

DisCon4Gen: A Semantic Alignment Framework Through Discriminative Conditioning for Generation

Caiwei Yang, Yanping Chen*, Shuai Yu

Engineering Research Center of Text Computing & Cognitive Intelligence,
College of Computer Science and Technology, Guizhou University
Guiyang 550025, Guizhou, China

{gs.cwyang21, ypench, gs.syu21}@gzu.edu.cn

*Correspondence: ypench@gzu.edu.cn

Abstract

Generative language models (GLMs) excel at text generation and instruction following, while discriminative language models (DLMs) provide strong semantic representations for understanding tasks. From a modeling perspective inspired by the manifold hypothesis, natural language semantics can be regarded as lying on a shared latent semantic manifold, while DLMs and GLMs learn different representations of the underlying semantic structure in their respective latent spaces. However, due to differences in training objectives and modeling strategies, these representation spaces are not naturally aligned, preventing discriminative representations from being directly utilized by GLMs through simple concatenation or linear transformation. To address this issue, a semantic alignment framework through **discriminative conditioning for generation** (DisCon4Gen) is proposed. Guided by this manifold-inspired modeling perspective, DisCon4Gen transforms DLM semantics into continuous conditions for GLMs. It further incorporates both global semantic signals and fine-grained discriminative cues to guide generation, enabling GLMs to better exploit external discriminative semantics. Experiments on five downstream datasets show that DisCon4Gen improves over strong supervised fine-tuning baselines and achieves competitive performance across both semantic understanding and generative reasoning tasks.

1 Introduction

Pre-trained language models (PLMs) mainly follow two paradigms: discriminative language models (DLMs) for semantic understanding and generative language models (GLMs) for text generation and reasoning (Devlin et al., 2019; Liu et al., 2019; Brown et al., 2020; Ouyang et al., 2022). These two paradigms provide complementary strengths: DLMs learn bidirectional semantic representations for understanding-oriented tasks, while GLMs pro-

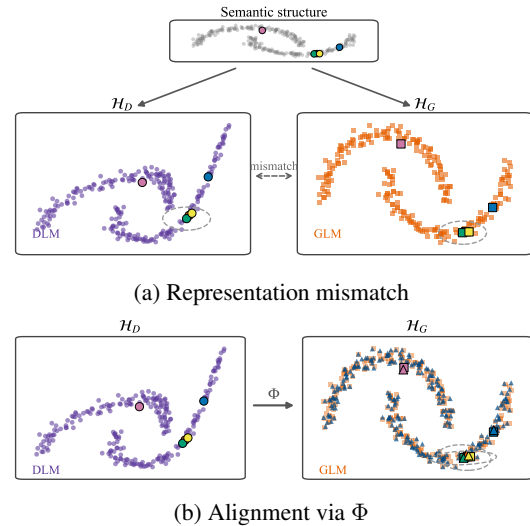


Figure 1: A manifold-inspired view of DLM-GLM alignment. (a) DLMs and GLMs encode related semantic structure in different representation spaces. (b) Φ transforms DLM representations into a GLM-compatible space.

duce fluent and task-adaptive outputs through autoregressive generation.

Recent studies suggest that representations learned by different foundation models may share partially aligned semantic structures, despite differences in architecture, modality, and training objective (Huh et al., 2024; Hosseini et al., 2024). This view is consistent with the manifold hypothesis, which provides a modeling perspective in which natural language semantics can be viewed as lying on a low-dimensional latent semantic manifold structure $\mathcal{M}$ (Psenka et al., 2024). Given an input text x , DLMs and GLMs can therefore be viewed as encoding the same semantic content into model-specific representations of the underlying semantic structure, denoted as $\mathcal{H}_D$ and $\mathcal{H}_G$, respectively. These representations need not share identical coordinates or local geometry. Similar observations have also been made in cross-modal representa-

tion alignment, where different models may share semantic concepts but organize them in different vector spaces (Li et al., 2024b).

Under this manifold-inspired modeling perspective, the integration of discriminative and generative paradigms can be viewed as learning a transformation Φ that maps discriminative representations from $\mathcal{H}_D$ into a GLM-compatible representation space $\mathcal{H}_G$. As illustrated in Figure 1, DLM and GLM representations may initially exhibit geometric and distributional mismatches (Figure 1a), while an appropriate transformation can reduce such mismatch and enable effective semantic integration (Figure 1b).

However, existing methods often rely on simple concatenation, soft prompting, or direct feature projection. Prefix tuning and prompt tuning introduce continuous prompts for generation, while LoRA provides lightweight adaptation of GLMs (Li and Liang, 2021; Lester et al., 2021; Hu et al., 2022). These approaches implicitly assume that external representations can be directly injected into GLMs, but overlook that differences in objectives, tokenization, and architecture can lead to substantial misalignment between DLM and GLM representation spaces. The same semantic content may be encoded with different coordinates, distributions, and local structures, creating barriers to direct transfer. Consequently, naive feature injection introduces additional signals without ensuring semantic alignment or generation compatibility.

To address these challenges, this paper proposes a semantic alignment framework through discriminative conditioning for generation (DisCon4Gen). Guided by the manifold-inspired modeling perspective, DisCon4Gen formulates alignment as a representation transformation that converts DLM representations into conditioning signals compatible with the GLM latent space. Furthermore, DisCon4Gen incorporates a multi-granular conditioning mechanism operating at multiple levels of semantics, capturing both global and fine-grained discriminative cues. This hierarchical strategy ensures that generative models can exploit the multi-scale semantic information encoded in discriminative representations, enabling more informed generation processes.

The main contributions are as follows:

- This paper identifies DLM–GLM integration as a representation alignment problem characterized by geometric and distributional mis-

match, viewed from a manifold-inspired modeling perspective.

- This paper proposes DisCon4Gen, which maps discriminative representations into generation conditions through representation alignment and multi-granular conditioning.
- Experiments on five downstream datasets demonstrate that DisCon4Gen achieves competitive performance on diverse generation and understanding tasks.

2 Related Work

Discriminative language models. Discriminative language models (DLMs), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021), and ELECTRA (Clark et al., 2020), learn contextual representations through bidirectional or masked language modeling objectives. They have achieved strong results on understanding-oriented tasks, including classification, named entity recognition, semantic matching, and retrieval. However, these models are mainly optimized for discrimination rather than autoregressive generation, making their representations difficult for generative modeling.

Generative language models. Generative language models (GLMs) model text through autoregressive next-token prediction and have become the dominant backbone for open-ended generation and instruction following. Representative models include GPT-style models (Radford et al., 2019; Brown et al., 2020), T5 (Raffel et al., 2020), PaLM (Chowdhery et al., 2023), Llama (Grattafiori et al., 2024), Qwen (Yang et al., 2024), and Mixtral (Jiang et al., 2024). Although instruction tuning improves their task adaptability (Ouyang et al., 2022; Chung et al., 2024), their hidden representations are primarily shaped by generation objectives and may lack explicit discriminative semantic guidance for fine-grained understanding.

Collaboration between DLMs and GLMs. Recent studies have explored combining discriminative and generative models to exploit their complementary strengths. Retrieval-augmented generation uses discriminative retrievers or rankers to provide external evidence for GLMs (Karpukhin et al., 2020; Guu et al., 2020; Asai et al., 2024; Yu et al., 2024), while other work studies representation-level integration through cross-attention or encoder-decoder style connections (Li

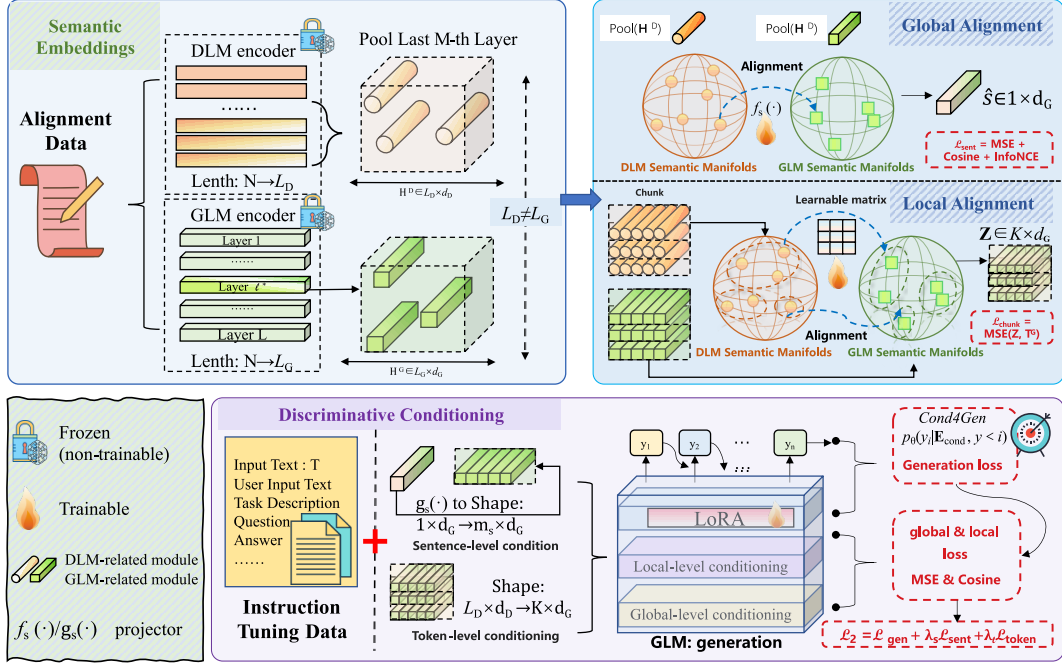


Figure 2: Overview of DisCon4Gen.

et al., 2022; BehnamGhader et al., 2024). However, most existing methods focus on task-level collaboration. In contrast, DisCon4Gen explicitly addresses the geometric mismatch between DLM and GLM representation manifolds, and learns multi-granular alignment before using discriminative representations as generation conditions.

3 Method

This section presents a semantic alignment framework that enables GLMs to leverage discriminative representations through geometric alignment and multi-granular conditioning.

DisCon4Gen follows a two-level semantic conditioning pipeline. Given an input text $\mathbf{T}$, the frozen DLM encodes it into contextual representations $\mathbf{H}^D$. At the global level, the sentence-level projector f_s maps the pooled discriminative representation into the GLM manifold, producing the aligned sentence representation $\hat{\mathbf{S}}$. This representation is further expanded by g_s into sentence-level conditions $\mathbf{C}_s$. At the local level, the token-level projector f_t and the latent resampler produce local semantic slots $\mathbf{Z}$, which are used as token-level conditions $\mathbf{C}_t$. Finally, $\mathbf{C}_s$, $\mathbf{C}_t$, and the prompt embedding $\mathbf{E}_{T'}$ are concatenated as $\mathbf{E}_{\text{cond}}$ and fed into the frozen backbone and trainable LoRA adapters for conditional generation.

3.1 Semantic Manifold Representation

To enable different model semantic alignment, DisCon4Gen first extracts contextual semantic representations from both the discriminative language model (DLM) and the generative language model (GLM). Given an input text sequence $\mathbf{T} = (t_1, t_2, \dots, t_N)$, this section defines the DLM and GLM representations used as the basis for alignment and conditioning.

Discriminative and Generative Semantic Manifolds. Given an input text $\mathbf{T}$, the discriminative tokenizer produces a token sequence of length L_D , which is encoded by the DLM encoder $D(\cdot)$ with N_D Transformer layers. Let $\mathbf{H}_D^{(l)} \in \mathbb{R}^{L_D \times d_D}$ denote the hidden states of the l -th layer. To obtain a high-level semantic representation, DisCon4Gen averages the last M layers:

$$\mathbf{H}^D = \frac{1}{M} \sum_{l=N_D-M+1}^{N_D} \mathbf{H}_D^{(l)} \in \mathbb{R}^{L_D \times d_D}, \quad (1)$$

where M is a hyperparameter and $\mathbf{H}^D = (\mathbf{h}_1^D, \dots, \mathbf{h}_{L_D}^D)^\top$ contains the fused contextual representations of DLM tokens.

For the GLM, the same text $\mathbf{T}$ is tokenized into a sequence of length L_G and processed by the decoder $G(\cdot)$ with N_G Transformer layers. Since final-layer states are often biased toward next-token prediction, an intermediate layer l^* is used as the

generative alignment space:

$$\mathbf{H}^G = G_{l^*}(\mathbf{x}) = \mathbf{H}_G^{(l^*)} \in \mathbb{R}^{L_G \times d_G}, \quad (2)$$

where $1 \leq l^* < N_G$, and $\mathbf{H}^G = (\mathbf{h}_1^G, \dots, \mathbf{h}_{L_G}^G)^\top$ denotes the intermediate-layer GLM representations. Due to different tokenizers and hidden dimensions, L_D and L_G , as well as d_D and d_G , are generally inconsistent.

3.2 Manifold Geometric Alignment

Due to heterogeneous tokenization, dimensions, and objectives, DLM and GLM representations are not directly token-aligned. DisCon4Gen therefore maps discriminative representations into the generative semantic manifold through global and local geometric alignment.

Global Sentence-Level Alignment. Global alignment aims to match the overall semantic intent between the DLM and GLM representations. Given the contextual embeddings $\mathbf{H}^D$ and $\mathbf{H}^G$, sentence-level representations are obtained by masked mean pooling over valid tokens:

$$\mathbf{S}^D = \text{Pool}(\mathbf{H}^D, \mathbf{a}^D) \in \mathbb{R}^{d_D}, \quad (3)$$

$$\mathbf{S}^G = \text{Pool}(\mathbf{H}^G, \mathbf{a}^G) \in \mathbb{R}^{d_G}, \quad (4)$$

where $\mathbf{a}^D$ and $\mathbf{a}^G$ denote the corresponding attention masks.

A two-layer MLP projector with GELU activation maps the DLM sentence representation into the GLM semantic manifold:

$$\hat{\mathbf{S}} = f_s(\mathbf{S}^D) \in \mathbb{R}^{d_G}. \quad (5)$$

To align $\hat{\mathbf{S}}$ with the target GLM representation $\mathbf{S}^G$, DisCon4Gen optimizes a joint sentence-level objective:

$$\mathcal{L}_{\text{sent}} = \lambda_m \mathcal{L}_{\text{mse}} + \lambda_c \mathcal{L}_{\text{cos}} + \lambda_n \mathcal{L}_{\text{nce}}, \quad (6)$$

where $\mathcal{L}_{\text{mse}}$ enforces point-wise geometric closeness, $\mathcal{L}_{\text{cos}}$ preserves directional consistency, and $\mathcal{L}_{\text{nce}}$ maintains batch-level discriminability by contrasting matched and unmatched DLM–GLM sentence pairs. The detailed formulations of these losses are provided in Appendix B.1.

Local Chunk-Level Alignment. To address tokenizer-induced length mismatches ($L_D \neq L_G$), DisCon4Gen employs a Latent Resampler to convert variable-length DLM representations into K fixed local semantic slots. The DLM hidden states are first projected into the GLM hidden space:

$$\mathbf{X} = f_t(\mathbf{H}^D) \in \mathbb{R}^{L_D \times d_G}. \quad (7)$$

Given K learnable latent queries $\mathbf{U} \in \mathbb{R}^{K \times d_G}$, attention pooling is performed over the projected sequence:

$$\mathbf{A} = \text{Softmax} \left(\frac{\mathbf{U}\mathbf{X}^\top}{\sqrt{d_G}}, \mathbf{a}^D \right) \in \mathbb{R}^{K \times L_D}, \quad (8)$$

where padding positions indicated by $\mathbf{a}^D$ are masked before softmax. The resulting local slots are computed as

$$\mathbf{Z} = \text{LayerNorm}(\mathbf{A}\mathbf{X}) \in \mathbb{R}^{K \times d_G}. \quad (9)$$

The GLM hidden states are divided into K consecutive chunks according to token order, and each chunk is mean-pooled to obtain the GLM-side local target:

$$\mathbf{T}^G = [\mathbf{t}_1^G; \dots; \mathbf{t}_K^G] \in \mathbb{R}^{K \times d_G}. \quad (10)$$

The latent queries are implemented as an ordered query bank, where each query has a fixed index. Since the GLM hidden states are also partitioned into K consecutive chunks in token order, the k -th latent slot $\mathbf{z}_k$ is explicitly supervised by the k -th GLM chunk representation $\mathbf{t}_k^G$. This index-wise supervision avoids permutation ambiguity and encourages each slot to capture the local semantic segment at the same relative position.

Local alignment is then imposed without strict token-to-token correspondence:

$$\mathcal{L}_{\text{chunk}} = \text{MSE}(\mathbf{Z}, \mathbf{T}^G). \quad (11)$$

Joint Objective. The overall training objective integrates both the global and local alignment signals:

$$\mathcal{L}_1 = \mathcal{L}_{\text{sent}} + \alpha \mathcal{L}_{\text{chunk}}, \quad (12)$$

where α is a balancing hyperparameter. Crucially, during this alignment stage, the parameters of both the DLM and GLM backbones are frozen. Only the projectors (f_s, f_t) and the latent queries ($\mathbf{U}$) are optimized.

3.3 Discriminative Conditioning

DisCon4Gen utilizes the aligned discriminative features to guide the GLM through a dual-conditioning tuning mechanism.

Dual-Conditioning Construction. The mapped global representation $\hat{\mathbf{S}}$ is first expanded into a sentence-level conditioning sequence through a projector $g_s(\cdot)$:

$$\mathbf{C}_s = g_s(\hat{\mathbf{S}}) \in \mathbb{R}^{m_s \times d_G}, \quad (13)$$

where $g_s(\cdot)$ applies a linear transformation $\mathbf{W}_s \hat{\mathbf{S}} + \mathbf{b}_s$ followed by reshaping, with $\mathbf{W}_s \in \mathbb{R}^{(m_s \cdot d_G) \times d_G}$ and $\mathbf{b}_s \in \mathbb{R}^{m_s \cdot d_G}$. The complete sentence-level transformation is:

$$\mathbf{S}^D \xrightarrow{f_s} \hat{\mathbf{S}} \xrightarrow{g_s} \mathbf{C}_s. \quad (14)$$

Meanwhile, the local slots $\mathbf{Z}$ from Section 3.2 are used as the token-level conditioning:

$$\mathbf{C}_t = \mathbf{Z} \in \mathbb{R}^{K \times d_G}. \quad (15)$$

Let $\mathbf{E}_{T'} \in \mathbb{R}^{L_x \times d_G}$ denote the GLM embedding of the generative prompt $\mathbf{T}'$. When $\mathbf{T}' = \mathbf{T}$, its length satisfies $L_T = L_G$. The final conditional input is formed by concatenating the sentence-level condition, token-level condition, and prompt embedding:

$$\mathbf{E}_{\text{cond}} = [\mathbf{C}_s; \mathbf{C}_t; \mathbf{E}_{T'}] \in \mathbb{R}^{(m_s + K + L_x) \times d_G}. \quad (16)$$

Given the target output sequence $\mathbf{y} = (y_1, y_2, \dots, y_{L_y})$, the GLM predicts target tokens autoregressively conditioned on $\mathbf{E}_{\text{cond}}$. The cross-entropy loss is masked for all conditioning and prompt positions, including the first $m_s + K + L_x$ positions. Therefore, the generation loss is computed only over the target tokens:

$$\mathcal{L}_{\text{gen}} = -\frac{1}{L_y} \sum_{i=1}^{L_y} \log P_{\theta}(y_i | \mathbf{E}_{\text{cond}}, y_{<i}), \quad (17)$$

where θ denotes the trainable parameters and $y_{<i}$ denotes the previously generated target tokens. This masking prevents the model from being optimized to reproduce the conditioning and prompt positions, so that the training signal is focused on target generation conditioned on the augmented input.

Joint Tuning Objective. To ensure that the generative manifold structurally internalizes the injected conditions, DisCon4Gen introduces two losses at the target layer l^* . The global loss $\mathcal{L}'_{\text{sent}}$ binds the prompt-region hidden representation to the injected sentence-level condition $\hat{\mathbf{S}}$, while the local loss $\mathcal{L}'_{\text{token}}$ aligns the token-prefix hidden states with the token-level conditioning $\mathbf{C}_t$. Both losses combine MSE and cosine constraints, with detailed formulations provided in Appendix B.2.

The final objective is formulated as

$$\mathcal{L}_2 = \mathcal{L}_{\text{gen}} + \lambda_s \mathcal{L}'_{\text{sent}} + \lambda_t \mathcal{L}'_{\text{token}}. \quad (18)$$

During this phase, only the semantic projectors, the conditioning projector $g_s(\cdot)$, and Low-Rank Adaptation (LoRA) modules are optimized, while the DLM and GLM backbones remain frozen.

4 Experiments and Analyses

4.1 Experimental Setup

ScienceQA (Saikh et al., 2022) is used as the alignment corpus to learn cross-model semantic correspondence, without using downstream task labels or output schemas (Appendix D); additional corpus ablations are provided in Appendix D.1. Downstream evaluation is conducted on five instruction-tuning datasets: CoNLL03 (Tjong Kim Sang and De Meulder, 2003), ACE05 (Doddington et al., 2004), and CoNLL04 (Roth and Yih, 2004) for semantic understanding, and CommonGen (Lin et al., 2020) and AESLC (Zhang et al., 2019) for generative reasoning.

Both alignment quality and downstream generation performance are evaluated. Alignment quality is measured by validation loss and cosine similarity. Since all downstream tasks are formulated as conditional generation, task-specific generation metrics are reported: micro-F1 (%) for CoNLL03, ACE05, and CoNLL04; ROUGE-1/2/L (%) (Lin, 2004) for AESLC; and BLEU-4, CIDEr, and SPICE (Papineni et al., 2002; Vedantam et al., 2015; Anderson et al., 2016) for CommonGen.

DisCon4Gen¹ is implemented with PyTorch and PEFT. Unless otherwise specified, we use Llama3-8B as the GLM and RoBERTa-base as the DLM. Both backbones are frozen during alignment and conditioning tuning; only the projectors, latent resampler, conditioning modules, and LoRA adapters are optimized. Models are trained for 3 epochs with AdamW using 8 sentence-level conditions and 8 token-level slots. Downstream results are averaged over three runs with different random seeds. More details are provided in Appendix C and Appendix F. Additional results with modern embedding encoders are provided in Appendix G.1.

4.2 Baselines

We compare DisCon4Gen with reported task-specific methods and our own supervised fine-tuning (SFT) baselines. For AESLC, we include PEGASUS and ROUGE-Ext from Sahu and Sanjeevi (2021), and active-learning baselines reported by LDCAL (Li et al., 2024a). For CommonGen, we compare with standard benchmark baselines, retrieval-enhanced generation, and constrained decoding methods (Lin et al., 2020; Wang et al., 2021; Lu et al., 2021). For semantic understanding tasks,

¹Code is available at <https://github.com/caiwyang/discon4gen-submission>.

Table 1: **Results on generative reasoning tasks.** Bold and underlined values denote the best and second-best results, respectively. Values in parentheses show the difference from DisCon4Gen. SFT denotes our LoRA-based supervised fine-tuning baselines. DisCon4Gen uses beam size 6 on CommonGen (other decoding details are in Appendix H).

Model	AESLC			Model	CommonGen		
	R-1	R-2	R-L		BLEU-4	CIDEr	SPICE
Existing Methods							
PEGASUS	37.68	21.25	36.51	GPT-2	23.74	12.18	23.57
RB(ext-seq)	<u>39.34</u>	<u>22.24</u>	<u>38.15</u>	T5-Large	31.96	15.13	28.86
Core-set	25.20	14.00	25.90	BART	31.83	13.98	27.99
BAS	28.50	15.10	27.50	RE-T5	<u>40.86</u>	<u>17.66</u>	<u>31.08</u>
LDCAL	30.10	16.90	29.30	NEUROLOGIC	28.60	15.60	29.60
Average	32.16(-11.10)	17.90(-8.75)	31.47(-10.41)	Average	31.40(-9.97)	14.91(-4.26)	28.22(-4.36)
Supervised Fine-Tuning (SFT with LoRA)							
Llama3-8B	34.15	19.73	33.55	Llama3-8B	28.20	15.78	29.47
Qwen3-8B	31.02	16.13	30.32	Qwen3-8B	21.35	14.65	30.02
Mistral-7B	20.92	9.38	20.29	Mistral-7B	24.98	12.71	25.12
Qwen2.5-1.5B	20.76	14.37	20.56	Qwen2.5-1.5B	21.15	10.82	25.77
Gemma-2-2B	19.15	12.79	18.97	Gemma-2-2B	19.79	10.72	22.11
Average	25.20(-18.06)	14.48(-12.17)	24.74(-17.14)	Average	23.09(-18.28)	12.94(-6.23)	26.50(-6.08)
Our Proposed Method (Llama3-8B+RoBERTa-base)							
DisCon4Gen	43.26	26.65	41.88		41.37	19.17	32.58

we include recent instruction-based, generalizable, reasoning-enhanced, and parameter-efficient information extraction systems (Wang et al., 2023; Zhou et al., 2024; Zaratiana et al., 2024; Huang et al., 2026; Li et al., 2024c; Gui et al., 2024; Liu et al., 2026; Li et al., 2025; Yang et al., 2025b; Jin et al., 2025). Unless otherwise specified, the main results use Llama3-8B as the GLM backbone and RoBERTa-base as the DLM backbone. The corresponding LoRA-SFT baselines use the same GLM backbone under the same instruction format and evaluation protocol for fair comparison. Details are provided in Appendix E.

4.3 Main Results

Results on generative reasoning tasks. Table 1 reports the results on AESLC and CommonGen. Compared with the average of existing methods, DisCon4Gen improves AESLC by 11.10, 8.75, and 10.41 points on ROUGE-1/2/L, and improves CommonGen by 9.97 BLEU-4, 4.26 CIDEr, and 4.36 SPICE. The gains over LoRA-based SFT baselines are even larger, especially on AESLC, where DisCon4Gen improves the average SFT performance by 17.14 ROUGE-L. These results suggest that simply adapting the GLM with LoRA is insufficient to fully capture input semantics, while aligned

discriminative conditions provide more effective semantic guidance for generation.

Results on semantic understanding tasks. Table 2 shows that DisCon4Gen achieves strong performance on semantic understanding tasks. It improves over the average of existing methods by 2.22, 1.79, and 3.91 F1 on CoNLL03, ACE05, and CoNLL04, respectively, and brings larger gains over LoRA-based SFT baselines. Although DisCon4Gen does not obtain the best score on every dataset, it remains competitive with task-specific methods, many of which are specially designed for particular extraction settings. In contrast, DisCon4Gen follows a unified alignment-and-conditioning framework, showing stable effectiveness across different tasks.

Overall analysis. Overall, DisCon4Gen achieves competitive performance against task-specific methods. The gains mainly come from injecting aligned discriminative semantics into the GLM, where sentence-level conditions provide global semantic intent and token-level conditions preserve local structural cues.

4.4 Ablation Study

To examine explicit alignment and discriminative conditioning, DisCon4Gen is compared with

Table 2: **Results on semantic understanding tasks.** Bold and underlined values denote the best and second-best results, respectively. Values in parentheses show the difference from DisCon4Gen. We report F1 scores for NER on CoNLL03 and ACE05, and RE on CoNLL04.

Model	NER		RE
	CoNLL03	ACE05	CoNLL04
<i>Existing Methods</i>			
InstructUIE	91.53	79.94	<u>78.48</u>
UniNER	93.30	86.69	–
GLiNER	92.60	82.80	–
ReasoningNER	94.40	<u>86.60</u>	–
KnowCoder	95.10	86.10	73.30
IEPILE	92.98	–	73.71
ProUIE	92.41	86.15	73.17
MR-UIE	92.61	–	73.55
B ² NER	92.56	83.04	–
Multi-LoRA	93.16	86.53	76.77
Average	92.77(-2.22)	84.73(-1.79)	74.83(-3.91)
<i>Supervised Fine-Tuning (SFT with LoRA)</i>			
Llama3-8B	91.49	76.85	56.37
Qwen3-8B	91.49	82.14	63.95
Mistral-7B	85.28	69.12	49.50
Qwen2.5-1.5B	86.39	69.44	58.02
Gemma-2-2B	80.89	63.90	45.05
Average	87.11(-7.88)	72.29(-14.23)	54.58(-24.16)
<i>Our Proposed Method</i>			
DisCon4Gen	<u>94.99</u>	86.52	78.74

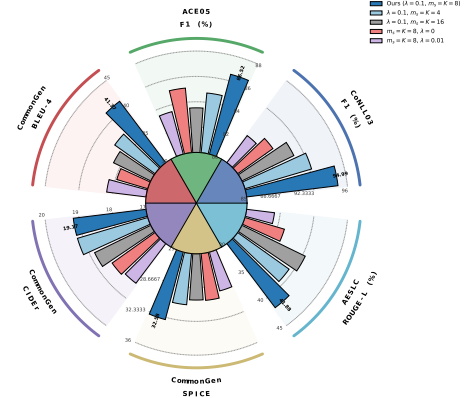
controlled variants from three aspects: conditioning components, alignment objectives, and tuning losses. *S only*, *T only*, *w/o LoRA*, and *w/o Init.* test the effects of different conditioning and training mechanisms, where *w/o Init.* corresponds to direct DLM-based condition injection without alignment pretraining. *w/o NCE*, *w/o Chunk*, *w/o S-Tune*, and *w/o T-Tune* remove different alignment or tuning objectives. All variants use the same setting, with full definitions in Appendix I.

Table 3 shows that both sentence-level and token-level conditions are necessary. Using only C_s or C_t underperforms the full model, indicating that global semantic guidance and local structural cues are complementary. Removing LoRA causes the largest degradation, especially on AESLC, suggesting that the generator must be adapted to effectively use the injected conditions. The drop of *w/o Init.* further confirms the importance of manifold geometric alignment as initialization.

The loss-level ablations show that both alignment and tuning objectives contribute to perfor-

Table 3: **Ablation results on representative datasets.** Full definitions are provided in Appendix I.

Type	Variant	CoNLL04 F1	AESLC R-L	CommonGen B-4
Cond.	<i>S only</i>	73.08	32.38	27.20
Cond.	<i>T only</i>	76.98	34.68	30.75
Train	<i>w/o LoRA</i>	58.46	11.83	23.35
Train	<i>w/o Init.</i>	71.43	32.75	34.78
Align.	<i>w/o NCE</i>	66.94	39.20	29.42
Align.	<i>w/o Chunk</i>	72.51	39.42	29.00
Tune	<i>w/o S-Tune</i>	73.31	38.86	28.93
Tune	<i>w/o T-Tune</i>	69.67	39.24	26.81
DisCon4Gen		78.74	41.88	41.37



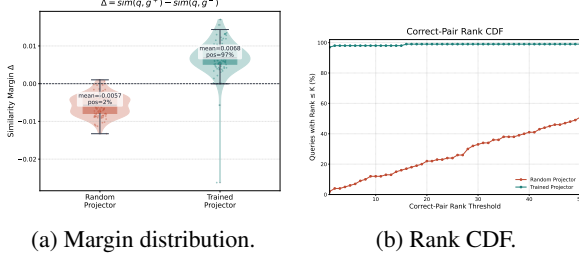


Figure 4: **Diagnostics for probing DLM-GLM manifold correspondence.** (a) The margin is defined as $\Delta = \text{sim}(\hat{\mathbf{q}}, \mathbf{g}^+) - \text{sim}(\hat{\mathbf{q}}, \mathbf{g}^-)$, where $\mathbf{g}^+$ is the matched GLM representation and $\mathbf{g}^-$ is the hardest negative. Positive margins indicate higher similarity to the matched pair. (b) The rank CDF reports the percentage of inputs whose matched GLM representation is ranked within top- K candidates.

gesting that overly long conditions may introduce redundant or noisy signals. The default setting is therefore $m_s = K = 8$.

Effect of joint tuning weight. The joint tuning weight λ controls the auxiliary conditioning objectives in $\mathcal{L}_2$. Small values such as 0 or 0.01 cause clear performance drops, indicating that simply concatenating aligned conditions is insufficient. The setting $\lambda = 0.1$ provides the best trade-off between semantic consistency and generation quality. These results confirm the importance of sufficient conditioning capacity and properly weighted joint tuning.

4.6 Verifying Manifold Alignment

This section verifies whether manifold geometric alignment makes DLM and GLM representations more consistent on the shared semantic manifold. As shown in Figure 4, a positive margin means that the projected DLM representation has higher similarity to its matched GLM representation than to the hardest negative. The random projector yields mostly negative margins, with mean -0.0057 and only 2% positive samples, while the trained alignment module shifts the margin above zero, with mean 0.0068 and 97% positive samples. The rank curve shows the same trend, where matched GLM representations are ranked near the top for almost all inputs after alignment. Figure 5 further shows that the diagonal matched-pair structure is preserved after projection, while off-diagonal stripe-like bias is reduced. These results suggest that the alignment module maps DLM representations into a GLM-compatible space while preserving input-specific semantic correspondence, rather than act-

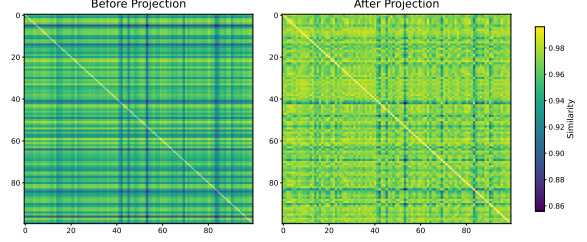


Figure 5: **Pairwise similarity matrices before and after manifold alignment.** The diagonal denotes matched DLM-GLM pairs. After alignment, the diagonal structure is preserved while off-diagonal bias is reduced.

Table 4: **Effect of GLM alignment layer.** Lower, middle, upper, and final layers correspond to $\lfloor N_G/4 \rfloor$, $\lfloor N_G/2 \rfloor$, $\lfloor 3N_G/4 \rfloor$, and N_G , respectively.

Layer	CoNLL04 F1	AESLC R-L	CommonGen B-4
Lower	62.30	35.79	33.43
Middle	78.74	41.88	41.37
Upper	71.43	38.95	40.60
Final	76.61	40.01	39.34

ing as an arbitrary projection. Additional cross-corpus alignment diagnostics are provided in Appendix D.2.

4.7 Alignment Layer Selection Analysis

The choice of the GLM alignment layer l^* determines which hidden space is used as the generative semantic manifold. Table 4 compares lower, middle, upper, and final GLM layers as alignment targets. The middle layer achieves the best performance on all three representative datasets, reaching 78.74 F1 on CoNLL04, 41.88 ROUGE-L on AESLC, and 41.37 BLEU-4 on CommonGen.

The lower layer performs the worst, suggesting that early GLM representations are not sufficiently semantic for manifold alignment. Upper and final layers perform better than the lower layer, but still underperform the middle layer. This indicates that high-level GLM states contain useful semantic information, but may also be more affected by autoregressive next-token prediction. Therefore, the middle layer provides a more stable alignment space for discriminative representations. The same trend is also observed with Qwen3-8B, as reported in Appendix G.2.

5 Conclusion

We introduced DisCon4Gen, a lightweight framework that converts discriminative representations

into GLM-compatible semantic conditions through sentence-level and chunk-level manifold alignment. Experiments show that DisCon4Gen consistently improves over supervised fine-tuning baselines across both semantic understanding and generative reasoning tasks. Further analyses suggest that the learned transformation improves input-specific semantic correspondence, supporting discriminative conditioning as an effective way to enhance generation with aligned semantic guidance.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant No. 2023YFC3304500, and the Key Technology R&D Program of Guizhou Province under Grant No. [2024]003.

Limitations

This work has several limitations. First, DisCon4Gen requires an additional alignment corpus to learn the transformation between DLM and GLM representation manifolds. Although both backbones are frozen and the alignment stage is lightweight, the quality and domain coverage of the alignment corpus may influence downstream performance. Second, our experiments are mainly conducted on English semantic understanding and generative reasoning tasks. Extending DisCon4Gen to multilingual settings, long-context generation, and more open-ended dialogue scenarios remains future work. Third, we use fixed-length sentence prefixes and token-level semantic slots for discriminative conditioning. This design is efficient, but may be less expressive for complex document-level semantics. Future work can explore adaptive or hierarchical semantic conditioning. Finally, while our analysis suggests that the learned transformation preserves input-specific semantic manifold correspondence, a deeper theoretical understanding of how manifold alignment affects generation quality is still needed.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. [SPICE: Semantic propositional image caption evaluation](#). In *Proceedings of the European Conference on Computer Vision 2016*, pages 382–398.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-RAG: Learning to](#)

[retrieve, generate, and critique through self-reflection](#). In *Proceedings of the International Conference on Learning Representations 2024*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [LLM2Vec: Large language models are secretly powerful text encoders](#). In *Proceedings of the Conference on Language Modeling 2024*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2023. [PaLM: Scaling language modeling with pathways](#). *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. [Scaling instruction-finetuned language models](#). *Journal of Machine Learning Research*, 25(70):1–53.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: Pre-training text encoders as discriminators rather than generators](#). In *Proceedings of the International Conference on Learning Representations 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2019*, pages 4171–4186.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. [The automatic content extraction \(ACE\) program – tasks, data, and evaluation](#). In *Proceedings of the International Conference on Language Resources and Evaluation 2004*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh

- Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Honghao Gui, Lin Yuan, Hongbin Ye, Ningyu Zhang, Mengshu Sun, Lei Liang, and Huajun Chen. 2024. [IEPile: Unearthing large scale schema-conditioned information extraction corpus](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics 2024*, pages 127–146.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. [Retrieval augmented language model pre-training](#). In *Proceedings of the International Conference on Machine Learning 2020*, pages 3929–3938.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTa: Decoding-enhanced BERT with disentangled attention](#). In *Proceedings of the International Conference on Learning Representations 2021*.
- Eghbal Hosseini, Colton Casto, Noga Zaslavsky, Colin Conwell, Mark Richardson, and Evelina Fedorenko. 2024. [Universality of representation in biological and artificial neural networks](#). *bioRxiv*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *Proceedings of the International Conference on Learning Representations 2022*.
- Hui Huang, Yanping Chen, Ruizhang Huang, Chuan Lin, and Yongbin Qin. 2026. [A reasoning paradigm for named entity recognition](#). In *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence 2026*, volume 40, pages 31140–31148.
- Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. [Position: The platonic representation hypothesis](#). In *Proceedings of the International Conference on Machine Learning 2024*, pages 20617–20642.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. [Mixtral of experts](#). *arXiv preprint arXiv:2401.04088*.
- Yu Jin, Jie Liu, and Shaowei Chen. 2025. [Multi-LoRA continual learning based instruction tuning framework for universal information extraction](#). *Knowledge-Based Systems*, 308:112750.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing 2020*, pages 6769–6781.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing 2021*, pages 3045–3059.
- Dongyuan Li, Ying Zhang, Zhen Wang, Shiyin Tan, Satoshi Kosugi, and Manabu Okumura. 2024a. [Active learning for abstractive text summarization via LLM-determined curriculum and certainty gain maximization](#). In *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2024*, pages 8959–8971.
- Jiaang Li, Yova Kementchedjhieva, Constanza Fierro, and Anders S  gaard. 2024b. [Do vision and language models share concepts? a vector space alignment study](#). *Transactions of the Association for Computational Linguistics*, 12:1232–1249.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. [BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation](#). In *Proceedings of the International Conference on Machine Learning 2022*, pages 12888–12900.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the Joint Conference of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing 2021*, pages 4582–4597.
- Zhongqiu Li, Shiquan Wang, Ruiyu Fang, Mengjiao Bao, Zhenhe Wu, Shuangyong Song, Yongxiang Li, and Zhongjiang He. 2025. [MR-UIE: Multi-perspective reasoning with reinforcement learning for universal information extraction](#). *Viciniagearth*, 2:17.
- Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024c. [KnowCoder: Coding structured knowledge into LLMs for universal information extraction](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics 2024*, pages 8758–8779.
- Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. [CommonGen: A constrained text generation challenge for generative commonsense reasoning](#). In *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing 2020*, pages 1823–1840.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81.

- Wenda Liu, Zhigang Song, Shuai Nie, Guangyao Liu, Lisung Chen, Binyu Yang, Yaran Chen, Peng Zhou, Hongzhen Wang, Yuchen Liu, Wenyue Hu, Jiaming Xu, Runyu Shi, and Ying Huang. 2026. [ProUIE: A macro-to-micro progressive learning method for LLM-based universal information extraction](#). In *Findings of the Association for Computational Linguistics 2026*, pages 21737–21750.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khachabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. [NeuroLogic A*esque decoding: Constrained text generation with lookahead heuristics](#). In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2022*, pages 780–799.
- Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. [NeuroLogic decoding: \(un\)supervised neural text generation with predicate logic constraints](#). In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2021*, pages 4288–4299.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [BLEU: A method for automatic evaluation of machine translation](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics 2002*, pages 311–318.
- Michael Psenka, Druv Pai, Vishal Raman, Shankar Sasstry, and Yi Ma. 2024. [Representation learning via manifold flattening and reconstruction](#). *Journal of Machine Learning Research*, 25(132):1–47.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI Technical Report*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Dan Roth and Wen-tau Yih. 2004. [A linear programming formulation for global inference in natural language tasks](#). In *Proceedings of the Conference on Computational Natural Language Learning 2004*, pages 1–8.
- Ankit Sahu and Sriram G. Sanjeevi. 2021. [Better fine-tuning with extracted important sentences for abstractive summarization](#). In *Proceedings of the International Conference on Communication, Computing and Industry 4.0 2021*, pages 1–6.
- Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. 2022. [ScienceQA: A novel resource for question answering on scholarly articles](#). *International Journal on Digital Libraries*, 23(3):289–301.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Conference on Computational Natural Language Learning 2003*, pages 142–147.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. [CIDEr: Consensus-based image description evaluation](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015*, pages 4566–4575.
- Han Wang, Yang Liu, Chenguang Zhu, Linjun Shou, Ming Gong, Yichong Xu, and Michael Zeng. 2021. [Retrieval enhanced model for commonsense generation](#). In *Findings of the Association for Computational Linguistics: Joint Conference of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing 2021*, pages 3056–3062.
- Kai Wang, Yanping Chen, Weizhe Yang, and Yongbin Qin. 2025. [A homogeneous multimodality sentence representation for relation extraction](#). *Information Fusion*, 118:102968.
- Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. [InstructUIE: Multi-task instruction tuning for unified information extraction](#). *arXiv preprint arXiv:2304.08085*.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. [Qwen2 technical report](#). *arXiv preprint arXiv:2407.10671*.
- Caiwei Yang, Yanping Chen, Shuai Yu, Ruizhang Huang, and Yongbin Qin. 2025a. [Incorporating edge sharpening and covariance attention for named entity recognition](#). *Neurocomputing*, 643:130402.
- Yuming Yang, Wantong Zhao, Caishuang Huang, Junjie Ye, Xiao Wang, Huiyuan Zheng, Yang Nan, Yuran

Wang, Xueying Xu, Kaixin Huang, Yunke Zhang, Tao Gui, Qi Zhang, and Xuanjing Huang. 2025b. [Beyond boundaries: Learning a universal entity taxonomy across datasets and languages for open named entity recognition](#). In *Proceedings of the International Conference on Computational Linguistics 2025*, pages 10902–10923.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. [RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs](#). In *Advances in Neural Information Processing Systems*, volume 37.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. [GLiNER: Generalist model for named entity recognition using bidirectional transformer](#). In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2024*, pages 5364–5376.

Rui Zhang, Honglak Lee, and Dragomir R. Radev. 2019. [This email could save your life: Introducing the task of email subject line generation](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics 2019*, pages 446–456.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. [UniversalNER: Targeted distillation from large language models for open named entity recognition](#). In *Proceedings of the International Conference on Learning Representations 2024*.

A Extended Preliminaries

A.1 Manifold-Based Alignment

Under the manifold hypothesis, natural language semantics can be regarded as lying on a low-dimensional latent semantic manifold $\mathcal{M}$, where semantically similar texts correspond to nearby points. Although natural language is observed as high-dimensional discrete sequences, different language models can be viewed as learning different coordinate systems or parameterizations of the same underlying semantic manifold.

Given an input text $x \in \mathcal{X}$, a discriminative language model maps it into a discriminative representation manifold $\mathcal{H}_D \subset \mathbb{R}^{d_D}$, while a generative language model maps it into a generative representation manifold $\mathcal{H}_G \subset \mathbb{R}^{d_G}$:

$$f_D : \mathcal{X} \rightarrow \mathcal{H}_D, \quad f_G : \mathcal{X} \rightarrow \mathcal{H}_G. \quad (19)$$

Both manifolds encode the same semantic content, but their coordinates and local geometry may differ due to different tokenizers, architectures, and training objectives. Therefore, semantic manifold

alignment can be formulated as learning a transformation

$$\Phi : \mathcal{H}_D \rightarrow \mathcal{H}_G, \quad (20)$$

which maps discriminative representations into the GLM-side semantic manifold.

The generic alignment objective can be written as

$$\mathcal{L}_{\text{align}}(\Phi) = \frac{1}{N} \sum_{i=1}^N d(\Phi(f_D(x_i)), f_G(x_i)), \quad (21)$$

where $d(\cdot, \cdot)$ denotes a distance or similarity-based discrepancy. In DisCon4Gen, this transformation is implemented through sentence-level and token-level projectors, which map DLM representations into GLM-compatible semantic coordinates.

A.2 Continuous Conditioning

A standard autoregressive language model factorizes the generation probability over discrete tokens. Given an input sequence $\mathbf{x} = (x_1, \dots, x_n)$ and a target sequence $\mathbf{y} = (y_1, \dots, y_T)$, the model defines

$$p(\mathbf{y} | \mathbf{x}) = \prod_{t=1}^T p(y_t | y_{<t}, \mathbf{x}). \quad (22)$$

The input sequence is first mapped into continuous embeddings

$$E(\mathbf{x}) = (\mathbf{e}_{x_1}, \dots, \mathbf{e}_{x_n}), \quad \mathbf{e}_{x_i} \in \mathbb{R}^{d_G}. \quad (23)$$

This discrete-token interface restricts the direct use of continuous semantic representations from external models. Continuous conditioning relaxes this constraint by prepending dynamically generated vectors $\mathbf{P} \in \mathbb{R}^{m \times d_G}$ to the input embeddings:

$$\tilde{\mathbf{E}} = [\mathbf{P}; E(\mathbf{x})] \in \mathbb{R}^{(m+n) \times d_G}. \quad (24)$$

The generation process is then conditioned on the augmented embedding:

$$p(\mathbf{y} | \mathbf{x}, \mathbf{P}) = \prod_{t=1}^T p(y_t | y_{<t}, \tilde{\mathbf{E}}). \quad (25)$$

In DisCon4Gen, $\mathbf{P}$ is not a set of free soft prompts, but is constructed from aligned discriminative representations, including sentence-level conditions $\mathbf{C}_s$ and token-level conditions $\mathbf{C}_t$.

B Details of Loss

B.1 Manifold Geometric Alignment Loss

Given a mini-batch of B samples, let $\hat{\mathbf{S}}_i = f_s(\mathbf{S}_i^D)$ denote the projected DLM sentence representation of the i -th sample, and let $\mathbf{S}_i^G$ denote the corresponding GLM sentence representation. The sentence-level MSE loss is defined as

$$\mathcal{L}_{\text{mse}} = \frac{1}{B} \sum_{i=1}^B \left\| \hat{\mathbf{S}}_i - \mathbf{S}_i^G \right\|_2^2. \quad (26)$$

The cosine loss is defined as

$$\mathcal{L}_{\text{cos}} = \frac{1}{B} \sum_{i=1}^B \left(1 - \cos \left(\hat{\mathbf{S}}_i, \mathbf{S}_i^G \right) \right), \quad (27)$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity.

For bidirectional contrastive alignment, the scaled similarity between the i -th projected DLM representation and the j -th GLM representation is

$$q_{ij} = \frac{\cos \left(\hat{\mathbf{S}}_i, \mathbf{S}_j^G \right)}{\tau}, \quad (28)$$

where τ is the temperature. The DLM-to-GLM and GLM-to-DLM contrastive losses are:

$$\mathcal{L}_{D \rightarrow G} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(q_{ii})}{\sum_{j=1}^B \exp(q_{ij})}, \quad (29)$$

$$\mathcal{L}_{G \rightarrow D} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(q_{ii})}{\sum_{j=1}^B \exp(q_{ji})}. \quad (30)$$

The bidirectional InfoNCE loss is:

$$\mathcal{L}_{\text{nce}} = \frac{1}{2} (\mathcal{L}_{D \rightarrow G} + \mathcal{L}_{G \rightarrow D}). \quad (31)$$

For local alignment, let $\mathbf{Z} \in \mathbb{R}^{K \times d_G}$ denote the resampled DLM local slots, and let $\mathbf{T}^G \in \mathbb{R}^{K \times d_G}$ denote the GLM-side chunk targets. The chunk-level alignment loss is:

$$\mathcal{L}_{\text{chunk}} = \text{MSE}(\mathbf{Z}, \mathbf{T}^G). \quad (32)$$

The complete manifold geometric alignment objective is:

$$\mathcal{L}_1 = \mathcal{L}_{\text{sent}} + \alpha \mathcal{L}_{\text{chunk}}, \quad (33)$$

where

$$\mathcal{L}_{\text{sent}} = \lambda_m \mathcal{L}_{\text{mse}} + \lambda_c \mathcal{L}_{\text{cos}} + \lambda_n \mathcal{L}_{\text{nce}}. \quad (34)$$

B.2 Discriminative Conditioning Loss

Given a target sequence $\mathbf{y} = (y_1, \dots, y_{L_y})$, the autoregressive generation loss is computed only over target tokens:

$$\mathcal{L}_{\text{gen}} = -\frac{1}{L_y} \sum_{i=1}^{L_y} \log p_{\Theta}(y_i \mid \mathbf{E}_{\text{cond}}, y_{<i}), \quad (35)$$

where $\mathbf{E}_{\text{cond}} = [\mathbf{C}_s; \mathbf{C}_t; \mathbf{E}_{T'}]$ denotes the dual-conditioned input embedding. Prefix and prompt positions are masked out during loss computation.

Let $\mathbf{h}_{\text{prompt}} \in \mathbb{R}^{d_G}$ denote the pooled prompt-region hidden representation at the target layer l^* , and let $\hat{\mathbf{S}} \in \mathbb{R}^{d_G}$ denote the aligned sentence-level representation. The sentence-level tuning loss is:

$$\mathcal{L}'_{\text{sent}} = \left\| \mathbf{h}_{\text{prompt}} - \hat{\mathbf{S}} \right\|_2^2 + \left(1 - \cos \left(\mathbf{h}_{\text{prompt}}, \hat{\mathbf{S}} \right) \right). \quad (36)$$

Let $\mathbf{H}_t^{(l^*)} \in \mathbb{R}^{K \times d_G}$ denote the token-prefix hidden states at layer l^* , and let $\mathbf{C}_t \in \mathbb{R}^{K \times d_G}$ denote the token-level condition. The token-level tuning loss is:

$$\mathcal{L}'_{\text{token}} = \frac{1}{K} \sum_{k=1}^K \left[\left\| \mathbf{H}_{t,k}^{(l^*)} - \mathbf{C}_{t,k} \right\|_2^2 + 1 - \cos \left(\mathbf{H}_{t,k}^{(l^*)}, \mathbf{C}_{t,k} \right) \right]. \quad (37)$$

The complete discriminative conditioning tuning objective is:

$$\mathcal{L}_2 = \mathcal{L}_{\text{gen}} + \lambda_s \mathcal{L}'_{\text{sent}} + \lambda_t \mathcal{L}'_{\text{token}}. \quad (38)$$

C Implementation Details

DisCon4Gen is implemented with PyTorch, Hugging Face Transformers, and PEFT. Unless otherwise specified, the main experiments use Meta-Llama-3-8B-Instruct as the GLM backbone and roberta-base as the DLM backbone. Additional backbone combinations are reported in Appendix F.

Data splits. For CoNLL03 and ACE05, we follow the data splits used by Yang et al. (2025a). For CoNLL04, we follow the split setting of Wang et al. (2025). For AESLC and CommonGen, we use the standard benchmark splits from the original datasets (Zhang et al., 2019; Lin et al., 2020). In all downstream experiments, the training and development sets are merged for model training, and the test set is used only for final evaluation. The split statistics are shown in Table 5.

Table 5: **Dataset split statistics.** Train and Dev are merged for training, and Test is used for final evaluation.

Dataset	Train	Dev	Test
CoNLL03	29,441	5,648	5,648
ACE05	24,132	3,171	2,956
CoNLL04	1,283	343	422
AESLC	14,436	1,960	1,906
CommonGen	32,651	993	1,497

Input format and output parsing. All tasks are formulated as conditional generation. For the alignment corpus, each instance is formatted as Context: ... Question: ... Answer: For NER and RE, model outputs are parsed into structured entity or relation tuples. Invalid outputs and unmatched formats are counted as incorrect predictions.

Manifold geometric alignment. During manifold geometric alignment, both the DLM and GLM backbones are frozen. Only the sentence-level projector f_s , token-level projector f_t , and latent resampler $\mathcal{R}$ are optimized. DLM token representations are obtained by averaging the last four hidden layers, and sentence representations are computed with masked mean pooling. On the GLM side, the middle layer is used as the generative semantic manifold unless otherwise specified.

The sentence-level and token-level projectors are implemented as two-layer MLPs with GELU activation and LayerNorm. The latent resampler uses learnable latent queries to compress variable-length DLM token representations into K token-level semantic slots. The coefficients for MSE, cosine, InfoNCE, and chunk-level losses are set to 1.0, 1.0, 0.1, and 0.5, respectively.

The alignment modules are trained for 3 epochs with AdamW, using a learning rate of 1×10^{-4} , weight decay 1×10^{-2} , gradient clipping with maximum norm 1.0, and a linear warmup schedule over 10% of total update steps. The maximum sequence length is 384.

Discriminative conditioning tuning. During discriminative conditioning tuning, the DLM and GLM backbones remain frozen. The alignment modules are initialized from the manifold geometric alignment checkpoint. The aligned sentence representation is expanded into sentence-level conditions through $g_s(\cdot)$, while token-level semantic slots are used as token-level conditions. These con-

Table 6: Default hyperparameters used in the experiments.

Hyperparameter	Value
Alignment epochs	3
Conditioning tuning epochs	3
Alignment learning rate	1×10^{-4}
Projector / condition learning rate	5×10^{-5}
LoRA learning rate	1×10^{-4}
Weight decay	1×10^{-2}
Gradient clipping	1.0
Warmup ratio	10% / 5%
Sentence-level conditions	8
Token-level slots	8
LoRA rank	8
LoRA alpha	16
LoRA dropout	0.05
Max source length	320
Max target length	64
Random seeds	42 / 43 / 44

ditions are concatenated with prompt embeddings and fed into the GLM.

LoRA is applied to the generator attention projections, including q_proj, k_proj, v_proj, and o_proj. The LoRA rank is 8, the scaling factor is 16, and the dropout rate is 0.05. The default setup uses 8 sentence-level conditions, 8 token-level slots, a maximum source length of 320, and a maximum target length of 64.

The model is trained for 3 epochs with AdamW. The learning rate is 5×10^{-5} for projectors and conditioning modules, and 1×10^{-4} for LoRA parameters. Weight decay is set to 1×10^{-2} , gradient clipping uses maximum norm 1.0, gradient accumulation is 4, and the warmup ratio is 5%. For each downstream dataset, we repeat discriminative conditioning tuning three times using different random seeds and report the averaged results in the main tables.

Training Cost and Hardware. All experiments were conducted on NVIDIA RTX A6000 GPUs with 48GB memory. Unless otherwise specified, all experiments were trained on a single GPU.

For the representative Qwen3-8B + BERT-base configuration, the trainable parameter statistics are summarized in Table 7. During manifold geometric alignment, both the DLM and GLM backbones remain frozen, and only the cross-model projectors and latent resampler are optimized, resulting in approximately 56.7M trainable parameters. During

Table 7: **Representative training cost and trainable parameter statistics.**

Training Phase / Dataset	Trainable Params	Training Time	GPU Memory
Alignment pretraining	56.7M	3.8 h	within 48GB
CoNLL04 tuning	215.4M / 8.52B	13 min	within 48GB
AESLC tuning	215.4M / 8.52B	2.7 h	up to $\sim$ 48GB
CommonGen tuning	215.4M / 8.52B	6.6 h	within 48GB

discriminative conditioning tuning, the DLM and GLM backbones are still frozen, and only LoRA adapters together with conditioning modules are updated, resulting in approximately 215.4M trainable parameters out of 8.52B total parameters.

GPU memory usage was monitored using `nvidia-smi`. The largest configuration, using Qwen3-8B with 512-token inputs, nearly saturated the 48GB memory capacity of the RTX A6000 GPU.

Since some historical runs did not record exact peak CUDA memory statistics, the reported memory values should be interpreted as approximate hardware requirements rather than profiler-level peak measurements.

In practice, the additional computational overhead of DisCon4Gen mainly comes from: (1) the forward pass of the frozen DLM encoder, and (2) the extended conditioning prefix sequence injected into the GLM.

Compared with full-parameter instruction tuning of large generative models, DisCon4Gen substantially reduces the number of trainable parameters while preserving strong downstream performance across both semantic understanding and controlled generation tasks.

Decoding. For structured extraction tasks, including CoNLL03, ACE05, and CoNLL04, greedy decoding is used to reduce format variation. For AESLC and CommonGen, beam search is used. Following the CommonGen decoding setting in NeuroLogic Decoding (Lu et al., 2021, 2022), the default beam size is set to 6 for CommonGen. To ensure fair comparison, all LoRA-SFT baselines are evaluated with the same decoding strategy and beam size as DisCon4Gen. No additional lexical constraints or task-specific constrained decoding are used for DisCon4Gen unless explicitly stated.

D Alignment Corpus Analysis

We further examine the influence of the alignment corpus. Since the alignment stage is designed to

Table 8: **Effect of different alignment corpora.** CoNLL04 is evaluated by F1, and AESLC by ROUGE-1/2/L.

Alignment Text	CoNLL04	AESLC		
	F1	R-1	R-2	R-L
ScienceQA	78.74	43.26	26.65	41.88
Downstream inputs only	78.67	44.13	26.56	39.80
WikiText-103	78.15	42.91	25.98	35.99
GLM-only LoRA	56.37	34.15	19.73	33.55

learn model-space correspondence rather than task-specific decision boundaries, ScienceQA is used as a general semantic alignment corpus. It contains natural language contexts, questions, and answers, but does not share task labels or output schemas with the downstream NER, relation extraction, email subject generation, or commonsense generation datasets.

To reduce potential leakage concerns, we ensure that the downstream training and test instances are not used in the alignment stage. The alignment corpus is only used to construct paired DLM and GLM representations of the same input text, and no downstream labels are involved. Therefore, the observed improvements are unlikely to come from memorizing downstream annotations.

The cross-task gains suggest that the learned transformation captures general DLM-GLM representation correspondence rather than task-specific patterns. Nevertheless, the quality and domain coverage of the alignment corpus may affect performance. The additional corpus ablations below further examine the robustness of the learned alignment across different unlabeled text sources.

D.1 Alignment Corpus Ablation

To further examine whether the effectiveness of DisCon4Gen depends on a specific alignment corpus, we compare ScienceQA with two alternative unlabeled text sources: downstream inputs without task labels or output schemas, and WikiText-103. The results are reported in Table 8.

Table 9: **Cross-corpus alignment diagnostics.** Higher values indicate stronger DLM–GLM correspondence.

Corpus	Pair Cos.	Raw Corr.	Proj. Corr.	Gain	R@1
ScienceQA	0.8929	0.5752	0.6991	+0.1238	1.0000
Downstream	0.9292	0.8739	0.9004	+0.0265	0.9800
WikiText	0.9200	0.7271	0.8369	+0.1099	1.0000

The results show that useful DLM–GLM correspondence can be learned from different unlabeled text sources. ScienceQA, downstream inputs, and WikiText-103 all substantially outperform the GLM-only LoRA baseline. This indicates that the gains of DisCon4Gen do not rely on downstream labels or output schemas. Nevertheless, the differences across alignment corpora also suggest that corpus characteristics can affect the quality of the learned representation alignment.

D.2 Cross-Corpus Alignment Diagnostics

We further evaluate the learned mapping on different text sources using representation-level diagnostics. Table 9 reports matched-pair cosine similarity, structural correspondence before and after projection, the resulting correspondence gain, and projected Recall@1.

Projection consistently improves structural correspondence across the three corpora, with gains ranging from 0.0265 to 0.1238. The projected representations also achieve near-perfect matched-pair retrieval, with Recall@1 reaching 0.98 or above. These results provide additional evidence that the learned transformation captures transferable cross-model correspondence rather than an arbitrary projection tied to a single alignment corpus.

E Baseline Details

AESLC baselines. For AESLC, PEGASUS and ROUGE-Ext are taken from Sahu and Sanjeevi (2021). PEGASUS is a pre-trained abstractive summarization model, while ROUGE-Ext fine-tunes PEGASUS using important sentences selected by ROUGE. Core-set, BAS, and LDCAL are taken from Li et al. (2024a). The Core-set and BAS results are not taken from their original papers, but from the unified AESLC evaluation in LDCAL.

CommonGen baselines. For CommonGen, GPT-2, T5-Large, and BART are standard benchmark baselines reported in the CommonGen benchmark (Lin et al., 2020). RE-T5 augments commonsense generation with retrieved knowledge (Wang et al., 2021), while NeuroLogic ap-

plies predicate-logic constrained decoding for controllable concept-to-sentence generation (Lu et al., 2021).

Semantic understanding baselines. For semantic understanding tasks, recent information extraction systems are included. InstructUIE formulates information extraction as instruction-guided generation (Wang et al., 2023). UniNER focuses on open-domain named entity recognition (Zhou et al., 2024). GLiNER is a generalizable NER model with arbitrary entity types (Zaratiana et al., 2024). ReasoningNER introduces reasoning chains for NER (Huang et al., 2026). KnowCoder represents extraction schemas in code-style structures (Li et al., 2024c). IEPile provides a large-scale schema-conditioned instruction corpus (Gui et al., 2024). ProUIE, MR-UIE, B²NER, and Multi-LoRA are also included as recent universal or parameter-efficient information extraction baselines (Liu et al., 2026; Li et al., 2025; Yang et al., 2025b; Jin et al., 2025).

SFT baselines. LoRA-based supervised fine-tuning baselines are implemented with recent open-weight LLMs, including Llama3-8B, Qwen2.5-1.5B, Qwen3-8B, Mistral-7B, and Gemma-2-2B. These baselines use the same instruction format, training data, and evaluation protocol as DisCon4Gen.

F Backbone Configuration Analysis

To examine the effect of different backbone combinations, DisCon4Gen is evaluated with multiple DLM and GLM configurations, including BERT-base and RoBERTa-base as discriminative encoders, and Qwen2.5-1.5B, Qwen3-8B, and Llama3-8B as generative backbones. The results are reported in Table 10.

Several observations can be made. First, stronger GLM backbones generally lead to better downstream performance. Llama3-8B consistently achieves the best overall results across most tasks, while Qwen2.5-1.5B shows noticeably weaker generation and extraction capability due to its smaller model capacity. Second, RoBERTa-base consistently outperforms BERT-base under the same GLM setting, suggesting that stronger discriminative semantic representations provide more effective conditioning signals for manifold alignment and generation.

Overall, these results indicate that DisCon4Gen

Table 10: **Effect of different DLM and GLM backbones.** Bold and underlined values denote the best and second-best results, respectively. Main experiments use Llama3-8B and RoBERTa-base.

Backbone		Semantic Understanding			AESLC			CommonGen		
GLM	DLM	CoNLL03	ACE05	CoNLL04	R-1	R-2	R-L	B-4	CIDEr	SPICE
Llama3-8B	BERT-base	93.21	87.14	<u>77.86</u>	<u>41.72</u>	<u>24.25</u>	<u>40.43</u>	37.97	<u>18.23</u>	<u>34.43</u>
Qwen3-8B	BERT-base	91.78	85.05	70.16	39.66	22.46	38.10	36.96	16.25	35.12
Qwen2.5-1.5B	BERT-base	80.10	63.91	69.20	30.92	19.38	20.29	31.57	14.65	34.41
Llama3-8B	RoBERTa-base	94.99	<u>86.52</u>	78.74	43.26	26.65	41.88	41.37	19.17	32.58
Qwen3-8B	RoBERTa-base	<u>92.36</u>	85.55	75.62	42.98	24.43	41.10	<u>39.62</u>	17.17	33.90
Qwen2.5-1.5B	RoBERTa-base	83.04	78.22	66.43	31.36	14.37	20.56	32.24	14.69	30.58

Table 11: **Results with modern embedding encoders.** CoNLL04 is evaluated by F1 and AESLC by ROUGE-1/2/L.

Encoder	GLM	CoNLL04		AESLC		
		F1	R-1	R-2	R-L	
Qwen3-Embedding-0.6B	Llama3-8B	65.96	38.60	22.61	37.78	
llama-nemotron-embed-1b-v2	Qwen3-8B	76.13	39.86	21.46	38.69	

is compatible with different DLM and GLM backbones and benefits from improvements on both the discriminative and generative sides. Unless otherwise specified, the main experiments use Llama3-8B as the GLM backbone and RoBERTa-base as the DLM backbone.

G Additional Backbone Robustness Analysis

G.1 Modern Embedding Encoders

The main backbone analysis uses BERT-base and RoBERTa-base as discriminative encoders. To examine whether DisCon4Gen can also operate with more recent LM-family embedding models, we additionally evaluate Qwen3-Embedding-0.6B and llama-nemotron-embed-1b-v2. The results are reported in Table 11.

The results demonstrate that DisCon4Gen can be instantiated with modern embedding encoders and is not restricted to BERT/RoBERTa-style DLMs. However, newer embedding encoders do not automatically yield stronger downstream performance. One possible explanation is that retrieval-oriented embedding models emphasize sentence-level semantic matching, whereas DisCon4Gen also relies on fine-grained token- and chunk-level structure for local alignment and conditioning. This result further highlights the importance of representation compatibility across heterogeneous encoder families.

Table 12: **Alignment layer selection with Qwen3-8B.** Middle-layer representations consistently provide the strongest alignment target.

Alignment Layer	CoNLL04 F1	AESLC R-L	CommonGen B-4
Lower	59.85	34.92	31.85
Middle	75.62	41.10	39.62
Upper	69.21	38.15	38.90
Final	72.84	39.45	39.11

G.2 Additional GLM Alignment Layer Analysis

The main layer-selection analysis uses Llama3-8B as the GLM backbone. To examine whether the observed trend is specific to this model, we repeat the experiment using Qwen3-8B. The results are reported in Table 12.

The Qwen3-8B results show the same overall trend as the Llama3-8B analysis: the middle layer provides the strongest alignment target, while the lower layer performs substantially worse and the upper and final layers remain below the middle layer. This consistency suggests that the preference for intermediate representations is not specific to a single GLM backbone. A possible interpretation is that lower layers contain less mature semantic information, whereas later layers are increasingly shaped by the autoregressive prediction objective.

H Decoding Strategy Analysis

Following prior work on constrained decoding for CommonGen, which commonly compares greedy decoding, sampling-based decoding, and beam-search variants (Lu et al., 2022), we examine the effect of decoding strategies under different decoding settings. Specifically, we evaluate greedy decoding, sampling, and beam search with varying beam sizes on CommonGen.

Table 13: **Decoding strategy analysis on Common-Gen.** Beam size 6 achieves the best overall performance and is used as the default setting in the main experiments.

Decoding	BLEU-4	CIDEr	SPICE
Greedy	38.59	18.01	31.64
Sampling	33.67	16.19	30.76
Beam size 4	40.16	15.81	31.85
Beam size 6	41.37	19.17	32.58
Beam size 8	39.96	18.53	31.09

As shown in Table 13, beam search generally performs better than greedy decoding and sampling, indicating that multi-candidate search better exploits the aligned discriminative conditions during concept-to-sentence generation.

Among the beam-search settings, beam size 6 achieves the best performance across all three metrics, with 41.37 BLEU-4, 19.17 CIDEr, and 32.58 SPICE. Increasing the beam size to 8 does not bring further gains and slightly decreases performance, suggesting that an excessively large search space may introduce less suitable generations. Therefore, beam size 6 is used as the default decoding setting for CommonGen in the main experiments.

I Ablation Variant Definitions

This section defines the controlled variants used in Table 3. These variants are designed to test whether the improvements come from explicit DLM-GLM alignment, multi-granular conditioning, and joint tuning, rather than from simply adding continuous prefixes or additional trainable parameters. All variants use the same GLM backbone, prefix length, training data, and evaluation protocol as DisCon4Gen.

Conditioning components. *S only* keeps only the sentence-level condition $\mathbf{C}_s$ and removes the token-level condition $\mathbf{C}_t$ from the conditional input:

$$\mathbf{E}_{\text{cond}} = [\mathbf{C}_s; \mathbf{E}_{T'}]. \quad (39)$$

T only keeps only the token-level condition $\mathbf{C}_t$ and removes $\mathbf{C}_s$:

$$\mathbf{E}_{\text{cond}} = [\mathbf{C}_t; \mathbf{E}_{T'}]. \quad (40)$$

These variants evaluate the individual contributions of global semantic conditioning and local semantic conditioning.

Training mechanism. *w/o LoRA* removes LoRA modules from the GLM and only trains the conditioning-related modules, while the GLM backbone remains frozen. This variant tests whether external semantic conditions alone are sufficient for adapting the generator.

w/o Init. keeps the same conditioning architecture but does not initialize the projectors from manifold geometric alignment. Instead, the projectors and conditioning modules are randomly initialized before discriminative conditioning tuning. This variant evaluates the importance of alignment-based initialization.

Alignment objectives. *w/o NCE* removes the contrastive loss $\mathcal{L}_{\text{nce}}$ from the global sentence-level alignment objective. The remaining sentence-level alignment relies on MSE and cosine constraints.

w/o Chunk removes the local chunk-level alignment loss $\mathcal{L}_{\text{chunk}}$ from the manifold geometric alignment objective, leaving only sentence-level alignment. These variants test whether batch-level discriminability and local geometric alignment are necessary for learning transferable discriminative conditions.

Joint tuning objective. *w/o S-Tune* removes the sentence-level tuning loss $\mathcal{L}'_{\text{sent}}$ from $\mathcal{L}_2$.

w/o T-Tune removes the token-level tuning loss $\mathcal{L}'_{\text{token}}$ from $\mathcal{L}_2$.

These variants examine whether preserving sentence-level and token-level conditions during discriminative conditioning tuning improves downstream generation.

J Conditioning Slot Intervention Analysis

To examine whether the local conditioning slots encode meaningful input-specific structure rather than merely increasing prefix length, we conduct two controlled interventions. *Permuted* randomly shuffles the order of token-level conditions $\mathbf{C}_t$, while *Cross-sample* replaces the conditions of the current input with those from another sample. The conditioning length and model architecture remain unchanged. The results are reported in Table 14.

Both interventions consistently reduce performance. Shuffling the local slots causes particularly large degradation, indicating that their ordering carries useful local semantic structure. Replacing the conditions with those from another input also hurts performance, showing that the conditions contain input-specific information. These results suggest

Table 14: **Intervention analysis of token-level conditioning slots.** CoNLL04 reports F1 and AESLC reports ROUGE-L.

Task	Alignment Corpus	Normal	Permuted	Cross-sample
CoNLL04	ScienceQA	78.74	47.67	64.83
CoNLL04	WikiText	73.15	54.25	67.75
AESLC	ScienceQA	41.88	28.54	35.45
AESLC	WikiText	35.85	26.04	33.40

that the local slots serve as structured semantic conditions rather than functioning only as additional continuous prefix tokens.