SENTENCE CURVE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models (LMs) are central to modern AI systems, and diffusion language models (DLMs) have recently emerged as a competitive alternative. Both use word embeddings not only as inputs but also, implicitly, as target representations. We argue that conventional targets primarily provide token-wise supervision and do not explicitly encourage coordinated prediction across neighboring positions, which may be particularly important under noisy parallel decoding. To address this, we propose *sentence curve*, a spline representation whose control points are shared across multiple target words, and introduce the sentence curve language model (SCLM), which extends DLMs to predict through this shared representation. Theoretically, we show that sentence curve prediction reweights token-wise gradients and characterize a trade-off between target-side coupling and representation capacity. Empirically, SCLM achieves state-of-the-art performance among DLMs on IWSLT14 and WMT14, surpasses autoregressive Transformers on WMT14, and trains effectively without knowledge distillation. On LM1B, SCLM further improves generic language generation in a standard-scale fully non-autoregressive setting and shows promising preliminary transfer to semi-autoregressive block diffusion.

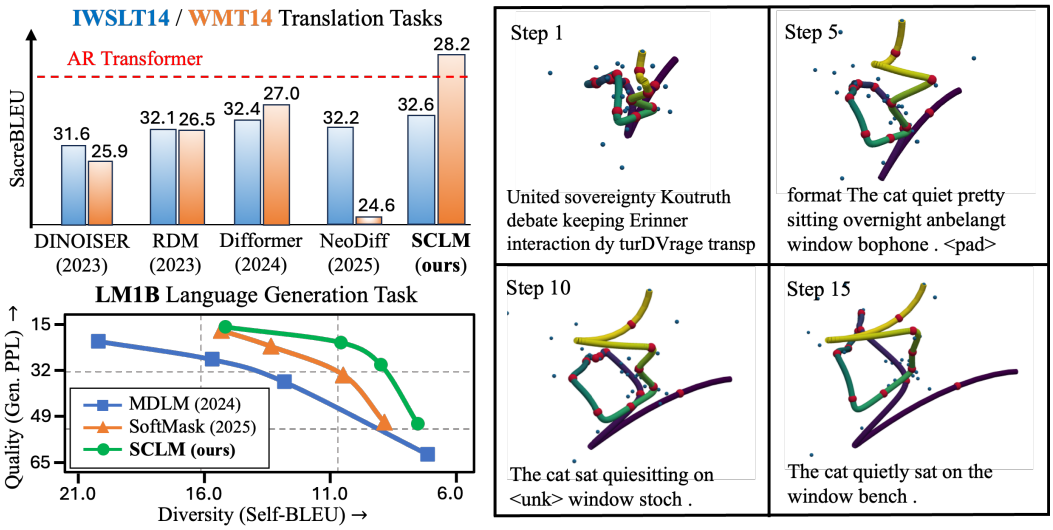


Figure 1: **Left**: Performance comparison of SCLM with prior models on translation tasks (top) and generic language modeling (bottom). **Right**: Example of sentence curve generation by SCLM in a translation task, where the reference sentence is *"The cat sat quietly on the window sill."*

1 INTRODUCTION

Language models (LMs) have become central to modern AI systems (Gemini et al., 2023; Achiam et al., 2023). Recently, non-autoregressive (non-AR) models, including diffusion language models (DLMs) (Sahoo et al., 2024; Nie et al., 2025), have emerged as promising alternatives to AR models, offering iterative refinement, bidirectional context, and parallel token generation. However, parallel prediction introduces the long-standing *multimodality problem*: when multiple valid target sequences exist, individually plausible token predictions may combine into an inconsistent sentence (Gu et al.,

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.

2017; Feng et al., 2025). Prior approaches mitigate this through sequence-level knowledge distillation (KD) (Kim & Rush, 2016), latent-variable modeling (Shu et al., 2020; Heo & Choi, 2023), or modified decoding (Lee et al., 2018), yet coordination across target positions remains an important challenge.

We revisit this problem from the perspective of the *target representation*. In conventional language modeling, each output position is supervised toward its corresponding target token, represented by a word embedding or output projection vector, which we call *target embedding vector* (Section 3). Although the backbone can exploit contextual information to coordinate predictions across target positions, this coordination largely relies on the availability of informative context. In non-AR and diffusion generation, where multiple tokens are predicted simultaneously from noisy or incomplete context, this source of coordination can be weakened. Meanwhile, the target representation itself still treats positions separately and provides no additional encouragement for coordinated prediction. We therefore ask whether introducing dependency directly into the target representation can provide an additional source of target-side dependency for coordinated generation.

To provide such target-side dependency, we seek an intermediate target representation in which neighboring words share variables while the original embedding sequence remains recoverable. A B-spline naturally provides such a structure: its curve is governed by a set of *control points*, and each point on the curve is represented through several nearby control points. We use these control points as intermediate representations of target word embeddings, so that neighboring words share part of their representation instead of being treated independently. We call this representation a ***sentence curve*** and introduce the ***sentence curve language model*** (SCLM), which predicts in the control point space and maps the result back to the original embedding space for the standard training objective, without modifying the backbone architecture. We further show that this shared representation reweights token-wise gradients, revealing a trade-off between the burden induced by stronger target-side coupling and the representation capacity available to accommodate token-specific corrections.

Empirically, SCLM achieves state-of-the-art performance among Gaussian DLMs on IWSLT14 (Cettolo et al., 2014) and WMT14 (Bojar et al., 2014) (Figure 1, top-left), and surpasses AR Transformers (Vaswani et al., 2017) on some tasks. SCLM also trains effectively without knowledge-distilled data and exhibits more coordinated target behavior in our analyses. Beyond machine translation, SCLM improves generic language modeling on LM1B against discrete DLMs including MDLM (Sahoo et al., 2024) and SoftMask (Hersche et al., 2025) (Figure 1, bottom-left), and shows promising results when extended to the block diffusion architecture BD3-LM across multiple block sizes. These results support that sentence curve prediction provides a useful form of target-side dependency across different DLM architectures and generation settings. Our contributions are:

- **Sentence Curve Target Representation**: We introduce *sentence curves*, where neighboring target embeddings share spline control points, explicitly introducing target-side dependency.
- **Sentence Curve Language Model**: We incorporate sentence curve prediction into non-AR and semi-AR language models without modifying their backbone architectures.
- **Theoretical Analysis**: We characterize how sentence curve mapping reweights token-wise gradients and how excessive coupling can increase the burden of token-specific corrections, revealing a coupling–capacity trade-off.
- **Empirical Validation**: We demonstrate improvements on machine translation and language modeling across multiple DLMs, together with analyses of coordinated target behavior.

2 Background: Diffusion Language Models

Let $\mathcal{V}$ denote the vocabulary and $E = [\mathbf{e}_1, \dots, \mathbf{e}_{|\mathcal{V}|}] \in \mathbb{R}^{d \times |\mathcal{V}|}$ the word embedding matrix. For a sentence $Y = (y_1, \dots, y_L)$, its embedding sequence is $E_Y = [\mathbf{e}_{y_1}, \dots, \mathbf{e}_{y_L}] \in \mathbb{R}^{d \times L}$, where L and d denote sequence length and embedding dimension, respectively.

Similar to diffusion models (Ho et al., 2020), DLMs consist of a forward process that progressively adds noise and a reverse process that removes it. Depending on whether these processes operate in continuous or discrete spaces, DLMs are broadly categorized into Gaussian and discrete variants.

In Gaussian DLMs, a sentence Y is mapped to an embedding sequence $E_Y = [\mathbf{e}_{y_i}]_{i=1}^L$, treated as the clean latent representation $[\mathbf{e}_i^0]_{i=1}^L$. For each latent $\mathbf{e}^0$, the forward and reverse processes are defined:

$$q(\mathbf{e}^t|\mathbf{e}^0) = \mathcal{N}\left(\mathbf{e}^t; \sqrt{\bar{\alpha}^t}\mathbf{e}^0, (1-\bar{\alpha}^t)\mathbf{I}_d\right),$$
$$p_\theta(\mathbf{e}^{t-1}|\mathbf{e}^t) = \mathcal{N}\left(\mathbf{e}^{t-1}; \mu_\theta(\mathbf{e}^t, t), (1-\alpha^t)\mathbf{I}_d\right),$$

where $t \in \{1, \dots, T\}$, α^t is the noise schedule, and $\bar{\alpha}^t = \prod_{i=1}^t \alpha^i$. Many Gaussian DLMs parameterize the denoiser by its clean embedding prediction $\hat{\mathbf{e}}_\theta^0(\mathbf{e}^t, t)$, from which the reverse mean is constructed. They are trained with an objective of the form

$$\mathcal{L}_g(y; \theta, E) = \mathbb{E}_{t,\mathbf{e}^t}\left[\|\mathbf{e}^0 - \hat{\mathbf{e}}_\theta^0\|^2\right] - \log p(y|E^\top \hat{\mathbf{e}}_\theta^0), \tag{1}$$

where the first diffusion loss reconstructs the clean embedding and the second anchor loss aligns the predicted latent with the corresponding discrete token.

Discrete DLMs instead operate directly in token space, treating $Y^0 := Y$ as clean categorical latent variables. Both forward and reverse processes use categorical distributions, and training minimizes a negative evidence lower bound (ELBO). In MDLM (Sahoo et al., 2024), this simplifies to a noise-weighted CE objective:

$$\mathcal{L}_m(y; \theta, W) = \mathbb{E}_{t,y^t}\left[\frac{\alpha^{t'}}{1-\alpha^t}\log p_\theta(\hat{y}^0|y^t, t)\right],$$
$$p_\theta(\hat{y}^0|y^t, t) = \text{softmax}\big(W^\top \mathbf{h}_\theta(y^t, t)\big), \tag{2}$$

where $W \in \mathbb{R}^{d\times|\mathcal{V}|}$ is the output projection matrix and $\mathbf{h}_\theta(y^t, t)$ is the backbone hidden state. We refer readers to Sahoo et al. (2024) for further details.

In non-AR generation, multiple target tokens are predicted in parallel from a shared, often noisy context. When multiple valid target sequences exist, this can make coordination across positions difficult and lead to the multimodality problem (Gu et al., 2017; Huang et al., 2022). Prior approaches such as KD (Kim & Rush, 2016) and latent-variable modeling (Shu et al., 2020; Heo & Choi, 2023) partially address this issue, motivating additional mechanisms for target-side dependency.

3 Token-wise Target Representation

We first clarify the target-side supervision in conventional LMs and DLMs. In Gaussian DLMs, Eq. 1 directly encourages the backbone output $\hat{\mathbf{e}}_\theta^0$ to reconstruct the clean word embedding $\mathbf{e}_y$. In discrete models, token likelihood is parameterized by the output projection matrix $W = [\mathbf{w}_1, \dots, \mathbf{w}_{|\mathcal{V}|}]$, where $p_\theta(y|\mathbf{h}_\theta) = \text{softmax}(W^\top \mathbf{h}_\theta)_y$. Each column $\mathbf{w}_y$ can therefore be viewed as a learned vector representation of target token y in the output space. Hereafter, we refer to both the clean word embedding $\mathbf{e}_y$ in Gaussian DLMs and the output projection vector $\mathbf{w}_y$ in discrete models as the *target embedding vector*.

For discrete models, let $H_\theta = [\mathbf{h}_{\theta,1}, \dots, \mathbf{h}_{\theta,L}]$ and $W_Y = [\mathbf{w}_{y_1}, \dots, \mathbf{w}_{y_L}]$, where W_Y collects the target embedding vectors corresponding to the target sequence Y. The standard token classification pipeline is:

Definition 1 (Token Classification Pipeline)**.**

$$H_\theta \to W^\top H_\theta \to \text{softmax}(W^\top H_\theta) \to \min_\theta CE_Y(H_\theta; W_Y, W),$$

where

$$CE_Y(H_\theta; W_Y, W) = -\frac{1}{L}\sum_{i=1}^L \log \frac{\exp(\mathbf{w}_{y_i}^\top \mathbf{h}_{\theta,i})}{\sum_{v\in\mathcal{V}}\exp(\mathbf{w}_v^\top \mathbf{h}_{\theta,i})}.$$

Thus, each position is supervised toward its corresponding target vector $\mathbf{w}_{y_i} \in W_Y$, while normalization is performed over the full vocabulary W.

Under unit-norm and local-isotropy assumptions on the target embedding vectors, we obtain the following geometric interpretation.

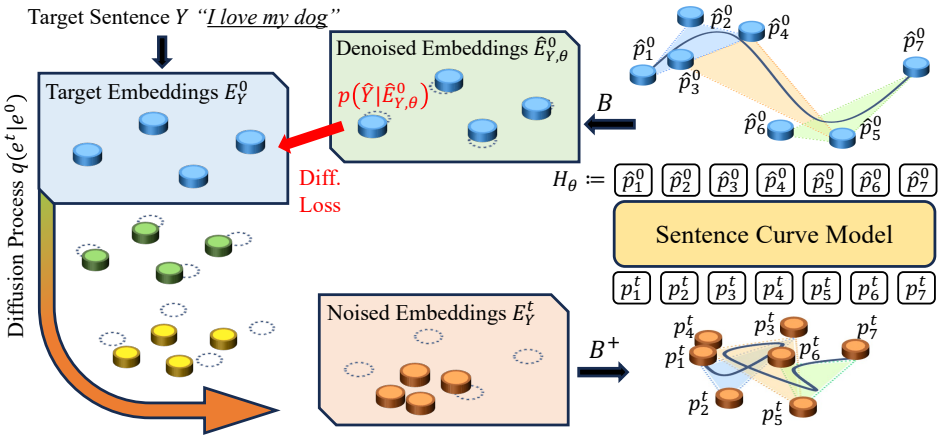


Figure 2: SCLM training at diffusion step t. Noised target embeddings are mapped to control points by B^+, denoised by the backbone, and mapped back by B before applying the original DLM objective. Only these input/output mappings modify the underlying DLM.

Proposition 1 (Target Embedding Alignment)**.** *Under Assumptions 1 and 2, the target embedding matrix W_Y is an optimal solution of the Token Classification Pipeline:*

$$W_Y = \arg\min_{H_\theta} CE_Y(H_\theta; W_Y, W). \tag{3}$$

The proof and assumptions are discussed in Appendix A.3. Proposition 1 formalizes that token-level classification may encourage each backbone output toward its target embedding vector.

The target embedding vector may capture dataset-level contextual regularities because it is learned across many occurrences of the same token. However, a single vector cannot reflect the occurrence-specific relationships among target tokens in each sentence. Accordingly, while contextual adaptation is handled by the backbone, the conventional target representation primarily provides individual token-wise supervision and does not explicitly encourage target tokens to be predicted in a coordinated manner. This distinction becomes particularly relevant in non-AR generation, where multiple tokens are predicted simultaneously from noisy or incomplete context.

4 Sentence Curve

In this section, we introduce sentence curves, motivated by probabilistic curve modeling for multi-step prediction (Hug et al., 2020). We first define the sentence curve representation and its relationship with word embeddings. We then incorporate sentence curve prediction into DLMs in Section 4.2 and analyze its learning behavior in Section 4.3.

4.1 Definition of Sentence Curve

We construct sentence curves using B-spline functions. A B-spline curve is defined as $\mathbf{z}(\gamma) = \sum_{j=1}^{N} b_\eta(\gamma)_j \mathbf{p}_j \in \mathbb{R}^d$, where $P = [\mathbf{p}_j]_{j=1}^N \in \mathbb{R}^{d\times N}$ contains N control points. The basis function $b_\eta(\gamma) \in [0,1]^N$, satisfying $\sum_j b_\eta(\gamma)_j = 1$, is a degree-$\eta$ B-spline basis that determines the contribution of neighboring control points to each location on the curve.

The parameter γ specifies a location on the curve. Given a sentence of length L, we evaluate the curve at predefined indices $\Gamma = (\gamma_1, \ldots, \gamma_L)$, yielding $Z_\Gamma = [\mathbf{z}(\gamma_1), \ldots, \mathbf{z}(\gamma_L)] \in \mathbb{R}^{d\times L}$. We align these sampled points with the target embedding sequence, $E_Y = Z_\Gamma$. Defining $B_{\Gamma,\eta} := [b_\eta(\gamma_1), \ldots, b_\eta(\gamma_L)] \in [0,1]^{N\times L}$ gives

$$E_Y = P_Y B_{\Gamma,\eta}. \tag{4}$$

We refer to the B-spline parameterized by P_Y as the *sentence curve* of Y. As illustrated in Figure 3, each word embedding is formed from several control points with different weights,

while each control point contributes to multiple word embeddings. Thus, the control point representation provides a shared intermediate representation of the word embedding sequence.

For the theoretical development, we consider sentence curve configurations for which the basis matrix $B \in \mathbb{R}^{N\times L}$ has full column rank, with $N \geq L$. The control point representation is then

$$P_Y = E_Y B^+, \qquad B^+ = (B^\top B)^{-1} B^\top.$$

Since $B^+ B = I_L$, the mapping is information preserving in exact arithmetic. The pseudoinverse can nevertheless be numerically sensitive for poorly conditioned spline configurations, as examined in Appendix D.1. We therefore regard it as an initial implementation choice rather than an essential component of sentence curve prediction. Hereafter, we omit the subscripts Γ and η from B.

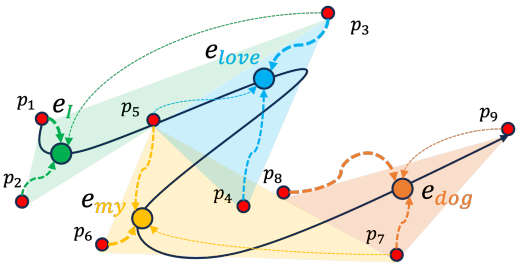


Figure 3: Illustration of control points and embeddings for the sentence *"I love my dog"* with $\eta = 3$. Small red dots denote control points ($\mathbf{p}$), and larger dots denote word embeddings ($\mathbf{e}$). The black curve is the spline defined by the control points, while colored dotted lines indicate their contributions to embeddings, weighted by $b_{\eta,j}$.

4.2 Sentence Curve Language Models

Based on Section 4.1, we reformulate the token classification pipeline by letting the backbone predict control point representations:

Definition 2 (Sentence Curve Prediction Pipeline)**.** *Given backbone outputs $H_\theta \in \mathbb{R}^{d\times N}$, sentence curve prediction follows*

$$H_\theta \to H_\theta B \to W^\top (H_\theta B) \to \mathrm{softmax}(W^\top (H_\theta B)) \to \min_\theta CE_Y(\theta).$$

Because the backbone output in this pipeline represents predicted control points, we hereafter denote it by $\hat{P}_\theta := H_\theta \in \mathbb{R}^{d\times N}$. Compared with Definition 1, $\hat{P}_\theta$ is mapped to the embedding space as $\hat{P}_\theta B$. By Proposition 1, this mapped output is encouraged toward the target embedding sequence. Because each control point contributes to multiple nearby target embeddings, their supervision becomes related through the shared control point representation. We refer to this as *target-side coupling*.

We incorporate this formulation into DLMs to obtain ***sentence curve language models*** (SCLMs). For Gaussian DLMs, given clean embeddings E_Y^0 and noised embeddings E_Y^t, we map E_Y^t to the control point space as $P_Y^t = E_Y^t B^+$. The backbone predicts denoised control points $\hat{P}_\theta(P_Y^t, t)$, which are mapped back by

$$\hat{E}_{Y,\theta}^0 = \hat{P}_\theta(P_Y^t, t) B.$$

The original DLM objective is subsequently applied to $\hat{E}_{Y,\theta}^0$. Thus, SCLM introduces only the $E \to P$ input mapping and the $P \to E$ output mapping while leaving the backbone architecture and training objective unchanged. The overall procedure is illustrated in Figure 2 and detailed in Algorithm 1. The same formulation can be applied to discrete DLMs by placing the sentence curve mappings before and after the backbone.

4.3 Theoretical Analysis of Sentence Curve Prediction

We analyze how the geometry of B affects learning under target-side coupling. For readability, we present simplified statements highlighting their main interpretations; full statements, derivations, and proofs are provided in Appendix A.3.

Gradient reweighting. Let $G := \nabla_{\hat{E}_{Y,\theta}} CE_Y(\theta) \in \mathbb{R}^{d\times L}$. Since $\hat{E}_{Y,\theta} = \hat{P}_\theta B$, the gradient propagated to the control point representation is $\nabla_{\hat{P}_\theta} CE_Y = GB^\top$. We define

$$I_B(G) := \|GB^\top\|_F^2,$$

which measures the magnitude, or *importance*, of a gradient pattern after propagation through the sentence curve mapping.

Lemma 1 (Gradient Reweighting Induced by B (Simplified)). *Let $Q := B^\top B$. For a normalized sequence-wide gradient pattern $G_{\text{seq}} = \mathbf{a}\mathbf{1}_L^\top/\sqrt{L}$ and token-local patterns $G_{\text{local},i} = \mathbf{a}\delta_i^\top$, with $\|\mathbf{a}\|_2 = 1$, their relative propagated importance is*

$$R(B) := \frac{I_B(G_{\text{seq}})}{\frac{1}{L}\sum_{i=1}^{L} I_B(G_{\text{local},i})} = \frac{\mathbf{1}_L^\top Q \mathbf{1}_L}{\text{tr}(Q)} \le \frac{s_{\max}^2(B)}{s_{\min}^2(B)}.$$

Thus, $R(B)$ measures how strongly B emphasizes a coordinated, sequence-wide gradient relative to an average token-local gradient. In conventional language modeling, where target positions are represented independently ($B = I$), the two have equal relative importance and $R(B) = 1$.

Target coupling burden. Sharing control points can make distinct token-wise corrections harder to realize. For a desired embedding correction ΔE, we therefore define

$$\mathcal{B}_B(\Delta E) := \min_{\Delta P:\, \Delta P B = \Delta E} \|\Delta P\|_F^2,$$

which measures the minimum control point-space change required to realize ΔE; we interpret this quantity as the *correction burden*.

Lemma 2 (Target Coupling Burden (Simplified)). *For full-column-rank $B = U\Sigma V^\top$ with singular values $s_1 \ge \cdots \ge s_L > 0$,*

$$\mathcal{B}_B(\Delta E) = \|\Delta E B^+\|_F^2 = \sum_{r=1}^{L} \frac{\|\Delta E \mathbf{v}_r\|_2^2}{s_r^2}, \qquad \sup_{\Delta E \neq 0} \frac{\mathcal{B}_B(\Delta E)}{\|\Delta E\|_F^2} = \frac{1}{s_{\min}^2(B)}.$$

Corrections along small-singular-value directions therefore incur greater burden. This is particularly relevant when neighboring targets share control points but require different token-wise corrections.

Representational enrichment. Additional representation capacity can alleviate this burden.

Remark 1 (Representational Enrichment (Simplified)). *For an enriched representation $\widetilde{B} = [B^\top, C^\top]^\top \in \mathbb{R}^{(N+M)\times L}$,*

$$\mathcal{B}_{\widetilde{B}}(\Delta E) \le \mathcal{B}_B(\Delta E), \qquad s_{\min}(\widetilde{B}) \ge s_{\min}(B).$$

Although increasing B-spline control points does not strictly satisfy this nested-enrichment setting, the expanded control point space may similarly reduce the correction burden through greater representation capacity. Together, Lemma 2 and Remark 1 motivate a *coupling–capacity trade-off*: spline degree controls the extent of local sharing, while the number of control points controls representation capacity. We examine this trade-off empirically in Appendix D.8.

5 Experiments and Results

Experiments cover machine translation and language modeling. For machine translation, we compare SCLM with Gaussian DLMs and analyze coordinated generation across target positions. For language modeling, we evaluate SCLM on LM1B against discrete DLMs, including MDLM and SoftMask, using generation-quality and diversity metrics. We also conduct preliminary BD3-LM experiments to examine transfer to a different DLM architecture and block-wise longer-context generation.

5.1 Experimental Setup

For machine translation, we use IWSLT14 and WMT14 English–German (Cettolo et al., 2014; Bojar et al., 2014) in both directions. For KD, we generate IWSLT14 distilled data using our AR Transformer and use the WMT14 distilled data from Gao et al. (2024). For language modeling, we use LM1B (Chelba et al., 2013). Preprocessing and dataset statistics are provided in Appendix C.1.

SCLM uses two sentence curve hyperparameters: the number of control points N and spline degree η. We set $N = \text{round}(LN_{\text{ratio}})$ and $\eta = \max(\text{round}(N\eta_{\text{ratio}}), 2)$, or use a fixed η_{fix}. Sampling

Table 1: SacreBLEU (↑) on IWSLT14 and WMT14. Best and second-best DLM results are shown in **bold** and underlined. Scores are from original papers unless marked by superscripts or evaluated by us (Transformer and Differmer). Superscripts */†/‡/§ indicate results reported in Ye et al. (2023a); Gao et al. (2024); Wu et al. (2023); Arriola et al. (2025b).

Model	IWSLT14 En→De	IWSLT14 De→En	WMT14 En→De	WMT14 De→En
AR Transformer (beam=1)	27.19	32.89	26.57	31.72
AR Transformer (beam=5)	28.24	33.83	27.71	33.00
DLM baselines				
DiffusionLM (Li et al., 2022)*	–	29.11	17.41	–
CDCD (Dieleman et al., 2022)†	–	–	19.70	25.40
SeqDiffuSeq (Yuan et al., 2022)	22.12	30.45	19.76	23.93
DINOISER (Ye et al., 2023b)	26.14	31.61	25.88	30.30
GENIE (Lin et al., 2023)‡	23.89	29.45	–	–
AR-Diffusion (Wu et al., 2023)	26.01	31.80	–	–
RDM (Zheng et al., 2023)*	–	32.14	26.54	–
MDLM (Sahoo et al., 2024)§	–	–	18.40	–
Differmer (Gao et al., 2024) w/o KD	22.99	28.58	–	–
Differmer (Gao et al., 2024) w/ KD	27.52	32.42	26.98	32.12
WDR (Heo et al., 2024)	26.26	31.83	–	–
E2D2 (Arriola et al., 2025b)	–	–	24.80	–
NeoDiff (Li et al., 2025)	–	32.20	24.64	–
SCLM (ours) w/o KD	26.04	31.07	–	–
SCLM (ours) w/ KD	**28.10**	**32.56**	**28.19**	**33.49**

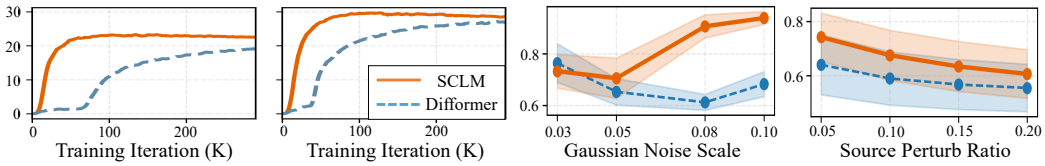


Figure 4: Additional IWSLT14 analyses. **Left:** Validation SacreBLEU without KD. **Right:** Distance correlation under conditioning perturbations.

locations Γ are L uniformly spaced points in $[0.01, 0.99]$, and B and B^+ are precomputed for each sentence length and spline configuration.

For machine translation, we build SCLM on Differmer (Gao et al., 2024). For LM1B, we evaluate fully non-AR SCLM on the MDLM/SoftMask framework (Sahoo et al., 2024; Hersche et al., 2025) and conduct preliminary semi-AR experiments on BD3-LM (Arriola et al., 2025a). Guided by Remark 1, we focus on $N_{\text{ratio}} \geq 2.0$ and select remaining spline configurations by validation performance. Sentence curve mappings introduce no trainable parameters. Detailed configurations and hyperparameters are provided in Appendix C.2 and Tables 5–6.

5.2 Conditional Generation: Machine Translation

Main translation results. We evaluate with SacreBLEU (Post, 2018)[1] on IWSLT14 and WMT14 in both directions. We report prior DLM results and re-evaluate Transformer and Differmer under our setting. As shown in Table 1, SCLM achieves the best DLM performance across all four tasks. In addition to absolute SacreBLEU scores, Appendix D.2 compares each DLM with its corresponding AR baseline, where SCLM still retains a strong relative advantage.

Compared with Differmer, SCLM gains $0.58/0.14$ SacreBLEU on distilled IWSLT14 En→De/De→En, but substantially larger $1.21/1.37$ gains on WMT14. It also surpasses our beam-5 AR Transformer on both WMT14 directions (28.19 vs. 27.71 and 33.49 vs. 33.00), while remaining

[1]Signature: `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1`

Table 2: Generic language modeling results on LM1B. **Top:** Fully non-AR generation across NFEs. **Bottom:** Preliminary semi-AR BD3-LM experiments varying active block size with $C = 128$ (left) and context length with block $= 4$ (right). We report Gen. PPL (↓), self-BLEU (↓), and MAUVE (↑).

Model	NFE=64			NFE=128			NFE=256			NFE=512		
	PPL	sBLEU	MAUVE	PPL	sBLEU	MAUVE	PPL	sBLEU	MAUVE	PPL	sBLEU	MAUVE
MDLM	62.81	7.20	0.0674	38.58	11.25	0.0708	33.54	14.45	0.0824	22.82	19.74	0.1204
SoftMask	**52.07**	8.75	0.0870	34.91	10.85	0.1111	26.21	13.57	0.1092	20.25	14.66	0.1274
SCLM (ours)	53.11	8.23	**0.1140**	**34.29**	10.15	**0.1320**	**24.41**	12.79	**0.1308**	**20.09**	14.11	**0.1508**

block	BD3-LM			SCLM (ours)		
	PPL	sBLEU	MAUVE	PPL	sBLEU	MAUVE
4	203.80	25.56	0.7837	**192.68**	26.25	**0.8421**
8	219.58	25.08	0.7303	**219.46**	25.47	**0.8008**
16	**224.87**	25.12	0.7440	242.34	24.58	**0.7593**

cont	BD3-LM			SCLM (ours)		
	PPL	sBLEU	MAUVE	PPL	sBLEU	MAUVE
128	203.80	25.56	0.7837	**192.68**	26.25	**0.8421**
256	158.71	31.13	0.8075	**156.46**	31.17	**0.8478**
512	**132.52**	36.71	0.7763	135.38	36.16	**0.8688**

substantially faster (Table 8). Without KD, SCLM reaches $26.04/31.07$ on IWSLT14, improving over Differmer ($22.99/28.58$) by $3.05/2.49$ points and showing a clear training advantage in Figure 4 (Left). Thus, the modest IWSLT14 gains under KD become much larger on the original data.

A possible explanation is the remaining complexity of the target distribution. KD reduces target variation and multimodality, potentially leaving less room for improvement on the relatively homogeneous IWSLT14 corpus (Cettolo et al., 2014). Without KD, greater target variation remains; similarly, the larger and more diverse WMT14 corpus (Bojar et al., 2014) may retain substantial variation even after distillation. This empirical pattern is consistent with our motivation that target-side coupling can be particularly useful under more complex or multimodal target distributions. Figure 1 (right) further visualizes sentence curve generation during denoising, where predicted control points progressively map toward the target embedding sequence. Additional examples are provided in Appendix D.6.

Coordinated generation analyses. We examine whether target-side dependency leads to more coordinated prediction across target positions. Our intuition is that, if positions are generated coordinately, perturbing the shared conditioning context should induce related rather than independent changes across them. We therefore measure statistical dependence among target-position responses using distance correlation (dCor) (Székely et al., 2007) under Gaussian perturbations to encoder hidden states and random source-token masking. As shown in Figure 4 (Right), SCLM generally yields higher dCor than Differmer, indicating more coordinated target-side responses. Appendix D.4 details the protocol and shows that the theoretically derived quantities $R(B)$ and $\kappa(B)$ correlate positively with dCor, providing empirical support for the connection between sentence curve geometry and coordinated prediction. Appendix D.5 further provides qualitative examples in which Differmer produces incompatible or locally inconsistent phrases, whereas SCLM yields more coherent outputs.

5.3 Generic Language Modeling Experiment Results

Fully non-autoregressive generation. We evaluate SCLM on LM1B against MDLM (Sahoo et al., 2024) and SoftMask (Hersche et al., 2025), using 1,024 samples over three seeds. We report Gen. PPL, self-BLEU (Zhu et al., 2018), and MAUVE (Pillutla et al., 2021), with Gen. PPL computed by `GPT2-Large`. We regard MAUVE as the primary metric, as prior DLM work shows that Gen. PPL can be spuriously driven very low by degenerate repetition (e.g., "love love love..."), despite poor diversity (Wang et al., 2025). As shown in Table 2, SCLM achieves the highest MAUVE at every NFE. Compared with SoftMask, it also obtains lower Gen. PPL at NFE=$128/256/512$, while its lower self-BLEU indicates greater sample diversity. These results extend the benefit of sentence curve prediction beyond conditional translation to generic language modeling.

Semi-autoregressive block diffusion. We further apply SCLM to BD3-LM to test transfer to a different generation architecture, using preliminary tiny-DiT experiments with 100K updates. Table 2 shows that SCLM improves MAUVE across all tested block sizes and context lengths, while Gen. PPL is mixed across settings. Because sentence curves are applied only within the active denoising block, their representation size remains bounded by the block size; the consistent MAUVE gains from

context lengths 128 to 512 therefore suggest a promising block-local route toward longer-context generation. Appendix D.8 further shows that performance depends jointly on the number of control points and spline degree, broadly supporting the coupling–capacity trade-off in Section 4.3.

6 Related Works

6.1 Curves in Machine Learning

Spline-based functions, including B-splines and Bézier curves, have been used in models for graph and 3D meshing (Fey et al., 2018), lane detection (Feng et al., 2022), and multi-step time-series prediction (Hug et al., 2020). Their local and overlapping basis functions provide a structural inductive bias. SCLM applies this idea to language modeling. Nearby target embeddings share more control points than distant ones, providing a sentence-independent positional prior that can be reused across the corpus. This locality is broadly consistent with the tendency of syntactic dependencies to occur over short distances (Niu et al., 2023). Unlike prior curve-based prediction methods, SCLM uses B-spline control points as an intermediate target representation of token sequences. To our knowledge, this is the first use of spline curves as sentence target representations for language modeling.

6.2 Target Reformulation and Why Sentence Curves?

Alternative target representations for language modeling remain underexplored. Word difference representations (WDR) (Heo et al., 2024) introduce dependency between neighboring target embeddings while preserving recoverability, but are used as auxiliary targets within an N-gram prediction framework. SCLM instead directly processes and predicts a shared control point representation while retaining an explicit mapping back to the original embedding sequence. This allows the original DLM objective to be preserved, unlike contextual representations such as BERT (Devlin et al., 2019), which are not explicitly designed to be invertible and would require an additional decoder.

Our analysis suggests that target-side coupling should be considered together with representation capacity: sharing variables can increase the burden of realizing token-specific corrections, while additional capacity can alleviate it. Simple smoothing operations and WDR introduce local dependency but do not provide a capacity control corresponding to its strength. Sentence curves provide both: spline degree controls sharing, while the number of control points controls representation capacity.

7 Conclusion and Discussion

We introduce sentence curves, a continuous target representation in which neighboring target embeddings share B-spline control points, and propose sentence curve language models (SCLMs) that predict through this shared representation. Our theoretical analysis shows that sentence curve mapping changes the geometry of token-wise gradients and that stronger target-side coupling can increase the burden of realizing distinct token-specific corrections unless sufficient representation capacity is provided. Empirically, SCLM achieves strong machine translation performance, including improvements over autoregressive Transformers on WMT14, and remains effective without knowledge distillation. On LM1B, SCLM also improves generic language generation in standard-scale fully non-autoregressive models and shows promising preliminary results when instantiated on the semi-autoregressive BD3-LM architecture. Together with the coordinated-generation analyses, these results support sentence curve prediction as a useful target-side inductive bias for diffusion language modeling.

As an initial study, several directions remain open. First, using more control points than tokens increases computation, motivating architectures that restrict control point processing to selected layers or skip redundant control points. Second, although our theory interprets spline degree as controlling target-side coupling and the number of control points as controlling representation capacity, a more explicit characterization of their trade-off could simplify hyperparameter selection. Finally, the current sentence-independent B-spline basis provides only a positional form of dependency; sentence-adaptive structures informed by syntax, semantics, or discourse may provide richer target-side coupling while preserving sufficient capacity for token-specific distinctions.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing and presentation, parts of mathematical formulation and verification, software implementation, experimental workflows. The machine translation and fully non-autoregressive language modeling implementations were developed primarily by the authors, while generative AI substantially assisted the implementation and experimentation of the BD3-LM extension. AI-generated or AI-assisted mathematical arguments, code, experimental analyses, and text were carefully reviewed by the authors; code used for reported experiments was executed and tested, numerical results and figures were checked against the underlying outputs, and mathematical claims were independently examined before inclusion. We take responsibility for all final text, claims, code, figures, and experimental results in this work.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide anonymized reference implementations and experiment configurations. Experimental settings and additional analyses are described in the main paper and appendix, while complete proofs of the theoretical results are provided in Appendix A.3.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

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*, 2025a.

Marianne Arriola, Yair Schiff, Hao Phung, Aaron Gokaslan, and Volodymyr Kuleshov. Encoder-decoder diffusion language models for efficient training and inference. *arXiv preprint arXiv:2510.22852*, 2025b.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia. Proceedings of the ninth workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th iwslt evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pp. 2–17, 2014.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

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.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. *arXiv preprint arXiv:2502.09622*, 2025.

Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17062–17070, 2022.

Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 869–877, 2018.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Empowering diffusion models on the embedding space for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4664–4683, 2024.

Gemini, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

DongNyeong Heo and Heeyoul Choi. Shared latent space by both languages in non-autoregressive neural machine translation. *arXiv preprint arXiv:2305.03511*, 2023.

DongNyeong Heo, Daniela Noemi Rim, and Heeyoul Choi. N-gram prediction and word difference representations for language modeling. *arXiv preprint arXiv:2409.03295*, 2024.

Michael Hersche, Samuel Moor-Smith, Thomas Hofmann, and Abbas Rahimi. Soft-masked diffusion language models. *arXiv preprint arXiv:2510.17206*, 2025.

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

Fei Huang, Tianhua Tao, Hao Zhou, Lei Li, and Minlie Huang. On the learning of non-autoregressive transformers. In *International conference on machine learning*, pp. 9356–9376. PMLR, 2022.

Ronny Hug, Wolfgang Hübner, and Michael Arens. Introducing probabilistic bézier curves for n-step sequence prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10162–10169, 2020.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv preprint arXiv:1802.06901*, 2018.

Bocheng Li, Zhujin Gao, and Linli Xu. Unifying continuous and discrete text diffusion with non-simultaneous diffusion processes. *arXiv preprint arXiv:2505.22165*, 2025.

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.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning*, pp. 21051–21064. PMLR, 2023.

Jinyang Liu, Yujia Zhai, and Zizhong Chen. Normalization of input-output shared embeddings in text generation models. *arXiv preprint arXiv:2001.07885*, 2020.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. Spherical text embedding. *Advances in neural information processing systems*, 32, 2019.

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.

Ruochen Niu, Yaqin Wang, and Haitao Liu. The cross-linguistic variations in dependency distance minimization and its potential explanations. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pp. 559–569, 2023.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pp. 8846–8853, 2020.

Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. 2007.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Quanquan Gu. Diffusion language models can perform many tasks with scaling and instruction-finetuning. *arXiv preprint arXiv:2308.12219*, 2023a.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. Dinoiser: Diffused conditional sequence learning by manipulating noises. *arXiv preprint arXiv:2302.10025*, 2023b.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*, 2022.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100, 2018.

A Mathematical Background

A.1 Notational Preliminaries

Let $\mathcal{V}$ denote the vocabulary and $E = [\mathbf{e}_1, \dots, \mathbf{e}_{|\mathcal{V}|}] \in \mathbb{R}^{d\times|\mathcal{V}|}$ the word embedding matrix. For a sentence $Y = (y_1, \dots, y_L)$, we denote its embedding sequence by

$$E_Y = [\mathbf{e}_{y_1}, \dots, \mathbf{e}_{y_L}] \in \mathbb{R}^{d\times L},$$

where L and d denote sequence length and embedding dimension.

For discrete models, $W = [\mathbf{w}_1, \dots, \mathbf{w}_{|\mathcal{V}|}] \in \mathbb{R}^{d\times|\mathcal{V}|}$ denotes the output projection matrix, $W_Y = [\mathbf{w}_{y_1}, \dots, \mathbf{w}_{y_L}]$ the corresponding target projection vectors, and $H_\theta = [\mathbf{h}_{\theta,1}, \dots, \mathbf{h}_{\theta,L}]$ the backbone outputs. We use θ for trainable model parameters.

For sentence curves, N denotes the number of control points, η the B-spline degree, and $\Gamma = (\gamma_1, \dots, \gamma_L)$ the sampling locations. The basis matrix is $B \in \mathbb{R}^{N\times L}$, the target control point representation is $P_Y \in \mathbb{R}^{d\times N}$, and its model prediction is $\hat{P}_\theta \in \mathbb{R}^{d\times N}$, with $E_Y = P_Y B$. We denote the Moore–Penrose pseudoinverse of a matrix B by B^+.

We use $\mathbf{1}_n \in \mathbb{R}^n$ for the all-ones vector, $\boldsymbol{\delta}_i^{(n)} \in \mathbb{R}^n$ for the i-th standard basis vector, and I_n for the $n \times n$ identity matrix. For a sentence Y, let

$$\Delta_Y = [\boldsymbol{\delta}_{y_1}^{(|\mathcal{V}|)}, \dots, \boldsymbol{\delta}_{y_L}^{(|\mathcal{V}|)}] \in \{0,1\}^{|\mathcal{V}|\times L}$$

denote its one-hot representation, so that $E_Y = E\Delta_Y$.

For matrices, $\|\cdot\|_F$ denotes the Frobenius norm, $\langle A, C\rangle_F = \mathrm{tr}(A^\top C)$ the Frobenius inner product, and $\mathrm{tr}(\cdot)$ and $\mathrm{rank}(\cdot)$ the trace and rank. For vectors, $\|\cdot\|_2$ denotes the Euclidean norm. We denote the singular values of A by $s_1(A) \geq \cdots \geq s_r(A)$ and its largest and smallest nonzero singular values by $s_{\max}(A)$ and $s_{\min}(A)$. For symmetric matrices, $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the extreme eigenvalues, and $A \succeq C$ means that $A - C$ is positive semidefinite.

When conditioning is present, X denotes the conditioning input, p_{data} the data distribution, q the forward diffusion process, and t the diffusion timestep.

A.2 Assumptions

Assumption 1 (Unit Norm Constraint)**.**

$$\|\mathbf{w}_i\|_2 = \|\mathbf{h}\|_2 = 1, \qquad \forall i \in \{1, \dots, |\mathcal{V}|\}.$$

This assumption constrains the target embedding vectors and backbone output to the unit sphere $\mathbb{S}^{d-1}$, allowing the analysis to focus on directional geometry rather than scale. Such spherical constraints have been used not only as mathematical simplifications, but also as effective modeling assumptions for understanding the geometry of learned text representations and their empirical behavior (Meng et al., 2019; Liu et al., 2020).

Assumption 2 (Local Isotropy on the Target Embedding Tangent Space)**.** *Given a target token y, decompose each target embedding vector as*

$$\mathbf{w}_i = a_i \mathbf{w}_y + \mathbf{u}_i, \qquad a_i = \mathbf{w}_y^\top \mathbf{w}_i, \qquad \mathbf{u}_i = \mathbf{w}_i - a_i \mathbf{w}_y.$$

We assume that the tangent components are locally centered conditioned on their similarity to the target:

$$\mathbb{E}[\mathbf{u}_i \mid a_i] = \mathbf{0}.$$

This assumption captures a locally symmetric geometry around the target embedding direction. Stronger assumptions, such as isotropic or approximately uniform embedding spaces, have been successfully used to make high-dimensional word embedding geometry analytically tractable and to explain empirical properties of learned representations (Arora et al., 2016). In particular, Arora et al. (2016) provide empirical support for the approximate uniform dispersion of latent word vectors and use this assumption to explain observed algebraic structure in word embeddings. Our assumption adopts this geometric intuition locally on the tangent space of the target embedding.

Assumption 3 (Full-Rank Sentence Curve Mapping). *The sentence curve basis matrix $B \in \mathbb{R}^{N\times L}$ satisfies*

$$\operatorname{rank}(B) = L.$$

Accordingly, $B^\top B$ is positive definite and $B^+ = (B^\top B)^{-1}B^\top$ satisfies $B^+B = I_L$.

A.3 Proofs of Theorems and Lemmas

Proposition 2 (Target Embedding Alignment). *Under Assumptions 1 and 2, the target embedding matrix $W_Y = [\mathbf{w}_{y_1}, \ldots, \mathbf{w}_{y_L}]$ is an optimal solution of the Token Classification Pipeline (Definition 1):*

$$W_Y = \arg\min_{H_\theta} CE_Y(\theta) = \arg\min_{H_\theta} \left[-\frac{1}{L}\sum_{i=1}^{L} \log \operatorname{softmax}(W^\top \mathbf{h}_{\theta,i})_{y_i} \right]. \tag{5}$$

Proof. Without loss of generality, we focus on the minimization for a single target token $y \in Y$:

$$\arg\min_{\mathbf{h}} -\log \operatorname{softmax}(W^\top \mathbf{h})_y,$$

where we omit θ for brevity.

To derive the optimal solution under the constraint $\|\mathbf{h}\|_2^2 = 1$ (Assumption 1), we use a Lagrange multiplier. Let

$$\begin{aligned} \mathcal{L}(\mathbf{h}, \lambda) &= f(\mathbf{h}) - \lambda(\|\mathbf{h}\|_2^2 - 1), \\ \nabla_{\mathbf{h}}\mathcal{L}(\mathbf{h}, \lambda) &= \nabla_{\mathbf{h}} f(\mathbf{h}) - 2\lambda\mathbf{h}, \end{aligned}$$

where

$$\begin{aligned} f(\mathbf{h}) &= -\log \operatorname{softmax}(W^\top \mathbf{h})_y \\ &= -\mathbf{w}_y^\top \mathbf{h} + \log \sum_k \exp(\mathbf{w}_k^\top \mathbf{h}). \end{aligned}$$

The stationary condition therefore satisfies

$$\nabla_{\mathbf{h}} f(\mathbf{h}) = 2\lambda\mathbf{h}.$$

Thus, $\nabla_{\mathbf{h}} f(\mathbf{h})$ is parallel to $\mathbf{h}$, and its tangent component with respect to $\mathbf{h}$ should vanish:

$$\nabla_{\mathbf{h}} f(\mathbf{h}) - \left(\mathbf{h}^\top \nabla_{\mathbf{h}} f(\mathbf{h})\right)\mathbf{h} = 0.$$

Since

$$\nabla_{\mathbf{h}} f(\mathbf{h}) = -\mathbf{w}_y + \sum_k p(y=k|\mathbf{h})\mathbf{w}_k,$$

the stationary condition becomes

$$-\mathbf{w}_y + \sum_k p(y=k|\mathbf{h})\mathbf{w}_k - \mathbf{h}^\top \left(-\mathbf{w}_y + \sum_k p(y=k|\mathbf{h})\mathbf{w}_k\right)\mathbf{h} = 0. \tag{6}$$

We now examine our conjecture $\mathbf{h}^* = \mathbf{w}_y$. Substituting it into Eq. 6 gives

$$\sum_k p(y=k|\mathbf{w}_y)\left(\mathbf{w}_k - (\mathbf{w}_y^\top \mathbf{w}_k)\mathbf{w}_y\right) = 0. \tag{7}$$

Given the target embedding vector $\mathbf{w}_y$, each target embedding vector can be decomposed into parallel and tangent components:

$$\mathbf{w}_k = (\mathbf{w}_y^\top \mathbf{w}_k)\mathbf{w}_y + \mathbf{u}_k, \qquad \mathbf{u}_k = \mathbf{w}_k - (\mathbf{w}_y^\top \mathbf{w}_k)\mathbf{w}_y.$$

Let $a_k = \mathbf{w}_y^\top \mathbf{w}_k$. At $\mathbf{h} = \mathbf{w}_y$, the softmax weight $p(y = k|\mathbf{w}_y)$ depends only on a_k. Therefore, by Assumption 2, $\mathbb{E}[\mathbf{u}_k|a_k] = 0$, the softmax-weighted tangent components cancel:

$$\sum_k p(y = k|\mathbf{w}_y)\mathbf{u}_k = 0.$$

Hence, Eq. 7 is satisfied at $\mathbf{h}^* = \mathbf{w}_y$.

Finally, $f(\mathbf{h})$ is convex because it consists of a linear term and a log-sum-exp term. At $\mathbf{h} = \mathbf{w}_y$, its gradient is parallel to $\mathbf{w}_y$, with

$$\nabla_{\mathbf{h}} f(\mathbf{w}_y) = \left(-1 + \sum_k p(y = k|\mathbf{w}_y)\mathbf{w}_y^\top \mathbf{w}_k \right) \mathbf{w}_y.$$

Since $\mathbf{w}_y^\top \mathbf{w}_k \leq 1$ under Assumption 1, this point satisfies the optimality condition of the corresponding unit-ball constrained convex problem and is therefore an optimal solution under the unit-norm constraint.

Applying the result to every position $i = 1, \ldots, L$ gives $H_\theta = W_Y$, completing the proof. □

Lemma 3 (Gradient Reweighting Induced by B)**.** *Let*

$$G := \nabla_{\hat{E}_{Y,\theta}} CE_Y(\theta) \in \mathbb{R}^{d \times L}$$

denote the gradient of the cross-entropy objective with respect to the predicted target embeddings, where $\hat{E}_{Y,\theta} = \hat{P}_\theta B$. Then,

$$\nabla_{\hat{P}_\theta} CE_Y(\theta) = GB^\top.$$

Let $Q := B^\top B$ and define the propagated gradient magnitude as

$$I_B(G) := \|GB^\top\|_F^2 = \|GQ^{1/2}\|_F^2.$$

Consider the normalized sequence-wide and token-local gradient patterns

$$G_{\text{seq}} = \frac{\mathbf{a}\mathbf{1}_L^\top}{\sqrt{L}}, \qquad G_{\text{local},i} = \mathbf{a}\delta_i^\top, \qquad \|\mathbf{a}\|_2 = 1.$$

Under Assumption 3, their relative propagated gradient magnitude satisfies

$$R(B) := \frac{I_B(G_{\text{seq}})}{\frac{1}{L}\sum_{i=1}^L I_B(G_{\text{local},i})} = \frac{\mathbf{1}_L^\top Q \mathbf{1}_L}{\text{tr}(Q)} \leq \frac{s_{\max}^2(B)}{s_{\min}^2(B)},$$

where $s_{\max}(B)$ and $s_{\min}(B)$ are the largest and smallest singular values of B, respectively.

Proof. By definition,

$$G = \nabla_{\hat{E}_{Y,\theta}} CE_Y(\theta).$$

For a scalar-valued function of a matrix, its differential is given by the Frobenius inner product between its gradient and the differential of its argument:

$$dCE_Y = \left\langle G, d\hat{E}_{Y,\theta} \right\rangle_F.$$

Since

$$\hat{E}_{Y,\theta} = \hat{P}_\theta B$$

and B is fixed,

$$d\hat{E}_{Y,\theta} = d\hat{P}_\theta B.$$

Therefore,

$$\begin{aligned} dCE_Y &= \left\langle G, d\hat{P}_\theta B \right\rangle_F \\ &= \text{tr}\left(G^\top d\hat{P}_\theta B \right) \\ &= \text{tr}\left(BG^\top d\hat{P}_\theta \right) \\ &= \left\langle GB^\top, d\hat{P}_\theta \right\rangle_F. \end{aligned}$$

Hence,

$$\nabla_{\hat{P}_\theta} CE_Y(\theta) = GB^\top.$$

The squared Frobenius norm of this propagated gradient is

$$\begin{aligned} I_B(G) &= \|GB^\top\|_F^2 \\ &= \operatorname{tr}\left(GB^\top BG^\top\right) \\ &= \operatorname{tr}\left(GQG^\top\right), \end{aligned}$$

where $Q = B^\top B$. Since Q is positive definite under Assumption 3,

$$\operatorname{tr}\left(GQG^\top\right) = \|GQ^{1/2}\|_F^2.$$

We next consider the sequence-wide gradient pattern

$$G_{\text{seq}} = \frac{\mathbf{a}\mathbf{1}_L^\top}{\sqrt{L}}.$$

Using $\|\mathbf{a}\|_2 = 1$,

$$\begin{aligned} I_B(G_{\text{seq}}) &= \operatorname{tr}\left(G_{\text{seq}} Q G_{\text{seq}}^\top\right) \\ &= \frac{1}{L}\operatorname{tr}\left(\mathbf{a}\mathbf{1}_L^\top Q \mathbf{1}_L \mathbf{a}^\top\right) \\ &= \frac{1}{L}\mathbf{1}_L^\top Q \mathbf{1}_L. \end{aligned}$$

For a token-local gradient pattern at position i,

$$G_{\text{local},i} = \mathbf{a}\delta_i^\top,$$

we similarly obtain

$$\begin{aligned} I_B(G_{\text{local},i}) &= \operatorname{tr}\left(\mathbf{a}\delta_i^\top Q \delta_i \mathbf{a}^\top\right) \\ &= \delta_i^\top Q \delta_i. \end{aligned}$$

Averaging over all L positions gives

$$\frac{1}{L}\sum_{i=1}^{L} I_B(G_{\text{local},i}) = \frac{1}{L}\sum_{i=1}^{L} \delta_i^\top Q \delta_i = \frac{\operatorname{tr}(Q)}{L}.$$

Therefore,

$$R(B) = \frac{\mathbf{1}_L^\top Q \mathbf{1}_L}{\operatorname{tr}(Q)}.$$

Finally, we bound this ratio using the eigenvalues of Q. Since $\|\mathbf{1}_L\|_2^2 = L$, the Rayleigh quotient gives

$$\mathbf{1}_L^\top Q \mathbf{1}_L \le L\lambda_{\max}(Q).$$

Moreover,

$$\operatorname{tr}(Q) = \sum_{r=1}^{L} \lambda_r(Q) \ge L\lambda_{\min}(Q).$$

It follows that

$$R(B) \le \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)}.$$

Because $Q = B^\top B$, its eigenvalues are the squared singular values of B:

$$\lambda_{\max}(Q) = s_{\max}^2(B), \qquad \lambda_{\min}(Q) = s_{\min}^2(B).$$

Therefore,

$$R(B) \le \frac{s_{\max}^2(B)}{s_{\min}^2(B)}.$$

□

Lemma 4 (Target Coupling Burden). *Let $B \in \mathbb{R}^{N\times L}$ have full column rank and consider the sentence curve mapping $E = PB$. For a desired target embedding correction $\Delta E \in \mathbb{R}^{d\times L}$, define its minimum control point correction burden as*

$$\mathcal{B}_B(\Delta E) := \min_{\Delta P:\, \Delta P B = \Delta E} \|\Delta P\|_F^2.$$

Let $B = U\Sigma V^\top$ be the compact SVD of B, with singular values $s_1 \geq \cdots \geq s_L > 0$. Then

$$\mathcal{B}_B(\Delta E) = \|\Delta E B^+\|_F^2 = \sum_{r=1}^{L} \frac{\|\Delta E \mathbf{v}_r\|_2^2}{s_r^2},$$

and

$$\sup_{\Delta E \neq 0} \frac{\mathcal{B}_B(\Delta E)}{\|\Delta E\|_F^2} = \frac{1}{s_{\min}^2(B)}.$$

Proof. Since B has full column rank, its Moore–Penrose pseudoinverse is

$$B^+ = (B^\top B)^{-1} B^\top, \qquad B^+ B = I_L.$$

Therefore,

$$\Delta P^* := \Delta E B^+$$

is a feasible control point correction, because

$$\Delta P^* B = \Delta E B^+ B = \Delta E.$$

We first show that ΔP^* has minimum Frobenius norm among all feasible corrections. Let ΔP be any correction satisfying $\Delta P B = \Delta E$. Multiplying both sides by B^+ gives

$$\Delta P B B^+ = \Delta E B^+.$$

Define

$$\Pi_B := B B^+,$$

which is the orthogonal projector onto the column space of B. The above identity can therefore be written as

$$\Delta P \Pi_B = \Delta E B^+.$$

Using

$$I_N = \Pi_B + (I_N - \Pi_B),$$

we can decompose any feasible correction as

$$\begin{aligned} \Delta P &= \Delta P \Pi_B + \Delta P (I_N - \Pi_B) \\ &= \Delta E B^+ + \Delta P (I_N - \Pi_B). \end{aligned}$$

The second component does not affect the target embedding correction, since

$$(I_N - \Pi_B) B = B - B B^+ B = 0.$$

Hence, $\Delta P(I_N - \Pi_B)$ represents an additional change in the control point space that produces no change after mapping through B.

Moreover, Π_B and $I_N - \Pi_B$ are orthogonal projectors, so the two components in the decomposition are orthogonal under the Frobenius inner product. Therefore,

$$\|\Delta P\|_F^2 = \|\Delta E B^+\|_F^2 + \|\Delta P (I_N - \Pi_B)\|_F^2 \geq \|\Delta E B^+\|_F^2.$$

The minimum is attained when the second component is zero. Thus,

$$\Delta P^* = \Delta E B^+, \qquad \mathcal{B}_B(\Delta E) = \|\Delta E B^+\|_F^2.$$

We next characterize this burden through the singular directions of B. Let

$$B = U\Sigma V^\top, \qquad \Sigma = \mathrm{diag}(s_1, \ldots, s_L)$$

be the compact SVD of B. Its pseudoinverse is

$$B^+ = V\Sigma^{-1}U^\top.$$

Therefore,

$$\begin{aligned}\mathcal{B}_B(\Delta E) &= \|\Delta E V\Sigma^{-1}U^\top\|_F^2 \\ &= \|\Delta E V\Sigma^{-1}\|_F^2 \\ &= \sum_{r=1}^{L}\frac{\|\Delta E\mathbf{v}_r\|_2^2}{s_r^2},\end{aligned}$$

where $\mathbf{v}_r$ denotes the r-th right singular vector of B. Thus, the control point correction required for each target-space direction is weighted inversely by the squared singular value associated with that direction.

Finally, we characterize the worst-case normalized correction burden. From the singular-direction decomposition above,

$$\mathcal{B}_B(\Delta E) = \sum_{r=1}^{L}\frac{\|\Delta E\mathbf{v}_r\|_2^2}{s_r^2}.$$

Since $V = [\mathbf{v}_1, \ldots, \mathbf{v}_L]$ is orthogonal, multiplication by V preserves the Frobenius norm, and therefore

$$\|\Delta E\|_F^2 = \|\Delta E V\|_F^2 = \sum_{r=1}^{L}\|\Delta E\mathbf{v}_r\|_2^2.$$

For any nonzero ΔE, define

$$\alpha_r := \frac{\|\Delta E\mathbf{v}_r\|_2^2}{\|\Delta E\|_F^2}.$$

By construction,

$$\alpha_r \geq 0, \qquad \sum_{r=1}^{L}\alpha_r = 1.$$

Hence, the normalized correction burden can be written as

$$\frac{\mathcal{B}_B(\Delta E)}{\|\Delta E\|_F^2} = \sum_{r=1}^{L}\alpha_r\frac{1}{s_r^2}.$$

That is, the normalized burden is a convex combination of $1/s_1^2, \ldots, 1/s_L^2$. Since

$$s_1 \geq \cdots \geq s_L = s_{\min}(B),$$

we have

$$\frac{1}{s_r^2} \leq \frac{1}{s_{\min}^2(B)} \qquad \forall r.$$

Therefore, for every nonzero target embedding correction,

$$\frac{\mathcal{B}_B(\Delta E)}{\|\Delta E\|_F^2} \leq \frac{1}{s_{\min}^2(B)}.$$

It remains to show that this upper bound is attainable. Let $\mathbf{v}_{\min}$ be a right singular vector associated with $s_{\min}(B)$, and choose

$$\Delta E = \mathbf{a}\mathbf{v}_{\min}^\top$$

for any nonzero $\mathbf{a} \in \mathbb{R}^d$. Since the right singular vectors are orthonormal,

$$\Delta E\mathbf{v}_r = \begin{cases}\mathbf{a}, & \mathbf{v}_r = \mathbf{v}_{\min}, \\ \mathbf{0}, & \text{otherwise.}\end{cases}$$

Thus, all the weight in the convex combination is assigned to the minimum-singular-value direction, yielding

$$\frac{\mathcal{B}_B(\Delta E)}{\|\Delta E\|_F^2} = \frac{1}{s_{\min}^2(B)}.$$

Hence, $1/s_{\min}^2(B)$ is both an upper bound for every feasible ΔE and is itself attained by a feasible correction. Consequently,

$$\sup_{\Delta E \neq 0} \frac{\mathcal{B}_B(\Delta E)}{\|\Delta E\|_F^2} = \frac{1}{s_{\min}^2(B)}.$$

□

Remark 2 (Representational Enrichment)**.** *Consider an enriched control point representation*

$$\widetilde{B} = \begin{bmatrix} B \\ C \end{bmatrix} \in \mathbb{R}^{(N+M)\times L},$$

where $C \in \mathbb{R}^{M\times L}$ introduces additional representation directions while preserving the original mapping B. Then,

$$\mathcal{B}_{\widetilde{B}}(\Delta E) \leq \mathcal{B}_B(\Delta E) \qquad \textit{for every } \Delta E,$$

and

$$s_{\min}(\widetilde{B}) \geq s_{\min}(B).$$

Thus, enriching the control point representation can reduce both the minimum correction burden for a given target embedding correction and its worst-case value characterized in Lemma 2.

Proof. From Lemma 2, the minimum control point correction burden can equivalently be written as

$$\mathcal{B}_B(\Delta E) = \|\Delta E B^+\|_F^2 = \operatorname{tr}\left(\Delta E (B^\top B)^{-1} \Delta E^\top\right).$$

Because B has full column rank, $B^\top B$ is positive definite.

For the enriched representation,

$$\widetilde{B}^\top \widetilde{B} = B^\top B + C^\top C.$$

Since $C^\top C$ is positive semidefinite,

$$\widetilde{B}^\top \widetilde{B} \succeq B^\top B.$$

Thus, representational enrichment increases the Gram matrix in the positive-semidefinite ordering. Since both matrices are positive definite, matrix inversion reverses this ordering:

$$(\widetilde{B}^\top \widetilde{B})^{-1} \preceq (B^\top B)^{-1}.$$

Applying the correction-burden expression to $\widetilde{B}$ gives

$$\mathcal{B}_{\widetilde{B}}(\Delta E) = \operatorname{tr}\left(\Delta E (\widetilde{B}^\top \widetilde{B})^{-1} \Delta E^\top\right).$$

Therefore,

$$\begin{aligned} \mathcal{B}_B(\Delta E) - \mathcal{B}_{\widetilde{B}}(\Delta E) &= \operatorname{tr}\left[\Delta E \left((B^\top B)^{-1} - (\widetilde{B}^\top \widetilde{B})^{-1}\right) \Delta E^\top\right] \\ &\geq 0, \end{aligned}$$

because

$$(B^\top B)^{-1} - (\widetilde{B}^\top \widetilde{B})^{-1} \succeq 0.$$

Hence,

$$\mathcal{B}_{\widetilde{B}}(\Delta E) \leq \mathcal{B}_B(\Delta E).$$

The same Gram-matrix relation also determines the smallest singular value. Since

$$s_{\min}^2(B) = \lambda_{\min}(B^\top B),$$

we have

$$\begin{aligned} s_{\min}^2(\widetilde{B}) &= \lambda_{\min}\left(B^\top B + C^\top C\right) \\ &\geq \lambda_{\min}(B^\top B) \\ &= s_{\min}^2(B). \end{aligned}$$

Since singular values are nonnegative,

$$s_{\min}(\widetilde{B}) \geq s_{\min}(B).$$

By Lemma 2, the worst-case normalized correction burden is $1/s_{\min}^2(B)$, and therefore

$$\frac{1}{s_{\min}^2(\widetilde{B})} \leq \frac{1}{s_{\min}^2(B)}.$$

□

Algorithm 1: Sentence Curve Language Model (SCLM) Training for Gaussian DLMs

Input: Data (X, Y), embedding matrix E, mappings B, B^+, DLM noise process q, backbone H_θ
Output: Updated model parameters

Sample $(X, Y) \sim p_{\text{data}}$;
Compute clean target embeddings: $E_Y^0 \leftarrow E\Delta_Y$;
Sample timestep t and noised embeddings: $E_Y^t \sim q(\cdot \mid E_Y^0)$;
// Map to control point space
$P_Y^t \leftarrow E_Y^t B^+$;
// Backbone prediction in control point space
$\hat{P}_{Y,\theta}^0 \leftarrow H_\theta(P_Y^t, t, X)$;
// Map back to target embedding space
$\hat{E}_{Y,\theta}^0 \leftarrow \hat{P}_{Y,\theta}^0 B$;
// Original DLM objective
Compute $\mathcal{L}(Y; \theta, E)$ using $\hat{E}_{Y,\theta}^0$ in Eq. 1;
Update trainable parameters;

B Details of Sentence Curve Language Models

Algorithm 1 summarizes the Gaussian-DLM implementation of SCLM described in Section 4.2. The noised target embeddings are first mapped to the control point space through B^+. The backbone then processes the resulting control point representation, and its final hidden states serve as the predicted denoised control points,

$$\hat{P}_{Y,\theta}^0 := H_\theta(P_Y^t, t, X).$$

The prediction is subsequently mapped back to the original target embedding space through B, after which the original DLM objective is applied without modification.

For discrete DLMs, the same principle is applied by inserting the $E \to P$ and $P \to E$ mappings around the backbone, while retaining the model-specific discrete noising process and training objective.

C Experimental Details

C.1 Details of Datasets

For machine translation, we use the IWSLT14 and WMT14 English–German datasets in both translation directions. We perform data cleaning, tokenization, and byte-pair encoding (BPE) using the Fairseq toolkit (Ott et al., 2019), closely following the preprocessing pipelines of prior DLM studies (Ye et al., 2023b; Gao et al., 2024). After preprocessing, IWSLT14 contains approximately 160K/7K/7K sentence pairs for the train/validation/test splits, respectively, with a vocabulary size of 10K. WMT14 contains approximately 4.5M/3K/3K sentence pairs with a vocabulary size of 40K.

For generic language modeling, we use the LM1B dataset and follow the preprocessing protocol of MDLM (Sahoo et al., 2024). The resulting dataset contains approximately 30M training sentences and 300K evaluation sentences.

C.2 Details of DLM Experiments

In this section, we describe the model configurations and training procedures used in our DLM experiments.

Machine translation. For machine translation, we build SCLM on Differmer (Gao et al., 2024) and use the `transformer_iwslt_de_en` and `transformer_wmt_en_de` architectures provided by Fairseq for IWSLT14 and WMT14, respectively.[2] We closely follow the official Differmer training procedure for both the baseline and SCLM models. To improve baseline tuning fairness, we additionally sweep the Differmer rescaling factor (RF) separately for each translation direction and

[2] https://github.com/zhjgao/differmer

Table 3: Basic model and training configurations for conditional machine translation. For the rescaling factor (RF), values are reported as En→De / De→En.

Hyperparameter	**IWSLT14**	**WMT14**
Model / Architecture	`transformer_iwslt_de_en`	`transformer_wmt_en_de`
# Encoder layers	6	6
# Decoder layers	6	6
d_{model}	512	512
d_{ff}	1024	2048
# Attention heads	4	8
Dropout	0.3	0.1
Label smoothing	0.1	0.1
Batch size	8K tokens	32K tokens
Optimizer	AdamW	AdamW
Learning rate	5×10^{-4}	5×10^{-4}
Gradient clipping	0.0	1.0
Training iterations	300K	300K
Checkpoint ensemble	5	5
Latent dimension	128	128
Diffusion steps	2K	2K
Rescaling factor (RF)	6.0 / 5.5	5.0 / 5.5
Self-conditioning	✓	✓
Reverse steps	20	20
Length beam size	5	7
Noise beam size	2	3

Table 4: Basic model and training configurations for LM1B generic language modeling. The fully non-autoregressive setting includes MDLM, SoftMask, and their SCLM variant, while the semi-autoregressive setting includes the preliminary BD3-LM and BD3-SCLM experiments.

Hyperparameter	**Fully Non-AR** **MDLM / SoftMask / SCLM**	**Semi-AR** **BD3-LM / SCLM**
Dataset	LM1B	LM1B
Model / Architecture	small DiT	tiny DiT
# DiT blocks	12	8
d_{model}	768	256
# Attention heads	12	8
Dropout	0.1	0.1
Global batch size	512 sentences	512 sentences
Context length	128	128 / 256 / 512
Active block length	–	4 / 8 / 16
Optimizer	AdamW	AdamW
Learning rate	3×10^{-4}	3×10^{-4}
Gradient clipping	1.0	1.0
Training iterations	200K	100K
Generation scheme	Fully non-AR	Semi-AR block diffusion
Checkpoint selection	EMA (0.9999)	Lowest validation NLL

select it according to validation performance. The selected values are $\mathrm{RF} = 6.0/5.5$ for IWSLT14 En→De/De→En and $\mathrm{RF} = 5.0/5.5$ for WMT14 En→De/De→En, respectively. Each SCLM inherits the task-specific RF selected for its corresponding Differmer baseline, while only sentence curve-specific hyperparameters are tuned separately. All other baseline-specific settings follow the official implementation.

The inference configuration uses 20 reverse steps together with the length and noise beam settings summarized in Table 3. Differmer and SCLM contain 38.9M parameters on IWSLT14 and 53.5M parameters on WMT14. The sentence curve mappings introduce no additional trainable parameters, although increasing the number of control points increases the number of states processed by the backbone.

Table 5: Selected SCLM configurations for conditional machine translation. Decimal values of η denote η_{ratio}, while integer values denote a fixed degree η_{fix}.

Hyperparameter	**IWSLT14** En→De	**IWSLT14** De→En	**IWSLT14** En→De w/o KD	**IWSLT14** De→En w/o KD	**WMT14** En→De	**WMT14** De→En
N_{ratio}	2.0	3.0	2.0	2.0	2.0	3.0
η	5	0.01	0.01	0.01	0.1	5

Table 6: Selected SCLM configurations for LM1B generic language modeling. The fully non-autoregressive setting is evaluated against MDLM and SoftMask, while the semi-autoregressive setting is evaluated with BD3-LM. For η, decimal values denote η_{ratio}, while integer values denote a fixed degree η_{fix}.

	Fully Non-AR	**Semi-AR (BD3-LM)**				
Hyperparameter	**MDLM / SoftMask** $C = 128$	$C = 128$ $L_{\text{block}} = 4$	$C = 128$ $L_{\text{block}} = 8$	$C = 128$ $L_{\text{block}} = 16$	$C = 256$ $L_{\text{block}} = 4$	$C = 512$ $L_{\text{block}} = 4$
N_{ratio}	3.0	2.5	2.5	3.0	3.0	2.5
η	5	0.25	0.25	0.25	0.25	0.25

LM1B generic language modeling. We consider both fully non-autoregressive and semi-autoregressive unconditional generation on LM1B. For the fully non-autoregressive setting, we follow the official MDLM implementation and training setup,[3] whose backbone is a small DiT (Peebles & Xie, 2023). We implement the SoftMask strategy (Hersche et al., 2025) on top of the same codebase using its standard method-specific configuration. MDLM, SoftMask, and SCLM use the same backbone scale, context length of 128, and 200K training iterations. These models contain approximately 139M parameters.

For the preliminary semi-autoregressive experiments, we build SCLM on BD3-LM and apply one sentence curve within each actively denoised block. We use a tiny-DiT backbone trained for 100K updates. We evaluate active block lengths $L_{\text{block}} \in \{4, 8, 16\}$ at total context length $C = 128$, and additionally fix $L_{\text{block}} = 4$ while varying $C \in \{128, 256, 512\}$ to examine longer-context behavior. The corresponding fully non-autoregressive and semi-autoregressive configurations are summarized in Table 4. The BD3-LM experiments are intended as preliminary evaluations of architectural generality and potential length scalability rather than standard-scale comparisons.

For the fully non-autoregressive generation evaluation, we generate 1,024 samples for each setting and repeat generation using three random seeds. We report Gen. PPL, MAUVE, and self-BLEU in the main text, with the corresponding standard deviations provided in Appendix D. For the preliminary BD3-LM experiments, we follow the block-diffusion generation setting and generate 256 samples using a single random seed.

SCLM hyperparameter selection. We separately tune sentence curve configurations for conditional machine translation and unconditional language generation. Guided by Remark 1, we focus on expanded control point spaces with $N_{\text{ratio}} \geq 2.0$. This provides a practical prior for narrowing the search space while retaining sufficient representation capacity under target-side coupling.

For machine translation and fully non-autoregressive LM1B experiments, we search $N_{\text{ratio}} \in \{2.0, 2.5, 3.0\}$, $\eta_{\text{ratio}} \in \{0.01, 0.05, 0.1\}$, and $\eta_{\text{fix}} \in \{5, 10\}$, and select the final configuration according to validation performance. The selected machine translation settings are reported in Table 5.

For the preliminary BD3-LM experiments, the actively denoised blocks are much shorter than the full sequences used in the other tasks. We therefore explore larger degree ratios, including $\eta_{\text{ratio}} \in \{0.25, 0.5\}$, so that each target embedding can still share a meaningful number of control points within the shorter block. Table 6 summarizes the selected configurations for both the fully non-autoregressive and semi-autoregressive LM1B experiments.

[3] https://github.com/kuleshov-group/mdlm

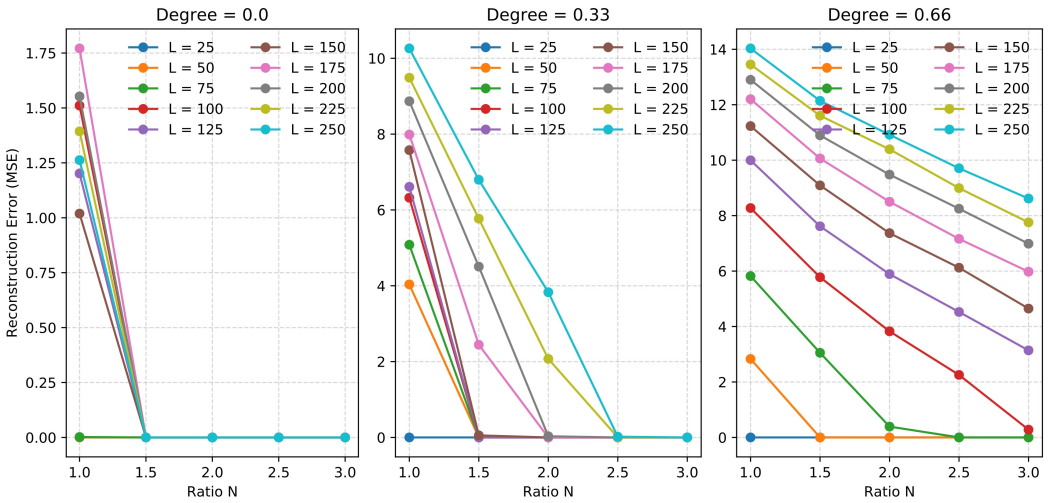


Figure 5: Finite-precision reconstruction error of the sentence curve mapping $E \xrightarrow{B^+} P \xrightarrow{B} \hat{E}$ under different sequence lengths L, control point ratios N_{ratio}, and B-spline degree ratios η_{ratio}.

D Additional Experimental Analyses

D.1 Numerical Stability of Sentence Curve Mapping

As discussed in Section 4.1, under Assumption 3, B has full column rank and therefore

$$B^+B = I_L.$$

Consequently, the sentence curve mapping is information preserving in exact arithmetic:

$$E \xrightarrow{B^+} P = EB^+ \xrightarrow{B} \hat{E} = EB^+B = E.$$

Nevertheless, finite-precision computation of the pseudoinverse can introduce small reconstruction errors when B is poorly conditioned. We therefore empirically examine the numerical stability of the mapping.

For each configuration, we sample length-L sequences from a standard Gaussian distribution, map them to the control point space using B^+, and reconstruct them using B. We compute the mean squared error (MSE) between the original sequence E and its reconstruction $\hat{E}$. We vary $L \in \{25, 50, \ldots, 225, 250\}$, $N_{\text{ratio}} \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$, and $\eta_{\text{ratio}} \in \{0.0, 0.33, 0.66\}$, and average the results over 100 independently sampled sequences. The minimum B-spline degree is fixed to $\eta = 2$; thus, $\eta_{\text{ratio}} = 0.0$ denotes this minimum-degree configuration.

Figure 5 shows that reconstruction error generally increases with sequence length and B-spline degree, while increasing the number of control points substantially improves numerical stability. In particular, configurations with fewer control points and higher spline degrees can exhibit noticeably larger finite-precision error, whereas settings with $N_{\text{ratio}} \geq 2.0$ and moderate η reconstruct the original sequence accurately over the examined length range. Higher B-spline degrees increase the overlap among basis functions, which can make the pseudoinverse more sensitive to finite-precision errors when the resulting basis matrix becomes poorly conditioned. Empirically, increasing the number of control points reduces this reconstruction error in our tested configurations. Importantly, these errors arise from the numerical computation of the pseudoinverse rather than from an intrinsic loss of information, since $B^+B = I_L$ under the full-rank assumption in exact arithmetic.

D.2 Comparison with Corresponding AR Baselines

Absolute translation scores reported across DLM studies can be affected by differences in the underlying model architecture, training setup, and decoding configuration. To complement the direct

Table 7: Comparison of each DLM with its corresponding AR Transformer counterpart. $\Delta_{\mathrm{AR}} = \mathrm{BLEU}_{\mathrm{DLM}} - \mathrm{BLEU}_{\mathrm{AR}}$. Higher Δ_{AR} indicates a smaller gap from the corresponding AR baseline. "–" denotes unavailable information.

			En→De			De→En		
Model	**Steps**	**Beam**	**DLM**	**AR**	$\boldsymbol{\Delta_{\mathrm{AR}}}$	**DLM**	**AR**	$\boldsymbol{\Delta_{\mathrm{AR}}}$
IWSLT14								
DiffusionLM (Li et al., 2022)	–	–	–	–	–	29.11	33.30	-4.19
SeqDiffuSeq (Yuan et al., 2022)	20	10	22.12	26.51	-4.39	30.45	33.81	-3.36
DINOISER (Ye et al., 2023b)	20	50	26.14	28.30	-2.16	31.61	33.61	-2.00
GENIE (Lin et al., 2023)	20	50	23.89	28.30	-4.41	29.45	33.61	-4.16
AR-Diffusion (Wu et al., 2023)	20	50	26.01	28.30	-2.29	31.80	33.61	-1.81
Differmer (Gao et al., 2024)	20	10	27.52	28.24	-0.72	32.42	33.83	-1.41
SCLM (ours)	20	10	**28.10**	28.24	**-0.14**	**32.56**	33.83	**-1.27**
WMT14								
DiffusionLM (Li et al., 2022)	–	–	17.41	26.85	-9.44	–	–	–
SeqDiffuSeq (Yuan et al., 2022)	20	10	19.76	26.20	-6.44	23.93	30.20	-6.27
DINOISER (Ye et al., 2023b)	20	50	25.88	26.85	-0.97	30.30	30.55	-0.25
MDLM (Sahoo et al., 2024)	–	–	18.40	25.20	-6.80	–	–	–
E2D2 (Arriola et al., 2025b)	–	–	24.80	25.20	-0.40	–	–	–
Differmer (Gao et al., 2024)	20	21	26.98	27.71	-0.73	32.12	33.00	-0.88
SCLM (ours)	20	21	**28.19**	27.71	**+0.48**	**33.49**	33.00	**+0.49**

comparison in Table 1, we therefore compare each DLM with the corresponding AR Transformer baseline reported in the same work whenever available. We define

$$\Delta_{\mathrm{AR}} := \mathrm{BLEU}_{\mathrm{DLM}} - \mathrm{BLEU}_{\mathrm{AR}},$$

so that a larger value indicates a smaller performance gap from the method's own AR counterpart, while a positive value indicates that the DLM surpasses it.

Table 7 additionally reports the available decoding settings for context. Although these settings are not fully standardized across previous studies, our decoding budget is comparable to or smaller than those used by many prior DLM baselines. For Differmer and SCLM, the reported beam size corresponds to the product of the length beam and noise beam.

The relative comparison leads to a similar conclusion as the absolute-score comparison. On IWSLT14, SCLM substantially narrows the gap to its corresponding AR Transformer, reaching within $0.14/1.27$ SacreBLEU on En→De/De→En. The result is particularly notable on WMT14: SCLM exceeds its own AR counterpart in both directions, by $0.48/0.49$ SacreBLEU. Thus, the strong WMT14 performance in Table 1 is not explained merely by using a stronger underlying AR baseline; the relative comparison also shows a pronounced advantage of SCLM over the corresponding autoregressive reference.

D.3 Generation Latency

Table 8 shows that sentence curve prediction introduces only modest additional inference cost over Differmer. In particular, increasing N_{ratio} tends to slightly increase latency because a larger number of control point states must be processed during denoising. Nevertheless, across the tested configurations, SCLM remains substantially faster than the autoregressive Transformer, retaining roughly 24–39% speedup depending on the dataset and sentence curve configuration. Thus, the improved translation performance of SCLM is obtained while largely preserving the parallel-generation efficiency of the underlying DLM.

D.4 Detailed Analysis of Coordinated Target Prediction

Distance correlation analysis. We provide additional details for the coordinated-generation analysis presented in Figure 4 (Right). Our goal is to examine whether predictions at different target positions respond coordinately when the conditioning information is perturbed. If multiple target

Table 8: Inference latency (seconds/sample) on IWSLT14 and WMT14 En→De, measured on a single NVIDIA A4000 GPU. Values in parentheses denote speedup relative to the AR Transformer.

Model	**IWSLT14 En→De**	**WMT14 En→De**
AR Transformer	0.3016 (0.00%)	0.4020 (0.00%)
Differmer	0.2191 (27.36%)	0.2473 (38.47%)
SCLM ($N_{\text{ratio}} = 2.0, \eta_{\text{ratio}} = 0.1$)	0.2201 (27.01%)	0.2477 (38.39%)
SCLM ($N_{\text{ratio}} = 2.5, \eta_{\text{ratio}} = 0.1$)	0.2298 (23.82%)	0.2540 (36.82%)
SCLM ($N_{\text{ratio}} = 3.0, \eta_{\text{ratio}} = 0.1$)	0.2271 (24.70%)	0.2663 (33.75%)
SCLM ($N_{\text{ratio}} = 2.0, \eta_{\text{ratio}} = 0.01$)	0.2265 (24.90%)	0.2443 (39.22%)
SCLM ($N_{\text{ratio}} = 2.0, \eta_{\text{ratio}} = 0.05$)	0.2396 (20.58%)	0.2537 (36.90%)
SCLM ($N_{\text{ratio}} = 2.0, \eta_{\text{fix}} = 5$)	0.2316 (23.23%)	0.2510 (37.56%)
SCLM ($N_{\text{ratio}} = 2.0, \eta_{\text{fix}} = 10$)	0.2350 (22.10%)	0.2590 (35.57%)

positions are strongly dependent in their response to the same input variation, their prediction changes should exhibit stronger statistical dependence.

We quantify this behavior using distance correlation (dCor) (Székely et al., 2007), which captures both linear and nonlinear statistical dependence. For each source–target example, we keep the decoder diffusion input fixed and repeatedly perturb the source-side conditioning information. We consider two perturbation types: (1) additive Gaussian noise to the encoder hidden states and (2) random perturbation of source tokens. For Gaussian perturbation, the noise magnitude is normalized by the average encoder hidden-state norm to avoid trivial scale effects.

For each perturbation level, we collect representations in the same target-position space for both models. For Differmer, we use the decoder's final hidden states, which directly correspond to the L target positions. For SCLM, the backbone's final hidden states represent predicted control points $\hat{P}_\theta$; we therefore map them back through B and use $\hat{P}_\theta B$, yielding the corresponding L target-position representations. We then compute pairwise dCor between target positions over repeated perturbations and average the off-diagonal entries of the resulting dCor matrix. Thus, although the two models internally operate on different representations, dCor is measured on the same L-position representation space. A larger value therefore indicates that different target positions tend to change more dependently in response to the same variation in the conditioning context. As shown in Figure 4 (Right), SCLM exhibits higher dCor than Differmer under both perturbation types, suggesting more coordinated target-side responses.

To summarize this behavior for comparison across different sentence curve configurations, we compute the normalized area under each dCor curve over the perturbation range shown in Figure 4 (Right), and average the resulting Gaussian-noise and source-perturbation AUCs. We refer to this scalar as *dCor AUC* below.

Theoretical quantities induced by the sentence curve mapping. We next compare the empirical dCor behavior with the gradient geometry derived in Lemma 1. Recall that, for the spline basis matrix B, the gradient propagated to the control point representation is $GB^\top$. Lemma 1 consequently defines

$$R(B) = \frac{\mathbf{1}^\top B^\top B \mathbf{1}}{\operatorname{tr}(B^\top B)},$$

which measures the propagated squared gradient magnitude of a normalized sequence-wide coordinated pattern relative to the average magnitude of a token-local pattern. Larger $R(B)$ therefore corresponds to relatively stronger gradient propagation for coordinated changes across target positions.

The same lemma also relates this quantity to the spectral condition number

$$\kappa(B) = \frac{s_{\max}(B)}{s_{\min}(B)}$$

through

$$R(B) \le \kappa(B)^2.$$

Table 9: Sentence curve geometry and coordinated target responses under different SCLM configurations on IWSLT14. $\bar{R}(B)$ and $\bar{\kappa}(B)$ denote length-aggregated theoretical quantities derived from Lemma 1. dCor AUC summarizes the empirical target-response coordination under the Gaussian and source perturbations shown in Figure 4 (Right).

Direction	$\mathbf{N}_{\text{ratio}}$	$\boldsymbol{\eta}$	$\bar{\boldsymbol{\kappa}}(\boldsymbol{B})$ / $\bar{\boldsymbol{R}}(\boldsymbol{B})$	**dCor AUC**
En→De	2.0	$\eta_{\text{ratio}} = 0.10$	1.4654 / 1.2935	.6759
	2.5	$\eta_{\text{ratio}} = 0.10$	1.3016 / 1.2032	.6772
	3.0	$\eta_{\text{ratio}} = 0.10$	1.2027 / 1.1387	.6569
	2.0	$\eta_{\text{ratio}} = 0.01$	1.1140 / 1.0263	.6763
	2.0	$\eta_{\text{ratio}} = 0.05$	1.1654 / 1.0830	.6703
	2.0	$\eta_{\text{fix}} = 5$	1.5698 / 1.3549	.6630
	2.0	$\eta_{\text{fix}} = 10$	4.9073 / 1.9215	.7376
De→En	2.0	$\eta_{\text{ratio}} = 0.10$	1.4061 / 1.2604	.6310
	2.5	$\eta_{\text{ratio}} = 0.10$	1.2687 / 1.1810	.6198
	3.0	$\eta_{\text{ratio}} = 0.10$	1.1807 / 1.1213	.6184
	3.0	$\eta_{\text{ratio}} = 0.01$	1.1042 / 1.0000	.6088
	3.0	$\eta_{\text{ratio}} = 0.05$	1.1098 / 1.0190	.6347
	3.0	$\eta_{\text{fix}} = 5$	1.0597 / 1.0298	.6128
	3.0	$\eta_{\text{fix}} = 10$	1.4825 / 1.2872	.6297

While $R(B)$ directly evaluates the particular coordinated direction considered in the lemma, $\kappa(B)$ characterizes the overall spectral disparity of the sentence curve mapping. Importantly, both quantities depend only on B and are therefore determined by the sentence curve configuration independently of the trained model, whereas dCor is an empirical property measured from model responses after training.

Because B varies with target length L, we aggregate these quantities over the empirical training target-length distribution. We report the arithmetic mean of $R(B)$ and the geometric mean of $\kappa(B)$, denoted by $\bar{R}(B)$ and $\bar{\kappa}(B)$, respectively.

Relation between theory and observed coordination. Table 9 shows that the theoretical quantities induced by the sentence curve mapping are positively associated with the coordinated target behavior measured by dCor. To summarize the relationship across both translation directions, we standardize each quantity within each direction and pool the configurations. The resulting Pearson correlations with dCor AUC are 0.690 for $\bar{R}(B)$ and 0.754 for $\bar{\kappa}(B)$; the corresponding Spearman correlations are 0.459 and 0.534, respectively.

These observations are consistent with the gradient-reweighting interpretation of Lemma 1. Sentence curve configurations that place relatively greater gradient magnitude on coordinated positional changes, or exhibit stronger spectral disparity, also tend to produce more coordinated target responses after training. The relationship is particularly clear for the Pearson correlation, although it is not perfectly monotonic across individual configurations.

We emphasize that this analysis does not establish that $R(B)$ or $\kappa(B)$ directly determines the behavior of the trained model. Lemma 1 characterizes the gradient geometry induced by the target representation, whereas dCor measures the behavior that emerges after optimization of the complete model. Nevertheless, the observed association provides empirical support that the gradient structure predicted by our theoretical analysis is reflected in the coordinated target prediction behavior observed in Figure 4 (Right).

D.5 Qualitative Translation Examples

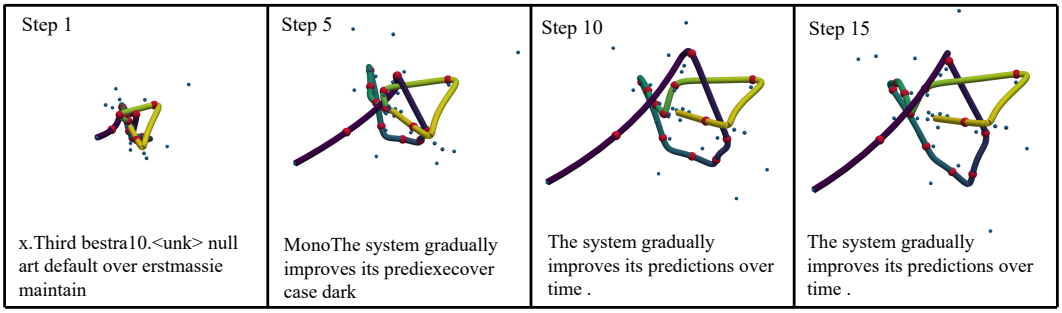
We provide qualitative examples from IWSLT14 models trained without knowledge distillation to complement the coordinated-generation analysis in Section 5.2. Table 10 compares outputs from Difformer and SCLM on representative examples.

Across these examples, Difformer sometimes combines incompatible local phrases or produces repeated or grammatically inconsistent constructions, such as *"any particular surface at any surface"*

Table 10: Qualitative comparison of Differmer and SCLM translation outputs on IWSLT14 without KD. Highlighted phrases illustrate locally inconsistent or repeated constructions in Differmer and their corresponding SCLM outputs.

	Ground-truth	**Differmer**	**SCLM (ours)**
1	*... project a light onto any surface and then , with a laser pointer , draw on it ...*	*... project ster down a light on any particular surface at any surface, and and then drawing with with a laser pointer on it ...*	*... project a light on top of any surface, and then draw up a laser pointer on it ...*
2	*Imagine you can have everybody make a small donation for one pixel .*	*Imagine that anyone else can give giving a small donation to a point of image .*	*Imagine that anyone can give a little donation for a picture .*
3	*But there are over 48 hours of video uploaded to youtube every minute .*	*But on youtube , they are uploaded more than 48 hours a minute .*	*But on youtube , there's over 48 hours of video a minute .*

Reference: *"The system gradually improves its predictions over time ."*



Reference: *"Accurate results depends on careful experimental design ."*

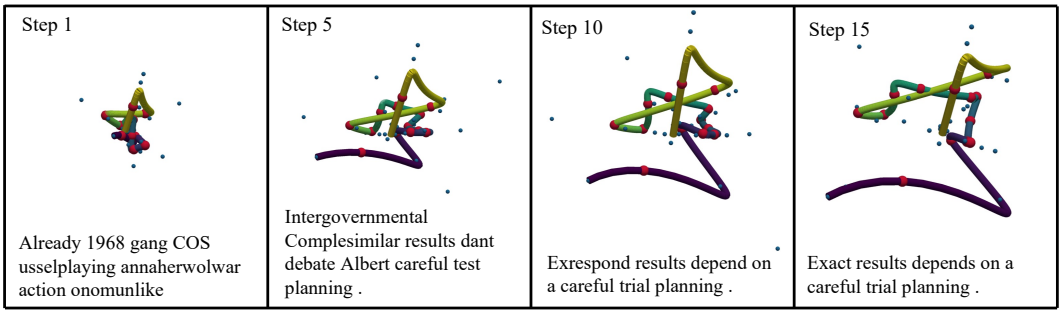


Figure 6: Extra examples of sentence curve generation by SCLM in the translation task. Blue dots represent the estimated control points.

and *"give giving"*. SCLM produces more coherent alternatives in the same cases. These examples are qualitative and do not by themselves establish a general performance advantage, but they provide an intuitive complement to the distance-correlation analysis by illustrating how more coordinated target prediction can appear in generated sentences.

D.6 Sentence Curve Generation Examples

In this section, we present additional sentence curve generation examples, analogous to the illustration in the right side of Figure 1. All those examples are generated using our best-performing WMT14 De→En checkpoint. Figure 6 shows two extra representative sentence curve generation histories. As illustrated, SCLM smoothly denoises the sentence curve from early diffusion steps and progressively refines it toward a curve that maps to the reference sentence embeddings. Together with the examples in Figure 1 and Figure 6, all visualizations are based on the intermediate denoising variables $\hat{\mathbf{e}}^t_{Y,\theta}$.

Table 11: Multi-seed generic language modeling results on LM1B. Each entry reports mean ± standard deviation over three generation seeds, with 1,024 generated samples per seed.

Model	Metric	NFE=64	NFE=128	NFE=256	NFE=512
MDLM	Gen. PPL ↓	62.81 ± 0.50	38.58 ± 1.74	33.54 ± 1.12	22.82 ± 0.17
	self-BLEU ↓	7.20 ± 0.15	11.25 ± 0.46	14.45 ± 0.25	19.74 ± 0.50
	MAUVE ↑	$.0674 \pm .0135$	$.0708 \pm .0268$	$.0824 \pm .0289$	$.1204 \pm .0301$
SoftMask	Gen. PPL ↓	52.07 ± 0.96	34.91 ± 1.10	26.21 ± 0.09	20.25 ± 0.26
	self-BLEU ↓	8.75 ± 0.27	10.85 ± 0.35	13.57 ± 0.11	14.66 ± 0.19
	MAUVE ↑	$.0870 \pm .0380$	$.1111 \pm .0330$	$.1092 \pm .0228$	$.1274 \pm .0026$
SCLM (ours)	Gen. PPL ↓	53.11 ± 2.15	34.29 ± 0.51	24.41 ± 1.14	20.09 ± 0.03
	self-BLEU ↓	8.23 ± 0.54	10.15 ± 0.28	12.79 ± 0.27	14.11 ± 0.41
	MAUVE ↑	$.1140 \pm .0190$	$.1320 \pm .0308$	$.1308 \pm .0468$	$.1508 \pm .0338$

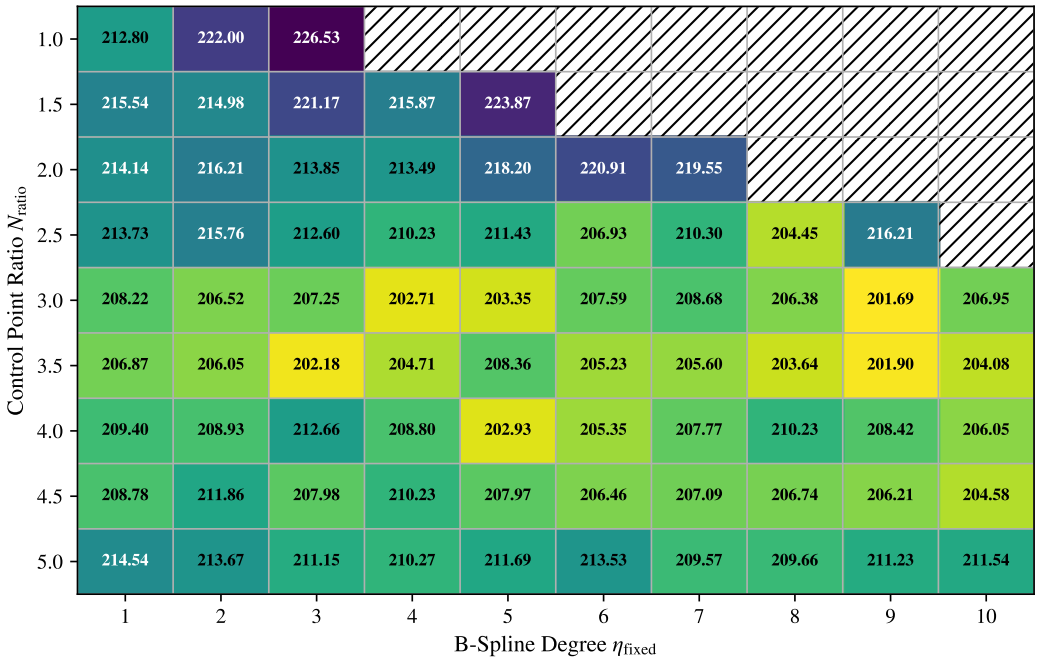


Figure 7: Fine-grained Gen. PPL (↓) landscape of SCLM on the preliminary BD3-LM experiment as the control point ratio N_{ratio} and B-spline degree η are varied. Each observed cell reports the three-seed mean.

D.7 Multi-seed Results for Fully Non-autoregressive Language Modeling

To assess the stability of the fully non-autoregressive LM1B results in Table 2, we repeat generation with three random seeds and generate 1,024 samples per seed. Table 11 reports the corresponding mean ± standard deviation for Gen. PPL, self-BLEU, and MAUVE.

The multi-seed evaluation preserves the trends reported in the main text. At NFE=64, SCLM exhibits slightly higher Gen. PPL than SoftMask, while achieving lower self-BLEU and higher MAUVE. At NFE=128/256/512, SCLM improves all three mean generation metrics over SoftMask.

D.8 Fine-grained Analysis of Coupling and Representation Capacity

To further examine the interaction between target-side coupling and representation capacity discussed in Section 4.3, we conduct a fine-grained sweep of sentence curve configurations in the preliminary BD3-LM setting. Each configuration is individually trained and evaluated with three random seeds.

Figure 7 reveals a non-monotonic interaction between the two sentence curve hyperparameters. Increasing the B-spline degree does not uniformly improve generation when the control point representation is small. In particular, stronger overlap among neighboring target positions can be detrimental when N_{ratio} is low. In contrast, configurations with a larger control point representation can accommodate substantially higher degrees while maintaining, and in several cases improving, generation quality. For example, the lowest observed three-seed Gen. PPL occurs at $N_{\text{ratio}} = 3.0$ and $\eta = 9$, with 201.69 ± 11.89, compared with 212.36 ± 5.72 for the BD3-LM baseline.

This behavior is broadly consistent with the theoretical picture developed in Section 4.3. Increasing the B-spline degree generally increases the extent to which neighboring target embeddings share control points, strengthening target-side coupling. Lemma 2 shows that such shared representations can increase the control point-space burden of realizing different token-specific corrections. At the same time, the representational-enrichment result (Remark 1) shows that enlarging the control point representation can provide additional flexibility for realizing these corrections. The heatmap exhibits a corresponding empirical tendency: stronger coupling is more viable when accompanied by sufficient control point capacity.