
Compressed and Smooth Latent Space for Text Diffusion Modeling

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Autoregressive language models dominate modern text generation, yet their se-
2 quential nature introduces fundamental limitations: decoding is slow, and main-
3 taining global coherence remains challenging. Diffusion models offer a promising
4 alternative by enabling parallel generation and flexible control; however, their
5 application to text generation is hindered by the high dimensionality of token-level
6 representations. We introduce COSMOS, a novel approach to text generation that
7 operates entirely in a compressed, smooth latent space tailored specifically for
8 diffusion. This space is learned using an autoencoder trained simultaneously for
9 token-level reconstruction and alignment with frozen activations from a pretrained
10 language encoder, providing robust semantic grounding and enabling effective
11 perturbation-based augmentations. Empirically, we demonstrate that text represen-
12 tations can be compressed by $8\times$ while maintaining generation quality comparable
13 to token-level diffusion models. Furthermore, increasing the latent sequence length
14 allows COSMOS to surpass both diffusion-based and autoregressive baselines. We
15 evaluate COSMOS on four diverse generative tasks including story generation,
16 question generation, summarization, and detoxification and compare it with various
17 generative paradigms. COSMOS achieves comparable or superior generation quality
18 while offering more than $2\times$ faster inference.

19 1 Introduction

20 Autoregressive (AR) language models are a de facto gold standard for text generation [28, 29, 6].
21 By factorizing the probability of a sequence into a product of conditional token probabilities, they
22 transform the global generation task into a series of local next-token prediction tasks that can be
23 optimized efficiently via teacher forcing [37].

24 The same sequential factorization, however, creates several structural bottlenecks. First, decoding is
25 inherently sequential: the time to produce a sequence grows linearly with its length because each
26 token must await the completion of its predecessor. Second, the process suffers from exposure bias: a
27 single early mistake contaminates the context for all subsequent predictions and cannot be corrected
28 without restarting generation [31, 2, 10]. Third, maximizing local log-likelihood encourages the
29 model to privilege fluency over factual accuracy, resulting in the well-documented phenomenon
30 of hallucination [20, 12]. Finally, because decisions are made token-by-token, the model lacks an
31 explicit global plan and therefore struggles to maintain long-range logical or narrative coherence
32 [39].

33 In stark contrast, computer vision has been reshaped by diffusion models [11, 26], particularly by la-
34 tent diffusion [32, 5, 4], which first compresses a high-resolution image or video into a compact latent
35 representation and then performs the diffusion process in that space. Operating in a low-dimensional
36 latent manifold reduces computational cost by orders of magnitude and enables breathtaking advances

in image and video synthesis. Yet recent works [13, 34] also show that diffusion is highly sensitive to the geometry of the latent space: poorly designed latent representations can destabilize training and degrade sample quality, underscoring the need for principled mechanisms to construct *diffusable*¹ representations.

Motivated by these insights, we revisit the foundations of text representation. We hypothesize that the token-level encoding used in contemporary language models is heavily overparameterized for the purpose of sequence generation. Building on the successes of latent diffusion in vision, we pose the following question: *How far can we compress textual information into a compact latent space while still matching or even exceeding the generative fidelity of conventional token-level representations?*

To answer this question, we develop an autoencoder that maps text into a lower-dimensional space and then train a diffusion model directly in that space. We show empirically that naively minimizing token-reconstruction loss yields a brittle latent geometry that hampers diffusion, whereas introducing robustness- and smoothness-oriented objectives produces a well-behaved manifold conducive to high-quality diffusion synthesis. Our experiments demonstrate that with the right training regime, latent diffusion not only rivals but in several settings surpasses traditional token-level baselines.

These findings challenge the prevailing assumption that token-level autoregression is indispensable for language generation and position latent-space diffusion as a powerful alternative paradigm for future large-scale language models. Our key contributions are as follows:

- We propose COSMOS — a training recipe for a **C**ompressed and **S**MOOTH latent **S**pace, that allows to train a diffusion model in more compact latent space while matching the quality of token-level diffusion baselines.
- We demonstrate that modestly scaling the number of latent vectors enables latent-space diffusion to surpass both token-level diffusion and autoregressive models on an unconditional text generation task.
- On standard text-generation benchmarks, the proposed latent diffusion achieves up to $2\times$ faster sampling than conventional token-level diffusion models, while matching or slightly exceeding them in quality and diversity.

2 Related work

Early efforts to bring **latent** diffusion models into natural language generation treat the diffusion module mainly as a *pre-processor* that supplies conditioning vectors for an autoregressive decoder. LD4LG [18] trains a diffusion model on compressed hidden states from BART, then feeds the generated latent vectors into the BART decoder to produce text. PLANNER [42] adopts a similar two-stage recipe: a fine-tuned BERT encoder produces a 16-token variational latent code, a diffusion model refines that code, and a GPT-2 decoder generates text from this code. While both systems improve controllability, their heavy reliance on powerful autoregressive decoders makes it difficult to isolate and evaluate the intrinsic generative capacity of the latent-space diffusion itself.

A complementary line of work applies diffusion directly to the continuous embeddings of pretrained encoders. TEncDM [33] demonstrates that full-length BERT representations can be modeled with Gaussian diffusion and decoded into coherent text without an intermediate autoregressive step. Crucially, however, TEncDM retains the original sequence dimensionality, leaving open a question: *How aggressively can such representations be compressed before generation quality collapses?*

Our study closes this gap. We devise a training procedure that shrinks BERT-level representations by up to $8\times$ while preserving, and in some cases enhancing, their suitability for latent-space diffusion.

3 Preliminary

Diffusion models [11, 26] learn a data distribution by reversing a progressive noising process. Given a clean sample $z_0 \sim p_{\text{data}}$, the forward dynamic corrupts the input with Gaussian noise whose magnitude is controlled by a continuous time index $t \in [0, 1]$:

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1 - \alpha_t} \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \quad (1)$$

¹Diffusable representations refer to latent spaces that permit effective modeling by a diffusion process.

where the noise schedule α_t is monotonically decreasing with $\alpha_0 = 1$ and $\alpha_1 \approx 0$. A neural denoiser $z_\theta(z_t, t)$ with learnable parameters θ is trained to invert this corruption by minimising the denoising-score-matching objective

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{z_0 \sim p_{\text{data}}, t \sim \mathcal{U}[0,1], \varepsilon \sim \mathcal{N}(0,I)} \left[\|z_0 - z_\theta(z_t, t)\|_2^2 \right]. \quad (2)$$

At inference time the model is applied iteratively from $t = 1$ to $t = 0$, gradually transforming pure noise into a realistic sample.

3.1 Latent diffusion

Latent Diffusion Models (LDMs) [32, 4] improve both compute efficiency and quality by learning the diffusion process in a compact latent space rather than in pixel or token space. An autoencoder with encoder E and decoder D is first trained to achieve high-fidelity reconstruction, $\hat{\mathbf{w}} = D(E(\mathbf{w})) \approx \mathbf{w}$, where $\mathbf{z} = E(\mathbf{w})$ denotes the low-dimensional latent. Eqs. (1)–(2) yield a diffusion model trained in latent space whose network can be shallower, and faster than a counterpart operating in the original space, yet still achieves comparable perceptual quality after decoding $D(\mathbf{z})$.

In the remainder of the paper we adopt this framework for textual data: we compress contextualised text representations into the latent space, and train a diffusion model that operates within this space.

4 Methodology

4.1 Overview

Frozen text encoder. We initialise the pipeline with a pretrained contextual encoder E_{text} (BERT-base [7] by default) that remains *frozen* during all subsequent training stages. For an input sequence of L tokens $\mathbf{w} = (w_1, \dots, w_L)$, the encoder produces a matrix of hidden states $\mathbf{h} = E_{\text{text}}(\mathbf{w}) \in \mathbb{R}^{L \times d}$, where $d = 768$. Each row supplies a *semantically rich* representation of its corresponding token [35]. These representations provide a high-fidelity starting point for the compression stage.

Compressor. To distill the variable-length hidden states into a compact set of latents, we employ a Perceiver Resampler architecture [1]. It is a 12-layer transformer [36] in which every block replaces self-attention with cross-attention. Concretely, let $\mathbf{u} \in \mathbb{R}^{N \times d}$ be a set of learnable vectors initialized randomly and kept at a fixed length $N \ll L$. In each block, these vectors act as **queries** ($Q = \mathbf{u}W_Q$), while **keys** and **values** are formed by projecting a concatenation of \mathbf{u} with the text encoder outputs ($K, V = [\mathbf{u}; \mathbf{h}]W_{K,V}$). Cross-attention therefore allows every vector u_i gather information from the entire sequence of hidden states and from other vectors \mathbf{u} , gradually refining \mathbf{u} into a semantically organized compressed representation. After the final block we obtain a *fixed-length* latent matrix $\mathbf{z} \in \mathbb{R}^{N \times d}$, which composes latent space for the diffusion process. For better interpretability of results we compress only along the sequence axis. Altering the embedding dimension d would require architectural changes in the diffusion network itself, conflating encoder effects with diffusion capacity and obscuring the variables we aim to isolate.

Latent normalization. Before starting the Gaussian diffusion process, we estimate global mean and standard deviation $(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \mathbb{R}^{N \times d}$ on a held-out corpus, and normalize each latent feature so that it has zero mean and unit variance, $\mathbf{z} \leftarrow (\mathbf{z} - \boldsymbol{\mu})/\boldsymbol{\sigma}$. This step allows us to run variance-preserving diffusion process.

Latent diffusion model. We train a Gaussian diffusion model in space formed by \mathbf{z} , following Eqs. (1)–(2). Because N is small, the diffusion model runs faster in this space.

Decompressor. Decompressor mirrors compressor in terms of architecture. It expands the fixed-length latent back to a sequence of L vectors $\hat{\mathbf{h}} \in \mathbb{R}^{L \times d}$, capped at $L_{\text{max}} = 512$ by default.

Token predictor. Finally, a linear projection followed by softmax converts each vector in $\hat{\mathbf{h}}$ into a probability distribution over the vocabulary, yielding the generated text.

In our work, we refer to the combination of the text encoder and compressor as the *encoder*, while the decompressor together with the token predictor constitutes the *decoder*.

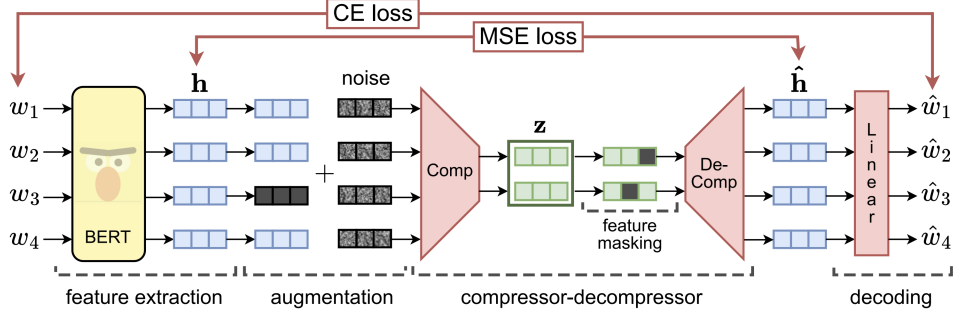


Figure 1: Overview of our training pipeline. A frozen BERT encoder extracts features, which are augmented before compression. A lightweight compressor–decompressor pair is trained with both token reconstruction (CE) and MSE objectives to produce compact and perturbation-resilient latent representations.

4.2 Learning a compact text latent space

The previous Section 4.1 outlined our end-to-end pipeline: a frozen contextual encoder supplies token representations, a compressor distils them into a fixed-length latent matrix, and a diffusion model operates solely in this latent space. We now zoom in on the *compression* and *decompression* stages.

4.2.1 How compact can text latents be?

Figure 2 reports token-level reconstruction accuracy on WIKIPEDIA dataset as a function of the latent sequence length N . The compressor outlined in §4.1 is trained using a token-level cross-entropy objective. Remarkably, a lightweight compressor equipped with only 12 transformer blocks is able to encode 512-token sequences into only $N = 16$ latents with 100% reconstruction accuracy, achieving a $32\times$ compression relative to initial 512 hidden states produced by BERT.

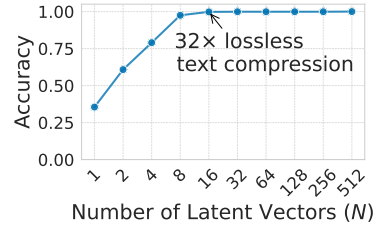


Figure 2: Token-level reconstruction accuracy on WIKIPEDIA (512 tokens) as a function of N .

These results paint an encouraging picture: if such a compact latent manifold can be modeled generatively, text generation could proceed in a space whose dimensionality is much smaller than that of the original token embeddings, promising dramatic speed-ups. However, as we show further, high reconstruction accuracy by itself does not imply that the latent manifold is suitable for generative modeling.

4.2.2 Learning robust compact text representations

The reconstruction study (§4.2.1) reveals that aggressive text compression can be loss-free. However, for a robust diffusion generation compressed space should satisfy additional requirements. Empirically, we observe that if the latent manifold lacks smoothness and robustness (Section 5), a Gaussian diffusion model fails to sample latents that yield high-quality texts. To improve diffusability of the latent space, we employ three complementary strategies to the autoencoder training. Figure 1 provides an overview of the proposed approach.

MSE regularisation on encoder activations. Alongside the cross-entropy loss between the original tokens and their reconstructions, we add a mean-squared error penalty between the frozen text encoder outputs \mathbf{h} and their reconstructions $\hat{\mathbf{h}}$. This auxiliary objective forces the compressor to preserve the semantics carried by the contextual representations \mathbf{h} .

Activation-space perturbations. In order to teach compressor to extract additional features from \mathbf{h} instead of just preserving its semantics, we apply perturb-and-recover training procedure. Concretely, we sample an augmented view \mathbf{h}' , pass it through the compressor–decompressor pipeline to obtain a reconstructed representation $\hat{\mathbf{h}}'$, and minimise $\text{MSE}(\mathbf{h}, \hat{\mathbf{h}}')$, forcing both compressor and decom-

163 pressor to remain invariant to the perturbation. Two perturbations are applied with equal probability
 164 within every batch:

- 165 (a) *Random masking*: 30% of the vectors of \mathbf{h} are set to zero, and
- 166 (b) *Gaussian noise*: after normalising \mathbf{h} with the pre-computed statistics, we inject noise via
 167 $\mathbf{h}' = \delta \mathbf{h} + \sqrt{1 - \delta^2} \varepsilon$ with $\delta = 0.7$ and $\varepsilon \sim \mathcal{N}(0, I)$.

168 These augmentations force the autoencoder to tolerate partial information loss and promote smooth
 169 interpolation between adjacent latents (see Section 5.1).

170 **Latent-space augmentation.** We also apply augmentation directly to the latent matrix \mathbf{z} . Since the
 171 number of latent vectors N can be an order of magnitude smaller than the token count L , masking
 172 whole latent vectors would annihilate too much information. Instead, during training we randomly
 173 zero out a fixed proportion p of the *individual features* inside each latent vector. This fine-grained
 174 sparsification encourages neighbouring features within every latent vector to store redundant cues
 175 about one another, so that the representation remains intelligible even when a subset of features is
 176 excised, thereby making the manifold more robust to small latent perturbations..

177 5 Latent-space properties that facilitate diffusion training

178 This section embarks on an exploration of the text-latent manifold, asking which of its intrinsic
 179 features govern the performance of a diffusion model. All experiments reported here employ
 180 autoencoders that compress 128-token texts into 16 latent vectors. The autoencoders are trained on
 181 the WIKIPEDIA dataset, whereas the diffusion models are optimised on the smaller yet high-quality
 182 ROCSTORIES [22] dataset. Across all experiments, we generate 1 000 samples and report each
 183 metric as the average over these samples. Our experiments spotlight two indispensable qualities —
 184 *manifold smoothness* and *resilience to perturbations*. The remainder of the section introduces an
 185 analysis that quantifies both attributes.

186 5.1 Smoothness of the latent manifold

187 Once the autoencoder has been trained, each text \mathbf{w} corresponds to a set of latent vectors \mathbf{z} that can
 188 be decoded back into the original text, $\mathbf{w} = D(\mathbf{z})$. Nonetheless, the latent space is far from fully
 189 charted: broad regions contain vectors to which no text has ever been assigned, leaving the behaviour
 190 of the data density there unknown. Because our diffusion model captures the distribution of texts
 191 via the distribution of these latents, yet observes only a finite subset during training, its ability to
 192 generalize depends critically on how smoothly that density varies across the manifold. Put differently,
 193 the degree to which a locally estimated score function extends to unexplored territory is dictated by
 194 manifold smoothness. Inspired by PLANNER [42], we investigate this property by conducting the
 195 following experiment, which simultaneously evaluates performance at points the model does not
 196 encounter during training.

- 197 1. Select two random texts from the training corpus and encode them as latent vectors
 198 $\mathbf{z}^{(1)}, \mathbf{z}^{(2)} \in \mathbb{R}^{N \times d}$.
- 199 2. Form a linear interpolation of the endpoints $\mathbf{z}^\mu = \mu \mathbf{z}^{(1)} + (1 - \mu) \mathbf{z}^{(2)}$, where $\mu \in [0, 1]$ and
 200 mid-range μ values steer \mathbf{z}^μ into regions unseen during training.
- 201 3. Apply diffusion noise (Equation (1)) to get \mathbf{z}_t^μ at different time steps t to measure how unseen
 202 regions influence diffusion at different noise levels.
- 203 4. Predict a clean latent $\hat{\mathbf{z}}_0$ from \mathbf{z}_t^μ and decode into text $\hat{\mathbf{w}}$.
- 204 5. Evaluate text plausibility with GPT-2 perplexity (PPL), averaged over 1 000 random endpoint
 205 pairs.

206 By analysing how perplexity (PPL) evolves with the interpolation coefficient μ and the diffusion
 207 timestep t , we obtain a fine-grained picture of manifold smoothness and model robustness far beyond
 208 the training support. Figure 3 shows that autoencoder trained only with cross-entropy (CE) loss
 209 and our robustness-oriented alternative behave identically when $\mu \rightarrow 0$ or $\mu \rightarrow 1$, confirming that
 210 both models successfully reconstruct train-time input latents. As we move away from the endpoints,
 211 perplexity rises for both encoders, but it escalates far more rapidly for the CE baseline — clear
 212 evidence that the latent manifold learned by our autoencoder is markedly smoother.

213 5.2 Reducing the train–inference mismatch

214 We empirically observe that when a Gaussian diffusion
 215 model is trained in the text latent space, a persistent *train–*
 216 *inference mismatch* often emerges: the latent vector $\hat{\mathbf{z}}$
 217 produced at sampling time can differ markedly from the
 218 latent vector the encoder would assign to its own decoded
 219 text, $E(D(\hat{\mathbf{z}})) \neq \hat{\mathbf{z}}$. This gap has two undesirable ef-
 220 fects. First, the decoder becomes unreliable because it
 221 must interpret latent vectors contaminated by the diffusion
 222 model. Second, the diffusion model repeatedly feeds itself
 223 inputs it never saw during training, compounding errors
 224 over time. To enable high-quality text generation under
 225 this mismatch, both the decoder and the diffusion model
 226 must be robust to perturbations in the latent space. In the
 227 following sections, we demonstrate through experiments how our training strategy (§4.2.2) fosters
 228 such robustness, effectively decreasing train–inference mismatch.

229 **Decoder robustness.** To assess how well the decoder
 230 tolerates perturbations, we inject Gaussian noise into latent
 231 vectors of real texts: $\mathbf{z}_{\text{noised}} = \mathbf{z} + \sigma \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, I)$,
 232 decode the perturbed vector to obtain $\hat{\mathbf{w}}$, and measure its
 233 BLEU score against the original text \mathbf{w} . Figure 4 shows
 234 that two components contribute most to decoder stability:
 235 (i) an explicit MSE loss on the decompressor’s output
 236 (§4.2.2), which prevents the final-layer norm from explod-
 237 ing, a common effect observed when training solely with
 238 cross-entropy loss [8], and (ii) latent masking (§4.2.2),
 239 which forces the compressor to distribute information
 240 more evenly among latent features and remain resilient to
 241 partial dropout. Notably, the decoder supplemented with
 242 described train-time modifications can reconstruct text almost perfectly even under substantial noise
 243 ($\sigma = 1$).

244 **Diffusion robustness during generation.** Next, we test
 245 the diffusion model’s sensitivity to small perturbations
 246 introduced *mid-trajectory*. For a late generation step t ,
 247 chosen from the second half of the reverse process, where
 248 latent representations are already semantically meaningful,
 249 we add Gaussian noise with magnitude ν and continue
 250 sampling. The deviation from the original trajectory is
 251 quantified by the MSE between the final estimates, $\|\hat{\mathbf{z}} -$
 252 $\hat{\mathbf{z}}^{\text{shifted}}\|_2^2$.

253 As illustrated in Figure 5, the diffusion model trained in
 254 our latent space is visibly more stable.

255 **Direct mismatch measurement.** Table 1 tracks the
 256 train–inference mismatch, quantified as the MSE between
 257 $\hat{\mathbf{z}}$ and $E(D(\hat{\mathbf{z}}))$. Each proposed refinement tightens the
 258 gap. The consistent drop in error confirms that the en-
 259 hanced training procedure makes the autoencoder more
 260 resilient to diffusion perturbations. Together, these re-
 261 sults demonstrate that our training recipe simultaneously
 262 strengthens the decoder and the diffusion model, thereby
 263 shrinking the train–inference gap.

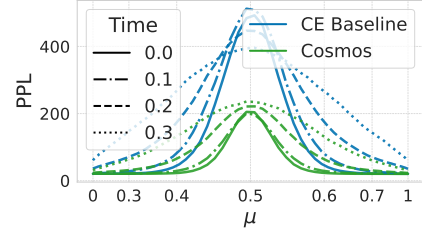


Figure 3: PPL of texts decoded from an interpolation of two latents for COSMOS and CE baseline.

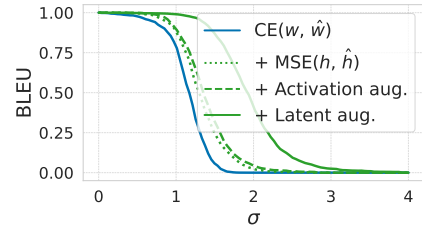


Figure 4: Decoder robustness to latent noising with sequential addition of training modifications.

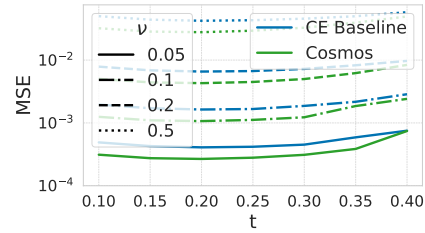


Figure 5: Evaluating diffusion model robustness under mid-trajectory noise injection.

Table 1: Train–inference mismatch for different autoencoder objectives.

Configuration	MSE ($\hat{\mathbf{z}}, E(D(\hat{\mathbf{z}}))$)
CE(w, \hat{w})	0.721
+ MSE(h, \hat{h})	0.619
+ Activation aug.	0.582
+ Latent aug.	0.499

6 Empirical study of latent representations for diffusion modeling

This section investigates how the design of autoencoder training strategies and the degree of latent space compression influence the quality of text generation via diffusion.

6.1 Empirical analysis of autoencoder training regimes

In this part of our study we assess the contribution of each proposed latent space augmentation to the final quality of the diffusion model. For the ablation study, we compress text from 128 BERT representations down to 16 latent vectors. All autoencoders are trained on the WIKIPEDIA dataset, while the diffusion models are trained on the ROCSTORIES [22] dataset.

Evaluation metrics. We evaluate the unconditional generation capacity of models with three complementary metrics that, together, capture textual quality, lexical diversity, and distributional fidelity. First, we report **perplexity (PPL)**, computed with GPT-2 LARGE [29]. Second, we quantify diversity using the score $\text{div}(y) = \prod_{n=2}^4 \frac{\# \text{ unique } n\text{-grams in } y}{\# n\text{-grams in } y}$, where y denotes the set of generated texts. Finally, because low perplexity can be achieved by simple or repetitive texts, we also compute the **MAUVE** score [27], which measures the distributional alignment between generated and reference texts and thus offers a broader view of generation quality. For every model we generate 1,000 samples and, when calculating MAUVE, draw an equally sized reference subset from the held-out test split. The entire evaluation procedure is repeated ten times with different random seeds, and we report the mean and standard deviation of the resulting scores across runs.

Results. Table 2 demonstrates that each proposed feature contributes significantly to the model performance. Remarkably, adding MSE penalty between the predicted and reference BERT activations alone boosts all quality metrics by about 50%. Subsequent augmentation of the BERT activations yields a further leap, underscoring the value of intermediate contextual, semantically grounded representations. This stage rises MAUVE to 0.767, already achieving the performance obtained when the diffusion model is trained on the full-length, uncompressed BERT representations (0.767 vs. 0.762 for TEncDM [33]). Finally, dropping random features from latent vectors pushes MAUVE beyond 0.83, lowers perplexity to 30.2, keeping diversity essentially unchanged.

These results provide a clear answer to the question posed in the introduction of this study: text representations can indeed be mapped into a more compact latent space, where a trained diffusion model performs comparably to traditional token-level counterparts without sacrificing the text generation quality. For all subsequent autoencoders we lock in the configuration highlighted in Table 2: the MSE penalty is kept, the random-masking rate is fixed at 0.3, Gaussian noising is applied with $\delta = 0.7$, and latent masking rate is set to 0.4.

Additionally, we validate all proposed autoencoder configurations by training the diffusion model on the larger WIKIPEDIA dataset. Detailed results of this evaluation are presented in Appendix C.2. Furthermore, we explore the use of a variational prior in the latent space and observe that it offers no clear advantage. A thorough analysis is provided in Appendix C.1.

Table 2: Comparison of autoencoder training regimes on text generation quality. Features are added cumulatively from top to bottom. The gray-highlighted row indicates the selected configuration used in the final model.

Configuration	MAUVE \uparrow	PPL \downarrow	Div \uparrow
CE(w, \hat{w})	0.294 _{.036}	71.5 _{.9}	0.298 _{.001}
+ MSE(h, \hat{h})	0.464 _{.024}	57.1 _{.7}	0.344 _{.002}
+ Random masking (rate)			
0.1	0.465 _{.029}	54.5 _{.4}	0.356 _{.003}
0.2	0.565 _{.023}	50.5 _{.5}	0.342 _{.001}
0.3	0.586 _{.015}	42.8 _{.7}	0.335 _{.003}
0.4	0.514 _{.011}	46.4 _{.3}	0.344 _{.002}
0.5	0.461 _{.025}	51.8 _{.7}	0.362 _{.005}
+ Gaussian noising (δ)			
0.5	0.679 _{.01}	36.8 _{.3}	0.310 _{.003}
0.6	0.724 _{.015}	35.6 _{.7}	0.324 _{.004}
0.7	0.767 _{.011}	33.6 _{.3}	0.328 _{.002}
0.8	0.725 _{.018}	36.4 _{.7}	0.333 _{.002}
0.9	0.664 _{.023}	44.3 _{.8}	0.294 _{.003}
+ Latent dropout (rate)			
0.1	0.771 _{.014}	33.9 _{.7}	0.325 _{.004}
0.2	0.781 _{.029}	32.6 _{.8}	0.322 _{.004}
0.3	0.812 _{.012}	31.9 _{.5}	0.328 _{.003}
0.4	0.836 _{.009}	30.2 _{.5}	0.322 _{.004}
0.5	0.724 _{.012}	34.8 _{.4}	0.320 _{.002}

6.2 Impact of scaling the number of latent vectors

In this section, we examine how diffusion generation quality varies when varying the number of latent vectors N . We keep the training pipeline fixed according to the hyperparameters detailed in Section 6.1. We do not modify the embedding dimension d , because it would necessitate architectural modifications to the diffusion model, making it difficult to isolate the impact of latent space diffusability from changes in model capacity and structure.

Figure 6 compares baseline cross-entropy compressor to the robustness-oriented compressor introduced in Section 4.2.2. We observe that for CE compressor decrease of N leads to a rapid degradation in quality. In contrast, the proposed autoencoder maintains high generation quality even under substantial compression. It achieves an $8\times$ reduction in latent sequence length while surpassing the quality of uncompressed representations.

Table 8 provides further insights by comparing generation quality across different numbers of latent vectors N . Although configuration $N = 16$ matches the quality of BERT representations baseline, we observe a notable quality gap compared to the configuration with $N=128$: the MAUVE score decreases from 0.940 to 0.836, while perplexity increases from 26.3 to 30.2. However, adopting a less aggressive compression to $N=32$ (a $4\times$ compression) results in only minor quality degradation, still significantly outperforming the baseline. Based on these observations, together with results from Section 7, we recommend limiting latent compression to approximately $4\times$ to optimally balance quality and computational efficiency.

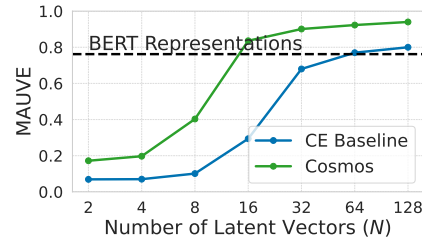


Figure 6: Impact of scaling N on diffusion generation quality.

Table 3: Impact of scaling the number of latent vectors N on unconditional generation quality.

N	MAUVE \uparrow	PPL \downarrow	Div \uparrow
Source	0.953	21.7	0.403
BERT repr.	0.762 _{.043}	29.1 _{.9}	0.295 _{.002}
2	0.172 _{.018}	109.5 _{.9}	0.344 _{.004}
4	0.197 _{.016}	70.1 _{.8}	0.354 _{.003}
8	0.403 _{.007}	51.9 _{.7}	0.327 _{.002}
16	0.836 _{.009}	30.2 _{.5}	0.322 _{.004}
32	0.901 _{.008}	27.3 _{.4}	0.346 _{.004}
64	0.923 _{.009}	26.7 _{.2}	0.347 _{.003}
128	0.940 _{.011}	26.3 _{.4}	0.346 _{.004}

7 Comparison across generative paradigms

We evaluate the performance of COSMOS on four distinct text generation tasks: unconditional story generation (ROCStories [22]), text summarization (XSum [24]), detoxification (ParaDetox [16]), and question generation (SQuAD2.0 [21]). For unconditional story generation we employ metrics discussed in Section 6.1. For text summarization and question generation, we report BERTScore (BS) [41]. The detoxification task is evaluated using BLEU-4 (BLEU). We further assess models’ efficiency by comparing inference time for unconditional generation on sequences with 128 and 512 tokens. Additional metrics and implementation details are provided in the Appendix C.3.

Baselines. We benchmark our proposed method against a diverse set of generative paradigms. These include Gaussian diffusion on embeddings (DiffuSeq [9], SeqDiffuSeq [40], AR-Diffusion [38]), simplex-based diffusion (TESS [19]), masked diffusion (SEDD [17]), alternative latent diffusion baselines (LD4LG [18], TEncDM [33]), and autoregressive models (GPT-2 [29] and GPT-Neo [3]). All models operate within a comparable parameter scale, ranging from approximately 100M to 200M. To ensure fairness, all baselines were faithfully reimplemented and trained on generative tasks using training protocols closely aligned with those described by their original authors.

Autoencoder setup across tasks. To perform latent-space diffusion, we begin by training autoencoders on WIKIPEDIA dataset, varying the input sequence length. For tasks with relatively short input, such as unconditional story generation and text detoxification, we design two variants of the autoencoder that map 128-token input into either 16 or 128 latent vectors, resulting in the $\text{COSMOS}_{N=16}$ and $\text{COSMOS}_{N=128}$ configurations, respectively. In contrast, conditional generation tasks such as summarization and question generation require substantially longer input contexts. For

Table 4: Comparison with autoregressive and diffusion baselines across four generative tasks. Inference time (in seconds) is reported for sequence lengths of 128 and 512 tokens. The best-performing scores are shown in **bold**, while the second-best scores are underlined.

Method	ROCStories			XSum	ParaDetox	SQuAD2.0	Time (s)	
	MAUVE \uparrow	PPL \downarrow	Div \uparrow	BS \uparrow	BLEU \uparrow	BS \uparrow	$L = 128$	$L = 512$
Source text	0.953	21.7	0.403	—	—	—	—	—
GPT2	0.789	<u>20.5</u>	0.252	0.690	0.677	0.680	1.2 _{.1}	<u>42.8</u> _{.1}
GPT Neo	0.720	19.9	0.258	0.621	0.610	0.665	<u>2.6</u> _{.1}	45.2 _{.2}
AR-Diffusion	0.066	41.8	0.101	0.568	0.647	0.569	226.4 _{.2}	602 _{.1}
DiffuSeq	0.086	50.5	0.124	0.588	0.679	0.563	215.9 _{.1.2}	1565 _{.7}
SeqDiffuSeq	0.103	29.3	0.137	0.617	0.688	0.574	100 _{.5}	601 _{.5}
TESS	0.061	22.4	0.163	0.627	0.693	0.667	1441 _{.2}	5984 _{.0}
SEDD	0.598	70.8	<u>0.336</u>	0.576	0.666	0.443	15.0 _{.1.1}	60.3 _{.1.0}
LD4LG	0.716	30.6	0.331	<u>0.702</u>	0.708	0.641	27.9 _{.5}	102 _{.2}
TEncDM	0.762	29.1	0.295	0.699	0.619	<u>0.703</u>	29.6 _{.1}	180 _{.3}
COSMOS _{$N=16$}	<u>0.836</u>	30.2	0.322	—	0.654	—	5.8 _{.1}	—
COSMOS _{$N=128$}	0.940	26.3	0.346	0.704	<u>0.694</u>	0.708	35.1 _{.1}	36.6 _{.1}

these settings, we employ a third autoencoder configured to compress 512-token sequences into 128 latent vectors. Accordingly, we do not report results for long-context tasks for COSMOS _{$N=16$} , as such aggressive compression fails to yield plausible outputs in these long-context scenarios. We adopt $N=128$ ($4\times$ compression) for long-context tasks as it strikes a favorable balance between generation quality and computational efficiency, as demonstrated in the scaling analysis in Section 6.2.

Results. As shown in Table 9, COSMOS _{$N=16$} achieves strong performance relative to its closest latent diffusion baseline, TEncDM. It consistently outperforms TEncDM across most evaluation metrics, with the exception of unconditional perplexity, while offering significantly faster generation. Scaling up to a bigger latent representation, COSMOS _{$N=128$} further advances generation quality. On ROCStories, it achieves a MAUVE score of 0.940, closely approaching the human reference score of 0.953, and substantially outperforming the best autoregressive baseline (GPT2), which reaches only 0.789. Although COSMOS _{$N=128$} lags behind GPT2 in perplexity, this gap reflects an evaluation bias: perplexity inherently favors autoregressive models, particularly when assessed using the same decoding objective they were trained on.

In generation tasks that involve longer input contexts, COSMOS _{$N=128$} consistently matches or slightly exceeds the performance of both diffusion-based and autoregressive baselines, while substantially reducing inference time. This efficiency stems from a core advantage of latent diffusion: unlike autoregressive decoding, where inference time grows linearly with sequence length, diffusion models operate with a fixed number of sampling steps. Although this cost is less favorable for short sequences, the benefits become increasingly pronounced as input length grows. This favorable scaling behavior makes latent diffusion as a promising direction for future research on efficient, long-context language generation.

8 Conclusion

In this paper, we present COSMOS, a novel approach to text generation that shifts the focus from high-dimensional token-level representations to a compact latent space tailored for diffusion modeling. By learning a compressed and smooth latent representation through carefully designed autoencoder objectives, COSMOS enables high-quality generation while reducing inference time. Our empirical results show that, COSMOS matches or outperforms traditional token-level diffusion and autoregressive baselines across diverse tasks. Our findings challenge the dominance of token-level models and highlight latent diffusion as a promising direction for building fast, high-quality language models.

References

- [1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning, 2022.
- [2] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [3] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, Mar. 2021.
- [4] A. Blattmann et al. Align your latents: High-resolution video synthesis with latent diffusion models. 2023.
- [5] A. Blattmann et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. 2023.
- [6] T. B. Brown, B. Mann, N. Ryder, and et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.
- [8] T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.
- [9] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.
- [10] T. He, J. Zhang, Z. Zhou, and J. Glass. Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation? *arXiv preprint arXiv:1905.10617*, 2019.
- [11] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [12] Z. Ji, G. Wang, B. Wang, and et al. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2023. to appear.
- [13] T. Kouzelis, I. Kakogeorgiou, S. Gidaris, and N. Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. *arXiv preprint arXiv:2502.09509*, 2025.
- [14] H. Laurençon, L. Saulnier, T. Wang, C. Akiki, A. Villanova del Moral, T. Le Scao, L. Von Werra, C. Mou, E. González Ponferrada, H. Nguyen, et al. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826, 2022.
- [15] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
- [16] V. Logacheva, D. Dementieva, S. Ustyantsev, D. Moskovskiy, D. Dale, I. Krotova, N. Semenov, and A. Panchenko. ParaDetox: Detoxification with parallel data. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [17] A. Lou, C. Meng, and S. Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

- [18] J. Lovelace, V. Kishore, C. Wan, E. Shekhtman, and K. Q. Weinberger. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36:56998–57025, 2023.
- [19] R. K. Mahabadi, H. Ivison, J. Tae, J. Henderson, I. Beltagy, M. E. Peters, and A. Cohan. Tess: Text-to-text self-conditioned simplex diffusion. *arXiv preprint arXiv:2305.08379*, 2023.
- [20] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1906–1919, 2020.
- [21] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. Deep learning-based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40, 2021.
- [22] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, 2016.
- [23] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In K. Knight, A. Nenkova, and O. Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics.
- [24] S. Narayan, S. B. Cohen, and M. Lapata. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*, 2018.
- [25] S. Narayan, S. B. Cohen, and M. Lapata. Don’t give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.
- [26] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning (ICML)*, 2021.
- [27] K. Pillutla, S. Swayamdipta, R. Zellers, J. Thickstun, S. Welleck, Y. Choi, and Z. Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.
- [28] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [30] P. Rajpurkar, R. Jia, and P. Liang. Know what you don’t know: Unanswerable questions for SQuAD. In I. Gurevych and Y. Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [31] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.
- [32] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [33] A. Shabalín, V. Meshchaninov, E. Chibulatov, V. Lapikov, R. Kim, G. Bartosh, D. Molchanov, S. Markov, and D. Vetrov. Tencdm: Understanding the properties of the diffusion model in the space of language model encodings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25110–25118, 2025.

- 493 [34] I. Skorokhodov, S. Girish, B. Hu, W. Menapace, Y. Li, R. Abdal, S. Tulyakov, and A. Siarohin.
494 Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.
- 495 [35] I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In A. Korhonen,
496 D. Traum, and L. Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association*
497 *for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for
498 Computational Linguistics.
- 499 [36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and
500 I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*,
501 30, 2017.
- 502 [37] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural
503 networks. *Neural Computation*, 1(2):270–280, 1989.
- 504 [38] T. Wu, Z. Fan, X. Liu, H.-T. Zheng, Y. Gong, J. Jiao, J. Li, J. Guo, N. Duan, W. Chen, et al. Ar-
505 diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information*
506 *Processing Systems*, 36:39957–39974, 2023.
- 507 [39] J. Ye, J. Gao, S. Gong, L. Zheng, X. Jiang, Z. Li, and L. Kong. Beyond autoregression: Discrete
508 diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024.
- 509 [40] H. Yuan, Z. Yuan, C. Tan, F. Huang, and S. Huang. Seqdiffuseq: Text diffusion with encoder-
510 decoder transformers. *arXiv preprint arXiv:2212.10325*, 2022.
- 511 [41] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text
512 generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.
- 513 [42] Y. Zhang, J. Gu, Z. Wu, S. Zhai, J. Susskind, and N. Jaitly. Planner: Generating diversified
514 paragraph via latent language diffusion model. *Advances in Neural Information Processing*
515 *Systems*, 36:80178–80190, 2023.

516 **Appendix**

517 A Dataset descriptions **14**

518 B Implementation details **14**

519 B.1 Latent diffusion pipeline 14

520 B.2 Training and inference configuration 14

521 B.3 Inference time benchmarking configuration 15

522 C Additional autoencoder training analysis **15**

523 C.1 Marginal utility of variational prior in text autoencoder 15

524 C.2 Empirical analysis of autoencoder training regimes on Wikipedia dataset 16

525 C.3 Additional details of comparison across generative paradigms 16

526 D Limitations **17**

527 E Societal Impact **17**

528 F Generation examples **17**

A Dataset descriptions

ROCStories This dataset [23] consists of five-sentence stories and serves as a well-established, small-scale, unconditional benchmark in language diffusion research. It contains a total of 98,161 instances, of which 88,161 are used for training, 10,000 for validation.

Wikipedia For large-scale experiments, we use the English Wikipedia subset from the ROOTS corpus [14]. The resulting dataset comprises 8.6 million sequences, with 50,000 held out for validation.

XSum This dataset [25] is used for abstractive summarization and comprises 204,000 BBC news articles paired with summaries covering diverse topics (e.g., sports, politics). It includes 204,045 training instances, 11,332 validation instances, and 11,334 test instances. We truncate input articles to 512 tokens and limit reference summaries to 50 tokens.

SQuAD2.0 This question-answering dataset [30] requires identifying the text span that answers a given question or indicating that no answer is possible from the provided context. For generative modeling, we reverse the task: given a context and an answer, the model generates the corresponding question. The dataset contains 130,319 training and 11,873 test instances. Contexts are truncated to 512 tokens.

ParaDetox For small-scale conditional generation experiments, we use the ParaDetox dataset [16], which comprises 19,766 pairs of toxic and neutral comments. Both context and target sequences are truncated to 40 tokens.

B Implementation details

B.1 Latent diffusion pipeline

With a diffusable latent manifold in hand (§4.2.2), we now describe the diffusion model that operates entirely within this space. Our setup follows Eq. (2): we train a denoiser that receives a noisy latent \mathbf{z}_t at diffusion time $t \in [0, 1]$ and predicts the corresponding clean latent \mathbf{z}_0 . The loss is the simple mean-squared error $\|\mathbf{z}_0 - z_\theta(\mathbf{z}_t, t)\|_2^2$, where z_θ shares parameters across all time-steps. In line with earlier work [18, 33], we employ *self-conditioning*: on 50% of training updates the model is fed its own previous estimate of \mathbf{z}_0 as an additional input. This iterative refinement enables the model, at inference time, to reuse its past predictions, producing markedly crisper and more coherent text. We adopt the noise schedule introduced by TEncDM [33]: $\alpha_t = \frac{1}{1 + \tan^2(t\pi/2) \cdot d^2}$, where parameter d controls the rate at which noise is injected during the diffusion steps. At inference time we use Euler solver with 200 uniform steps. The denoiser itself inherits the TEncDM [33] architecture. It is a 12-layer Transformer with 12 attention heads and a hidden size of 768. In total, the network comprises roughly $\approx 130\text{M}$ parameters.

Conditional generation. For sequence-to-sequence generation, we train a diffusion model in a conditional setting. The model learns to reconstruct a noisy latent representation of the target text, conditioned on the latent representation of the source text generated by our autoencoder. Consequently, the source text’s vector representation is also leveraged in a compressed format, enhancing both training and inference efficiency. The diffusion model is conditioned on the source latent, which is first processed by a 12-layer transformer encoder. This encoder extracts relevant features and derives a suitable conditioning representation from the source latent. These encoded source representations are then injected into the diffusion model using cross-attention mechanisms. For conditional tasks, we initiate the training of our diffusion model by initializing its denoiser weights with those pretrained on an unconditional task.

B.2 Training and inference configuration

All models are trained on 8 NVIDIA A100 GPUs. Detailed training configurations and approximate durations for both $\text{COSMOS}_N = 128$ and $\text{COSMOS}_N = 16$ are provided in Table 5.

	ROCStories	Wikipedia	XSum	SQuAD2.0	ParadetoX
Optimizer			AdamW		
Learning Rate			2e-4		
(β_1, β_2)			(0.9, 0.98)		
Warmup Steps			1000		
Learning Rate Schedule			Constant		
Weight Decay			0.01		
Gradient Clipping			1		
EMA Decay			0.9999		
Batch Size			1024		
Training Steps	200k	500k	100k	100k	10k
Max Seq Length	80	512	64	64	40
Max Context Length	—	—	512	512	40
Sampling steps			200		
Schedule parameter	5	3	7	7	9
Autoencoder Training Time	—	14h	—	—	—
Diffusion Training Time	1d 5h	2d 12h	1d 1h	1d 9h	14h

Table 5: Training details for COSMOS across different datasets.

B.3 Inference time benchmarking configuration

We measure the generation time for a batch of 512 samples per model on a single NVIDIA A100 GPU using the `bfloat16` data type. Inference settings for all baseline models follow the default configurations of their respective repositories, except for SEDD, where we adopt 32 diffusion steps in accordance with the original paper’s setup for matching autoregressive quality. We report the mean and standard deviation across five independent runs.

C Additional autoencoder training analysis

C.1 Marginal utility of variational prior in text autoencoder

In this section, we explain why incorporating a variational prior does not significantly enhance the performance of our text autoencoder.

In a classical Variational Autoencoder (VAE), the encoder maps each input to the parameters of a Gaussian latent distribution, *i.e.*, a mean vector μ and a (diagonal) variance vector σ^2 . Training minimises a reconstruction term together with the Kullback–Leibler (KL) divergence between the encoder distribution and an isotropic prior $\mathcal{N}(0, I)$. For a latent matrix $z \in \mathbb{R}^{N \times d}$, the KL term is

$$D_{\text{KL}}(\mathcal{N}(z; \mu, \sigma^2) \parallel \mathcal{N}(0, I)) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^d (\mu_{ij}^2 + \sigma_{ij}^2 - \log \sigma_{ij}^2 - 1).$$

Our text autoencoder is trained with the following objective:

$$\mathcal{L} = \text{CE}(w, \hat{w}) + \text{MSE}(h, \hat{h}) + \beta D_{\text{KL}}(\mathcal{N}(z; \mu, \sigma^2) \parallel \mathcal{N}(0, I)),$$

where $\text{CE}(w, \hat{w})$ is the token-level cross-entropy, $\text{MSE}(h, \hat{h})$ matches reconstructed contextual representations, and $\beta \geq 0$ balances reconstruction fidelity against latent regularisation.

During training, the decoder receives a stochastic latent sample

$$z_s = \mu + \sigma \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I),$$

so that the KL term nudges (μ, σ) towards the prior. Increasing β strengthens this pressure, producing a smoother latent geometry at the cost of higher reconstruction error; decreasing β does the opposite as encoder tends to push latents away from each other.

In our experiments, we observed that the performance is highly sensitive to the choice of β . Table 6 presents the results for different values of β .

The best perplexity is achieved at $\beta = 0.01$, but the gain over the baseline ($\beta = 0$) is marginal (33.6 \rightarrow 32.6). For larger β the model collapses: at $\beta = 1$, perplexity explodes and MAUVE falls.

This limited improvement can be attributed to the decoder’s robustness to Gaussian noise in the latent space. As illustrated in Figure 4, the decoder maintains reasonable performance even when sampling from the prior with $\sigma = 1$, indicating that the latent space is sufficiently robust without strong KL regularization.

Alternative regularisers, notably *latent masking*, yield larger and more robust improvements. We therefore omit the KL term in the final model and rely solely on latent masking to shape the representation space.

Table 6: Impact of the KL weight β on diffusion generation quality. Arrows indicate the preferred direction.

β	MAUVE \uparrow	PPL \downarrow	DIV \uparrow
0	0.767	33.6	0.328
0.0001	0.733	39.6	0.329
0.001	0.764	33.3	0.328
0.01	0.765	32.6	0.326
0.1	0.658	39.6	0.329
1	0.011	677.1	0.587

C.2 Empirical analysis of autoencoder training regimes on Wikipedia dataset

We also conduct a large-scale ablation on the WIKIPEDIA dataset with $N = 16$ latents and max sequence length $L = 128$, following a setup analogous to that in section 6.1. Results, presented in Table 7, reveal trends consistent with those observed in the small-scale ablation. Notably, applying the proposed training modifications leads to substantial improvements in evaluation metrics. The final model approaches the performance of the TEncDM baseline [33], while operating in a latent space that is 8 \times smaller.

Additionally, we investigate how scaling the latent space length on WIKIPEDIA affects performance. We experiment with latent sequence lengths ranging from 16 to 128. As expected, we observe steady performance improvements with longer latent sequences. Results are provided in Table 8.

Table 7: Comparison of autoencoder training regimes on text generation quality on WIKIPEDIA. Features are added cumulatively from top to bottom.

Configuration	MAUVE \uparrow	PPL \downarrow	Div \uparrow
CE(w, \hat{w})	0.056	385.6	0.590
+ MSE(h, \hat{h})	0.069	344.4	0.625
+ Random masking (rate)			
0.3	0.081	289.1	0.594
+ Gaussian noising (δ)			
0.7	0.103	205.0	0.588
+ Latent dropout (rate)			
0.4	0.112	186.7	0.581

Table 8: Impact of scaling the number of latent vectors N on unconditional generation quality on WIKIPEDIA.

N	MAUVE \uparrow	PPL \downarrow	Div \uparrow
Source	0.953	21.7	0.403
BERT repr.	0.109	173.1	0.562
16	0.112	186.7	0.581
32	0.149	134.4	0.579
64	0.157	128.1	0.582
128	0.173	118.2	0.592

C.3 Additional details of comparison across generative paradigms

We also present extended results for our conditional generation tasks using a broader variety of evaluation metrics. For XSUM and SQUAD2.0, we report ROUGE scores [15], a standard metric that evaluates text quality based on n-gram overlap with reference outputs. For PARADETOX, we additionally include the J-score, which is defined as the product of *style accuracy*, *fluency*, and *content preservation*. Extended results are summarized in Table 9.

Table 9: Comparison with autoregressive and diffusion baselines across three generative tasks. The best-performing scores are shown in **bold**, while the second-best scores are underlined.

Method	ParaDetox		Xsum		SQuAD2.0	
	BLEU \uparrow	J-Score \downarrow	R-1/2/L \uparrow	BS \uparrow	R-L \uparrow	BS \uparrow
GPT2	0.677	0.604	0.283/0.082/0.218	0.690	<u>0.332</u>	0.680
GPT Neo	0.610	0.492	0.231/0.045/0.171	0.621	0.245	0.665
AR-Diffusion	0.647	0.465	0.268/0.059/0.206	0.568	0.185	0.569
DiffuSeq	0.679	0.475	0.189/ 0.130 /0.136	0.588	0.186	0.563
SeqDiffuSeq	0.688	0.486	0.286/0.067/0.213	0.617	0.194	0.574
TESS	<u>0.694</u>	<u>0.587</u>	0.317/ <u>0.116</u> / 0.264	0.627	0.339	0.667
SEDD	0.666	0.001	0.200/0.033/0.138	0.576	0.086	0.443
LD4LG	0.708	0.580	0.303/0.100/0.246	0.708	0.211	0.641
TEncDM	0.619	0.496	<u>0.319</u> /0.107/0.253	<u>0.699</u>	0.323	<u>0.703</u>
COSMOS _{N=16}	0.649	0.497	—	—	—	—
COSMOS _{N=128}	<u>0.694</u>	0.554	0.328 /0.114/ <u>0.258</u>	0.694	0.339	0.708

D Limitations

There are several limitations to our study that point to promising directions for future work. First, jointly training the autoencoder and diffusion model remains an open research direction. Such an approach may significantly improve training efficiency. Second, the latent dimensionality d is held fixed in all experiments, because changing d would necessitate redesigning diffusion backbone; a systematic sweep of this hyper-parameter is therefore left to dedicated follow-up work. Finally, we report results with relatively small backbones of roughly 130M parameters to keep ablations rapid and compute budgets fair; scaling the architecture is an orthogonal engineering effort that is expected to reinforce the empirical trends observed here.

E Societal Impact

Our research introduces an alternative modeling approach for language, centered on latent continuous diffusion. We hold the view that this method does not bring about significant new societal risks exceeding those already connected with current language models.

F Generation examples

To give a qualitative sense of model behaviour, we first present representative unconditional generations: Table 10 shows random samples from COSMOS_{N=128} and the TEncDM baseline on ROCStories, while Table 11 does the same for Wikipedia. We then move to the conditional setting, providing sequence-to-sequence outputs for three benchmarks: summaries for XSum in Table 12, detoxified rewrites for ParaDetox in Table 14, and question-generation examples for SQuAD 2.0 in Table 13.

Table 10: Randomly generated samples for ROCStories dataset.

TEncDM	COSMOS _{N=128}
Amy wanted to buy a new dress for the dance! She shopped around for ten minutes. She tried on a dress that was too big. Amy loved that the dress was too grab for her size. She found that it was a little too long to wear!	Amy and her friend Sue were excited about sixth grade together. The three of them visited Amy's online shoe store. Amy's friend Beth arrived at the shoe store and Sue tried some dresses. The girls checked out and looked at all the collection. Amy felt like she was being an idiot.
Maggie was a nasty girl in school. When she moved to a small town, she had no Mae friends. After school, sheared out and made no friends. Sometimes they called her back, and they left her alone. She finally understood why she needed to be friends.	Jay was sitting at home bored. He was looking for an activity to keep him active. He decided to play a game of basketball. He played basketball for an hour. He was able to have a fun time that afternoon.
My car was becoming very worn out. I went to my mechanic to get checked out. He told me that I had a flat tire. I went to NCaven to get it fixed. I still made it to work in no time.	Sam was working late. He didn't have money to pay all his bills. He was having trouble getting paid. Sam decided to quit his work job. He was able to get another job to pay his bills.
My daughter is moving to a new school next year. She was nervous about moving to this new location. I am afraid of a leaker of all of her relationships. We figured it out and made some alterations on the team. This is where she will be graduating high school.	Debra hasn't written a check in weeks. She's been very stressed and frustrated with paying bills. Last week, she acted horribly during her check. The bank quickly ran a check - up. To her shock, she found out that she owed nearly a million in cash.
John went to the library. The librarian told him to read more books. John went to the booktore. He tried to read 6 books. He couldn't decide which to buy.	Tina never thought she needed her son to be cool enough to swim. So she begged him to do so while he was supposed to. So he practiced and swam all the time. But when it was time to go outside, he constantly got sore. Tina was too stressed to let him swim for 3 hours.

Table 11: Randomly generated samples for the Wikipedia dataset using COSMOS_{N=128}.

The press said working conditions for the match did not have to be finalised. Even the organizers had planned for the date of the match to be 19 August 2017 to accommodate a crowd of some 500, 000 people watching the match.
Bob Hall served as Foundation founding president from 1990 to 1991. In 1993, Hansen was elected President of the College of Bymphoschipegeons. Upon his retirement from this position, he also served as the chairman of the Scientific Advisory Committee of the College of Medical Surgeonsgeons (IMS). In 1996, publisher magazine selected Bob Hall as his candidate for the next U. S. presidential nomination for 2000 and 2001. He died October 26, 2013, at his home in Tampa, Florida from the causes of kidney failure at the age of 71, caused by a cerebral kidney failure in 1980. He is buried at the
He was excluded from the committee because Johnson refused to approve the text of the bill, and two senators refused to attend discussing the legal provisionss of the Seventh Amendment, but invited senior officials to hear it. Both the Congress House and the Weed Hermans challenged the committee's request to have the Seventh Amendment be argued in the Supreme Court's decision during the subsequent legislative hearings. Both attributed the premise premisel of Johnson's Amendment to internal law, which allocated the Congress to adopt executive decrees beginning in January 1699. The law, for its part, was however applied by the majority of the
Folk music typically contains a mix of German, Russian, Russian, Iranian, Egyptian, Norwegian, Turkish, Jewish and Tabad music, though typically the songs are from a specific religion or background. Icelandmar00edk, for instance, distinguish genderly distinguishes folk music from the multi - diverse genres, primarily rock and jazz.. Folk music is consequently not subegoaticly distinct from rock, which in turn, which consists primarily of music of folk including jazz, blues, nor rock, and industrial rock. Due to the diversity of genres in these genres, certain new genres of music also exist. Pandit music, such as
In 1974, Parker became the Director of the Office of Virginia's Chamber of Trade and Commerce, a position he would hold under increased pressure from his fellow Democratic opponent. He oversaw the J. Howard administration, which lasted for 11 years headed by William J. P honorer.

Table 12: Samples from XSum summarization dataset. Parts of the articles are omitted for brevity.

<p>Article: Two-year-old Lane Thomas Graves had been playing in the sand near the resort's Seven Seas Lagoon when he was dragged underwater by the creature... The lighthouse has been installed near to where the attack occurred... A Disney spokesperson said they hoped the monument would spread awareness for the Lane Thomas Foundation, which also uses the lighthouse as its logo. Who is liable for alligator boy's death? "The lighthouse sculpture has been installed to help spread awareness of the Lane Thomas Foundation, which was established to provide assistance and support to families whose children need organ transplants," Walt Disney World said in a statement.</p>	
<p>Reference: Walt Disney World has unveiled a lighthouse memorial for a young boy who was killed by an alligator while on holiday at the Florida theme park.</p>	
TEncDM	COSMOS _{N=128}
A statue of a Florida boy who died after being rescued from a beachquarina during water has been honoured by US television provider Disney.	Walt Disney Disney has unveiled a lighthouse statue in memory of a young boy who died when he was stabbed by an alligator at a Florida resort in Florida.
<p>Article: The Sky Blues currently play in Coventry's Ricoh Arena but had a long dispute with the stadium's previous owners... In a statement, Rugby Borough Council said its leader and the council's executive director and head of planning had met with Coventry City in March. "The club requested the meeting to understand how the council would deal with any planning application for potential stadium sites in the borough of Rugby," it said. It said the plans would need to be finalised by September to be included in the council's local plan, but added that a site had yet to be identified. Peter Ward, from Sky Blues Supporters' Consultative Group, said he was pleased to hear that things were "moving" with the club's search for a new home. "It's good that finally there is some evidence things</p>	
<p>Reference: Planners in Rugby have revealed they have been in talks with Coventry City Football Club about building a stadium in the borough.</p>	
TEncDM	COSMOS _{N=128}
Coventry City Council says it is looking on whether potential plans for a 00a33m Super League stadium in Coventry.	Coventry City fans say they will meet with the city council over a proposed move from the club club to a new stadium.

Table 13: Samples from SQuAD2.0 question generation dataset.

<p>Context: Temporal measurement has occupied scientists and technologists, and was a prime motivation in navigation and astronomy. Periodic events and periodic motion have long served as standards for units of time. Examples include the apparent motion of the sun across the sky, the phases of the moon, the swing of a pendulum, and the beat of a heart. Currently, the international unit of time, the second, is defined by measuring the electronic transition frequency of caesium atoms (see below). Time is also of significant social importance, having economic value (time is money) as well as personal value, due to an awareness of the limited time in each day and in human life spans.</p>	
<p>Answer: Temporal measurement.</p>	
<p>Question: What has been a prime motivation in astronomy and navigation?</p>	
TEncDM	COSMOS _{N=128}
What has a significant role in surveying units of time?	What have technologists filled time in astronomy?
<p>Context: In Book 11 of his Confessions, St. Augustine of Hippo ruminates on the nature of time, asking, "What then is time? If no one asks me, I know: if I wish to explain it to one that asketh, I know not. He begins to define time by what it is not rather than what it is, an approach similar to that taken in other negative definitions. However, Augustine ends up calling time a 201cdistention201d of the mind (Confessions 11.26) by which we simultaneously grasp the past in memory, the present by attention, and the future by expectation.</p>	
<p>Answer: St. Augustine of Hippo.</p>	
<p>Question: Who commented on the nature of time in Book 11 of his confessions?.</p>	
TEncDM	COSMOS _{N=128}
Who wrote " " nature of timefessions ?	Which philosopher ruminates about explaining the nature of time?

Table 14: Random samples for the detoxification task on the ParaDetox dataset.

Input: fucking imagine obama just put the hands up and keep a good distance between he and cruz .	
Reference: Imagine Obama just put the hands up and distanced himself with Cruz.	
TEncDM	COSMOS_{N=128}
2018 obama just put the hands up and keep a good distance between him and cruz.	imagine obama just put the hands up and keep a good distance between he and cruz.
Input: you just summed up how fucking stupid politics is in one comment	
Reference: You just summed up how politics is in one comment.	
TEncDM	COSMOS_{N=128}
you just sum up up how bad politics is in one comment	you just sum up up how how stupid politics is in one comment
Input: only churches give a crap it seems.	
Reference: only churches care it seems.	
TEncDM	COSMOS_{N=128}
Only churches care bad it it seems..	only churches give a thing it seems.
Input: second , maybe if you douches would read the damn edit summaries , this wouldnt happen .	
Reference: second , maybe if you would read the edit summaries , this wouldnt happen .	
TEncDM	COSMOS_{N=128}
201d, maybe if you would read the editmaries.	If you read the edit summaries, this wouldn't happen.
Input: what we witnessed here today was a fucking miracle, and i want you to acknowledge that.	
Reference: What we witnessed here today was a miracle, and I want you to acknowledge that.	
TEncDM	COSMOS_{N=128}
What we witnessed here today was a miracle, and I want you to acknowl- edge that.	what we witnessed here today was a a miracle, and i want to acknowl- edge that.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We claim that COSMOS learns a compressed and smooth latent representation through carefully designed autoencoder objectives, enabling high-quality text generation with reduced inference time. Empirically, COSMOS matches or surpasses traditional token-level diffusion and autoregressive baselines across a diverse set of generative tasks.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The discussion of limitations is located in the Appendix D.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not have theorems in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We thoroughly describe our method in Section 4.2.2 and provide a complete set of hyperparameters in Appendix B. We use publicly available datasets and will make our code publicly available upon publication.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use publicly available datasets. We add code to the supplementary materials and will make our code publicly available upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All details could be found in Appendix A and Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Following prior work, we do not report error bars for comparison results across all tasks.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: These details could be found in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our research does not involve human participants and does not introduce any new datasets; all data used are publicly available.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss societal impact in Appendix E.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: We do not release any pretrained models or datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We provide all the information about the datasets in Appendix A. We also cite the BERT paper whose representations we use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We release only our source code which we thoroughly comment.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: Our study does not rely on crowdsourced data and does not involve any research with human participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: Our study does not involve crowdsourcing or any research involving human participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- 967 • Depending on the country in which research is conducted, IRB approval (or equivalent)
968 may be required for any human subjects research. If you obtained IRB approval, you
969 should clearly state this in the paper.
- 970 • We recognize that the procedures for this may vary significantly between institutions
971 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
972 guidelines for their institution.
- 973 • For initial submissions, do not include any information that would break anonymity (if
974 applicable), such as the institution conducting the review.

975 **16. Declaration of LLM usage**

976 Question: Does the paper describe the usage of LLMs if it is an important, original, or
977 non-standard component of the core methods in this research? Note that if the LLM is used
978 only for writing, editing, or formatting purposes and does not impact the core methodology,
979 scientific rigorousness, or originality of the research, declaration is not required.

980 Answer: [NA]

981 Justification: LLMs are not utilized in any of the scenarios described.

982 Guidelines:

- 983 • The answer NA means that the core method development in this research does not
984 involve LLMs as any important, original, or non-standard components.
- 985 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
986 for what should or should not be described.