

# BBScore: A Brownian Bridge Based Metric for Assessing Text Coherence

Zhecheng Sheng<sup>\*</sup>, Tianhao Zhang<sup>\*</sup>, Chen Jiang<sup>\*</sup>, Dongyeop Kang

University of Minnesota, Twin Cities  
Minneapolis, Minnesota 55455  
{sheng136, zhan7594, jian0649, dongyeop}@umn.edu

## Abstract

Measuring the coherence of text is a vital aspect of evaluating the quality of written content. Recent advancements in neural coherence modeling have demonstrated their efficacy in capturing entity coreference and discourse relations, thereby enhancing coherence evaluation. However, many existing methods heavily depend on static embeddings or focus narrowly on nearby context, constraining their capacity to measure the overarching coherence of long texts. In this paper, we posit that coherent texts inherently manifest a sequential and cohesive interplay among sentences, effectively conveying the central theme, purpose, or standpoint. To explore this abstract relationship, we introduce the "BBScore," a novel reference-free metric grounded in Brownian bridge theory for assessing text coherence. Our findings showcase that when synergized with a simple additional classification component, this metric attains a performance level comparable to state-of-the-art techniques on standard artificial discrimination tasks. We also establish in downstream tasks that this metric effectively differentiates between human-written documents and text generated by large language models under a specific domain. Furthermore, we illustrate the efficacy of this approach in detecting written styles attributed to diverse large language models, underscoring its potential for generalizability. In summary, we present a novel Brownian bridge coherence metric capable of measuring both local and global text coherence, while circumventing the need for end-to-end model training. This flexibility allows for its application in various downstream tasks.

## 1 Introduction

Text coherence is the quality of a written or spoken text that makes it internally consistent, logical, and easily understandable to the reader or listener. It encompasses the flow and connectivity of ideas, the logical progression of information, and the smooth transition between sentences and paragraphs. Evaluating the coherence of text requires the evaluator to differentiate between a set of individual utterances and naturally flowing discourses such as co-references or grammatical relationships (??).

In linguistic theory, the evaluation of text coherence can be performed at both the local and global levels (??). Local coherence focuses on the interrelation of information within

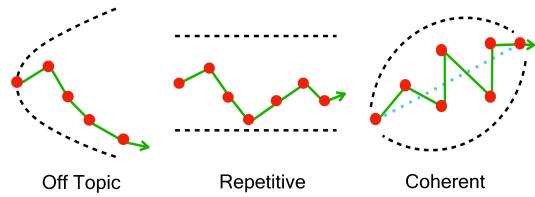


Figure 1: Utterance flow in latent space

a specific unit of discourse and its connection to the preceding unit. Conversely, global coherence refers to the degree to which the various components of discourse maintain a cohesive representation of the main topic. Despite the emergence of neural coherence models, most prior works focused on static representations of their local neighboring texts. (????)

This paper aims to capture *global coherence* from text, a sequential and cohesive relationship among the sentences or words, representing the main idea, goal, opinion, and other pertinent aspects regarding a coherent text. This relationship may vary across different situations, individuals, and even over time for the same individual. To simplify this concept and enhance its accessibility, we assume that this relationship remains consistent within a fixed domain or dataset. Also, we consider a coherent text should encompass the following properties: (1) A main idea should be kept throughout the text (2) Sentences near the beginning and ending part of the article should emphasize the main idea and (3) Sentences near the middle part can go a little bit far from the main idea, but still be controlled.

We present a novel metric that can capture global text coherence using the Brownian bridge framework, which has been used to analyze animal movements (?) or long-text generation (?). Brownian bridges serve as continuous-time stochastic models for movement, where the probability of being in a particular area is conditioned on the starting and ending locations, the time elapsed between those points, and the mobility or speed of movement (?). More specifically, we assume each step of this movement can be viewed as a sentence or word within the text, and the trajectory of the movement can represent different levels of coherence. Three such examples are given in Figure 1. In this paper, our

<sup>\*</sup>These authors contributed equally.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

contributions and findings are summarized as follows:

- We introduce  $\sigma_m$  in the Brownian bridge as a parameter that characterizes the text style.
- We design a flexible score measurement based on the Brownian bridge that can be used to evaluate the overall coherence of a text and naturally fit into any downstream tasks.
- We show our score can capture both global and local coherence and can effectively differentiate large language models generated and human-written text.

## 2 Related Work

Pre-neural models have predominantly relied on entity-based methods (??) to model coherence in text, as they closely tie to linguistic theory. This class of methods builds upon the Centering theory (?), which assumes that each sentence contains a centered entity that focuses the attention of the entire discourse. These methods provide insights into how information is connected at the local level but often fall short of capturing global coherence. With recent advancements in neural models such as the Transformer (?), efforts have been made to improve these methods using deep network architectures (???), integrating various lexical information to enhance performance.

Previous studies have revealed that Transformer language models struggle to effectively capture coherence structures (?). To address this limitation and guide language models in learning text dynamics, representation learning methods have incorporated auto variational encoders (?) or contrastive learning techniques (?), which aim to model sentence embeddings by leveraging information from neighboring utterances. However, these approaches yield static representations that overlook the dynamics of the entire text. In a recent study, ? proposed an innovative method utilizing the Brownian bridge to model long-range text dependencies. The proposed method exhibited superior performance in generating long and coherent discourses compared to the baseline GPT2 model. In this paper, we extend their findings by introducing a likelihood measurement and conducting experiments to assess its efficacy in capturing both global text coherence and local coherence.

The Shuffle test (?) has been widely utilized as a common practice for assessing coherence models. This artificial task requires the model to distinguish between an original document and its randomly shuffled counterpart. ? proposes that it should be treated as a probe—an evaluation task that allows models to be assessed without explicit supervision. Notably, they argue that the Shuffle test is not limited to local coherence analysis and can be extended to longer sequences, such as in the k-Blocked Shuffle test (??), where the block size can provide insights into a model’s ability to discern global coherence or identify main topics and subtopics.

## 3 Preliminary

**Brownian Bridge** A Brownian bridge is a stochastic process that modifies the standard Brownian motion by requiring an extra constraint (??). A standard Brownian bridge

$\{B(t)|t \in [0, 1]\}$  is a Gaussian stochastic process such that  $E[B(t)] = 0$  for all  $t \in [0, 1]$  and  $E[B(t_1)B(t_2)] = \min(t_1, t_2) - t_1t_2$  for all  $t_1, t_2 \in [0, 1]$ .

However, in our paper, we modified the standard Brownian bridge with a constant called diffusion coefficient and denoted it as  $\sigma_m^2$  to encode the information of the domain-specific property. Similar methods are also used to model animal mobility (?) and human mobility (?), but to the best of our knowledge, it is the first to use this assumption for evaluating text coherence. Formally, this modified Brownian bridge  $\{S(t)|t \in [0, T]\}$  follows a normal distribution  $S(t) \sim \mathcal{N}(\mu(t), \sigma^2(t)\mathbf{I})$ , for  $t \in [0, T]$ , where

$$\mu(t) = a + \frac{t}{T}(b - a), \quad \sigma^2(t) = \frac{t(T-t)}{T}\sigma_m^2.$$

$S(0) = a \in \mathbb{R}^n$  and  $S(T) = b \in \mathbb{R}^n$  are the starting and ending location respectively, and  $n \in \mathbb{N}$  is a fixed constant.

**Brownian Encoder via Contrastive Learning** To map utterances from the observable space into the designed latent space, we followed the practice from the previous work (?). The weights from pretrained GPT2 (?) are frozen while at the top of its hidden states, a trainable multi-layer perceptron is appended. The training details of the encoder are as follows. Given a sequence of data points  $X = \{x_1, \dots, x_N\}$ , we randomly draw batches  $\mathcal{B}$  consisting of randomly sampled triplets  $(x_{i_1}, x_{i_2}, x_{i_3})$  with  $1 \leq i_1 < i_2 < i_3 \leq T$  and optimize the encoder by minimize the corresponding contrastive objective function:

$$\mathcal{L} = \mathbb{E}_X \left[ -\log \frac{\exp(d(x_{i_1}, x_{i_2}, x_{i_3}; f_\theta))}{\sum_{(x_{i_1}, x_{i_2}, x_{i_3}) \in \mathcal{B}} \exp(d(x_{i_1}, x_{i_2}, x_{i_3}; f_\theta))} \right]$$

with  $d(x_{i_1}, x_{i_2}, x_{i_3}; f_\theta)$  defined by:

$$-\frac{\|f_\theta(x_{i_2}) - (1 - \frac{i_2 - i_1}{i_3 - i_1})f_\theta(x_{i_1}) - \frac{i_2 - i_1}{i_3 - i_1}f_\theta(x_{i_3})\|_2^2}{2\sigma^2},$$

where  $f_\theta$  denotes the encoder and  $\sigma^2 = \frac{(i_2 - i_1)(i_3 - i_2)}{i_3 - i_1}$ . The loss function can be viewed as maximizing towards the triplets sampled from a Brownian bridge process while minimizing the function for triplets sampled from a different sequence. We implemented the encoder training step based on code from (?). As one can find that during the training, the diffusion coefficient  $\sigma_m^2$  does not show up in the  $\sigma^2$ , and instead, this information is encoded into the function,  $f_\theta$ . After the training is finished, we can approximate the diffusion coefficient  $\sigma_m^2$  based on the training set and the encoder  $f_\theta$  which will be discussed in the following section.

## 4 Proposed Methodology: Brownian Bridge Score for Text Coherence

In this section, we introduce how to approximate the diffusion coefficient and how to design our Brownian bridge score (BBScore). First of all, as shown in Figure 2, we display the general procedure to use our BBScore: Given a domain-specific dataset  $\mathcal{D}_{train}$  and train the encoder  $f_\theta$  with this

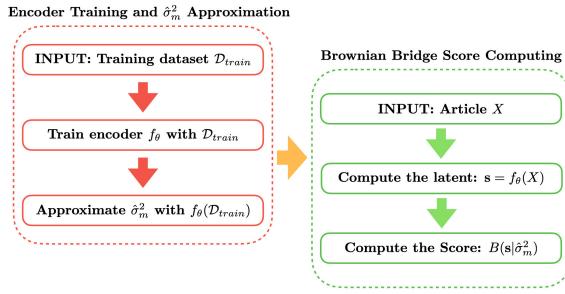


Figure 2: Procedure to generate the Brownian bridge score,  $B(s|\hat{\sigma}_m^2)$ .

dataset. Then, approximate the diffusion coefficient  $\sigma_m^2$  with all the latents  $f_\theta(D_{train})$ . After this, we can compute the BBScore of any given article which is assumed to be in the same domain as the training dataset  $D_{train}$ . The overall procedure to compute the BBScore is shown in the green dash box. Next, we will introduce how to approximate  $\sigma_m^2$  and construct BBScore  $B(\cdot|\sigma_m^2)$ .

The approximation and construction are both inspired by the likelihood of Brownian bridge. Given a Brownian bridge sequence  $\mathbf{s} = (s_1, s_2, \dots, s_{T(\mathbf{s})})$ , where  $s_i \in \mathbb{R}^n$ , for  $i = 2, 3, \dots, (T(\mathbf{s}) - 1)$  and  $T(\mathbf{s})$  is the length of  $\mathbf{s}$ , it satisfies:  $s_i \sim N(\mu_i, \sigma_i^2 \mathbf{I})$ , where  $\mathbf{I}$  is the  $n \times n$  identity matrix and

$$\mu_i = s_1 + \frac{i-1}{T(\mathbf{s})}(s_{T(\mathbf{s})} - s_1), \quad \sigma_i^2 = \frac{(i-1)(T(\mathbf{s})-i)}{T(\mathbf{s})}\sigma_m^2.$$

To estimate  $\sigma_m^2$  with the training dataset  $D_{train}$ , we consider the likelihood: for  $1 < i < T(\mathbf{s})$ , the likelihood of location  $s_i$  is:

$$\mathcal{L}_i = \frac{1}{2\pi\sigma_i^2} \text{Exp}\left(-\frac{1}{2\sigma_i^2} \|s_i - \mu_i\|^2\right).$$

Then the product of the log-likelihood of  $\mathcal{L}_i$  is

$$\ln\left(\prod_{i=2}^{T(\mathbf{s})-1} \mathcal{L}_i\right) = -\sum_{i=2}^{T(\mathbf{s})-1} \left(\ln(\alpha_i(\mathbf{s})\hat{\sigma}_m^2) + \frac{\beta_i(\mathbf{s})}{\sigma_m^2}\right), \quad (1)$$

where for  $i = 2, 3, \dots, T(\mathbf{s}) - 1$

$$\alpha_i(\mathbf{s}) = \frac{2\pi(i-1)(T(\mathbf{s})-i)}{T(\mathbf{s})-1}, \quad \beta_i(\mathbf{s}) = \frac{(T(\mathbf{s})-1)\|s_i - \mu_i\|^2}{2(i-1)(T(\mathbf{s})-i)}.$$

Next, by setting the derivative of Eq (1) to be zero, we can find the best approximation of  $\sigma_m^2$  with maximum likelihood:  $\hat{\sigma}_m^2(\mathbf{s}) = \frac{1}{T(\mathbf{s})-2} \sum_{i=2}^{T(\mathbf{s})-1} \beta_i(\mathbf{s})$ . Therefore, we can approximate the  $\sigma_m^2$  with the training set:

$$\hat{\sigma}_m^2 = \sum_{\mathbf{s} \in f_\theta(D_{train})} \hat{\sigma}_m^2(\mathbf{s}).$$

And to further simplify our method, we use the average of  $n$  dimension of the approximated  $\hat{\sigma}_m^2$  as the diffusion coefficient used in the BBScore. Besides, we show the necessity of such approximation by a simple sensitivity analysis, e.g. the standard Brownian bridge  $\sigma_m^2 = 1$  cannot capture the

coherence of the article (See Appendix C). Next, we define the BBScore which is inspired by the log-likelihood:

$$B(\mathbf{s}|\hat{\sigma}_m^2) = \frac{|\sum_{i=2}^{T(\mathbf{s})-1} \ln(\alpha_i(\mathbf{s})\hat{\sigma}_m^2) + \frac{\beta_i(\mathbf{s})}{\hat{\sigma}_m^2}|}{T(\mathbf{s})-2}. \quad (2)$$

Based on the likelihood (Eq (1)), this measure should show the following relation with text coherence: the more the text is coherent, the smaller the value of  $B(\mathbf{s}|\hat{\sigma}_m^2)$  will be.

## 5 Experiments

### 5.1 Research Questions

Throughout our experiments, we address two research questions as follows:

- While designed for global coherence, does BBScore capture both global and local coherence effectively in a synthetic setting? (Section 5.4)
- Can BBScore detect texts where the departure from desired coherence are not manipulated and recognize their differences? (Section 5.5)

### 5.2 Dataset

In this paper, our focus is on the WikiSection dataset (city) as introduced in (?). This dataset comprises Wikipedia articles related to global cities, sourced from Wikipedia dumps. Each segment of text within these articles is meticulously labeled with its corresponding section class and word count.

For our research, we make use of the pre-processed WikiSection data provided by ?. This curated dataset encompasses 2,165 articles in the training set and an additional 658 articles in the test set, all classified into 4 distinct section classes. Unlike the approach taken by ?, we deliberately exclude any section-related information during the training of the encoder. This decision is aimed at preventing the inadvertent disclosure of supplementary information and enhancing the model's overall applicability.

We then alter the original text to create material for each experiment. For the purpose of artificial tasks, we permute the original corpus to create various shuffled versions. As for downstream tasks, we employ large generative language models to complete each section. This is achieved by providing the model with the respective section title and the first few sentences as a prompt.

Additional details regarding the creation of the data are elaborated within the context of each specific experimental setup.

### 5.3 Methods for Comparison

We compare our proposed BBScore against the following coherence metrics:

**Entity Grid** ? is the most recognized entity-based approach. For each input document, it creates a two-way contingency table to track the entity appearance in each sentence. The issue with this count-based approach is the grids can easily become sparse as the length of the document increases or when the text is informal (?). The algorithm struggles to identify meaningful entity transitions across sentences. This issue

persists even in the subsequent neural version of Entity Grid. We include these methods in the comparison as the benchmark to validate the efficacy of BBSScore. We use Stanford’s CoreNLP to annotate the documents and the implementation in coheoka library\* to obtain the Entity Grid score.

**Unified Coherence** ? is a neural-based entity-grid method that incorporates sentence grammar, inter-sentence coherence relations, and global coherence patterns into a common neural framework, and it reaches state-of-the-art results in many artificial tasks. However, this model architecture is trained end-to-end with a pairwise ranking loss, and in our opinion, these pairwise constraints can help the model reach a good performance in some tasks, but limits its application and generalization, since in the real world not all the incoherent article is an exact shuffled version of a coherent one.

**BBSScore** Figure 2 shows a framework for the computation of BBSScore. For example, in the WikiSection task, we train the encoder with the original training dataset and approximate the diffusion coefficient with the latents of the training dataset. The score is supposed to represent the deviation from the latent Brownian trajectory from training data.

**BBSScore + Classifier** We use global BBSScore and local moving window BBSScore with window sizes (defined in Appendix D) being 1, 2, 4, and 8 as features showing how coherent the text is. A three-layer perceptron then takes those features as input and outputs predictions for designed tasks.

#### 5.4 Artificial Tasks

**Global Discrimination** The evaluation of coherence can be effectively assessed through the Shuffle test (??). This evaluation method involves randomly permuting sentences within a discourse to create an incoherent document and compared to the original one. By employing this task, the overall coherence of the text can be effectively evaluated. To ensure the robustness and sensitivity of our proposed coherence scoring mechanism, we also incorporated the k-Blocked Shuffle test, which introduces varying levels of incoherence into the documents. In this test, we first trained the Brownian encoder using the WikiSection training set and then mapped both the original and shuffled documents into the latent space, generating their respective BBSScores.

To create the dataset, we generated a single random shuffle for each document and constructed a mixture of raw and shuffled documents, forming the basis of a binary classification task. We used BBSScores to discern between coherent and incoherent text without additional supervision. To further align with established methodologies (???), we extended our experimentation to encompass a larger number of shuffles. Specifically, we generated 20 random shuffles for each document, both within the training and test sets. Furthermore, to maintain the integrity of the evaluation process, any documents that matched their original copy were omitted from the analysis. The statistics of the resultant dataset for 20 shuffle global discrimination tasks are shown in Table 1.

\*<https://github.com/kigawas/coheoka>

		Global		Local	
		Data	#Pairs	Data	#Pairs
Train	$\mathcal{D}_{b=1}$	36,574		$\mathcal{D}_{w=1}$	20,762
	$\mathcal{D}_{b=2}$	22,738		$\mathcal{D}_{w=2}$	26,165
	$\mathcal{D}_{b=5}$	17,110		$\mathcal{D}_{w=3}$	27,069
	$\mathcal{D}_{b=10}$	11,356		$\mathcal{D}_{w=1,2,3}$	73,996
Test	$\mathcal{D}_{b=1}$	8,600		$\mathcal{D}_{w=1}$	6,255
	$\mathcal{D}_{b=2}$	4,280		$\mathcal{D}_{w=2}$	7,924
	$\mathcal{D}_{b=5}$	3,837		$\mathcal{D}_{w=3}$	8,176
	$\mathcal{D}_{b=10}$	2,193		$\mathcal{D}_{w=1,2,3}$	22,355

Table 1: Statistics for the shuffle task dataset

**Local Discrimination** Detecting incoherent documents in the global discrimination task is relatively straightforward; however, assessing local coherence at the document level presents a more challenging task. To evaluate the effectiveness of our coherence scoring approach for local coherence, we constructed datasets using local permutation windows.

Following the methodology outlined in ?, we randomly sampled a set of windows, denoted as  $w$ , from the entire text. Each window consisted of a fixed size of three sentences, ensuring there was no overlap between any of the windows. Subsequently, we randomly shuffled the order of sentences within each window. We repeated the shuffle procedure for 20 times, excluding any duplicate sample.

We generated three separate datasets for the local discrimination task, namely  $\mathcal{D}_{w=1}$ ,  $\mathcal{D}_{w=2}$ , and  $\mathcal{D}_{w=3}$ . Each dataset had a different window size, which determined the degree of evaluation for local coherence. Figure 3 demonstrates examples of incoherence caused by local shuffles.

#### 5.5 Downstream Task

As noted in a previous study (?), there is a limited correlation between model performance on artificial tasks and their performance on downstream tasks. Since BBSScore is not restricted by a well-defined structure, such as pairwise input, we further explore its potential application areas. We assess our method on two novel downstream tasks: a task aimed at distinguishing LLM-generated articles from human-written ones and another task to detect which LLM the given input texts are from.

**AI Differentiation** For this task, we use the WikiSection dataset as human-written texts. To obtain LLM-generated texts with similar structures as the human-written ones, we use the first few sentences in each section of the WikiSection article as prompts, ask LLM to complete each section, and then combine those sections into one generated article. We do not compare to ? in this task because it requires the text length to be aligned between the pair. We still include the Entity Grid as the baseline.

**LLM Detection** In order to exam the potential generalizability of BBSScore, we form a domain detection task where texts generated by different large language models can be viewed as different domains based on our assumption. Our

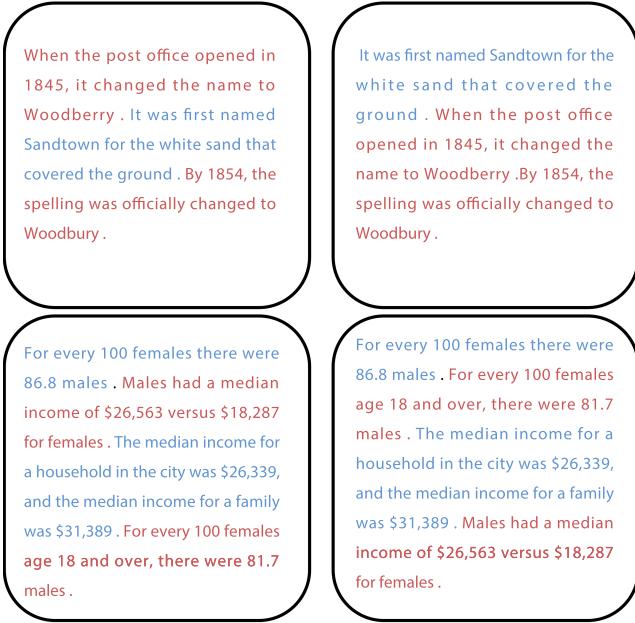


Figure 3: Example window pair for local discrimination task. **Left** is from the locally shuffled article and the **right** is from the original article.

primary inquiry revolves around the sensitivity of  $\hat{\sigma}_m^2$  in distinguishing the employed large language model during the text generation process when provided with input texts.

## 6 Results

### 6.1 Global Discrimination Task

We calculate the BBSScore for the original training and test set as well as the ones with various shuffle windows, and from their distribution we can conclude coherent text ought to have a lower BBSScore, which implies that they are closer to the hypothetical Brownian bridge process. We also rank the scores from all the articles and compute the AUC score for coherent and incoherent separation. The general performance of our model on shuffled tasks is shown in Figure 4. The AUC for Block 1 shuffle reaches 0.835 with the raw BBSScore and remains at 0.738 for Block 10 shuffle where the incoherence is harder to detect.

Next, we calculate the pairwise accuracy score for each document with scores derived from different methods, and the result is shown in Table 2. As the block size increases in the block shuffle task, the task becomes more challenging. This is because the shuffled article exhibits greater local coherence, but overall, it still lacks sufficient global coherence. The performance of the basic Entity Grid score on the test set shows significant variation with changes in block sizes. Consequently, to successfully accomplish this task, the metric should consider both local and global coherence while maintaining a balance between them.

From the result of the basic BBSScore, we can find that with an encoder trained on the original Train dataset and diffusion coefficient approximated with latents of the Train

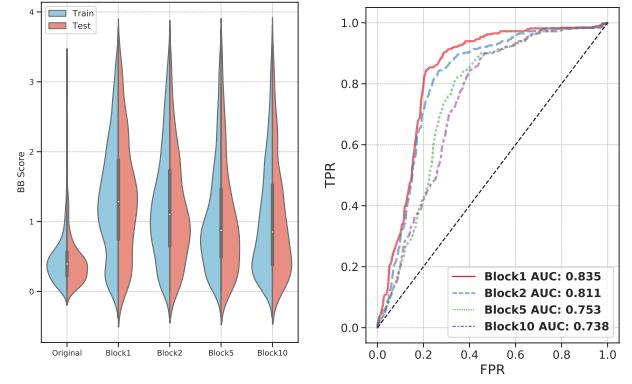


Figure 4: **Left:** BBSScore distribution with different size of shuffle blocks. **Right:** Test set BBSScore-based AUC with different size of shuffle blocks

dataset, its performance is still very stable even if the task becomes harder. After we add a simple linear classifier after the BBSScore, its performance exceeds (?) and becomes better when the block size increases. Therefore, we conclude the BBSScore has a better evaluation of global coherence while still capturing local coherence. To further explore and verify the ability of BBSScore on local coherence evaluation, we implement the local coherence discrimination task in the following section.

### 6.2 Local Discrimination Task

We compute pairwise scores for each document and its locally shuffled counterpart. The accuracy reported is based on whether the model can assign a lower BBSScore to the original document compared to the locally shuffled one in each pair. The results can be found in Table 3. As expected, the Unified Coherence Model (?) demonstrates strong performance across different locally shuffled datasets, given that their model incorporates sentence alignment and deliberately learns to differentiate between document pairs. Their model includes a dedicated module for capturing inter-sentence relations and modeling local coherence. In contrast, our approach focuses more on global coherence rather than supervising local windows. Nonetheless, our supervised model, BBSScore+CLF, demonstrates the fundamental ability to evaluate local coherence.

### 6.3 AI Generated Text Differentiation

In this section, we mainly focus on two different training strategies: (1) Pairwise discrimination; and (2) General AI discrimination.

For the pairwise discrimination task, we design a data pair, e.g. (original doc, language model generated doc) as an input to test whether we can distinguish which one was generated by humans. An example is shown in Appendix A, and in this pair, the document from WikiSection has a smaller BBSScore than the AI-generated document, so we can successfully detect the AI-generated one. Later, we also

Methods	Train				Test			
	$\mathcal{D}_{b=1}$	$\mathcal{D}_{b=2}$	$\mathcal{D}_{b=5}$	$\mathcal{D}_{b=10}$	$\mathcal{D}_{b=1}$	$\mathcal{D}_{b=2}$	$\mathcal{D}_{b=5}$	$\mathcal{D}_{b=10}$
ENTITYGRID (?)	79.17	86.20	74.51	60.70	85.73	82.79	75.81	64.65
UNIFIEDCOHERENCE (?)	<b>99.75</b>	98.60	97.10	96.21	<b>99.73</b>	97.86	96.90	96.09
BBScore	76.29	75.12	73.04	73.12	83.39	80.71	79.36	78.66
BBScore + CLF	99.12	<b>98.95</b>	<b>98.68</b>	<b>98.54</b>	98.92	<b>98.46</b>	<b>98.25</b>	<b>98.50</b>

Table 2: Global Discrimination Task Results on WikiSection

Methods	Train				Test			
	$\mathcal{D}_{w=1,2,3}$	$\mathcal{D}_{w=1}$	$\mathcal{D}_{w=2}$	$\mathcal{D}_{w=3}$	$\mathcal{D}_{w=1,2,3}$	$\mathcal{D}_{w=1}$	$\mathcal{D}_{w=2}$	$\mathcal{D}_{w=3}$
ENTITYGRID (?)	61.25	55.37	62.94	65.44	60.18	53.04	60.83	66.67
UNIFIEDCOHERENCE (?)	90.84	84.02	83.64	89.59	87.00	77.47	82.98	87.87
BBScore	57.85	47.17	55.84	63.10	57.66	50.29	60.15	64.06
BBScore + CLF	69.68	61.69	69.00	75.04	67.12	55.20	67.86	75.66

Table 3: Local Discrimination Task Results on WikiSection.  $\mathcal{D}_{w=1,2,3}$  represents the joint set of all three other datasets.

train a simple classifier with the BBScore on this pairwise classification task, and the complete result is listed in Table 4, where we use six different large language models to generate the document.

General AI discrimination task is a harder problem. With any document as an input, we train BBScore+CLF to predict whether humans or the AI write this document, the classification accuracy are shown in Table 5. We also compute the AUC score as shown in Figure 5. The AUC corresponds to different large language models that do not vary a lot which implies our model can finish the general AI discrimination task in a stable way.

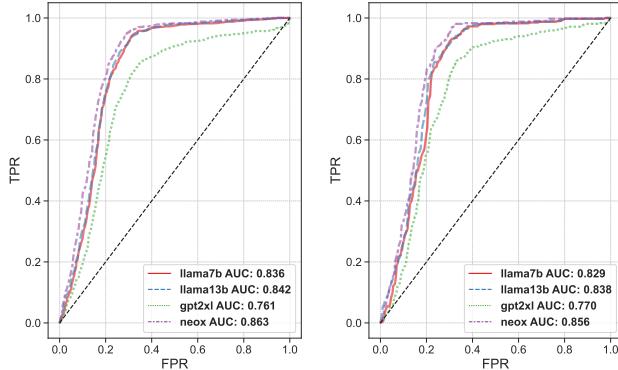


Figure 5: **Left** and **Right** correspond to **Train** and **Test** BBScore-based AUC with different LLMs, respectively.

#### 6.4 LLM Detection

For this task, we initiate by training a Brownian encoder utilizing a composite corpus consisting of articles generated

by diverse LLMs and the original articles. Subsequently, we derive estimates for  $\sigma_{m_k}^2$  from the training set corresponding to LLM  $k$ . Then we calculate  $\sigma_m^2(s)$  conditional on the input text  $s$  obtained from the test set. Moving forward, we evaluate the distributional similarity between  $\sigma_m^2(s)$  and  $\sigma_{m_k}^2$ , and normalize the outcomes for each  $s$  to ensure the values are on the scale from 1 to 2. The outcomes, as calculated by the Wasserstein distance, are illustrated in Figure 6. In the results, we observe that 5 out of the 6 test texts are in close proximity (among the top 2) to their respective source LLMs, except for the text generated by GPT-NeoX. This illustrates that the comparison of the distribution of  $\sigma_m^2$  effectively distinguishes a narrower range of LLMs to be considered. Given that the input texts pertain to the same subject and variations introduced by different LLMs can be subtle, we find this technique valuable in identifying subdomains in the condition of training with an extensive collection of text and can potentially be generalized into other domains. To examine the generalizability in terms of out of domain performance, we also test the WikiSection encoder on the GCDC dataset (?) (See Appendix B).

#### 6.5 Trajectory in Latent Space

In this section, we visualize the trajectories in the latent space of texts from different generation processes mapped by the trained Brownian encoder. Our goal is to examine whether the assumption about latent coherence, as depicted in Figure 1, holds in general. We test text from unseen WikiSection data, text generated by two large language models - ChatGPT and LLaMA2-13B, and text created by sentence-level permutation. The results are shown in Figure 7. The latents are interpolated to uniform length of documents and then averaged over all documents at each sentence index. From the visualization, we observe that randomly shuffled text follows a flat trajectory where sentences are not in logical order,

Methods	Train						Test					
	♣	♠	♦	♥	◊	♡	♣	♠	♦	♥	◊	♡
ENTITYGRID	54.43	36.99	18.64	25.23	43.50	25.86	59.57	36.17	25.88	32.19	39.40	22.85
BBScore	76.96	85.07	82.21	82.42	73.07	75.29	77.75	83.74	81.16	81.82	75.63	75.13
BBScore + CLF	<b>83.21</b>	<b>85.51</b>	<b>88.04</b>	<b>88.03</b>	<b>92.19</b>	<b>91.89</b>	<b>83.01</b>	<b>86.17</b>	<b>86.18</b>	<b>86.73</b>	<b>92.02</b>	<b>91.89</b>

Table 4: AI Generated Text Differentiation Results on WikiSection. ♣: GPT2XL ♠: GPT-NeoX ♦: LLaMA-7B ♥: LLaMA-13B ◊: LLaMA2-7B ♡: LLaMA2-13B

Dataset	LLM					
	♣	♠	♦	♥	◊	♡
Train	77.33	83.42	84.55	82.80	85.12	86.80
Test	76.44	82.65	82.53	81.20	84.53	85.35

Table 5: General AI Discrimination Task Results of BB-Score + CLF on WikiSection. ♣: GPT2XL ♠: GPT-NeoX ♦: LLaMA-7B ♥: LLaMA-13B ◊: LLaMA2-7B ♡: LLaMA2-13B

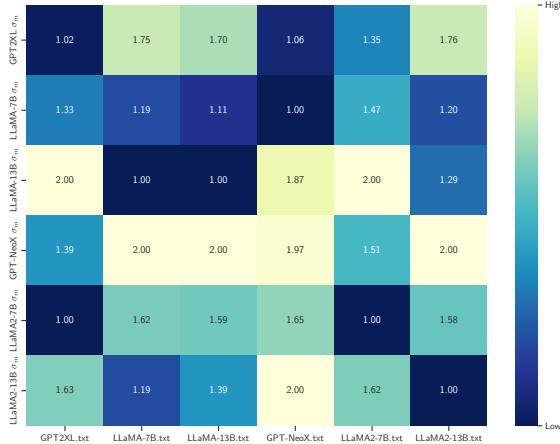


Figure 6: Normalized Wasserstein Distances. Each column represents the input texts are generated by a specific model.

and similar sentences could occur at any location within the document, thus disrupting the flow of the text. Conversely, text written by humans or generated by LLMs tends to follow a trajectory more akin to a Brownian bridge in the latent space. The movement of the latents typically starts from lower values with a jump and then progresses to higher values, accompanied by fluctuations in the middle. Concurrently, we also notice that the variance of the latents initially increases and then diminishes towards the end, particularly in human-written text. This observation aligns with the properties of a Brownian bridge process. These findings validate our assumption that coherent text resembles a Brownian bridge in the latent space, as per our modeling process.

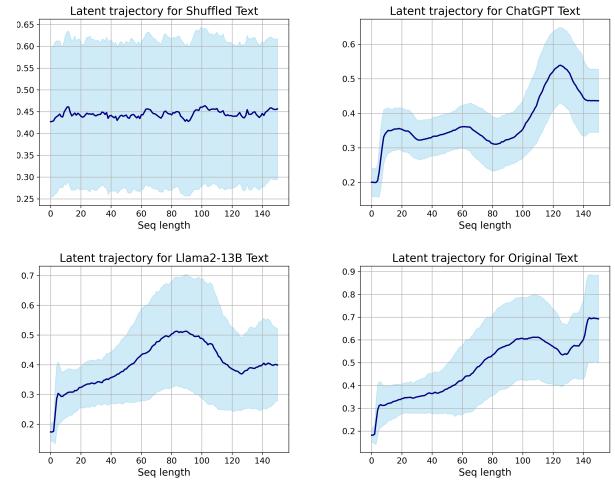


Figure 7: Latent trajectories for text from different generation processes.

## 7 Discussion and Limitations

In this paper, we propose the BBScore as a simple yet effective metric for measuring text coherence. Unlike previous entity-based approaches that rely on linguistic insights, the BBScore naturally learns text coherence through a contrastive objective during training. According to (?), three factors contribute to discourse coherence: the organization of discourse segments, the intention or purpose of the discourse, and attention or focused items. While well-organized discourse segments are common in formal writing, the abundance of informal text poses challenges in conveying the main topic or intention through entity transitions. In such cases, explicit guidance for learning text coherence becomes more difficult, necessitating a self-supervised approach.

In Section 5.3, we demonstrate that BBScore could be calculated at the sentence level, and by manipulating the sentence window sizes, we can generate a combination of features that capture coherence at different discourse units. Despite its simple network structure, BBScore achieves state-of-the-art results on global discrimination tasks in standard coherence assessment experiments, highlighting its potential for evaluating global text coherence. Additionally, even without a specific structure for capturing local coherence, our model still performs well in evaluating local text coherence. We believe that the diffusion coefficient encodes important

domain-specific features that aid in text coherence evaluation. More specifically, we show it can be leveraged for Human-AI discrimination and used to compare generation qualities of different models with respect to coherence. From the LLM detection task results, we find when training with an inclusive corpus,  $\hat{\sigma}_m^2$  can potentially be utilized to identify sub-domains. Our results also demonstrate that while Entity Grid is a simple and effective measure in artificial settings, it falters in downstream tasks.

While our approach has yielded positive results, we also identify several limitations. In this paper, our focus has been on a single dataset to showcase the usability of BBSScore. Although we have demonstrated its ability to discern different LLM-generated content by training with a cross-domain corpus, the obtained results are not entirely satisfactory and still require further robustness validation. In light of this, we acknowledge that the assumption we have made, linking  $\hat{\sigma}_m^2$  to specific domain/style, is rather restrictive in its general applicability. Future endeavors will involve expanding the parameter space by introducing higher-dimensional variance estimates. Furthermore, given BBSScore's demonstrated capacity to differentiate between LLM-generated text and human-authored text, we aspire to establish BBSScore as a more general and well-defined metric for comparing various LLMs in our forthcoming research. Lastly, BBSScore's correlations with formal Human evaluation should be examined.

## 8 Conclusion

Overall, BBSScore presents a novel perspective on text coherence and has demonstrated its efficacy on artificial tasks involving deliberately induced incoherence. Additionally, we illustrate the practical utility of BBSScore in a natural context, where unintentional deviations from desired coherent text occur. Serving as an intermediate computed score, BBSScore holds the potential to become a valuable feature in numerous real-world applications, including tasks related to Human-AI discrimination. In contrast to the intricate network architectures employed in neural entity-grid models (e.g., Multi-layer LSTM), our approach utilizes a simple three-layer perceptron with BBSScore as input for classification tasks, devoid of any crafted loss function. Remarkably, the experiments show this approach attains comparable, and in some cases, even superior results. Further refinement of our existing method promises to be an intriguing avenue for future research.

## Appendix

### A Example of pairwise discrimination on AI discrimination tasks

In the pairwise discrimination on AI discrimination tasks, for each data pair (original doc, AI-generated doc), we will compute the BBSScore separately as we did in the coherence task, the local/global discrimination task. An example is shown in Figure S2, here the AI-generated document uses the first sentence of the original doc as a prompt and generates the rest document with LLaMA7b. In this example, the original document has a lower BBSScore and the task on this data pair is marked as successful.

**ORIGINAL DOC: BBscore = 0.238**

[ ABSTRACT ] Richmond Heights is a city in Cuyahoga County, Ohio, United States. The population was 10,546 at the 2010 census. [ HISTORY ] Richmond Heights was founded as the Village of Claribel in 1917, but was later renamed as Richmond Heights in 1918. [ GEOGRAPHY ] Richmond Heights is located at (41.558183, -81.503651). Richmond Heights borders Euclid on the west, Lyndhurst and South Euclid on the south, Highland Heights on the east, and Willoughby Hills to the north. According to the United States Census Bureau, the city has a total area of , of which is land and is water. [ DEMOGRAPHICS ] 82.7% spoke English, 4.8% Russian, 3.1% Spanish, 1.9% Slovene, 1.7% Italian, 1.2% Chinese, and 1.1% Croatian. Of the city's population over the age of 25, 38.5% held a bachelor's degree or higher.

**LLAMA7B DOC: BBscore = 0.519**

[ ABSTRACT ] Richmond Heights is a city in Cuyahoga County, Ohio, United States. The population was 10,135 at the 2010 census. The city was incorporated [ HISTORY ] Richmond Heights was founded as the Village of Richmond Heights in 1923. The village was named for the Richmond Heights neighborhood in St. Louis, Missouri, which [ GEOGRAPHY ] Richmond Heights is located at (41.558183, -81.503999). According to the United States Census Bureau, the village has a total area of 0.3 square miles (0.7 km<sup>2</sup>), all of it land. As of the census of 2000, there were 1,000 people, 391 households, and 286 families residing in the village. [ DEMOGRAPHICS ] 82.7% spoke English, 4.8% Russian, 3.1% Spanish, 1.8% German, 1.4% French, 1.3% Italian, 1.2% Polish, 1.1% Arabic, 1.0% Ukrainian, 0.9% Yiddish, 0.8% Hebrew, 0.7% Chinese, 0

PROMPT

Figure 8: An example of the pairwise AI discrimination

### B Domain generalizability of Brownian encoders

We use the WikiSection encoder to encode GCDC texts and obtain the corresponding BBscores. A three-layer perceptron is then trained on the BBscores for a three-class classification task on the GCDC dataset. The results are shown in Table S1.

Dataset	Domain			
	Enron	Clinton	Yahoo	Yelp
Train	47.67	43.11	49.54	51.45
Test	47.50	41.50	42.64	49.25

Table 6: Three-class Classification Task Results on GCDC Dataset with the WikiSection Encoder.

### C Diffusion coefficient $\hat{\sigma}_m^2$ analysis

As shown in Figure S1, it describes the AUC score for the blocksize=1 shuffle test under different diffusion coefficients. It shows our current approximation of the diffusion coefficient (marked by the red dashed line) can give us a better result but not the best (marked by the olive dashed line). Moreover, it also shows, the standard Brownian bridge  $\hat{\sigma}_m^2=1$  shows a poor result AUC score=1 which emphasizes the necessity of this diffusion coefficient approximation.

### D BBSScore defined with a shifting window

The basic BBSScore is defined as:

$$B(\mathbf{s}|\hat{\sigma}_m^2) = \frac{|\sum_{i=2}^{T(\mathbf{s})-1} \ln(\alpha_i(\mathbf{s})\hat{\sigma}_m^2) + \frac{\beta_i(\mathbf{s})}{\hat{\sigma}_m^2}|}{T(\mathbf{s}) - 2}. \quad (3)$$

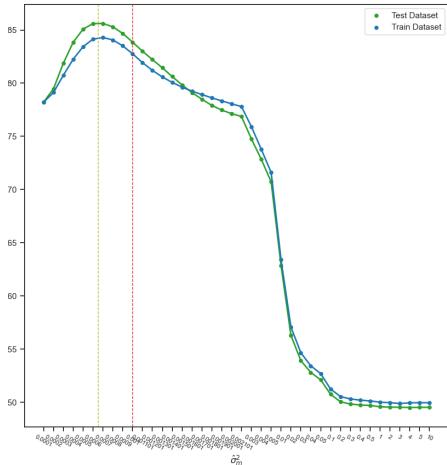


Figure 9: Diffusion coefficient analysis: The AUC score for shuffle test with block 1 under different  $\sigma_m^2$ , and the red dash line corresponds to the  $\hat{\sigma}_m^2$  approximated with the train dataset.

We also test BBSScore with a shifting window to capture the local coherence property: given a shifting window size  $2w+1$ ,  $w \in \mathbb{N}$ , the shifting window BBSScore  $B_w(\mathbf{s}|\hat{\sigma}_m^2)$  is defined as,

$$B_w(\mathbf{s}|\hat{\sigma}_m^2) = \frac{|\sum_{i=w+1}^{T(\mathbf{s})-w} \ln(\alpha_{i,w}(\mathbf{s})\hat{\sigma}_m^2) + \frac{1}{\hat{\sigma}_m^2}\beta_{i,w}(\mathbf{s})|}{T(\mathbf{s}) - 2w} \quad (4)$$

where for  $i = w+1, \dots, T(\mathbf{s})-w$

$$\alpha_{i,w}(\mathbf{s}) = 2\pi \frac{w(w+1)}{2w+1}, \quad \beta_{i,w}(\mathbf{s}) = \frac{(2w+1)||s_i - \mu_i||^2}{2w(w+1)}$$

and

$$\mu_i = s_{i-w} + \frac{w+1}{2w+1}(s_{i+w} - s_{i-w}).$$

## E Training details

For all the experiments mentioned in the main paper, we used a hidden dimension size of 128 for the multi-layer perceptron appended to the GPT2 encoder and an output dimension size of 8 for the fully connected layer. The Brownian encoders were then trained using the contrastive objective function via the SGD optimizer, with learning rate of  $1 \times 10^{-4}$ , momentum of 0.9, and batch size of 32. The GPT2-based encoder was trained on 1 node with 2 A100 GPUs and 32 GB of memory.

esse dicta hic nobis, consecetur blanditiis voluptas voluptate corporis ipsa. Illo mollitia maiores, quas veritatis consecetur asperiores preferendis repellendus ipsum illo deleniti, fuga reprehenderit molestias obcaecati atque rerum velit recusandae corrupti preferendis incident. Officiis iure cupiditate molestias consequatur ducimus similius asperiores vero, fugiat temporibus exercitationem preferendis aliquid. Doloremque cumque dolores, eum ipsam doloribus, corrupti eaque a asperiores presentium dolore qui iste quos nobis quidem ab, porro id animi molestias reprehenderit rem, id aspernatur nemo consequuntur dolor eos inventore soluta? Animi ad quia, minus dolorem

aspernatur quam in nobis ea eveniet quis sint unde delectus, corporis molestias ea atque cum veritatis ab provident. Quaerat ex cum beatae atque quis, beatae officis at ipsa architecto error quisquam enim, nihil architecto corporis voluptatem eaque repudiandae, excepturi aliquid molestiae in incident ea sapiente tenetur accusantium distinctio. Officia nisi odio iste aut, amet quasi sed consecetur assumenda voluptatum quia veniam ipsa excepturi? A sequi assumenda inventore esse deleniti eveniet corrupti ab corporis, nulla culpa illo voluptatibus quam perspicatis at tenetur iste veritatis optio beatae. Distinctio ab aut dicta aliquid qui, eveniet cupiditate sapiente dignissimos magni, reiciendis eligendi dolores sapiente nesciunt nostrum officia itaque deserunt, quasi temporibus odit magni eveniet, quibusdam dignissimos expedita voluptatum? Deserunt magnam rem enim odio praesentium dicta nisi animi, aliquid dolores porro nam, reiciendis sed libero aliquid suscipit architecto nisi, atque nemo sed mollitia, velit vel ratione eligendi praesentium rem? Mollitia vel earum asperiores ad inventore nihil veritatis culpa, veritatis dolorum quasi debitis harum similius nulla, quia fugit odio incident maiores harum reprehenderit molestias et facere, illo reprehenderit officiis sed ipsa exercitationem quos officia labore ab facere, praesentium animi at minus nam. Aliquam ab eos, laudantium omnis praesentium nihil recusandae quae quas beatae doloribus officiis doloremque sunt, molestias corporis ad tempora maxime esse quaerat, voluptas necessitatibus inventore incident sequi facilis. Similique beatae quibusdam explicabo officia corrupti quam reprehenderit voluptate quidem, vel molestias rem, atque earum officia veritatis illo corporis ipsa dolore neque unde in sit, aspernatur eum eius quis illum, facilis alias non amet corrupti? Modi laboriosam doloribus necessitatibus, possimus aut sed consecetur facere, magnam cum ullam amet suscipit sunt itaque voluptatibus officia blanditiis accusantium laudantium, vitae alias ad expedita aperiam vel quo accusantium nihil numquam similius molestiae, eaque ullam officia velit perspicatis voluptatum numquam eum nam ad? Accusantium corporis ea autem, reprehenderit ipsam at saepe pariatur cumque voluptatem consecetur quaerat est amet? Voluptatum voluptate sint ipsam id necessitatibus similius ab pariatur ipsa error, rem voluptas incident possimus sapiente autem, harum cumque dolore obcaecati, animi reprehenderit numquam exercitationem omnis nihil perspicatis eaque sapiente neque architecto. Optio veniam soluta adipisci quam laborum, pariatur sit incident quae autem inventore natus preferendis, recusandae ullam amet a quaerat sapiente preferendis ipsam voluptatibus consequuntur soluta. Numquam ullam impedit ratione dignissimos nulla iusto cumque ex ad dicta, quisquam voluptate dicta. Necessitatibus architecto laudantium quaerat perspicatis dicta aut quisquam placeat tempora exercitationem expedita, non amet distinctio blanditiis est ipsa suscipit consequatur veritatis iure, nostrum praesentium sint obcaecati laudantium repellat repudiandae ducimus deserunt officiis non, aliquam accusantium magni voluptatibus illum laudantium obcaecati impedit autem iusto architecto? Quis animi itaque, quibusdam similius officiis, dignissimos aspernatur iure in deleniti inventore ab quidem. Odit enim placeat corporis, tenetur esse iste nisi harum labore, nulla accusantium preferendis aliquid reiciendis eaque blanditiis numquam sunt deleniti nostrum tempora, exercitationem officiis placeat animi nihil voluptates reprehenderit odio est. Accusantium obcaecati officiis ipsam laborum vitae asperiores doloremque, incident labore dolorem eum assumenda exercitationem ab ratione quidem harum, vero explicabo rem consequatur laborum adipisci nulla laudantium, sit

inventore error earum. Vitae quidem ut aspernatur numquam re-  
iciendis architecto incident molestiae at ea accusantium, maiores  
voluptatum porro?