by Wang et al. (2022). In this framework, a key structural assumption is imposed on the latent space, namely an isotropic covariance structure represented by $\Sigma = \mathbf{I}_d$, where $\mathbf{I}_d$ denotes the d-dimensional identity matrix. Consequently, for an arbitrary starting point $\mathbf{s}_0$ at time $t = 0$ and an ending point $\mathbf{s}_T$ at time $t = T$, the marginal distribution of $\mathbf{s}_t$ at time t is given by Equation 1.

Consider any triplet of observations $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ with $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathcal{X}$. The goal is to ensure that $f_\theta(\mathbf{x}_2)$ follows the above marginal distribution with starting point $f_\theta(\mathbf{x}_1)$ and ending point $f_\theta(\mathbf{x}_3)$. For a sequence of observations $(\mathbf{x}_0, \ldots, \mathbf{x}_T)$, let

Marginal distribution of $\mathbf{s}_t$

$$\mathbf{s}_t \mid \mathbf{s}_0, \mathbf{s}_T \sim N\left((1-t/T)\,\mathbf{s}_0 + (t/T)\mathbf{s}_T, [t(T-t)/T]\mathbf{I}_d\right). \quad (1)$$

Encoder contrastive loss

$$L_{\mathrm{CL}} = \mathrm{E}\left[-\log \frac{\exp(d(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T; f_\theta))}{\sum_{(\mathbf{x}_0, \mathbf{x}_{t'}, \mathbf{x}_T)\in B} \exp(d(\mathbf{x}_0, \mathbf{x}_{t'}, \mathbf{x}_T; f_\theta))}\right], \quad (2)$$

$$d(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T; f_\theta) = -\frac{\|T f_\theta(\mathbf{x}_t) - (T-t) f_\theta(\mathbf{x}_0) - t f_\theta(\mathbf{x}_T)\|_2^2}{2t(T-t)}.$$

Log likelihood of Σ

$$\ell(\Sigma|\{\bar{\mathbf{s}}_i\}_{i=1}^n) = \frac{1}{2}(d\log(2\pi)\sum_{i=1}^n (T_i-1) - d\sum_{i=1}^n \log(|\Sigma_{T_i}|) \quad (3)$$

$$-\log(|\Sigma|)\sum_{i=1}^n (T_i-1) - \sum_{i=1}^n \mathrm{tr}(\Sigma^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)\Sigma_{T_i}^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top)).$$

Log density of $\bar{\mathbf{s}}$

$$\log p(\bar{\mathbf{s}}|\Sigma) = -\frac{d(T-1)}{2}\log(2\pi) - \frac{d}{2}\log(|\Sigma_T|) \quad (4)$$

$$-\frac{(T-1)}{2}\log(|\Sigma|) - \frac{1}{2}\mathrm{tr}(\Sigma^{-1}(\mathbf{s}-\boldsymbol{\mu})\Sigma_T^{-1}(\mathbf{s}-\boldsymbol{\mu})^\top).$$

BBScoreV2 of $\bar{\mathbf{s}}$

$$\mathcal{B}^+(\bar{\mathbf{s}}|\widehat{\Sigma}) = \log p(\bar{\mathbf{s}}|\Sigma)/[d(T-1)]. \quad (5)$$

Figure 2: Key Equations in the BBScoreV2 Formulation.

$B = \{(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T)\}$ be a batch consisting of randomly sampled positive triplets $(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T)$ with $0 < t < T$. Then, the CL loss function L_{CL} is defined by Equation 2.

To further investigate the structural assumption ($\Sigma = \mathbf{I}_d$) employed during encoder training, particularly given the importance of latent space correlation structure for downstream tasks, we conducted ablation studies. Specifically, we tested two different encoders: 1) CL encoder with AnInfoNCE loss: a CL loss designed by Rusak et al. (2024) to keep learning the covariance matrix Σ during training, and 2) a negative-log-likelihood based method (SP Encoder) which is purely based on fitting the temporal distribution of the bridge process.

3.3 Alignment in latent space

To evaluate trajectories within the stochastic latent space, we propose a method to approximate the inherent correlation structure and assess both spatial and temporal properties of the encoded latents. For an input sequence $\bar{\mathbf{s}} = (s_0, \ldots, s_T)$ with $s_t \in \mathbb{R}^d$ for $t = 0, 1, \ldots, T$, we capture temporal dependence using standard BBs. To account for structural dependence among the components, we consider d independent standard BBs $B_1(t), \ldots, B_d(t)$ over the interval $[0, T]$. At each time t, the sequence is modeled as $s_t = \mu_t + \mathbf{W}(B_1(t), \ldots, B_d(t))^\top$, where $\mathbf{W} \in \mathbb{R}^{d\times d}$ is a transformation matrix and $\mu_t = s_0 + (t/T)(s_T - s_0)$ represents the mean at time t. The structural dependence is captured by $\Sigma = \mathbf{W}\mathbf{W}^\top$. Let $\mathbf{s} = (s_1, \ldots, s_{T-1})$ denote the sequence excluding the start and end points, and let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{T-1})$ be the corresponding means.

The proposed BBScoreV2 is based on the likelihood function of the input sequences, with Σ being the only unknown parameter. The following proposition presents the likelihood function. For the detailed proof, please check Appendix B

Proposition 1. *Let $\Sigma_T \in \mathbb{R}^{(T-1)\times(T-1)}$ be the covariance matrix with entries $[\Sigma_T]_{s,t} = s(T-t)/T$.*

For n independent input sequences $\bar{\mathbf{s}}_1, \ldots, \bar{\mathbf{s}}_n$ with lengths $T_1+1, \ldots, T_n+1$, generated by the same $\mathbf{W}$ (or equivalently, Σ), and then the log-likelihood function is defined in Equation 3.

By Proposition 1, given the input sequences, the maximum likelihood estimate (MLE) of Σ is.

Proposition 2. *Under the setting of Proposition 1, the MLE of Σ given $\{\bar{\mathbf{s}}_i\}_{i=1}^n$ is*

$$\widehat{\Sigma} = \Big(\sum_{i=1}^n (T_i-1)\Big)^{-1}\Big(\sum_{i=1}^n (\mathbf{s}_i-\boldsymbol{\mu}_i)\Sigma_{T_i}^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top\Big).$$

The definition of the BBScoreV2 is therefore derived from the MLE of Σ. Consider the sequence $\bar{\mathbf{s}} = (s_0, \ldots, s_T)$, with $\mathbf{s}$ and $\boldsymbol{\mu}$ defined as before. To evaluate the coherence of the sequence from

a domain with unknown parameters Σ, a natural approach is to compute its density under the assumed model. If $\bar{\mathbf{s}}$ is a BB with covariance Σ, then by Proposition 1, the log density of $\bar{\mathbf{s}}$ is given by Equation 4. To remove the length sensitive term in the log density, we design a standardized score for practical purposes, and define the score as following:

Definition (BBScoreV2)**.** *Let $\widehat{\Sigma}$ be the estimate of Σ from Proposition 2. The metric BBScoreV2 is defined as*

$$\mathcal{B}^{+}(\bar{\mathbf{s}}|\widehat{\Sigma}) = \log p(\bar{\mathbf{s}}|\Sigma)/[d(T-1)].$$

Given an accurate estimate $\widehat{\Sigma}$ of the true covariance Σ, and assuming the input sequence $\bar{\mathbf{s}}$ originates from a BB process with covariance Σ, a lower BBScoreV2 value signifies a decreased likelihood of $\bar{\mathbf{s}}$ being generated under Σ. Conversely, if the representation encodes a better temporal and structural information, such as encoded from a more coherent article, the probability density will be higher, resulting in a larger BBScoreV2.

In a summary, BBScoreV2 is novel in two key aspects. First, by utilizing the temporal covariance matrix Σ_T, the BBScoreV2 captures the time-dependent structure inherent in the sequence, which is essential for accurately assessing sequence temporal property, such as coherence. Second, the inclusion of the covariance matrix Σ allows the BBScoreV2 to account for structural dependencies among the latent dimensions, providing a more comprehensive evaluation of the sequence's adherence to the assumed stochastic process.

4 Experiments and Problems

To understand the spatial and temporal information encoded in stochastic representations, we experimentally designed latent space visualization experiments. Subsequently, we evaluate BBScoreV2 to demonstrate its utility in downstream tasks that leverage this encoded information. Our experiments are designed to address the following three key research questions (Q):

- **Q1**: *How is stochastic representation learning achieved, and what makes it effective?* In Section 5.1, we analyze the spatial structure of the latent space.
- **Q2**: *Can BBScoreV2 capture correct temporal information and assess document coherence?* In Section 5.2, we examine its performance on standard shuffle tasks (indicative of temporal understanding) and also comparing the coherence of articles of varying lengths—an evaluation that current state-of-the-art methods often cannot perform effectively.
- **Q3**: *Can we use BBScoreV2 to detect AI-generated text from human-written ones ?* In Section 5.3, we explore whether BBScoreV2 can effectively distinguish between human-written text and text generated by AI. We also compare its performance with other baselines.

To validate the above question, we design the following experiments. Moreover, in Section C, we describes the dataset utilized in these experiments and how we construct the input.

Global discrimination. We employed the Shuffle Test (Barzilay and Lapata, 2005; Moon et al., 2019) to assess BBScoreV2's ability to evaluate temporal information and discriminate global coherence. It involves randomly permuting sentences within a document to create an incoherent version, which is then compared against the original. Specifically, for each article, we generated 20 unique shuffled copies by permuting entire sentence blocks of varying sizes (1, 2, 5, and 10 sentences).

Mixed Shuffled test. Building upon the standard Shuffle Test, we introduced a more challenging variant called the Mixed Shuffle Test. In this setup, BBScoreV2 of an original (unshuffled) article is compared against BBScoreV2 of shuffled articles drawn from the entire dataset, rather than solely against its own shuffled versions. A robust and general-purpose scoring mechanism should consistently identify the original, unshuffled article as more coherent in these broader comparisons.

Human-AI text discrimination. We leverage the HC3 Q&A dataset (Guo et al., 2023) to train the encoder exclusively on human-generated answers, and subsequently apply it to unseen Q&A pairs generated by both humans and ChatGPT. After deriving the stochastic representations, we compute the BBScoreV2 for each Q&A pair. We evaluate multiple encoder backbones to examine the impact of the raw embeddings. Additionally, we train an encoder on the WikiSection dataset and evaluate it using the Wikipedia subset of HC3. Experiments are conducted under both the full Q&A and answer-only settings to determine if the BBScoreV2 can ef-

fectively discriminate between ChatGPT-generated and human-written texts.

5 Results

5.1 Latent space structure analysis

Theoretically, transformer-based LLMs are argued to map articles to a latent representation that tends to form clusters. The structural properties of these clusters are believed to reflect underlying similarities and properties present in the original articles (Geshkovski et al., 2023).

Experimentally, we also find such clustered property. We first visualized the raw embeddings of each article from the frozen GPT-2 model. In Figure 3 (A.1), these embeddings are projected onto their joint first two principal components (PCs), derived using PCA computed from the latents of unshuffled articles. The color gradient, from light to dark red, represents the token's sequential position within the article, from beginning to end. Notably, as shown in (A.2) and (A.3) for shuffled versions of an article, the distinct clustering property persists. This persistence, despite the disruption of sequential order, suggests that these raw LM embeddings do not clearly and inherently encode temporal information. To further substantiate this, Figure 3 (B) plots the mean values of the first two PCs for the embeddings of each article, illustrating a tendency for articles from the same dataset to cluster together based on their raw LM embeddings.

Subsequently, to determine the information learned by the MLP layers in our CL encoder, we analyzed its outputted stochastic representations. Figure 3 (C.1) displays these MLP-processed latents projected via PCA. Here, a color gradient from light to dark blue indicates the component's position within the article's sequence (from begin to the end). This visualization reveals a clear temporal progression in the latent space for the original, unshuffled article. In stark contrast, Figures (C.2) and (C.3), which depict shuffled versions of the same article, demonstrate that the CL encoder's representations clearly reflect this violation of temporal order; the clear sequential pattern observed in (C.1) is visibly disrupted. Furthermore, Figure 3 (D) presents the projection of latent trajectories for all articles. This visualization further validates our assertion that the CL encoder effectively learns and represents temporal sequence information, unlike the raw LLM embeddings.

Based on these findings, we show that the CL encoder effectively encodes temporal information into the representation. Furthermore, by evaluating the temporal structure, we can infer properties of the original articles—such as coherence—which are quantified by BBScore$^+$ and will be systematically discussed in the following sections.

5.2 Article coherence evaluation

As shown in Tables 1, we first implement global discrimination tasks on WikiSection. In this task, BBScoreV2 significantly outperforms the BBScore and SOTA results. (See Appendix D for more details on methods we compared to.) The SOTA method, developed using a complex network structure and trained on unshuffle-shuffle data pairs, serves as a robust baseline. Our results demonstrate that BBScoreV2 surpasses the SOTA method in global discrimination tasks with larger block sizes, underscoring its potential to capture more globalized temporal properties.

In shuffle tasks, most current high-performance methods, including the SOTA approach, rely on pairwise training and are unable to effectively compare articles of different lengths, as these models are typically constructed based on sentence-wise matching and comparisons. However, in the Mixed Shuffle test which evaluate the metric robustness across different articles, as shown in Table 1, BBScoreV2 surpasses these SOTA method by generating a metric that can be compared across different articles. We use the basic entity-grid method (Barzilay and Lapata, 2005) as a baseline and the result highlights that our score enables article-wise comparison. It also demonstrates significant potential in more complex tasks. Additionally, BBScoreV2 outperforms the BBScore in this article-wise comparison, underscoring a key contribution of our design—mitigating the effect of article length on score evaluation. This property allows for a more general comparison across diverse articles.

We also explore the effect of different LLM backbones. We tested our model using LLaMA3-1B and LLaMA3-3B, with GPT2-124M which is the LLM model used in the main section. As summarized in Table 1, we find: 1) In global shuffle task, LLaMA3-3B outperforms both GPT2-124M and the SOTA method, demonstrating its effectiveness in capturing global sequence structure; 2) In Mixed Shuffle Task, LLaMA3-3B surpasses GPT2-124M for smaller blocks (b=1), but its performance decreases for larger blocks (b=2, b=5, b=10). This suggests a trade-off where larger models excel at

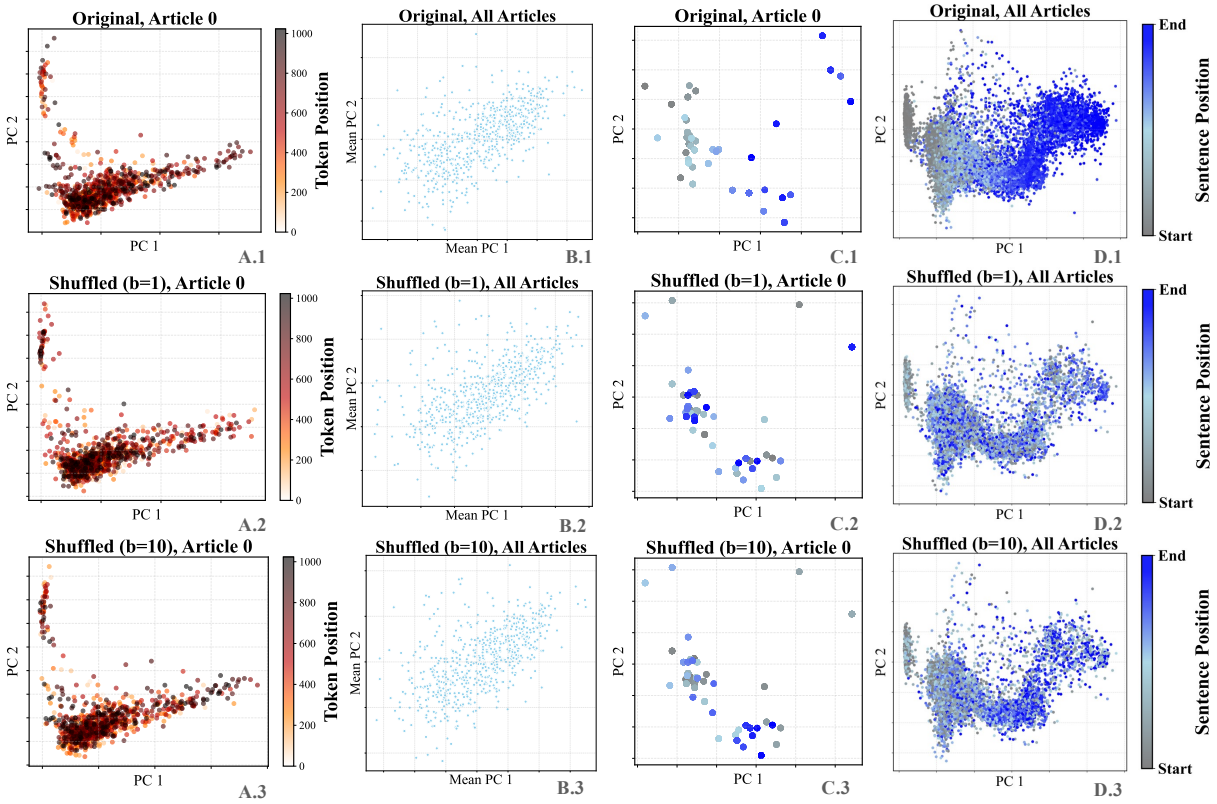

Figure 3: PCA analysis of raw LM embeddings and CL encoder representations. (A.1) Projection of raw LM embeddings for an unshuffled article onto the first two PCs. The color gradient, from light to dark red, indicates the sequential position of each token within the article. (A.2, A.3) raw LM embeddings for shuffled versions of the same article. (B) Mean PC1 and PC2 values for all articles are plotted, with each article represented by a dot. (C.1) Latent representations from the CL encoder for an unshuffled article, where the color gradient (light to dark blue) signifies the component's position in the article sequence. (C.2, C.3) Latent patterns observed in shuffled versions of the same article. (D) Visualization of the latent trajectories for all articles.

capturing local details (b=1) but might sacrifice robustness for global structures (b=10). This insight highlights an intriguing direction for future exploration — different LMs may facilitate learning stochastic representations in task-specific ways.

Moreover, we evaluate the robustness of the encoded stochastic representation on a broader dataset. As shown in Table 2, we train the encoder on WikiText and evaluate it on WikiSection (see Appendix C for details and comparisons about the datasets). The results indicate that our method remains highly robust in this O.O.D. setting, suggesting that the structural and temporal information captured by our model reflects fundamental patterns that generalize across different datasets.

5.3 Human-AI discrimination tasks

In this task, we hypothesize that human writing, compared to AI-generated text, displays temporal dynamics and structural patterns similar to those observed in other human-written articles. Specifically, we propose that an encoder trained on a human-written dataset will more accurately capture the characteristics of human writing than those of AI-generated text, resulting in a higher likelihood for human-authored content. As shown in Figure 4, BBScoreV2 consistently outperforms BBScore across all experimental settings. Notably, GPT2 (124M) surpasses the larger backbone models, suggesting that the quality of the learned stochastic representation does not necessarily improve with increased model size.Instead, it is the MLP module that plays the central role in shaping the stochastic representation. Among the models evaluated, GPT2 features a smaller hidden dimension of 768, whereas both LLaMA3 and Qwen3 utilize larger hidden dimensions of 2048. It implies that the hidden dimension of the backbone model may influence performance on this task.

Next, we use a WikiSection trained encoder to detect ChatGPT-generated answer in the Wikipedia subset of HC3. The results are shown in Table 3

Methods	Acc. (Shuffle Task)				Acc. (Mixed Shuffle Task)			
	$\mathcal{D}_{b=1}$	$\mathcal{D}_{b=2}$	$\mathcal{D}_{b=5}$	$\mathcal{D}_{b=10}$	$\mathcal{D}_{b=1}$	$\mathcal{D}_{b=2}$	$\mathcal{D}_{b=5}$	$\mathcal{D}_{b=10}$
ENTITY GRID (Barzilay and Lapata, 2005)	85.73	82.79	75.81	64.65	46.10	52.29	53.69	63.02
UNIFIED COHERENCE (Moon et al., 2019)	**99.73**	97.86	96.90	96.09	–	–	–	–
BBSCORE (Sheng et al., 2024)	83.39	80.71	79.36	78.66	22.37	24.94	23.84	19.69
BBSCOREV2 (GPT2-124M)	99.03	98.11	98.02	98.17	94.78	**89.24**	**79.64**	70.83
BBSCOREV2 (LLAMA3-1B)	99.16	98.37	97.99	97.87	94.53	87.86	76.95	**71.13**
BBSCOREV2 (LLAMA3-3B)	99.57	**98.74**	**98.14**	**98.74**	**94.97**	86.34	73.88	68.87

Table 1: Results of Global shuffle tasks on WikiSection. $\mathcal{D}_{b=i}$, $i = 1, 2, 5, 10$ refers to datasets constructed with varying levels of block shuffling.

Methods	Shuffle Test tasks (O.O.D.)			
	$\mathcal{D}_{b=1}$	$\mathcal{D}_{b=2}$	$\mathcal{D}_{b=5}$	$\mathcal{D}_{b=10}$
UNIFIED COHERENCE	60.02	9.63	44.80	66.51
BBSCORE	70.32	72.09	76.84	77.73
BBSCOREV2	**91.30**	**87.22**	**86.14**	**88.18**

Table 2: O.O.D. Task. Encoder was trained on the WikiText and evaluated on Shuffle Test tasks using the same WikiSection data to assess their performance.

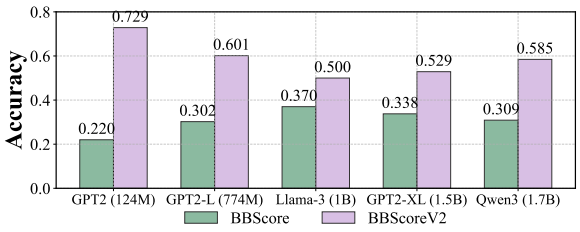

Figure 4: Compare different LM backbones.

under both Q&A and answer-only settings. To further highlight the flexibility and competitive performance of BBScoreV2 compared to LLM-based models, we assessed the perturbation discrepancy metric proposed in DetectGPT (Mitchell et al., 2023), which has high performance in AI detection tasks. Our results reveal that BBScoreV2 surpasses DetectGPT when using a comparable number of model inferences. DetectGPT's performance is influenced by a hyperparameter—the number of perturbations—which directly affects both the number of model inferences and the computational complexity. As shown in Table 7 in the Appendix, we tested cases with 1 and 10 perturbations. With 1 perturbation, DetectGPT's accuracy was approximately 64%, lower than BBScoreV2's 70%, while requiring twice the number of model inferences per text. With 10 perturbations, DetectGPT's accuracy increased to 84%, but this required 11 model inferences per text, making it significantly more computationally intensive than BBScoreV2.

Methods	HC3 (w/o Q&A)	HC3 (w/ Q&A)
BBSCORE	37.53	31.47
DETECTGPT	64.30	63.30
BBSCOREV2	**70.67**	**69.71**

Table 3: Accuracy of the Human-AI discrimination task.

5.4 Ablation analysis on CL encoder

As previously discussed, the CL encoder relies on a critical assumption of the independence and homogeneity among the dimensions of the encoded sequence which is $\Sigma = \mathbf{I}_d$. To further examine this assumption, we employ two alternative methods:

1) A likelihood-based encoder, **SP Encoder** (see Appendix A) whose loss function is defined based on the likelihood of the Brownian bridge:

$$L_{\text{NLL}} = \sum_{j=1}^{m}\sum_{i=1}^{n_j}(T_i - 1)\log(|\Sigma_j|) + \sum_{j=1}^{m}\sum_{i=1}^{n_j}\text{tr}(\Sigma_j^{-1}(\mathbf{s}_i^\theta - \boldsymbol{\mu}_i^\theta)\Sigma_{T_i}^{-1}(\mathbf{s}_i^\theta - \boldsymbol{\mu}_i^\theta)^\top). \tag{6}$$

2) A contrastive loss-based encoder, whose loss function is **AnInfoNCE** Rusak et al. (2024) which is capable of learning Σ during training The loss function $L_{\text{AnInfoNCE}}$ is defined with the same formate as CL loss (2), however, we replace the metric $d(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T)$ by a trainable metric $d^*(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T; f_\theta)$ defined as:

$$d^*(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T; f_\theta) = -\frac{\|f_\theta(\mathbf{x}_t) - \frac{T-t}{T}f_\theta(\mathbf{x}_0) - \frac{t}{T}f_\theta(\mathbf{x}_T)\|^2_{\hat{\Lambda}}}{2t(T-t)/T}.$$

and $\hat{\Lambda}$ is a trainable diagonal scaling matrix. Let

$\mathbf{v} = f_\theta(\mathbf{x}_t) - \frac{T-t}{T} f_\theta(\mathbf{x}_0) - \frac{t}{T} f_\theta(\mathbf{x}_T)$, then its corresponding norm is defined as:

$$\|\mathbf{v}\|^2_{\hat{\Lambda}} = \mathbf{v}^{\mathrm{T}} \cdot \hat{\Lambda} \cdot \mathbf{v}$$

As shown in Table 4, neither the likelihood-based nor the CL encoder with AnInfoNCE loss yields significant improvements in the shuffle test. This suggests that the MLP layers do not capture meaningful structural correlations across latent dimensions—a phenomenon also noted by Wang et al. (2022)—and instead primarily reconstruct temporal information, which validate our assumptions on CL encoder training. The lack of performance improvement may also suggest that the pertinent correlation structure is likely inherent either within the statistical properties of the article domain or already captured within the high-dimensional embedding space of the pre-trained language model as we seen in the cluster analysis in Fig 3.

Loss Type	$\mathcal{D}_{b=1}$	$\mathcal{D}_{b=2}$	$\mathcal{D}_{b=5}$	$\mathcal{D}_{b=10}$
AnInfoNCE	94.63	91.05	92.13	91.70
Likelihood	94.42	92.90	90.77	86.69
Ours	99.03	98.11	98.02	98.17

Table 4: Comparison of model performance with different loss function on WikiSection Dataset.

5.5 Computation efficiency analysis

We specifically analyze the computation efficiency of BBScoreV2, as shown in Figure 5, the y-axis represents computation time, while the x-axis indicates article length. The theoretical computational complexity of BBScoreV2 is $O(T^2)$, primarily due to matrix multiplications inherent in its definition. This complexity is fundamental to fully leveraging temporal information for sequence evaluation. Empirically, the observed computation time is slightly better than the theoretical prediction, thanks to the computational acceleration. These results demonstrate that BBScoreV2 is not only feasible for real-time applications but also retains its robust evaluation capabilities.

6 Conclusion

In this paper, we present both a theoretical and empirical investigation into the structural and temporal properties encoded in stochastic representations of latent trajectories for NLP tasks. We analyze and visualize these properties, and introduce BBScoreV2—a novel, length-invariant metric

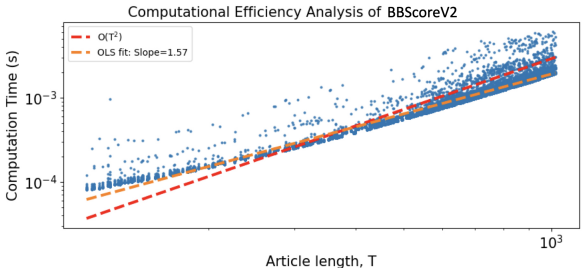

Figure 5: The computation time of BBScoreV2 for different article lengths. It reveals a quadratic relationship (experimentally 1.57, theoretically 2) between article length and computation time, with each article processed in approximately $\sim 10^{-3}$ seconds.

designed to quantify such information. First we present the learned representations recovers the time dependency of the input sequences. Then validated through shuffled and mixed-shuffle tests, we show that BBScoreV2 exhibits strong performance in capturing temporal structure and generalizes effectively to out-of-distribution tasks, suggesting that these properties reflect domain-independent textual signals. Moreover, BBScoreV2 shows promising capability in distinguishing human-written from AI-generated text by leveraging encoded structural and temporal features.

Looking ahead, we aim to extend BBScoreV2 to multi-domain tasks such as domain identification, and to exploit its length-insensitive nature to develop generative models that maintain semantic coherence across varying sequence lengths. Its computational efficiency (see Fig. 5) also makes it suitable for large-scale applications. Finally, inspired by Albergo et al. (2023); Albergo and Vanden-Eijnden (2023), we plan to explore more expressive bridge processes to further enhance the representational capacity of the latent space and enable richer downstream analysis and generation.

7 Limitations

Our current study is constrained by limited computational resources and the lack of human-annotated data, which prevents us from evaluating BBScoreV2 against human preference—a key limitation in assessing its alignment with human judgment. Additionally, in the Human-AI discrimination task, we were unable to evaluate it on a broader range of datasets or conduct more extensive comparisons across more baselines. These limitations suggest directions for future work involving large-scale human evaluation and broader benchmarking.

References

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. 2023. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*.

Michael Samuel Albergo and Eric Vanden-Eijnden. 2023. Building Normalizing Flows with Stochastic Interpolants. In *The Eleventh International Conference on Learning Representations*.

Sebastian Arnold, Benjamin Schrauwen, Verena Rieser, and Katja Filippova. 2019. SECTOR: A Neural Model for Coherent Topic Segmentation and Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 241–253, Hong Kong, China. Association for Computational Linguistics.

Regina Barzilay and Mirella Lapata. 2005. Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148, Ann Arbor, Michigan. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Yuntian Deng, Volodymyr Kuleshov, and Alexander Rush. 2022. Model Criticism for Long-Form Text Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11887–11912, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Conor Durkan, Iain Murray, and George Papamakarios. 2020. On Contrastive Learning for Likelihood-free Inference. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2771–2781. PMLR.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. 2023. The emergence of clusters in self-attention dynamics. In *Advances in Neural Information Processing Systems*, volume 36, pages 57026–57037. Curran Associates, Inc.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Jon S Horne, Edward O Garton, Stephen M Krone, and Jesse S Lewis. 2007. Analyzing animal movements using Brownian bridges. *Ecology*, 88(9):2354–2363.

Sungho Jeon and Michael Strube. 2022. Entity-based Neural Local Coherence Modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7787–7805, Dublin, Ireland. Association for Computational Linguistics.

Shafiq Joty, Muhammad Tasnim Mohiuddin, and Dat Tien Nguyen. 2018. Coherence Modeling of Asynchronous Conversations: A Neural Entity Grid Approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 558–568, Melbourne, Australia. Association for Computational Linguistics.

Alice Lai and Joel Tetreault. 2018. Discourse Coherence in the Wild: A Dataset, Evaluation and Methods. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 214–223, Melbourne, Australia. Association for Computational Linguistics.

Bingbin Liu, Pradeep Ravikumar, and Andrej Risteski. 2021. Contrastive learning of strong-mixing continuous-time stochastic processes. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3151–3159. PMLR.

Aviya Maimon and Reut Tsarfaty. 2023. A Novel Computational and Modeling Foundation for Automatic Coherence Assessment. *arXiv e-prints*, arXiv:2310.00598.

Emile Mathieu, Adam Foster, and Yee Teh. 2021. On Contrastive Representations of Stochastic Processes. In *Advances in Neural Information Processing Systems*, volume 34, pages 28823–28835. Curran Associates, Inc.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer Sentinel Mixture Models. *Preprint*, arXiv:1609.07843.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Han Cheol Moon, Tasnim Mohiuddin, Shafiq Joty, and Chi Xu. 2019. A Unified Neural Coherence Model. 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)*, pages 2262–2272, Hong Kong, China. Association for Computational Linguistics.

Bernt Øksendal and Bernt Øksendal. 2003. *Stochastic differential equations*. Springer.

Tanya Reinhart. 1980. Conditions for Text Coherence. *Poetics Today*, 1(4):161–180.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. 2024. Infonce: Identifying the gap between theory and practice. *Preprint*, arXiv:2407.00143.

Zhecheng Sheng, Tianhao Zhang, Chen Jiang, and Dongyeop Kang. 2024. BBScore: A Brownian Bridge Based Metric for Assessing Text Coherence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14937–14945.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR*, abs/1807.03748.

Rose E Wang, Esin Durmus, Noah Goodman, and Tatsunori Hashimoto. 2022. Language modeling via stochastic processes. In *International Conference on Learning Representations*.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Comput. Surv.*, 56(4).

Qinghua Yi, Xiaoyu Chen, Chen Zhang, Zhen Zhou, Ling Zhu, and Xin Kong. 2024. Diffusion models in text generation: a survey. *PeerJ Computer Science*, 10:e1905.

Heng Zhang, Daqing Liu, Qi Zheng, and Bing Su. 2023. Modeling Video As Stochastic Processes for Fine-Grained Video Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2225–2234.

Hao Zou, Zae Myung Kim, and Dongyeop Kang. 2023. A Survey of Diffusion Models in Natural Language Processing. *Preprint*, arXiv:2305.14671.

A Appendix: SP Encoder

A.1 Definition

Consider a multi-domain problem with m domains $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_m$ each associated with domain-specific true structural parameters $\Sigma_1, \Sigma_2, \ldots, \Sigma_m$, respectively. For each domain $\mathcal{X}_j$, we have n_j independent raw inputs $\mathbf{x}_{j1}, \ldots, \mathbf{x}_{jn_j}$. We define the encoded sequences as $\bar{\mathbf{s}}_{ji}^{\theta} = f_\theta(\mathbf{x}_{ji})$ for $j = 1, \ldots, m$ and $i = 1, \ldots, n_j$, where f_θ is the encoder parameterized by θ. When the encoder parameters reach their optimal values θ^*, the sequences $[\bar{\mathbf{s}}_{ji}^{\theta^*}]_{i=1}^{n_j}$ are expected to be i.i.d. samples from BBs with parameters Σ_j for each domain $\mathcal{X}_j$.

We employ the negative log-likelihood (NLL) as the loss function to train the encoder. According to Proposition 1, for each θ, the negative log-likelihood for domain $\mathcal{X}_j$ depends on Σ_j and the inputs $[\mathbf{x}_{ji}]_{i=1}^{n_j}$ through the expression $\sum_{i=1}^{n_j}(T_i - 1)\log(|\Sigma_j|) + \sum_{i=1}^{n_j}\mathrm{tr}(\Sigma_j^{-1}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})\Sigma_{T_i}^{-1}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})^\top)$. We consider the following training process.

Batch Processing: We divide the inputs $[\mathbf{x}_{ji}]_{i=1}^{n_j}$ into several batches. For each batch $\mathcal{B}$, we compute the batch loss using the current estimate $\widehat{\Sigma}_j$ of Σ_j: $\sum_{i\in\mathcal{B}}\mathrm{tr}(\widehat{\Sigma}_j^{-1}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})\Sigma_{T_i}^{-1}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})^\top)$. This loss function measures how well the encoded sequences fit the assumed BB model with the current structural parameter estimate.

Handling Large Sequences: When the sequence lengths T_i are large, computing the full loss can be computationally intensive. To address this, we randomly sample a triplet of time points $t = (t_1, t_2, t_3)$ with $1 \le t_1 < t_2 < t_3 \le T_i - 1$. We extract the corresponding sub-matrices $[\mathbf{s}_i^{\theta}]_t$ and $[\boldsymbol{\mu}_i^{\theta}]_t$ of size $d \times 3$ from $\mathbf{s}_i^{\theta}$ and $\boldsymbol{\mu}_i^{\theta}$, respectively. Let $[\Sigma_{T_i}]_t$ be the 3×3 sub-matrix of Σ_{T_i} corresponding to the selected time points. The loss for each i in the batch becomes $\mathrm{tr}(\widehat{\Sigma}_j^{-1}([\mathbf{s}_i^{\theta}]_t - [\boldsymbol{\mu}_i^{\theta}]_t)[\Sigma_{T_i}]_t^{-1}([\mathbf{s}_i^{\theta}]_t - [\boldsymbol{\mu}_i^{\theta}]_t)^\top)$. This approach reduces computational complexity while still capturing temporal dependencies at selected time points.

Updating Structural Parameters: After processing all batches for $\mathcal{X}_j$, we update the estimate of Σ_j using the MLE: $\widehat{\Sigma}_j = [\sum_{i=1}^{n_j}(T_i - 1)]^{-1}[\sum_{i=1}^{n_j}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})\Sigma_{T_i}^{-1}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})^\top]$. This update aggregates information from all sequences in the domain to refine the structural parameter estimate.

Regularization for Stability: To stabilize the training process, we regularize $\widehat{\Sigma}_j$ by blending it with a scaled identity matrix. We compute the average variance $\widehat{\sigma}_j^2$ and update $\widehat{\Sigma}_j$ as follows, using a small regularization parameter $\epsilon > 0$: $\widehat{\Sigma}_j = (1 - \epsilon)[\sum_{i=1}^{n_j}(T_i - 1)]^{-1}[\sum_{i=1}^{n_j}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})\Sigma_{T_i}^{-1}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})^\top] + \epsilon\widehat{\sigma}_j^2\mathbf{I}_d$ with $\widehat{\sigma}_j^2 = [\sum_{i=1}^{n_j}(T_i - 1)d]^{-1}[\sum_{i=1}^{n_j}\mathrm{tr}((\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})\Sigma_{T_i}^{-1}(\mathbf{s}_i^{\theta} - \boldsymbol{\mu}_i^{\theta})^\top)]$. This regularization shifts $\widehat{\Sigma}_j$ slightly towards isotropy, improving numerical stability during optimization.

Total Empirical Loss Function: After iterating over all domains, the total empirical loss function

becomes

$$L_{\text{NLL}} = \sum_{j=1}^{m}\sum_{i=1}^{n_j}(T_i-1)\log(|\Sigma_j|) + \sum_{j=1}^{m}\sum_{i=1}^{n_j}\text{tr}\Big(\Sigma_j^{-1}(\mathbf{s}_i^{\theta}-\boldsymbol{\mu}_i^{\theta}) \cdot \Sigma_{T_i}^{-1}(\mathbf{s}_i^{\theta}-\boldsymbol{\mu}_i^{\theta})^{\top}\Big).$$

Minimizing this loss over θ encourages the encoder to produce sequences that align with the assumed stochastic process model across all domains.

A.2 Training Details

The WikiSection SP Encoder was trained on 1 A100 GPU for about 10 hours using the training set of WikiSection for 100 epochs. We used SGD optimizer and set the learning rate to be *1e-9*. The ϵ in the loss function L_{NLL} is chosen as *1e-7*. The WikiText SP Encoder was trained on 4 A100 GPUs for roughly 20 hours for 4 epochs with WikiText dataset. For this dataset, we trained with AdamW optimizer with learning rate *1e-9* and batch size *32*. The ϵ in the loss function L_{NLL} is chosen as *1e-3*. Other hyperparameters can be accessed from the configuration file in the submitted code. Our empirical results show incorporating $\hat{\sigma}_j$ into the $\widehat{\Sigma}_j$ makes no significant results in the downstream tasks, thus we disregard $\hat{\sigma}_j$ during encoder training.

A.3 Hyper-parameter Tuning

While training the SP Encoder, we experimented with different ϵ in L_{NLL} to see its impact on the performance of the trained encoder. Note that ϵ determines the perturbation added to the matrix $\widehat{\Sigma}$. The eigenvalues of the initial $\widehat{\Sigma}$ range from 10^{-6} to 10^{-1}, with the majority of which lying in $[10^{-3}, 10^{-5}]$. Thus we tested the following three different ϵ:

- Large $\epsilon = 10^{-3}$ that is larger that most eigenvalues of $\widehat{\Sigma}$.
- Medium $\epsilon = 10^{-5}$ that is about the same scale of most eigenvalues of $\widehat{\Sigma}$.
- Small $\epsilon = 10^{-7}$ that is smaller than most eigenvalues of $\widehat{\Sigma}$.

We choose the small ϵ based on the performance.

B Proof

B.1 Proof of Proposition 1

Proof. We fix the start and end points s_0 and s_T and calculate the likelihood function of the input sequence $\mathbf{s}$.

Given that $s_t - \mu_t = \mathbf{W}(B_1(t), \ldots, B_d(t))^{\top}$, and considering the independence of $B_1(t), \ldots, B_d(t)$ along with the properties of the standard BB, we have for any $t, t' \in \{1, 2, \ldots, T-1\}$: $\text{E}[s_t - \mu_t] = 0$, $\text{Var}[s_t] = [\Sigma_T]_{t,t}\Sigma$ and $\text{Cov}[s_t, s_{t'}] = [\Sigma_T]_{t,t'}\Sigma$. Therefore, the vectorized form of $\mathbf{s} - \boldsymbol{\mu}$ follows a multivariate normal distribution:

$$\text{vec}(\mathbf{s} - \boldsymbol{\mu}) \sim N(0, \Sigma_T \otimes \Sigma),$$

where $\text{vec}(\cdot)$ denotes vectorization and $\otimes$ represents the Kronecker product.

Using the likelihood function of the multivariate normal distribution, we have:

$$L(\Sigma|\bar{\mathbf{s}}) = (2\pi)^{-d(T-1)/2}|\Sigma_T \otimes \Sigma|^{-1/2} \cdot \exp\Big[-\tfrac{1}{2}\text{vec}(\mathbf{s}-\boldsymbol{\mu})^{\top}[\Sigma_T \otimes \Sigma]^{-1}\text{vec}(\mathbf{s}-\boldsymbol{\mu})\Big]$$

Using properties of the Kronecker product, we have $|\Sigma_T \otimes \Sigma| = |\Sigma_T|^d|\Sigma|^{T-1}$ and then

$$\begin{aligned}
&\text{vec}(\mathbf{s}-\boldsymbol{\mu})^{\top}[\Sigma_T \otimes \Sigma]^{-1}\text{vec}(\mathbf{s}-\boldsymbol{\mu}) \\
&= \text{vec}(\mathbf{s}-\boldsymbol{\mu})^{\top}[\Sigma_T^{-1} \otimes \Sigma^{-1}]\text{vec}(\mathbf{s}-\boldsymbol{\mu}) \\
&= \text{vec}(\mathbf{s}-\boldsymbol{\mu})^{\top}\text{vec}(\Sigma^{-1}(\mathbf{s}-\boldsymbol{\mu})\Sigma_T^{-1}) \\
&= \text{tr}((\mathbf{s}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\mathbf{s}-\boldsymbol{\mu})\Sigma_T^{-1}) \\
&= \text{tr}(\Sigma^{-1}(\mathbf{s}-\boldsymbol{\mu})\Sigma_T^{-1}(\mathbf{s}-\boldsymbol{\mu})^{\top}).
\end{aligned}$$

Therefore, the likelihood function becomes:

$$L(\Sigma|\bar{\mathbf{s}}) = (2\pi)^{-d(T-1)/2}|\Sigma_T|^{-d/2}|\Sigma|^{-(T-1)/2} \cdot \exp[-\text{tr}(\Sigma^{-1}(\mathbf{s}-\boldsymbol{\mu})\Sigma_T^{-1}(\mathbf{s}-\boldsymbol{\mu})^{\top})/2].$$

Taking the logarithm, the log-likelihood function is:

$$\ell(\Sigma|\bar{\mathbf{s}}) = -\tfrac{d(T-1)}{2}\log(2\pi) - \tfrac{d}{2}\log|\Sigma_T| - \tfrac{(T-1)}{2}\log|\Sigma| - \tfrac{1}{2}\text{tr}\Big(\Sigma^{-1}(\mathbf{s}-\boldsymbol{\mu})\Sigma_T^{-1}(\mathbf{s}-\boldsymbol{\mu})^{\top}\Big).$$

For n independent input sequences $\bar{\mathbf{s}}_1, \ldots, \bar{\mathbf{s}}_n$ with lengths $T_1 + 1, \ldots, T_n + 1$, generated by the same Σ, the total likelihood is:

$$L(\Sigma|\{\mathbf{s}_i\}_{i=1}^n) = \Pi_{i=1}^n L(\Sigma|\mathbf{s}_i).$$

Then the total log-likelihood function is

$$\begin{aligned}\ell(\Sigma|\{\mathbf{s}_i\}_{i=1}^n) &= \sum_{i=1}^n \ell(\Sigma|\mathbf{s}_i) \\ &= -\frac{d\sum_{i=1}^n (T_i-1)}{2}\log(2\pi) \\ &\quad -\frac{d}{2}\sum_{i=1}^n \log(|\Sigma_{T_i}|) \\ &\quad -\frac{\sum_{i=1}^n (T_i-1)}{2}\log(|\Sigma|) \\ &\quad -\frac{1}{2}\sum_{i=1}^n \mathrm{tr}\Big(\Sigma^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i) \\ &\qquad \cdot \Sigma_{T_i}^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top\Big).\end{aligned}$$

□

B.2 Proof of Proposition 2

Proof. To find the MLE of Σ, we need to minimize the negative log-likelihood function, which is equivalent to minimizing:

$$\begin{aligned}g(\Sigma) &= \sum_{i=1}^n (T_i-1)\log|\Sigma| \\ &\quad + \sum_{i=1}^n \mathrm{tr}\Big(\Sigma^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)\Sigma_{T_i}^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top\Big)\end{aligned}$$

Since $\Sigma = \mathbf{W}\mathbf{W}^\top$ is positive definite, we can compute the gradient of $g(\Sigma)$ with respect to Σ. Note that:

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\Sigma}\log|\Sigma| = \Sigma^{-1}, \\ &\frac{\mathrm{d}}{\mathrm{d}\Sigma}\mathrm{tr}\Big(\Sigma^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)\cdot\Sigma_{T_i}^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top\Big) \\ &= -\Sigma^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)\Sigma_{T_i}^{-1}\cdot(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top\Sigma^{-1}.\end{aligned}$$

We compute the gradient:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\Sigma}g(\Sigma) &= \Big(\sum_{i=1}^n (T_i-1)\Big)\Sigma^{-1} \\ &\quad -\Sigma^{-1}\Big(\sum_{i=1}^n (\mathbf{s}_i-\boldsymbol{\mu}_i)\Sigma_{T_i}^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top\Big) \\ &\quad \cdot\Sigma^{-1}.\end{aligned}$$

Setting the gradient to zero for minimization, we have:

$$\begin{aligned}\widehat{\Sigma} &= \Big(\sum_{i=1}^n (T_i-1)\Big)^{-1} \\ &\quad \cdot\Big(\sum_{i=1}^n (\mathbf{s}_i-\boldsymbol{\mu}_i)\Sigma_{T_i}^{-1}(\mathbf{s}_i-\boldsymbol{\mu}_i)^\top\Big)\end{aligned}$$

As shown, the MLE estimate for Σ is obtained. □

C Datasets

WikiSection: We use dataset introduced in (Arnold et al., 2019) which contains selected Wikipedia articles on the topic of global cities and have clear topic structures. Each article in this collection follows a pattern certain sections such as abstract, history, geographics and demographics. The training split contains 2165 articles and the test split has 658 articles.

HC3: The Human ChatGPT Comparison Corpus (HC3) (Guo et al., 2023) includes comparative responses from human experts and ChatGPT, covering questions from various fields such as open-domain, finance, medicine, law, psychology and Wikipedia. We construct the input by concatenating the *Question* and *Answers* together as a single document and label whether it is ChatGPT generated by the source of the answers. We also use the data without Q&A settings and only treat the answer part as a single document.

WikiText: WikiText language modeling dataset (Merity et al., 2016) is a much larger set of verified good and featured articles extracted from Wikipedia compared to WikiSection,we further compare these two dataset (Section C) and show that there is only $\sim 1\%$ potential overlap in topics. We used *WikiText-103-v1* collection in specific for experiments. This dataset encompass over 100 million tokens from 29,061 full articles. The dataset is assessible through Huggingface [1].

Difference between WikiSection and WikiText The **WikiSection** dataset comprises 2,165 articles describing cities from Wikipedia, while **WikiText** includes 29,061 featured or high-quality articles covering a broader range of topics. The **WikiSection** dataset is most similar to the "places" category in **WikiText**, which contains approximately 500 articles. To ensure dataset exclusivity, we used string match to check the overlapping. The regular expression query we used is `'(a|the) ([\w\s]*)?(city|town) in'` as it is contained in 1,721 articles out of 2,165 in WikiSection dataset. Using the same query, we examined the WikiText dataset and checked the intersection of first word of the article from both search result. After manually getting rid of false positives, there are around 30 documents found overlap in both datasets. We

[1] https://huggingface.co/datasets/EleutherAI/WikiText_document_level

argue that with that amount of ($\sim$0.1%) contamination, **WikiSection** can be considered out of domain of **WikiText**.

D Other scores used in this paper

Entity Grid Barzilay and Lapata (2005) is the most recognized entity-based approach. It creates a two-way contingency table for each input document to track the appearance of entities in each sentence. We use Stanford's CoreNLP to annotate the documents and the implementation provided in the Coheoka library[2] to obtain the Entity Grid score.

Unified Coherence Moon et al. (2019) presents a neural-based entity-grid method that integrates sentence grammar, inter-sentence coherence relations, and global coherence patterns, achieving state-of-the-art results in artificial tasks.

BBScore Sheng et al. (2024) introduces BBScore, and also check the main text for a comprehensive comparison between BBScore and BBScoreV2.

E Human-AI comparison test

Table 5 presents the performance of BBScoreV2 computed with different $\widehat{\Sigma} \in \mathbb{R}^d$, while Table 6 shows the performance of the BBScore with various $\widehat{\sigma} \in \mathbb{R}$, where the subscript indicates the dataset used for approximation.

The clear improvement over the BBScore demonstrates that accurately capturing structural and temporal information can significantly enhance the model's accuracy. Table 7 display the performance of DetectGPT with more inferences which significantly improves its performance while also takes much longer time to infer.

[2] https://github.com/kigawas/coheoka

	Human AI comparison			Human AI comparison with Q&A		
	Human ($\widehat{\Sigma}_{human}$)	Human ($\widehat{\Sigma}_{ai}$)	Human ($\widehat{\Sigma}_{wiki}$)	Human ($\widehat{\Sigma}_{human}$)	Human ($\widehat{\Sigma}_{ai}$)	Human ($\widehat{\Sigma}_{wiki}$)
AI ($\widehat{\Sigma}_{human}$)	70.07	70.55	-	69.00	69.60	-
AI ($\widehat{\Sigma}_{ai}$)	59.98	61.52	-	58.19	59.74	-
AI ($\widehat{\Sigma}_{wiki}$)	-	-	**70.67**	-	-	**69.71**

Table 5: Combined accuracy of human AI comparison and human AI comparison with Q&A

	Human AI comparison			Human AI comparison with Q&A		
	Human ($\widehat{\sigma}_{human}$)	Human ($\widehat{\sigma}_{ai}$)	Human ($\widehat{\sigma}_{wiki}$)	Human ($\widehat{\sigma}_{human}$)	Human ($\widehat{\sigma}_{ai}$)	Human ($\widehat{\sigma}_{wiki}$)
AI ($\widehat{\sigma}_{human}$)	35.99	**45.13**	-	35.04	**38.84**	-
AI ($\widehat{\sigma}_{ai}$)	26.37	37.05	-	33.73	38.12	-
AI ($\widehat{\sigma}_{wiki}$)	-	-	37.53	-	-	31.47

Table 6: Human-AI Task Results with BBScore (Sheng et al., 2024).

	Human AI comparison		Human AI comparison with Q&A	
Number of Perturbations	1	10	1	10
Number of LLM Inferences	Number of Perturbations + 1			
Accuracy	64.30	84.89	63.30	83.13

Table 7: Human-AI Task Results with DetectGPT (Mitchell et al., 2023). As a comparison, BBScoreV2 only requires one LLM inference.