REPURPOSING PRE-TRAINED LLMS AS HIGH FIDELITY CONTINUOUS TEXT AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

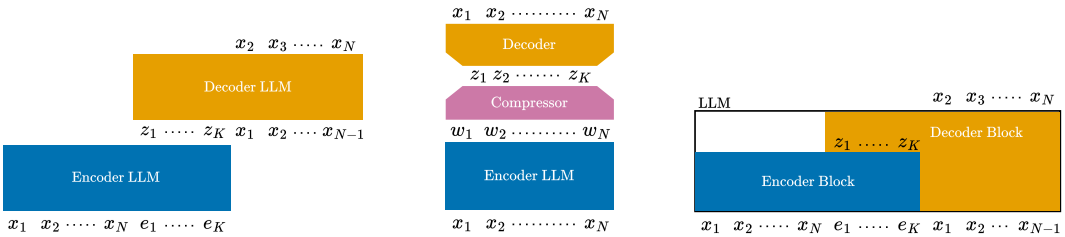
Next-token prediction has enabled highly fluent autoregressive language models, but it represents global structure only indirectly through sequential factorization. In contrast, high-fidelity autoencoders have become a standard primitive in image generation, enabling generative models to operate over continuous latent spaces; text lacks a comparably faithful continuous representation. We propose LLMAE, a method for repurposing a pretrained decoder-only language model as a continuous text autoencoder by exposing the activations of an intermediate layer as a fixed-length latent bottleneck. LLMAE achieves this interface with structured attention masks and LoRA adaptation, leveraging the generative prior of the original LLM. Across two backbones (270M Gemma 3 and 0.5B Qwen2.5), LLMAE reconstructs sequences up to 1024 tokens with high accuracy, reproducing up to 97% of documents verbatim. We demonstrate downstream utility by training a lightweight diffusion model that generates detailed image captions directly in the frozen LLMAE latent space. By mapping text into a fixed-length continuous latent space, our approach provides an effective substrate for downstream adaptation while benefiting from the fluency of the original LLM.

1 INTRODUCTION

Autoregressive LLMs dominate text generation, demonstrating remarkable efficacy and fluency, in part by capitalizing on the inherently sequential nature of language. However, this strict token-by-token factorization necessitates local decoding decisions, fundamentally constraining the model's capacity for global foresight and planning (Yao et al., 2023). Furthermore, downstream adaptation remains expensive and non-trivial (Hu et al., 2022), as traditional fine-tuning often induces "catastrophic forgetting" (Kotha et al., 2023) of the LLM's original generative prior (Holtzman et al., 2019). Non-autoregressive (NAR) generation has been studied to circumvent this sequential bottleneck, yet these methods often struggle to model word-order dependencies explicitly, leading to text that lacks the grammatical correctness and fluency characteristic of autoregressive models (Li et al., 2023). A prominent framework for NAR generation is text diffusion, however, token-level diffusion remains non-trivial, requiring specific pre-training objectives and carefully designed training/sampling procedures (Lou et al., 2023; Dieleman et al., 2022).

Drawing inspiration from the visual domain, where high-fidelity autoencoders are a standard primitive (Rombach et al., 2022), we propose LLMAE (LLM AutoEncoder), a faithful text autoencoder that maps sequences of any length into a fixed-length continuous representation. Non-autoregressive generative models are built to operate on fixed-shape tensors, and freezing the autoencoder enables easier downstream modeling without the cost or catastrophic forgetting of token-level fine-tuning. LLMAE partitions a single lightweight LLM into an integrated encoder and decoder (Fig. 1c) by utilizing structured attention masks and utilizing LoRA (Hu et al., 2022) for lightweight fine-tuning. Specifically, activations at an intermediate transformer layer aggregate abstract information to serve as latent encodings, while subsequent layers function as a fluent decoder. This enables high-fidelity reconstruction while benefiting from the fluency of the original model. We also propose an optional single-layer, identity-initialized additive codec over the latent sequence, which learns a locally smooth latent interface under KL regularization while improving the reconstruction quality further. We instantiate LLMAE on two lightweight backbones from different model families, Gemma-3-270M (Team et al., 2025) and Qwen2.5-0.5B (Qwen Team, 2024), training only 10.7M of 278.8M and 23.8M of 494M parameters. Our main contributions in this paper are the following:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Soft Prompting (b) Token Compressor (c) LLMAE (Ours)

Figure 1: **Related LLM-based autoencoding architectures.** (a) Soft prompting: an LLM encodes input tokens $x_1 \dots x_N$ with learnable query tokens $e_1 \dots e_K$; its activations at e form the latent $z_1 \dots z_K$, an input for a separate LLM. (b) Token compressor: output token embeddings are compressed into $z_1 \dots z_K$ by a separate module. (c) LLMAE (ours): LLM split by attention masks into an Encoder Block and a Decoder Block; $z_1 \dots z_K$ are the activations of e at an intermediate layer.

LLMAE Framework: We introduce an approach for repurposing pre-trained LLMs as text autoencoders by establishing a fixed-length continuous latent bottleneck at an intermediate layer.

Reconstruction Analysis: We demonstrate near-perfect word-level fidelity on text sequences up to 1024 tokens long ($\approx$ 4.6K characters or 760 words), with up to 97.4% of documents reproduced verbatim under strict exact match, significantly outperforming previous text autoencoders. We provide detailed ablations of latent depth, latent capacity, and the KL/codec components for two backbones, identifying the intermediate-layer bottleneck as the core enabling factor and confirming the "entropy valley" (Skean et al., 2025) as its best depth.

Downstream Generative Utility: We train a latent text diffusion model for detailed image captioning on frozen LLMAE latents. Our lightweight diffusion model (111M trainable parameters) generates fluent, high-quality captions. We remain behind frontier VLMs, which fine-tune multi-billion-parameter backbones on far larger corpora, but outperform some substantially larger autoregressive captioners such as LLaVA-1.5-7B (Liu et al., 2024) and Share-Captioner (Chen et al., 2024).

2 Related Work

We discuss lines of work related to this paper: context compression in LLMs, text autoencoding architectures, evidence for what different LLM layers encode, and diffusion modeling for text.

Context Compression. We establish a bottleneck in an intermediate layer of the LLM and use that bottleneck as the latent representation. This is similar to work on context compression, which adapts pre-trained LLMs to condense long prompts into compact representations. Early methods demonstrated that learnable continuous tokens, whether only at the input layer via Soft Prompting (Lester et al., 2021) or also integrated into intermediate layers via Prefix Tuning (Li & Liang, 2021), can compress context and steer model behavior. This concept evolved into structural techniques that cache LLM internal activations for compression, *e.g.*, Gist Tokens (Mu et al., 2023). The standard causal attention mask is modified so that query tokens can only attend as far back as the activations for the Gist tokens, forming a bottleneck that effectively represents context but has trouble scaling to longer sequences without architecture modification (Petrov et al., 2025). ICAE (Ge et al., 2023) fine-tunes an LLM to compress context into soft prompts that are decoded by a frozen LLM for reconstruction and task completion. Our experiments show that using activations of an intermediate layer provides better reconstruction quality and is achievable using a single lighter weight LLM.

Text Autoencoder Architectures. Approaches in related recent literature on text autoencoder architecture generally fall into two categories. The "soft prompt" approaches (Fig. 1a) can use a variety of models for encoder and decoder. ICAE (Ge et al., 2023) fine-tunes an LLM with LoRA as an encoder and uses a fixed LLM as the decoder. PLANNER (Zhang et al., 2023) takes the first K activations from the final layer of a fine-tuned BERT encoder as the latent and uses an autoregressive GPT-2 decoder to decode back to text.

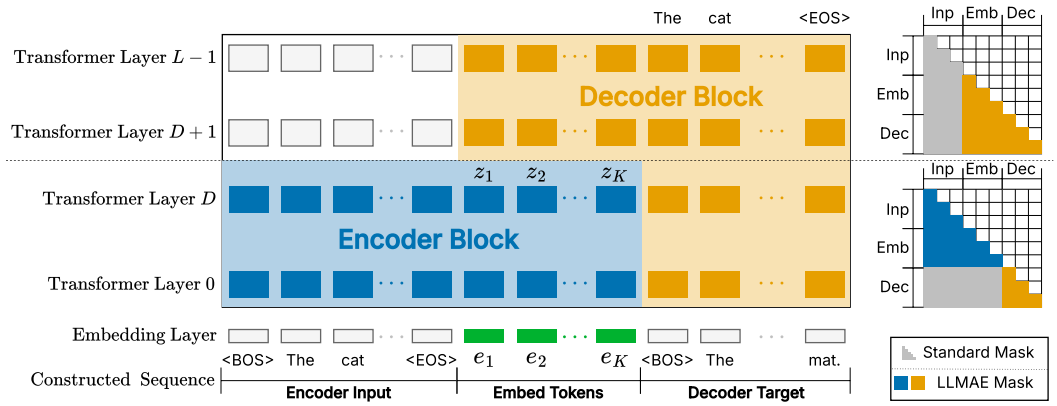


Figure 2: **Latent bottleneck via attention masking.** Tokens of one color form a causal segment attending only to the matching segment of the previous layer. Encoder block (blue): Embed tokens attend to Input. Decoder block (orange): Output tokens attend only to the latents z_i and previously decoded Output tokens. This establishes an autoencoding flow within an autoregressive backbone.

The other category of approach (Fig. 1b) compresses LLM output activations into latents for diffusion. LD4LG (Lovelace et al., 2023) repurposes pretrained encoder-decoder backbones (such as BART, Lewis et al., 2020) by utilizing a Perceiver (Jaegle et al., 2021) style query-based compression mechanism to gather information into a fixed-length latent sequence. These compressed tokens are then used to condition the autoregressive decoder for short-form text generation. COSMOS (Meshchaninov et al., 2025) follows a related approach, taking semantically rich hidden representations from a frozen BERT encoder and utilizing a Perceiver Resampler architecture (Alayrac et al., 2022) to compress the variable-length sequence into a fixed-size latent matrix. This fixed-size matrix is later expanded by a symmetric decompressor back into corresponding BERT tokens, which are then linearly projected into vocabulary probabilities for token prediction.

Our approach (Fig. 1c) partitions a single LLM for both encoding and decoding, simplifying modeling compared to soft-prompting, and uses the same LLM to reduce the sequence length instead of adding on a separate compressor and decoder.

LLM Representations. Research exploring the characteristics of LLM layers suggests the existence of an "entropy valley," where autoregressive models exhibit a distinct dip in entropy and a peak in representational accuracy at intermediate layers (Skean et al., 2025; Lad et al., 2025). LLMAE builds on this finding by placing its bottleneck at an intermediate layer, and our depth ablation (Table 3) supports it: deep intermediate layers carry the information a bottleneck needs in an accessible form.

Text Diffusion. Diffusion for text is in part inspired by the success of Diffusion models (Ho et al., 2020) for images where it often operates on continuous latent representations obtained from pretrained autoencoders (Rombach et al., 2022), where recent work shows potential advantages for more semantically grounded encoders (Zheng et al., 2025). Text diffusion can work via discrete noising processes on tokens, (Shi et al., 2024; Nie et al., 2025; Von Rütte et al., 2025; Lou et al., 2023), by Gaussian noise on continuous token embeddings (Li et al., 2022; Dieleman et al., 2022; Shabalin et al., 2025), or by Gaussian noise on latent embeddings that encode multi-token or long-form text (Lovelace et al., 2023; Zhang et al., 2023; Lovelace et al., 2024; Meshchaninov et al., 2025). Diffused latent embeddings are commonly decoded autoregressively for fluent output, but most such work has only targeted short texts. Challenges include achieving low latent-to-text conversion error and smoothness in the latent distribution as highlighted in (Zhang et al., 2023; Meshchaninov et al., 2025). Learning latent representations that can effectively encode long passages and work for diffusion is still an open challenge (Arriola et al., 2025) that we address in this paper.

3 Method

LLMAE turns a pretrained LLM into a text autoencoder. Its input is a text of up to N tokens; its output is a fixed-length latent $\mathbf{z}$ of K vectors, which the same LLM decodes back into the text. To obtain $\mathbf{z}$, we append K learnable "Embed" tokens to the input and read their activations at an

intermediate layer D. A modified attention mask splits the network there: the layers up to D serve as the encoder, and the layers after D serve as the decoder, which can attend to $\mathbf{z}$ but to nothing else about the input, so the reconstruction must pass through the latent (Fig. 2). The LLM is adapted with LoRA, and a small additive codec refines $\mathbf{z}$. The rest of this section describes the model, the codec, the training objective, and the training details.

Model Details. We partition the depth L of the model into two functional blocks: an Encoder Block ($0 \leq l \leq D$) and a Decoder Block ($D < l \leq L-1$). At the transition layer D, the activations corresponding to the K sequence positions where the learnable Embed tokens e were placed define the latent representation $\mathbf{z} \in \mathbb{R}^{K \times d}$. Therefore, $\mathbf{z}$ represents the transformation of the original Embed tokens through the depth of the Encoder Block. At layer $D+1$ within the subsequent Decoder Block, the masking constraints are relaxed to allow the Output tokens to attend to the resulting latent states $\mathbf{z}$ while maintaining a strict mask against the original Input sequence. Since the Output tokens always remain isolated from the Input tokens, this configuration ensures that the reconstruction process is channeled through the latent bottleneck. For more details, we specify the attention mask modification in Table 8 in App. C. Formally, this partition can be expressed as:

$$\mathbf{z} = \text{Encoder}(\mathbf{x});\ \hat{\mathbf{x}} = \text{Decoder}(\mathbf{z}, \hat{\mathbf{x}}_{<i}),$$

where $\hat{\mathbf{x}}_{<i}$ denotes the tokens already decoded. For encoding and downstream modeling, $\mathbf{z}$ can be cached as the latent representation of the input. At inference time for decoding, we inject the latent $\mathbf{z}$ as input to layer $D+1$ corresponding to the Embed tokens, resulting in an inference-time complexity comparable to standard LLM decoding.

Additive Codec Refinement. LoRA adapts the backbone through a low-rank update to its attention projections in every layer. This keeps adaptation cheap and steers the generative prior, but it proves challenging to transform the latent's coordinate statistics. That matters on Gemma, whose raw latent inherits the backbone's activation geometry, with a few coordinates orders of magnitude larger than the rest (Table 2). We therefore introduce a small codec that acts on the latent alone. It consists of single-layer Transformer blocks ($\mathcal{T}_{\text{enc}}, \mathcal{T}_{\text{dec}}$) with per-dimension learnable scalar multipliers ($\boldsymbol{\alpha} \in \mathbb{R}^d$) initialized to zero, so that it starts as the identity and learns only a residual correction:

$$\mathbf{z}_{\text{codec}} = \mathbf{z} + \boldsymbol{\alpha}_{\text{enc}} \odot \mathcal{T}_{\text{enc}}(\mathbf{z}); \quad \hat{\mathbf{z}} = \mathbf{z}_{\text{codec}} + \boldsymbol{\alpha}_{\text{dec}} \odot \mathcal{T}_{\text{dec}}(\mathbf{z}_{\text{codec}})$$

The multipliers are shared across all K tokens but learned per hidden dimension d. We enforce the replacement $\mathbf{z} \leftarrow \hat{\mathbf{z}}$ to ensure the decoder conditions on the refined latent, stashing the detached $\mathbf{z}_{\text{codec}}$ as the definitive latent representation for KL regularization during training and downstream inference. Because the codec is additive by design, we finetune it jointly over the LLMAE backbone.

3.1 Training Objectives

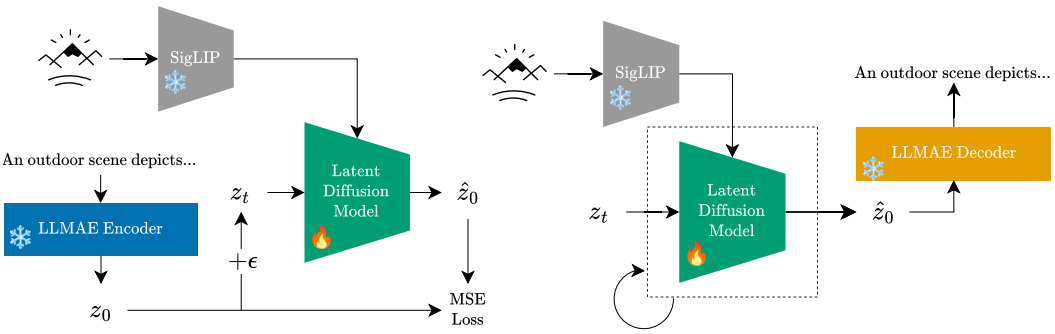
The reconstruction task is formulated by constructing a composite sequence $\mathbf{s}$ that concatenates the input source, the introduced Embed tokens, and the reconstruction target. Let $\mathbf{x} = \{x_1, \ldots, x_N\}$ denote the input sequence and $\mathbf{e} = \{e_1, \ldots, e_K\}$ represent the learnable Embed tokens. Shorter sequences are padded to reach the N token limit. The full training sequence is constructed as:

$$\mathbf{s} = [\mathbf{x}_{\text{enc}}] \parallel [\mathbf{e}] \parallel [\mathbf{x}_{\text{dec}}]$$

where $\mathbf{x}_{\text{enc}}$ and $\mathbf{x}_{\text{dec}}$ are identical sequences of tokens $x_i \in \mathcal{V}$. The total loss is a weighted sum:

$$\mathcal{L}_{\text{LLMAE}} = \mathcal{L}_{\text{NTP}} + \lambda_z \mathcal{L}_z + \lambda_{\text{ref}} \mathcal{L}_{\text{ref}},$$

where $\mathcal{L}_{\text{NTP}}$ is the next-token prediction cross-entropy loss of the decoder over the N reconstruction positions $\mathbf{x}_{\text{dec}}$; $\mathcal{L}_z = D_{\text{KL}}\big(q(\mathbf{z}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big)$ is the KL divergence of the latent from a standard normal prior, with q a Gaussian fitted per latent coordinate across the batch (Eq. 1 in App. D); $\mathcal{L}_{\text{ref}} = D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)$ is the KL divergence of the adapted model's next-token distribution π_θ from that of the frozen pretrained backbone LLM π_{ref}, before any adaptation, summed over the full vocabulary at each reconstruction position; and λ_z, λ_{ref} are hyperparameters. In the codec stage, $\mathcal{L}_z$ is applied to $\mathbf{z}_{\text{codec}}$. Gradients of $\mathcal{L}_{\text{NTP}}$ flow back through the decoding layers into the latent and through the entire encoding block and the Embed tokens. The latent prior term regularizes the latent representations into a smooth manifold. The reference term encourages linguistic fluency by anchoring the LLMAE decoder to the pretrained backbone; π_{ref} is evaluated with a forward pass on the target output tokens alone, without the input or the Embed tokens.



(a) Training (b) Inference

Figure 3: **Latent Diffusion for Captioning.** A lightweight denoising score network conditioned on frozen SigLIP image features through cross-attention. LLMAE remains frozen throughout.

3.2 Training Details

We employ $N{=}1024$ input tokens and $K{=}256$ Embed tokens on both backbones. On Gemma we found it helpful to adopt a curriculum schedule, sorting samples by ascending length to stabilize early convergence before longer reconstruction; Qwen instead trains best on shuffled data (App. F). We trained on 8 NVIDIA RTX PRO 6000 Blackwell GPUs; the full training takes $\approx$ 22 hours. Full hyperparameters are listed in App. K.

Training Data. We train on a curated corpus of 1.4M samples drawn from two sources: 1M from Pile-uncopyrighted (Biderman et al., 2023) for general web text and 400K from C4-RealNewsLike (Raffel et al., 2020) for higher quality prose. We collect each source stratified by length for a balanced training set: documents are binned into 41 buckets of 100 characters.

4 Experiments

Our experiments focus on three core objectives: reconstruction fidelity, representational smoothness, and downstream generative utility. We first evaluate LLMAE's autoencoding performance against existing baselines (Sec. 4.1.4). We then analyze the distribution of the latent space (Sec. 4.2), evaluating the impact of KL regularization and the additive codec, before conducting systematic ablations on the latent bottleneck (Sec. 4.3). Finally, we demonstrate the representation's utility (Sec. 4.4) by training a latent diffusion model for detailed image captioning.

4.1 Reconstruction Quality

4.1.1 Baselines

We compare against two publicly released text autoencoders: ICAE (Ge et al., 2023), using its `v2` Stage 1 checkpoint for autoencoding, and COSMOS (Meshchaninov et al., 2025), whose released checkpoint encodes up to 512 tokens into 512 latent tokens (no sequence-length reduction) and was not trained on longer inputs.

4.1.2 Evaluation Datasets

We evaluate on three held-out test sets of samples drawn from different text domains to assess generalization across writing styles and content types. All three sources are disjoint from the training corpus: C4-News-Stratified is deduplicated by content hash against all 1.4M training samples, OpenWebText uses the held-out test split shared by COSMOS, and the CreationMMBench references are GPT-4o-generated text from an external benchmark. The full construction protocol is detailed in App. K. Since the COSMOS baseline (Meshchaninov et al., 2025) is limited to a 512-token length, we report results on two test sets per source in Table 1: a medium-length set (up to 512 tokens) and a long-length set (up to 1024 tokens). For calibration, 1024 tokens corresponds to roughly 4,600 characters or 760 words of English prose. Each setting is evaluated on 500 held-out samples, with

Table 1: Reconstruction by domain and length (top: up to 512 tokens; bottom: up to 1024). BLEU-4↑, exact match (EM)↑, word edit distance (W-ED)↓; definitions in App. J.

			C4-News-Stratified			OpenWebText			CreationMMBench		
Method	**Latent**	**#Par.**	**BLEU**	**EM**	**W-ED**	**BLEU**	**EM**	**W-ED**	**BLEU**	**EM**	**W-ED**
ICAE	128×4096	7.46B	0.831	24.2%	14.1%	0.711	22.0%	26.5%	0.859	18.6%	10.9%
COSMOS	512×768	356.5M	0.783	6.0%	17.3%	0.773	3.6%	19.4%	0.712	0.0%	26.7%
LLMAE (Ours)											
Qwen	256×896	494M	**0.999**	**96.6%**	0.12%	**0.997**	**93.8%**	**0.17%**	**0.998**	**92.6%**	**0.15%**
Gemma	256×640	278.8M	**0.999**	92.6%	**0.05%**	0.995	80.6%	0.58%	0.994	83.9%	0.81%
ICAE	256×4096	7.46B	0.710	13.4%	25.8%	0.602	8.0%	41.3%	0.744	8.8%	22.4%
LLMAE (Ours)											
Qwen	256×896	494M	**1.000**	**97.4%**	**0.005%**	**0.998**	**93.2%**	**0.17%**	**0.999**	**82.4%**	**0.10%**
Gemma	256×640	278.8M	0.995	70.6%	0.26%	0.988	67.4%	1.8%	0.991	68.8%	1.5%

the exception of CreationMMBench at 512 tokens (311 samples, the subset of the benchmark that fits the limit).

C4-News-Stratified. We sample from the C4 (Raffel et al., 2020) corpus with the *RealNewsLike* filter for news-like domains, stratified into the same 41 character-length buckets used for training so that the evaluation spans short through full-length inputs evenly; we refer to this set as C4-News-Stratified. Besides Table 1, this is the test set used to measure reconstruction quality in this paper.

OpenWebText. OpenWebText (Gokaslan et al., 2019) is an open-source reproduction of the WebText dataset used to train GPT-2. We sample from the specific held-out test split shared by COSMOS (Meshchaninov et al., 2025), which allows us to evaluate our model on the distribution used to train the COSMOS autoencoder checkpoint.

CreationMMBench. CreationMMBench (Fang et al., 2025) offers a test benchmark of image-grounded long-form text generation tasks spanning four categories: Literary Writing, Common Functional Writing, Professional Functional Writing, and Creative Multimodal Understanding. We evaluate on the `reference_answer_by_gpt4o` field from the benchmark dataset, generated by GPT-4o.

4.1.3 METRICS

LLMAE is intended as a near-lossless interface: downstream models consume the latent in place of the text. We therefore report exact match (EM) and word edit distance (W-ED) as direct fidelity measures, alongside BLEU-4 for comparability with prior autoencoding work. BLEU-4 above 0.99 leaves little headroom and does not separate a perfect reconstruction from one with a single wrong word, whereas EM is all-or-nothing over the whole document. W-ED measures how far a miss actually is. All three operate on whitespace-delimited words, making them independent of any model-specific subword tokenizer and therefore directly comparable across the Gemma- and Qwen-based LLMAE instantiations, the Mistral-based ICAE, and the BERT-based COSMOS. We report additional similarity and perplexity metrics in App. J, along with formal metric definitions.

4.1.4 RECONSTRUCTION PERFORMANCE

We demonstrate near-perfect reconstruction in Table 1, maintaining BLEU-4 scores near or exceeding 0.99, significantly outperforming both ICAE (Ge et al., 2023) and COSMOS (Meshchaninov et al., 2025). Beyond BLEU, LLMAE reproduces the majority of documents *verbatim* (EM up to 97.4% at 1024 tokens) with word edit distances two orders of magnitude below the baselines. We observe that ICAE's reconstruction quality degrades as sequence length increases, with its BLEU score dropping from 0.831 to 0.710 on the C4-News-Stratified split. This demonstrates that LLMAE provides a more robust bottleneck than the soft-memory compression used in ICAE, despite being an

order of magnitude more parameter-efficient (278.8M vs. 7.46B parameters). Of those 278.8M parameters only 10.7M are trained on Gemma: 737K LoRA adapters, 164K latent token embeddings, and a 9.8M additive codec, so the codec accounts for most of the trainable budget. Notably, we also attempted to train an ICAE-style autoencoder using the lightweight Gemma backbone used in LLMAE, but were unable to achieve high-quality reconstruction (Appendix I), suggesting that simply applying soft-prompt compression to smaller models is insufficient for high quality reconstruction. Component-wise ablations of the KL regularization and the additive codec are reported in Table 2: on Gemma the 9.8M-parameter codec closes much of the remaining gap to perfect reconstruction, while on Qwen the bottleneck alone is already near-lossless but is closed further.

4.2 Latent Space Distribution

Table 2: Latent statistics and reconstruction across the training stages of both backbones.

	Latent distribution				Reconstruction		
Stage	**Mean**	**Std**	**Max**	D_{KL} (↓)	**BLEU**	**EM**	**W-ED**
Gemma-270M, NTP only	7.068	277.61	7296.0	38556.0	0.978	53.6%	1.8%
+ KL	0.164	8.58	1328.0	35.36	0.967	29.8%	2.2%
+ KL + Codec	-0.038	5.08	236.0	**11.69**	**0.995**	**70.6%**	**0.26%**
Qwen2.5-0.5B, NTP only	0.044	2.21	24.4	2.39	0.999	92.8%	0.07%
+ KL	-0.008	1.04	11.6	0.22	1.000	96.8%	0.008%
+ KL + Codec	-0.007	0.97	9.1	**0.12**	**1.000**	**97.4%**	**0.005%**

We evaluate the distribution statistics of the latent space $Z \in \mathbb{R}^{M\times K\times d}$, where $M = 500$ is the number of test samples, $K = 256$ is the latent token length, and d is the latent dimension (640 for Gemma, 896 for Qwen). We report the latent distributional statistics in Table 2 by computing global empirical moments mean, standard deviation, and maximum absolute value, pooled across all n, t, d indices. The rows are the three training stages: $\mathcal{L}_{\text{NTP}}$ alone; $\mathcal{L}_z$ and $\mathcal{L}_{\text{ref}}$ added with $\mathcal{L}_z$ on the raw latent $\mathbf{z}$ (+ KL); and the codec added with $\mathcal{L}_z$ moved to $\mathbf{z}_{\text{codec}}$ (+ KL + Codec), whose row reports the statistics of $\mathbf{z}_{\text{codec}}$. We observe that default (NTP-only) training activations exhibit highly irregular distributions on Gemma; KL regularization and the codec normalize these statistics effectively, alongside a net improvement in reconstruction. On Gemma, applying the KL to the raw latent alone trades exact-match fidelity (EM $53.6\% \rightarrow 29.8\%$), which the codec recovers and surpasses (70.6%). The two backbones differ sharply in raw latent geometry: Gemma's stage-1 activations are dominated by a few extreme outlier coordinates (Std 277, Max 7296) which are normalized effectively utilizing KL regularization and the codec, whereas Qwen's are already near-unit-scale (Std 2.2).

Table 3: Reconstruction vs. latent layer depth.

Depth	**BLEU** (↑)	**EM** (↑)	**W-ED** (↓)
Gemma-270M (18 layers)			
2	0.001	0.0%	274%
5	0.589	14.8%	87.8%
8	0.702	15.0%	55.6%
11	**0.978**	**53.6%**	**1.8%**
14	0.222	0.0%	279%
Qwen2.5-0.5B (24 layers)			
3	0.012	0.0%	322%
7	0.873	13.6%	97.7%
11	0.998	90.4%	0.11%
15	**0.999**	**92.8%**	**0.07%**
19	0.635	9.4%	61.3%

Table 4: Reconstruction vs. latent token count.

#Tokens	**BLEU** (↑)	**EM** (↑)	**W-ED** (↓)
Gemma-270M			
64	0.376	8.4%	163%
128	0.216	1.8%	434%
256	**0.978**	53.6%	**1.8%**
512	0.958	54.6%	7.9%
1024	0.916	**70.0%**	84.5%
Qwen2.5-0.5B			
64	0.397	0.0%	260%
128	0.931	34.4%	5.1%
256	0.999	92.8%	0.07%
512	**1.000**	**98.6%**	0.005%
1024	**1.000**	**98.6%**	**0.003%**

4.3 Latent Bottleneck Ablations

The ablations in Tables 3 and 4 are run at the NTP-only stage of Table 2, without the latent KL or the codec, so that each varied factor is isolated, and use each backbone's own data ordering (curriculum on Gemma, shuffled on Qwen). We systematically vary the intermediate latent depth D and the

Table 5: Detailed image captioning. Our model (w/ Gemma) uses only 111M trainable parameters.

Method	# Params	Length	VLM (↑)	CapArena-Auto (↑)	RefCLIP (↑)	PPL (↓)
Autoregressive Methods						
Gemini-1.5-Pro	–	166.5	5.90	56.17	0.454	19.8
LLaVA-1.5-7B	7.1B	74.7	2.89	-94.00	0.439	10.9
Qwen2-VL-2B	2.2B	117.4	4.64	-48.67	0.454	14.0
VLV	4.2B	217.6	4.30	-38.17	0.452	9.6
Share-Captioner	8.4B	176.5	3.34	-74.83	0.439	10.6
Latent Diffusion: on frozen latent space with frozen SigLIP for image conditioning						
w/ Gemma LLMAE	819M	164.8	3.44	-62.00	0.446	26.0
w/ Qwen2.5 LLMAE	1.1B	163.6	3.06	-50.50	0.449	24.0
w/ COSMOS	943M	161.5	2.24	-88.33	0.440	48.5

number of latent tokens K on both backbones. As shown in Table 3, we observe a clear "entropy valley" effect that replicates across backbones: reconstruction peaks at a deep intermediate layer ($\sim 2/3$ depth: layer 11 of 18 on Gemma; layer 15 of 24 on Qwen), whereas shallow and final-stage layers are not effective. We also evaluate the bottleneck width K in Table 4. On Gemma, reconstruction is best at $K{=}256$ and degrades slightly at larger budgets; on Qwen, fidelity instead saturates, with $K{=}512$ reaching 98.6% exact-match and $K{=}1024$ adding nothing further, indicating the usable latent budget is backbone-dependent rather than universal. Low budgets ($K{=}64$) suffer information collapse on both (App. H). An input-level soft prompt on the same backbone and data reaches BLEU 0.393 at best (App. I), and an early-layer bottleneck does not work (0.001 at layer 2), whereas moving the read-out to a deep intermediate layer reaches 0.978. The LoRA-rank sweep is reported in App. G.

4.4 Downstream Utility: Latent Diffusion for Image Captioning

We demonstrate downstream utility on detailed image captioning, a multimodal task that remains largely unexplored via latent diffusion despite the paradigm's dominance in text-to-image synthesis (Rombach et al., 2022). Keeping both the visual and linguistic representations frozen (Fig. 3), we train only a lightweight diffusion model. By operating on a global latent representation, we can utilize the standard MSE denoising score matching objective:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{z}_0,\, t,\, \boldsymbol{\epsilon}}\Big[\big\|\hat{\mathbf{z}}_0(\mathbf{z}_t, t, I) - \mathbf{z}_0\big\|^2\Big],$$

where $\mathbf{z}_0$ is the LLMAE latent of the caption, $\mathbf{z}_t$ its noised version at diffusion time t with noise $\boldsymbol{\epsilon}$, $\hat{\mathbf{z}}_0(\cdot)$ the score network's prediction of the clean latent, and I the SigLIP features of the image. The score network is a 12-layer transformer, directly adapted from (Meshchaninov et al., 2025), details are provided in Appendix K. We train one diffusion model per backbone, on the $256{\times}640$ latents of the Gemma LLMAE and on the $256{\times}896$ latents of the Qwen2.5 LLMAE; in both cases the latents stay frozen and are decoded into text by the frozen LLMAE decoder during inference (App. B). For image conditioning, a linear layer projects SigLIP (Zhai et al., 2023) patch tokens to the score network width, fed via cross-attention between the score network intermediate layers and these projected visual features.

Training data and evaluation details: We use the VLV-6M dataset (Zhang et al., 2025) to train our latent diffusion model. This is a 6 million subset of LAION-Aesthetic with captions generated by Gemini-2-Flash. To assess the captioning performance, we evaluate our model on the *CapArena-Auto* benchmark (Cheng et al., 2025). CapArena is a human-preference benchmark for detailed captioning to establish a preference-based Elo ranking. CapArena-Auto serves as an automated proxy for this protocol; it utilizes a set of 600 test images from the DOCCI dataset (Onoe et al., 2024) and employs GPT-4o as a pairwise judge. This automated protocol was shown to achieve a 94.3% correlation with human model rankings while significantly reducing evaluation overhead (Cheng et al., 2025). We include four metrics to capture distinct measurements on the same 600-image test set used by CapArena:

- **CapArena-Auto Score:** The mean normalized win rate against 3 reference VLM captioners.
- **VLM Judge (0-6):** We adopt the system prompt with a 7-point rubric from (Zhang et al., 2025) (Fig. 5). This system comprehensively evaluates the coverage of image elements, spatial layout consistency, and the absence of hallucinations. We use `gpt-4o-2024-08-06` as the VLM judge.
- **RefCLIPScore:** The harmonic mean of CLIP similarities between (caption, image) and (caption, reference) using `OpenCLIP ViT-B/32`, capturing semantic alignment (Hessel et al., 2021).
- **Perplexity (PPL):** of generated captions, using GPT-2-Large, to measure the linguistic fluency.

We evaluate against three representative models from the CapArena leaderboard: Gemini-1.5-Pro (Team et al., 2024) is the best performer, LLaVA-1.5-7B (Liu et al., 2024) as a poor performer, and Qwen2-VL-2B (Wang et al., 2024) as a relatively lightweight VLM. We also compare against dedicated autoregressive captioning models, Share-Captioner (Chen et al., 2024) and VLV (Zhang et al., 2025). Notably, all models fine-tune the underlying LLM backbone, whereas our approach maintains a completely frozen LLMAE autoencoder and a frozen SigLIP image encoder.

With the Gemma LLMAE, our lightweight latent diffusion model achieves a VLM Judge score of 3.44, outperforming both the 7.1B LLaVA-1.5 (2.89) and the specialized Share-Captioner (3.34), while remaining behind the stronger VLM captioners (Qwen2-VL-2B, VLV, and Gemini-1.5-Pro). Generative reliability is high: no caption is marked as non-informative (score 0) and 87.5% achieve a score of 2 or higher, which designates a caption as "informative" with valid semantic grounding under the VLV rubric. Substituting the Qwen2.5 LLMAE under the same protocol improves the pairwise CapArena-Auto score, RefCLIPScore and perplexity, but lowers the VLM Judge score (81.0% of captions at 2 or higher, four scored 0), so the two backbones trade off between the pairwise and the rubric judge. Substituting our autoencoder with COSMOS (Meshchaninov et al., 2025), while maintaining the identical diffusion architecture and parameters, which itself is adapted from their original work, results in a significant performance degradation, confirming that the high-fidelity LLMAE autoencoder is better suited for this task. A representative caption scored $6/6$ by the VLM judge is shown in Fig. 6 in Appendix A, alongside a failure case; a granular analysis of the failure modes there reveals that the model demonstrates high fidelity in describing aesthetic scenes and atmospheric nuances, reflecting the training data, but fails on text and symbols in the image.

5 Limitations

Despite the strong performance and efficiency of LLMAE, limitations remain. First, due to using a fine-tuned LLM, LLMAE requires autoregressive sampling in decoding that causes sequential latency. Encoding is still parallelizable. Because the masking protocol respects causal ordering in each layer, the framework is compatible with standard KV caching (Pope et al., 2023). Second, we did not perform any experiments for sequence lengths longer than 1024 tokens of text, roughly 4.6K characters or 760 words. Finally, we observe a fundamental constraint involving the balance of autoencoding fidelity, sequence-length reduction, and latent smoothness. By prioritizing high-resolution, near-lossless reconstruction, LLMAE creates a dense latent manifold (App. E) that may require a strong conditional signal for generative modeling. Future research may help address some of these concerns without sacrificing the reconstruction quality.

6 Conclusion

We introduced LLMAE, a framework for repurposing pre-trained decoder-only language models as high-fidelity continuous text autoencoders with a fixed-length latent. What makes this work is *where* the attention-mask bottleneck is placed: activations read from a deep intermediate layer reconstruct documents of up to 1024 tokens almost verbatim. We utilize low-rank adaptation to perform parameter-efficient fine-tuning. The depth optimum sits near two-thirds of the stack on both backbones (layer 11 of 18 on Gemma, 15 of 24 on Qwen), an empirical confirmation of the "entropy valley" that gives a principled basis for extracting dense latents for text reconstruction. Furthermore, our results in latent text diffusion for detailed image captioning validate the downstream utility of this representation. Ultimately, LLMAE offers a parameter-efficient text autoencoder for grounding continuous generative models in the rich priors of LLMs, enabling flexible architectures for multimodal tasks and contributing to more structurally aware, globally consistent text generation.

AI USE STATEMENT

In this work, we used generative AI tools to design and provide feedback on the research methodology and experiments, to implement methods, to support qualitative analysis of samples, and to interpret results. We have not used generative AI tools to generate synthetic datasets, to propose or refine hypotheses, to assist with translation, or to clean and reformat datasets; the training captions are synthetic datasets published by prior work, as cited. Developing theoretical models or conceptual frameworks, formulating mathematical claims, providing ingredients for proofs and writing proofs are not applicable to this work. Additionally, we used generative AI tools to create and modify figures, suggest experimental parameters, create and edit software code, summarize and analyze existing literature, identify relevant literature and research gaps, brainstorm, search for information, draft sections of the appendix and edit the paper to improve readability. We have reviewed all AI-assisted work: code was generated carefully in a structured manner with AI, tested by the authors with empirical checks, every reported number was produced by our own runs or taken from the cited source, and AI-assisted literature summaries were checked against the original papers. We take responsibility for the final content of this work, including its originality, text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Architectural details and the full hyperparameters of the LLMAE autoencoder and of the latent diffusion captioning model are given in App. K (Tables 20 and 21), together with the construction protocol of the reconstruction test sets; the attention masks that define the bottleneck are specified in Table 8. The metric definitions are given in App. J, the VLM judge prompt in Fig. 5. All backbones, datasets and the image encoder are publicly available, and these materials are sufficient to replicate our experiments. In line with our commitment to open-source research, we will make our code and model weights publicly available.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pp. 370–387. Springer, 2024.

Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. Caparena: Benchmarking and analyzing detailed image captioning in the llm era. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14077–14094, 2025.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Xinyu Fang, Zhijian Chen, Kai Lan, Lixin Ma, Shengyuan Ding, Yingji Liang, Xiangyu Zhao, Farong Wen, Zicheng Zhang, Guofeng Zhang, et al. Creation-mmbench: Assessing context-aware creative intelligence in mllms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 447–456, 2025.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. *arXiv preprint arXiv:2309.10105*, 2023.

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference? *Advances in Neural Information Processing Systems*, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 7871–7880, 2020.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Yifan Li, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Diffusion models for non-autoregressive text generation: A survey. *arXiv preprint arXiv:2303.06574*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36:56998–57025, 2023.

Justin Lovelace, Varsha Kishore, Yiwei Chen, and Kilian Weinberger. Diffusion guided language modeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14936–14952, 2024.

Viacheslav Meshchaninov, Egor Chimbulatov, Alexander Shabalin, Aleksandr Abramov, and Dmitry Vetrov. Cosmos: Compressed and smooth latent space for text diffusion modeling. *arXiv preprint arXiv:2506.21170*, 2025.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, et al. Docci: Descriptions of connected and contrasting images. In *European Conference on Computer Vision*, pp. 291–309. Springer, 2024.

Aleksandar Petrov, Mark Sandler, Andrey Zhmoginov, Nolan Miller, and Max Vladymyrov. Long context in-context compression by getting to the gist of gisting. *arXiv preprint arXiv:2504.08934*, 2025.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Alexander Shabalin, Viacheslav Meshchaninov, Egor Chimbulatov, Vladislav Lapikov, Roman Kim, Grigory Bartosh, Dmitry Molchanov, Sergey Markov, and Dmitry 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, pp. 25110–25118, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann Lecun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *International Conference on Machine Learning*, pp. 55854–55875. PMLR, 2025.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini,

Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL `https://arxiv.org/abs/2503.19786`.

Dimitri Von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Tiezheng Zhang, Yitong Li, Yu-cheng Chou, Jieneng Chen, Alan Yuille, Chen Wei, and Junfei Xiao. Vision-language-vision auto-encoder: Scalable knowledge distillation from diffusion models. *arXiv preprint arXiv:2507.07104*, 2025.

Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua Susskind, and Navdeep Jaitly. Planner: Generating diversified paragraph via latent language diffusion model. *Advances in Neural Information Processing Systems*, 36:80178–80190, 2023.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

APPENDIX: REPURPOSING PRE-TRAINED LLMS AS HIGH FIDELITY CONTINUOUS TEXT AUTOENCODERS

OUTLINE

A: **Captioning Model Quality Analysis**
A detailed qualitative analysis, identifying high aesthetic fidelity but critical failures in symbolic grounding and OCR.

B: **Captioning: Global Content is Determined in Latent Space**
Stochastic decodes of one latent versus decodes of independently sampled latents.

C: **Attention Masking Strategy**
Detailed breakdown of the attention visibility matrix for LLMAE.

D: **LLMAE Latent Smoothness**
Latent robustness to Gaussian noise perturbations and D_{KL} calculation equation.

E: **LLMAE Latent Interpolation**
Linear interpolation examples to observe phrase-level recombination and syntactic preservation.

F: **Data Ordering Ablation**
The length curriculum versus shuffled training data on both backbones.

G: **Adapter Rank and Compressing K further**
The LoRA-rank sweep on both backbones, and what compressing a trained $K{=}256$ latent to 64 tokens shows.

H: **Reconstruction by Input Length**
The latent-budget sweep broken out by input length, showing where each K succeeds and fails.

I: **ICAE-style Autoencoder with Gemma-270M**
Comparative baseline ablation when using the ICAE architecture instead of LLMAE.

J: **Exact Reconstruction Fidelity Metrics**
Exact-match and edit-distance metric definitions with a per-category error breakdown.

K: **Method Details**
Detailed specifications for LLMAE and the latent diffusion model for captioning.

A CAPTIONING MODEL QUALITY ANALYSIS

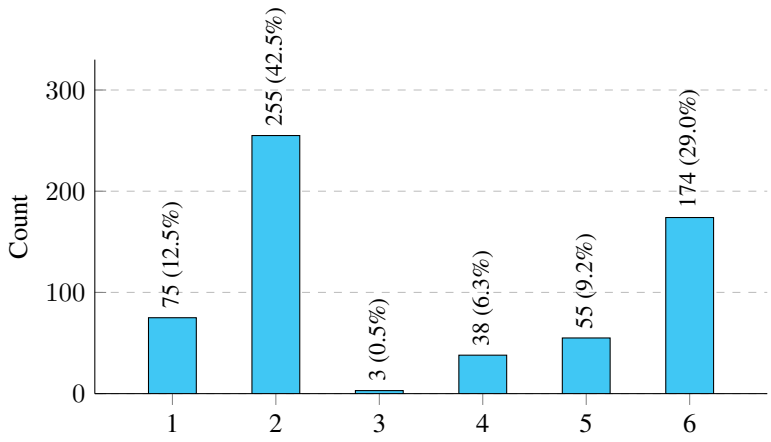


Figure 4: Distribution of VLM-Judge scores across the evaluation dataset.

We provide the distribution of VLM Judge Scores of our captioning model on the CapArena test set in Fig. 4. Fig. 6 shows a caption scored $6/6$ by the judge, and Fig. 7 one scored $1/6$.

VLM Score Judge Prompt:

Your role is to serve as an impartial and objective evaluator of an image caption generated by a Large Multimodal Model (LMM). Based on the single image input, assess the caption on three main criteria:

1. Coverage of image elements – how well the caption mentions the salient objects, their attributes, actions, and contextual details.
2. Absence of hallucinations – the caption must not invent objects, attributes, counts, spatial relations, or other details not present or implied by the image.
3. Object spatial layout consistency – whether spatial relationships (left/right, above/below, front/behind, center, background/foreground) are described accurately.
    - Any incorrect or invented spatial relation is a hallucination.
    - Omitting an obvious spatial relation reduces coverage.
    - Stating a relation that is ambiguous or uncertain in the image is also a hallucination.

Evaluation protocol:
Start with a brief explanation of your evaluation process. Then assign one rating using the scale below. Output only the rating number—no extra text, symbols, or commentary.

6 Comprehensive coverage, correct spatial layout, no hallucinations
5 Very informative, correct spatial layout, no hallucinations, minor omissions
4 Moderate coverage, correct spatial layout, no hallucinations, several omissions
3 Limited coverage, minimal spatial detail, no hallucinations
2 Informative but contains at least one hallucination (object or spatial)
1 Limited coverage and at least one hallucination (object or spatial)
0 Not informative and/or multiple hallucinations

Figure 5: System prompt used for the VLM Score Judge evaluation, adapted exactly from (Zhang et al., 2025).

A.1 Categorical Performance Analysis

To analyze the failure modes and performance benchmarks of our captioning model, we utilized an LLM Judge to aggregate and categorize the qualitative feedback and numerical scores from the full set of per-image VLM Judge explanations generated by GPT-4o. The following LLM-generated summary highlights a significant performance gap between aesthetic scene interpretation and symbolic grounding:

- **Environmental and Aesthetic Fidelity (Primary Strength):** The model demonstrates high fidelity in describing natural landscapes and organic textures (Scores: 5-6). It utilizes an advanced vocabulary to describe environmental nuances such as moss-covered surfaces, leaf translucency, and specific floral species. Furthermore, it excels in atmospheric reasoning, correctly identifying lighting conditions such as dappled sunlight, diffused overcast light, and artificial nighttime illumination.
- **Compositional and Spatial Awareness:** There is strong evidence of high-level compositional understanding. The model consistently identifies camera perspectives (e.g., macro, low-angle, aerial) and correctly interprets the spatial relationship between foreground framing elements and background subjects. This suggests an effective global awareness of the scene's visual hierarchy.
- **Symbolic and OCR Grounding (Critical Failure):** Systematic failure occurs in tasks requiring the decoding of textual or symbolic data (Scores: 1-2). The model frequently exhibits "symbolic hallucination," where it confidently generates plausible but entirely incorrect text for signs, murals, and labels (e.g., misreading *MAVERICK TENNIS* as *Convention Tennis Tennis* in Fig. 7). This indicates a lack of robust low-level OCR integration.
- **Quantitative Reliability and Individuation:** The model struggles with discrete object counting and instance individuation. It consistently misreports quantities (e.g., reporting "ten birds" for significantly larger flocks) and occasionally hallucinates secondary objects to "fill" the semantic space of a scene.



VLM Judge Score: 6/6

Caption: *A full shot captures a wooden bench positioned amidst a natural, forest-like setting. The wooden bench, the focal point of the image, prominently features a curved back, armrests, and vertical slats that enhance its classic design. It rests on a rectangular stone platform, flanked by two stout, light-colored bases. The ground surrounding the bench or platform is covered in small, light gray gravel, creating a neutral and textured surface. The background consists of dense foliage, predominantly leafless trees and bushes, indicating a natural and serene environment. These branches reach towards the sky, lending a sense of depth and enclosure to the scene.*

The lighting appears soft and diffused, casting gentle shadows across the scene, suggesting an overcast day. The dominant colors are muted greens, greys, and a muted touch of somber tones. The texture varies from the smooth, weathered wood of the bench to the rough stone and the smooth bark of the trees and branches.

There are no people or animals visible in the frame. The overall mood is peaceful, evoking a sense of quiet contemplation within the natural surroundings. The color palette is dominated by shades of brown, beige, and grey, which contribute to the earthy and muted aesthetic of the scene.

Figure 6: **Latent Diffusion with LLMAE.** Qualitative demonstration of detailed captioning ability.



Caption: *"The image is a close-up shot of Convention Tennis Tennis against a blue corrugated backdrop. The prominent lettering "CorrespondingON" features a stylized depiction of a horse head with a mane and tail, both outlined in bright orange with a subtle gradient highlight. The horse is rendered in profile, facing right, with its head raised, giving a slight frown. The backdrop is a vibrant blue, vertically ribbed metal, creating vertical shadows that suggest it is daytime. Below the horse's emblem, the words "VICK" are printed in a classic, bold orange font. Below the horse, the words "Correspondence TNINS" are written in the same orange type, slightly smaller font. The overall lighting is bright and even, emphasizing the contrast between the blue metal backdrop and the vivid colors. The mood conveyed is sleek and bold, appropriate for school team visual's nod to tennis players."*

VLM Judge Score: 1/6

VLM Judge Explanation: The caption contains several inaccuracies and hallucinations. It incorrectly describes the text as "Convention Tennis Tennis" and "CorrespondingON," while the actual text is "MAVERICK TENNIS." It also mentions "Correspondence TNINS" and "VICK," which are not present. The description of the horse's mane and tail is inaccurate, as the image only shows the horse's head. The spatial layout and lighting description are mostly correct, but the caption's errors significantly impact its accuracy.

Figure 7: Qualitative demonstration of the captioning model's lack of grounding.

- **Domain-Specific Technical Accuracy:** Performance degrades in industrial or specialized contexts due to "false specificity." While the model correctly identifies general categories (e.g., "vehicle"), it often hallucinates incorrect brand names, model numbers, or mechanical

components (e.g., attributing multiple turrets to single-turret tanks), suggesting a limit to its fine-grained classification capabilities.

- **Linguistic Consistency:** A recurring failure mode involves "lexical stuttering" or repetitive token generation within long-form descriptions (e.g., *"fleece and fleece"* or *"low-angle, low-angle"*). This suggests periodic breakdowns in the model's self-attention mechanism during the autoregressive generation of complex captions.

B CAPTIONING: GLOBAL CONTENT IS DETERMINED IN LATENT SPACE

Sec. 4.4 decodes each sampled latent with the frozen LLMAE decoder, treating the decode as a rendering step over a latent that already fixes the content. We test this directly on 24 evaluation images, using the Gemma captioning model under the sampling protocol of Table 5. For each image we compare two sources of randomness, each contributing five samples and therefore ten pairwise comparisons. In the first we sample a single latent and decode it five times with stochastic sampling (temperature 1.0), varying only the decode seed. In the second we sample five independent latents for the same image and decode each once, holding the decode seed fixed so that only the latent varies. Differences between two captions are measured as word-level edit distance normalized by the longer caption (0 identical, 1 entirely different). If global content were decided during decoding, the first condition would diverge; if it is determined by the latent, only the second should.

Table 6: Caption variation by source of randomness, over 24 evaluation images.

Varying	Held fixed	Mean	Median	Per-image range	Identical pairs
Decode seed	Latent	0.016	0.010	0.000-0.063	29.2%
Latent	Decode seed	**0.829**	**0.823**	0.746-0.902	0.0%

Table 6 reports the result. Varying the decode seed changes almost nothing: a mean edit distance of 0.016, with 29.2% of decode pairs byte-identical and all five decodes identical on 5 of the 24 images. Varying the latent changes nearly everything: a mean of 0.829, roughly $50\times$ larger, with no identical pair anywhere in the condition. The separation holds on every image individually, as the per-image ranges do not overlap (0.000-0.063 against 0.746-0.902), so it is not an artifact of averaging. Decoding stochastically rather than greedily from the same latent is equally inconsequential (mean 0.011). Table 7 shows both conditions for one image: two decodes of one latent agree on 175 of 176 words, differing only in *toward* versus *towards*, whereas two independent latents produce captions that remain grounded in the same scene — the squirrel, the tile surface, the white pillar, the surrounding greenery — but differ in which details are selected, where they appear, and the phrasing throughout. Content and structure are therefore properties of the latent, not of the decoding process, which complements the near-exact reconstruction of Table 1 and the noise robustness of Table 9.

Table 7: Both conditions for one evaluation image. Bold marks the only difference (top) and the points of divergence (bottom).

Same latent, two stochastic decodes — 175 of 176 words identical (word ED 0.006)	
Decode 1	. . . the primary subject of the image, is facing forward with its body turned slightly **toward** the camera. It has a long tail and a bushy. . .
Decode 2	. . . the primary subject of the image, is facing forward with its body turned slightly **towards** the camera. It has a long tail and a bushy. . .
Two independent latents, same image (word ED 0.804)	
Latent 1	A medium shot captures a squirrel standing on a **tile surface near a white pillar, with a glimpse of a green lawn and mulch**. The squirrel, **the primary subject of the image, is facing forward with its body turned slightly toward the camera. It has a long tail and a bushy tail**. . .
Latent 2	A medium shot captures a squirrel standing on a **tile patio, with a white column and support element in the outdoor setting**. The squirrel, **positioned towards the lower center of the frame, has a bushy gray fur and tan tail**. . .

Table 8: Attention masking strategy managing visibility between the encoding Input, Embed tokens, and decoder Output targets. Rows denote queries; columns denote keys. For a query token, **Visible** keys are subject to causal masking to maintain autoregressive consistency, while **Masked** keys prevent information flow. Layer 0 serves as a unique interface where the Output tokens are permitted to attend to the Embed tokens, effectively providing a global initialization for the reconstruction segment.

Block (Layers)	Query	Key: Input	Key: Embed	Key: Output
Input Layer	Output	Masked	Visible	Visible
(Layer 0)	Embed	Visible	Visible	Masked
Encoder Block	Output	Masked	Masked	Visible
(Layers 1 to D)	Embed	Visible	Visible	Masked
Decoder Block	Output	Masked	Visible	Visible
(Layers $D+1$ to $L-1$)	Embed	Masked	Visible	Masked

C Attention Masking Strategy

Table 8 provides a detailed breakdown of the attention masking strategy for LLMAE.

D LLMAE Latent Smoothness

To measure prior alignment in Table 2, we treat each coordinate (t, d) as a marginal distribution q_{td} and compute its empirical statistics over the M test samples, $\hat{\mu}_{td} = \frac{1}{M}\sum_{n=1}^{M} z_{n,t,d}$ and $\hat{\sigma}^2_{td} = \frac{1}{M}\sum_{n=1}^{M}(z_{n,t,d} - \hat{\mu}_{td})^2$. Same as the KL regularization loss calculation, we compute D_{KL} analytically using Eq. 1.

$$D_{\mathrm{KL}}(q_{td} \,\|\, \mathcal{N}(0,1)) = \frac{1}{2}\left(\hat{\sigma}^2_{td} + \hat{\mu}^2_{td} - 1 - \ln \hat{\sigma}^2_{td}\right) \tag{1}$$

During training the same divergence serves as $\mathcal{L}_z$, with $\hat{\mu}_{td}$ and $\hat{\sigma}^2_{td}$ estimated over the samples of the current batch and Eq. 1 averaged over all $K \times d$ coordinates (t, d).

We further investigate the robustness of LLMAE autoencoding through noise perturbations: we apply additive i.i.d. Gaussian noise to every scalar dimension: $z'_{n,t,d} = z_{n,t,d} + \sigma\epsilon_{n,t,d}$, where $\epsilon_{n,t,d} \sim \mathcal{N}(0,1)$. As shown in Table 9, the generative prior of Gemma 3 effectively denoises the latent signal, maintaining high performance up to $\sigma = 0.5$.

Table 9: Quality w.r.t. noise augmentation (Gemma, 1024-token setting).

Noise σ	BLEU (↑)	PPL (↓)	BERT (↑)
0	0.995	24.4	0.999
0.1	0.995	24.5	0.999
0.25	0.991	24.7	0.998
0.5	0.927	31.2	0.993

We further probe error tolerance directly across autoencoders by perturbing encoded latents in z-space at larger magnitudes and measuring the word similarity of the decoded text against each model's own clean (σ=0) decode (Table 10). Within the stage-1 (NTP-only) family, tolerance rises monotonically with LoRA rank. We note that the codec contracts the latent scale (Table 2), so a fixed absolute σ corresponds to a larger *relative* perturbation for the codec latent; the values below are therefore comparable within, but not directly across, latent scales.

Table 10: Latent error tolerance: word similarity of the decoded text vs. each model's clean decode, under i.i.d. Gaussian z-space noise. (Qwen unless noted.)

Autoencoder	σ=1.0	σ=1.5	σ=2.0
NTP only, r=8	0.9952	0.9337	0.6364
NTP only, r=16	0.9967	0.9839	0.8519
NTP only, r=32	**1.0000**	**0.9975**	**0.9730**
KL + Codec	0.9994	0.8528	0.2627
Gemma KL + Codec	0.9947	0.9426	0.5731

E LLMAE Latent Interpolation

To better understand the geometry of the learned text autoencoder, we perform a simple latent interpolation experiment. Given two input texts x_A and x_B, we encode them with the LLMAE encoder $E(\cdot)$ into latent representations

$$z_A = E(x_A), \qquad z_B = E(x_B),$$

and decode a midpoint linear interpolant:

$$z_{\text{mid}} = \frac{z_A + z_B}{2}$$

This experiment is not intended to show that Euclidean interpolation corresponds to semantic interpolation. Rather, it probes whether intermediate latent states remain decodable and what kinds of structure are preserved between distant texts. We observe a desirable property of the autoencoder that the interpolated latents often decode into recognizable, locally coherent text rather than complete garbage. However, the decoded midpoint is usually not a meaningful semantic average. Instead, it frequently preserves the global syntactic frame of one or both endpoints while swapping entities, modifiers, actions, or causal clauses. This suggests that the latent space supports high-resolution phrase-level recombination, even though linear paths through it do not reliably preserve semantic consistency. We provide examples in Table 11.

F Data Ordering Ablation

Data ordering. Table 12 shows that the short-to-long length curriculum used in the Gemma recipe is backbone-sensitive: it improves Gemma (EM $30.6\% \rightarrow 53.6\%$) but harms Qwen (EM $36.6\% \rightarrow 1.2\%$). We therefore train Qwen with shuffled data.

G Adapter Rank and Post-hoc Compression

Both experiments in this section are run at the NTP-only stage of Table 2, without the latent KL or the codec, with the data ordering fixed to shuffled on both backbones so that the adapter rank is the only variable of the sweep.

Table 13: Teacher-forced cross-entropy loss on the held-out set falls with LoRA rank on both backbones, while Gemma reconstruction peaks at r=8.

	Gemma-270M				**Qwen2.5-0.5B**			
Rank	**Loss (↓)**	**BLEU**	**W-ED**	**EM**	**Loss (↓)**	**BLEU**	**W-ED**	**EM**
4	0.4243	0.709	37.1%	0.4%	0.9806	0.467	94.9%	1.4%
8	0.0831	**0.966**	**10.4%**	**30.6%**	0.0207	0.940	7.9%	36.6%
16	0.0119	0.533	116%	0.4%	0.0044	0.996	0.26%	73.8%
32	**0.0060**	0.832	15.8%	19.8%	**0.0010**	**0.999**	**0.07%**	**92.8%**

Table 11: Linear interpolation in the text autoencoder latent space. Midpoints generally remain fluent and often preserve sentence-level structure, but they do not correspond to clean semantic averages. Instead, they exhibit phrase-level recombination: sometimes producing coherent hybrids and sometimes producing binding errors.

Pair	x_A	x_B	**Decoded midpoint**
Entity and object swap	A tired tabby cat stretched across the warm windowsill while rain tapped against the glass and a half-finished cup of tea cooled beside an open novel.	A tired golden retriever stretched across the warm windowsill while rain tapped against the glass and a half-finished cup of coffee cooled beside an open newspaper.	A tired **golden cat** stretched across the warm windowsill while rain tapped against the glass and a half-finished cup of **coffee** cooled beside an open **novel**.
Action and scene swap	The young violinist stood beneath the theater lights, closed her eyes, and played a slow melody while the audience listened in complete silence.	The young magician stood beneath the theater lights, raised his hands, and released a cloud of silver confetti while the audience cheered in surprise.	The young **violinist** stood beneath the theater lights, **raised his hands**, and released a **slow melody while confetti** while the audience **complete silence surprise**.
Scientific domain swap	During the biology lab, the students carefully observed yeast cells under a microscope, recorded changes in their growth, and argued that temperature was affecting the culture.	During the astronomy workshop, the students carefully observed distant galaxies through a telescope, recorded changes in their brightness, and argued that dust was affecting the measurement.	During the **astronomy lab**, the students carefully observed **distant galaxies** under a **telescope**, recorded changes in their **brightness**, and argued that **temperature** was affecting the **measurement**.
Opposing policy claims	The city should expand protected bike lanes because safer streets reduce traffic injuries, encourage commuting without cars, and make neighborhoods easier to navigate.	The city should remove protected bike lanes because narrower roads slow emergency vehicles, frustrate local businesses, and make neighborhoods harder to navigate.	The city should **expand** protected bike lanes because **safer roads slow emergency injuries**, **frustrate commuting without cars**, make make **harder to navigate**.
Different domains, similar tone	After years of drought, the farmer knelt in the cracked field, pressed a handful of dry soil between his fingers, and wondered whether the next storm would arrive in time.	After years of training, the astronaut floated beside the station window, watched Earth turn silently below her, and wondered whether the next signal would arrive in time.	After years of **training**, the **astronaut floated beside the station window**, **pressed Earth turn of dry her between his fingers**, the next **signal** would arrive in time arrive in time.
Completely different texts	A legal memorandum explains that the contract cannot be enforced because the signature was forged, the deadline had expired, and the witness testimony contradicted the filing.	A fairy-tale paragraph describes a child following a blue firefly through an enchanted forest, discovering a hidden door in an oak tree, and hearing music from an invisible castle.	A legal memorandum explains **paragraph describes** contract cannot be enforced because **firefly through forged enchanted forest**, discovering a **hidden door** in witness testimony contradicted the filing. **music from an invisible castle**.

Teacher-forced loss misranks generators. Table 13 varies the LoRA rank on both backbones and reports, alongside reconstruction quality, the teacher-forced cross-entropy loss on the held-out set: at every position of the reconstruction segment the model is given the reference prefix and scored on its predicted distribution over the next token. This loss decreases monotonically with rank on *both* backbones, yet Gemma's reconstruction peaks at $r{=}8$ and degrades at $r{=}16$ (BLEU $0.966 \rightarrow 0.533$,

Table 12: Effect of data ordering during training (two epochs).

Backbone	Ordering	BLEU	W-ED	EM (%)
Gemma-270M	curriculum	**0.978**	**1.8%**	**53.6%**
Gemma-270M	shuffled	0.966	10.4%	30.6%
Qwen-0.5B	curriculum	0.754	28.1%	1.2%
Qwen-0.5B	shuffled	**0.940**	**7.9%**	**36.6%**

Table 14: Compressing a trained $K{=}256$ latent to 64 tokens outperforms training $K{=}64$ directly.

Model	Latent	BLEU	W-ED	EM
$K{=}64$ trained directly	64	0.397	260%	0.0%
$256{\to}64{\to}256$ post-hoc compressor	64	0.978	1.2%	34.2%
$K{=}256$ uncompressed reference	256	0.999	0.07%	92.8%

EM $30.6\% \to 0.4\%$). On Qwen the two agree, improving strictly to $r{=}32$. Reconstruction is scored on the model's own greedy output, where every token after the first is conditioned on the tokens the model itself produced, whereas the teacher-forced loss conditions every position on the reference. The two measurements coincide only while the generated prefix matches the reference. A lower loss means the model places more probability on the reference token given the correct context; it does not mean the argmax is correct at every token. Nor does the loss say anything about what happens after a mistake, because a prefix containing the model's own error never occurs during teacher-forced evaluation. The practical consequence is what we act on: teacher-forced cross-entropy on the held-out set is a biased proxy for reconstruction, so every configuration in this paper is selected by generation quality on held-out documents rather than by loss.

Compress a good latent rather than train a small one. Training the bottleneck directly at $K{=}64$ fails on both backbones (Table 4). On Qwen the reconstructions reach BLEU 0.397 with a word edit distance of 260% and no document reproduced verbatim: the decoder does not produce a shortened or paraphrased version of the input but drifts into text unrelated to it and several times its length. The question is whether this is a capacity limit of 64 tokens or a failure to learn such a code through the LLM bottleneck, and Table 14 separates the two. We take the trained $K{=}256$ Qwen autoencoder and freeze it entirely. On its latents we fit a Perceiver-style compressor: 64 learned queries cross-attend to the 256 latent tokens through six cross-attention blocks (8 heads, each followed by self-attention and a feed-forward layer) to produce a $64{\times}896$ code, and a mirror-image decompressor with 256 learned queries expands the code back to $256{\times}896$; the two together have 310M parameters and are trained for one epoch over the cached latents of the 1.4M training documents. The only objective is the mean squared error between the expanded code and the original latent, computed over all $256{\times}896$ entries after per-coordinate standardization; the compressor never sees a token, the decoder, or a cross-entropy loss. At inference the text is encoded to 256 tokens, compressed to 64, expanded back to 256 and decoded by the untouched decoder. The 64-token code decodes to BLEU 0.978, word edit distance 1.2% and 34.2% exact match: not the 92.8% of the uncompressed latent, but a faithful document where direct training at $K{=}64$ produced none. The latent learned at full width is therefore compatible with a global MSE compression to a quarter of its length, whereas the encoder trained at $K{=}64$ never finds a usable code. The difficulty of small K lies in learning the code through the bottleneck, not in the information a 64-token latent can carry.

H Reconstruction by Input Length

Table 4 reports the latent budget sweep aggregated over the stratified test set. Because the bottleneck holds a fixed K regardless of input length, it is worth asking whether the best K shifts with document length. Table 15 breaks the same Gemma models out by input length, using four character-length buckets of the stratified 1024-token test set.

The per-bucket optimum does shift with length: $K{=}1024$ is marginally better on the longest inputs but degrades sharply on short ones (BLEU 0.677 below 1K characters), while $K{=}64$ and $K{=}128$ are capacity-limited throughout. $K{=}256$ is the only setting that remains strong in every bucket, which

Table 15: Gemma reconstruction (BLEU-4) by input length; buckets are character counts on the stratified test set.

Latent tokens	**≤1K chars** (n=126)	**1-2K** (n=129)	**2-3K** (n=125)	**3-4K** (n=120)
64	0.573	0.298	0.293	0.342
128	0.336	0.162	0.184	0.184
256	**1.000**	**0.998**	0.973	0.937
512	0.998	0.914	0.984	0.937
1024	0.677	0.995	**1.000**	**0.998**

is why it is optimal in aggregate for a fixed-length autoencoder intended to serve documents of any length. All models here are trained under the same regime, so this analysis varies the evaluation length only; it does not ablate the training length distribution or the maximum input length.

The two degradations have different characters. The K=1024 loss on short inputs is dominated by a failure to terminate: of the 126 inputs below 1K characters, 37 fall below BLEU 0.5, and 30 of those run past the end of the document into degenerate filler, with a median reconstruction 15.6× the length of the reference. These cases concentrate on the shortest inputs (mean reference length 210 characters, against 594 for the rest of the bucket), where the latent budget most overshoots the content. K=64 fails in the opposite way: it terminates correctly and preserves topic and document structure, but substitutes words and phrases throughout, including proper nouns — the signature of a capacity limit rather than a length mismatch. Table 16 gives one example of each.

Table 16: Characteristic failure modes by latent budget and input length. Bold marks corrupted spans.

Short input, 72 characters. BLEU: K=64 1.000, K=256 1.000, K=1024 0.489.	
Reference	We are having a Christmas Party!! Come join us with some Holiday Cheer!!
K=256	We are having a Christmas Party!! Come join us with some Holiday Cheer!!
K=1024	We are having a Christmas**, however** Come join us with some Holiday**,**!!
Long input, 3,269 characters (excerpt). BLEU: K=64 0.265, K=256 0.985, K=1024 1.000.	
Reference	. . . said he has no regrets about taking a break from his successful academic career for the rough and tumble of a Greek election campaign, his second in a year. Eleftheriadis is running for To Potami. . .
K=256	. . . said he has no regrets about taking a break from his successful academic career for the rough and tumble of a Greek election campaign, his second in a year. Eleftheriadis is running for To Potami. . .
K=64	. . . said he has no regrets about taking **what** break from his successful academic career for the rough and tumble of a Greek election campaign, his second in a year. **Elefther is the a running change** To Potami. . .

I ICAE-STYLE AUTOENCODER WITH GEMMA-270M

To evaluate the utility of the LLMAE architecture, we implemented a baseline following the In-context Autoencoder (ICAE) (Ge et al., 2023) framework. This baseline utilizes a soft prompting paradigm where the K latent tokens serve as a compressed "memory tokens" prefix to the input tokens for autoregressive decoding. We instantiated this model using the same Gemma-270M backbone and training data we use for LLMAE.

Our experiments indicate that this soft prompting approach is insufficient for high-fidelity text autoencoding for the lightweight Gemma backbone. This performance bottleneck remained non-trivial with or without employing curriculum learning (sorting training samples by sequence length). The inability of the ICAE-style prefix to recover fine-grained semantic and syntactic details suggests that simply treating latents as a soft prompt does not provide the necessary structural guidance for long-form reconstruction, at least in the Gemma 270M backbone. We provide the reconstruction evaluations in Table 17.

Table 17: Reconstruction quality for ICAE-style training on Gemma-270M

# Tokens	Curriculum Learning	BLEU (↑)	PPL (↓)	BERT (↑)
256	No	0.014	10.4	0.785
256	Yes	0.087	267410	0.801
512	Yes	0.393	204.1	0.922

Table 18: Perplexity (GPT-2-Large, ↓) and BERTScore F1 (↑) for the settings of Table 1. Reported in the appendix: BERTScore truncates long inputs, and degenerate reconstructions can attain lower perplexity than the source text.

	C4-News-Strat.		OpenWebText		CreationMMBench	
Method	**PPL**	**BERT**	**PPL**	**BERT**	**PPL**	**BERT**
Source Text	30.8	–	27.5	–	16.9	–
ICAE	36.8	0.981	30.0	0.963	17.7	0.983
COSMOS	66.9	0.979	32.6	0.982	23.3	0.951
LLMAE-Gemma	30.9	0.999	27.6	0.999	17.1	0.998
LLMAE-Qwen	31.2	1.000	28.8	0.999	17.4	0.999
Source Text	24.1	–	21.9	–	13.3	–
ICAE	30.2	0.968	23.0	0.948	14.4	0.970
LLMAE-Gemma	24.4	0.999	22.1	0.998	13.4	0.998
w/o Codec	26.9	0.993	24.7	0.993	14.6	0.994
LLMAE-Qwen	24.2	1.000	22.0	1.000	13.6	1.000
w/o Codec	24.2	1.000	22.0	1.000	13.4	1.000

J Exact Reconstruction Fidelity Metrics

Table 18 reports perplexity (GPT-2-Large) and BERTScore F1 for the settings of Table 1. To complement the BLEU-4, exact-match, and edit-distance results there, we report direct, token-level fidelity metrics computed on the same model outputs and test sets. All metrics operate on whitespace-delimited words, making them independent of any model-specific subword tokenizer and therefore directly comparable across the Gemma- and Qwen-based LLMAE instantiations, the Mistral-based ICAE, and the BERT-based COSMOS. We report the per-category error analysis in Table 19.

Metric definitions. For each test sample, let $\mathbf{w} = (w_1, \ldots, w_n)$ denote the reference word sequence and $\hat{\mathbf{w}} = (\hat{w}_1, \ldots, \hat{w}_m)$ the reconstruction word sequence, obtained by whitespace splitting.

- **Exact match (EM):** the fraction of samples for which the reconstruction equals the reference in full after whitespace normalization (runs of whitespace collapsed to single spaces). This is a strict, binary, whole-document criterion with no partial or prefix credit: a sample scores 1 only if the reconstruction is identical to the reference from the first character to the last, and a single incorrect character anywhere in the document yields 0. EM is therefore the fraction of documents reproduced verbatim end-to-end; an error in the final word is penalized exactly as heavily as an error in the first.
- **Word edit distance (W-ED):** the Levenshtein distance between $\mathbf{w}$ and $\hat{\mathbf{w}}$ (minimum number of word insertions, deletions, and substitutions), normalized by n and averaged over samples.

Category breakdown. For the per-category analysis in Table 19, every *reference* word is assigned to exactly one category, in priority order: *number* (contains a digit), *punctuation* (no alphanumeric characters), *capitalized* (begins with an uppercase letter; a proxy for named entities that conservatively also includes sentence-initial words), *rare* (lowercase word occurring at most twice among the references of that evaluation set), and *common* (all remaining words). A reference word counts as an error if the alignment marks it as substituted or deleted. The per-category error rate is the number of errored words divided by the total reference words in that category, pooled over all samples.

Table 19: Word-error rate (↓) of LLMAE by reference-word category on the C4-News-Stratified evaluation sets. Errors show no systematic concentration in numbers, capitalized words (named-entity proxy), punctuation, or rare tokens. On the same categories, the baseline error rates range from 0.12-0.36 (ICAE) and 0.02-0.38 (COSMOS).

Setting	Number	Punctuation	Capitalized	Rare	Common
C4-News-Stratified (512)	0.0011	0.0028	0.0019	0.0009	0.0004
C4-News-Stratified (1024)	0.0069	0.0028	0.0046	0.0057	0.0029

Interpretation. Exact match compounds sharply with document length even at fixed per-token fidelity: a single imperfect word among ∼500 zeroes the match, however low the word edit distance (Table 1). The per-document error distribution of the Gemma LLMAE confirms that misses are near-misses, consistently across all three domains: at 1024 tokens, 67-71% of documents reconstruct verbatim, 84-87% are within 2 word edits of the reference, 94-95% are within 5, and the median non-exact document differs in only 2-3 words. The baselines' low EM arises from qualitatively different error modes: ICAE's errors are content substitutions whose rate grows with position in the document (on every domain the word-error rate in the second half exceeds the first half, e.g. 14.7% vs. 22.8% on C4-News-Stratified at 1024 tokens), consistent with the length degradation in Table 1, whereas COSMOS almost never reproduces a document verbatim (94-100% of documents contain at least one word error) largely because its BERT-tokenizer decoding pipeline inserts spaces around intra-word punctuation and drops line breaks when detokenizing (e.g., *op-ed→op - ed*, *2014's→2014 ' s*, *27.8→27. 8*). The characters are largely preserved but word boundaries are not, so word-level metrics register errors on nearly every document.

K Method Details

K.1 LLMAE

We describe the Gemma recipe; the Qwen2.5-0.5B LLMAE follows the same recipe and differs only in the entries for which Table 20 lists two values (latent layer 15, LoRA rank 32 with alpha 64, shuffled data ordering, and 23.8M trainable parameters). LLMAE trains Gemma-3-270M-PT (not instruction-tuned) with LoRA on all attention projections for 2 epochs over 1.4M stratified text samples (1M Pile-uncopyrighted + 400K C4-RealNewsLike). The reconstruction test sets contain 500 held-out samples per dataset and length setting, with the exception of CreationMMBench at 512 tokens (311 samples): the source benchmark provides 765 GPT-4o reference answers in total, and we retain only those within the 512-token limit under all evaluated tokenizers (Gemma, BERT, and LLaMA) so that every baseline is evaluated on identical texts. The C4-News-Stratified test sets are constructed to be explicitly disjoint from the training corpus: every candidate text is checked by content hash against all 1.4M training samples and excluded on any match. The other two sources cannot overlap with the training data by construction: OpenWebText uses the held-out test split shared by COSMOS, and the CreationMMBench references are GPT-4o-generated text from an external benchmark. Within each set, the evaluated subset is a fixed deterministic draw (a seeded permutation of the set, truncated to the reported sample count), so every method in Table 1 is scored on identical texts. Across the six settings the evaluated documents span up to 962 tokens with a mean of 438; the 1024-token ceiling is set by the least token-efficient tokenizer among those evaluated rather than by LLMAE. On Gemma, LLMAE utilizes LoRA adapters (737K parameters), 256 latent token embeddings (164K parameters), and an additive Codec (9.8M), totalling 10.7M trainable parameters. The input layout is `[input tokens | 256 latent tokens | input copy]` with the LM loss computed only on the reconstruction segment. Texts are sorted by length (curriculum learning), output logits are regularized with KL toward the frozen reference model, and Gaussian noise (σ=0.1) is added to the latent activations at layer 11 during training to encourage a smooth latent space. The model is then fine-tuned for 2 additional epochs either with the latent KL toward $\mathcal{N}(0, I)$ on the raw latent (the + KL row of Table 2) or, for our final model, with a lightweight additive Codec inserted at layer 11 whose refined latents are regularized toward $\mathcal{N}(0, I)$; the coefficients of each stage are listed in Table 20. The full parameter configuration is listed in Table 20.

Table 20: LLMAE hyperparameters for both backbones. Values shared by both are given once.

Parameter	**Gemma-3-270M**	**Qwen2.5-0.5B**
Input text length	1024 tokens	
Latent tokens	256	
Latent layer	11 (12 of 18)	15 (16 of 24)
Latent dimension	640	896
Trainable parameters	10.7M	23.8M
LoRA / latent embeddings / codec	0.74M / 0.16M / 9.8M	4.3M / 0.23M / 19.3M
LoRA		
Rank r / alpha	8 / 16	32 / 64
Dropout	0.05	
Target modules	`q_proj, k_proj, v_proj, o_proj`	
Saved modules	`embed_tokens`	
Training		
Dataset	1M Pile-uncopyrighted + 400K C4-RealNewsLike	
Total samples	1,400,000	
Epochs	2	
Batch size	32 (4 per GPU $\times$ 8 GPUs)	
Gradient accumulation steps	4 (effective batch 128)	
Learning rate	5×10^{-5}	
Latent noise std (σ)	0.1	
Data ordering	curriculum (sorted by length)	shuffled
Precision	bfloat16	
Regularization		
Reference-model KL coefficient	1×10^{-4}	
Latent KL coefficient (+KL stage, no codec)	1×10^{-3}	
Latent KL coefficient (with codec)	0; the codec KL below regularizes the codec embedding	
Codec fine-tuning		
Additional epochs	2	
Codec KL weight	1×10^{-3}	
Codec learning rate	1×10^{-4}	
Data ordering	shuffled	

K.2 Latent Diffusion for Image Captioning

The score network and diffusion process are based on the architecture utilized by COSMOS (Meshchaninov et al., 2025), incorporating three primary modifications to align with the LLMAE latent space. First, the architecture is resized to the LLMAE latent dimensionality: 256×640 for the Gemma LLMAE (10 attention heads of size 64) and 256×896 for the Qwen2.5 LLMAE (14 heads of size 64), with a SwiGLU FFN width of 2560 in both cases. Second, we introduce image conditioning by projecting SigLIP features through a shared linear layer ($1152 \to 640$ or $1152 \to 896$). These features are integrated via a per-block cross-attention sublayer utilizing RMSNorm pre-norms on both query and key/value sequences. A learnable scalar residual gate, initialized to zero, is applied to these sublayers to allow the model to begin as a purely unconditional denoiser and gradually learn to incorporate image signals. Third, classifier-free guidance (CFG) is implemented by replacing the image conditioning with a learned null embedding with a 0.1 probability during training, enabling CFG extrapolation during inference. All other structural and procedural components—including the Tanh VP-SDE schedule ($d = 5$), U-Net skip connections, x_0-prediction parameterization, self-conditioning, per-dimension latent normalization, and the Euler reverse solver—remain identical to the original COSMOS diffusion implementation. In our primary configuration, the autoencoder is our pre-trained LLMAE (278M; Gemma-3-270M backbone with a 9.8M Codec) operating on a compact 256×640 latent space, and the score network is a 12-layer transformer with hidden dimension 640 (111M parameters), giving a total inference footprint of $\sim$819M parameters with only the score network trained. For the Qwen2.5 LLMAE (494M; Qwen2.5-0.5B backbone with a 19.3M Codec) the score network widens to hidden dimension 896 (183M parameters), for a total of $\sim$1,105M with 183M (16.6%) trained; all other settings are identical and the two models are trained and evaluated under the same protocol. Replacing LLMAE with the COSMOS autoencoder (356M; 208M en-

coder, 149M decoder) expands the latent space to 512×768, which requires a wider score network (hidden dimension 768, 159M parameters) to match; the total system grows to $\sim$943M parameters, with the trained fraction increasing from 111M to 159M (13.6% to 16.9% of total). The full parameter configuration is listed in Table 21.

Table 21: Hyperparameters for latent diffusion captioning, per LLMAE backbone. Values shared by both are given once.

Parameter	**Gemma LLMAE**	**Qwen2.5 LLMAE**
Score network		
Latent space (tokens × dim)	256×640	256×896
Hidden size × layers × heads	$640 \times 12 \times 10$	$896 \times 12 \times 14$
FFN intermediate size	2560	
Trainable parameters	111M	183M
Total inference footprint	819M	1,105M
Image conditioning		
Vision encoder	SigLIP SO400M patch14-384 (frozen, 428M params)	
Encoder output	729 patches × 1152 dims	
Conditioning projection	Linear($1152 \to 640$)	Linear($1152 \to 896$)
Cross-attention gate init	0.0	
CFG null embedding	Learnable, zero-init	
Cond drop probability	0.1	
CFG scale (inference)	3.0	
Diffusion process		
Noise schedule	Tanh VP-SDE, $d = 5$	
x_0-prediction	Yes	
Self-conditioning	Yes	
Latent normalization	Per-(token, dim) mean/std	
Inference steps	250	
Training		
Batch size	64 per GPU × 8 GPUs	
Training iterations	226,000	
Learning rate	2×10^{-4} (2M-step cosine to 2×10^{-5}, halted at 226K; constant in effect)	
Warmup steps	2,000	
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.98$, weight decay 10^{-5})	
EMA decay (tracked)	0.9999	0.99995
Inference weights	Raw score-network weights (EMA not applied)	
Precision	float32	