DiFRa: A Unified Framework for Harmonizing Semantic Diversity and Factual Consistency in Question-Answer Generation

Zhenqin Li[1], Shengyong Ding[2], Shuangyin Li[1]*
[1]School of Computer Science, South China Normal University, Guangzhou, China
[2]Faculty of Data Science, City University of Macau, Macau, China
zhenqinli@m.scnu.edu.cn, syding@cityu.edu.mo, shuangyinli@scnu.edu.cn

Abstract

Question-Answer Generation (QAG) is essential for alleviating the cold-start problem in domain-specific large language model (LLM) post-training, where high-quality data is severely scarce. Effective training samples include rich semantic diversity and rigorous factual consistency. Thus, it is necessary to consider the inherent tension between semantic breadth and factual fidelity. However, it is extremely challenging to trade off semantic diversity against factual consistency, in that generalization across the semantic space must be achieved effectively and reliably, and factual integrity must be ensured as well. To address this issue, we propose an effective framework, namely DiFRa, that integrates continuous concept diffusion with discrete knowledge graph constraints to balance semantic diversity and factual consistency. Specifically, the proposed DiFRa models discrete concepts as a continuous latent distribution to sample embeddings that capture rich semantic variations, and constructs a refined knowledge graph as explicit factual constraints. Then, a diversity and consistency aware mechanism is designed to dynamically integrate both embeddings and the knowledge graph for QA pairs generation. Furthermore, we introduce SeFa, which harmonizes semantic entropy and consistency scores to quantify the trade-off between diversity and correctness. Extensive experiments demonstrate that DiFRa consistently outperforms the baseline models, validating its efficacy in reconciling the tension to generate semantically diverse and factually consistent QA pairs. The source code is publicly available [1].

1 Introduction

Question-Answer Generation (QAG) is critical for generating high-quality QA pairs, which are essential for many tasks, particularly for mitigating data cold-start problems in the post-training of large language models (LLMs) for vertical domains. Unlike open domains with abundant training data, specialized domains often suffer from severe data scarcity. This scarcity hinders effective LLM adaptation, since LLMs pre-trained on general-domain data fail to transfer to domain-specific scenarios. Therefore, QAG becomes crucial for generating high-quality QA pairs to alleviate data cold-start in domain-specific LLM post-training.

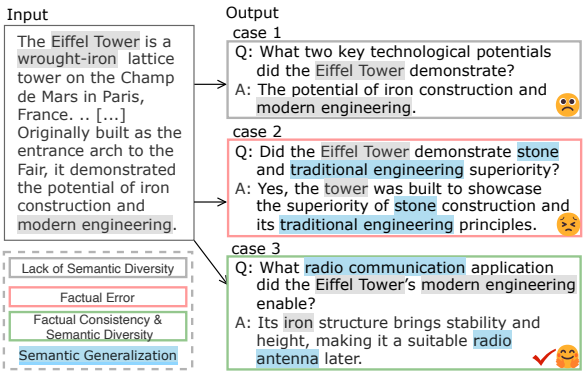

Figure 1: The first case reveals the problem that simple generation lacks expressive diversity. The second case shows that unconstrained generalization results in factual errors. The third case illustrates the trade-off between semantic diversity and factual consistency.

In general, high-quality QA pairs are distinguished by rich semantic diversity and rigorous factual consistency. In this context, *Semantic Diversity* refers to the representation of a rich and varied distribution of information across the generated QA pairs, whereas *Factual Consistency* denotes objective veracity, ensuring alignment with real-world facts. Collectively, these properties form the foundation of effective training samples, enhancing model generalization and robustness, respectively.

Theoretically, model training can be viewed as solving an underdetermined system, where generated QA pairs act as constraints that narrow the solution space. From the perspective of linear alge-

*Corresponding author.
[1]https://github.com/zqinli/DiFRa

Findings of the Association for Computational Linguistics: ACL 2026, pages 29857–29875

bra, semantic diversity of the QA pairs introduces linearly independent constraints that increase the system's *Rank*. Such semantic diversity is vital for mitigating feature redundancy, since highly correlated samples induce rank deficiency by reducing informational uniqueness. Conversely, factual consistency of the QA pairs ensures that newly added constraints do not bring in contradictions of facts. Mathematically, without such factual consistency, the augmented matrix would result in system incompatibility (i.e., $\text{rank}(A) < \text{rank}([A \mid b])$), where A and b represent the feature and target components of the combined set of original and generated QA pairs, rendering the system jointly infeasible. Fundamentally, achieving a robust and generalizable solution requires jointly optimizing the two properties, since maximizing effective rank to boost information gain risks introducing noise, which manifests as incompatible constraints. Thus, effective QAG must reconcile the tension between semantic breadth and factual fidelity, where the generated QA pairs are semantically expansive enough to reduce system uncertainty, yet factually rigorous sufficient to ensure validity. Achieving this balance is critical for effective training, ensuring convergence to a robust solution that is unhindered by redundant or contradictory constraints.

However, it is the inherent trade-off between semantic diversity and factual consistency that presents two primary challenges. Firstly, **how can the generation process effectively and reliably achieve generalization on the semantic space?** As illustrated in the first case of Figure 1, an emphasis on conservatism during QAG often leads to trivial, direct reproduction of source spans. While factually consistent, the generated QA pairs offer limited information gain due to their redundancy, as they lack the semantic variation needed to facilitate model adaptation. Secondly, **how can factual constraints be rigorously integrated into the QAG process to ensure factual integrity?** As shown in the second case of Figure 1, unconstrained augmentation directly contradicts source facts. This leads to samples with rich semantic variations but severe factual inconsistencies, thus injecting noise into the generated QA dataset. In fact, the third case of Figure 1 is recommended, which introduces independent constraints that maximize information gain while maintaining strict factual compatibility.

Unfortunately, although there are some existing studies on QAG, these works either prioritize diversity (Eo et al., 2023; Zhang and Yang, 2023; Shahgir et al., 2025) or consistency (Yao et al., 2022; Gabburo et al., 2023; Schmidt et al., 2024). Thus, resolving the inherent contradiction between semantic diversity and factual consistency remains a critical goal for the task of QAG.

To address these issues, we propose **DiFRa**, a novel framework that integrates continuous concept **DiF**fusion for semantic diversity with discrete knowledge g**Ra**ph constraints for the purpose of factual consistency. Specifically, a concept construction and diffusion module is introduced, which employs diffusion models to map discrete concepts into a continuous latent distribution, enabling the sampling of diverse concept embeddings with rich semantic variations. Meanwhile, a factual constraint construction module is developed to establish a refined knowledge graph that serves as the basis for rigorous constraints. Finally, through a well-designed diversity and consistency aware mechanism, these continuous embeddings and discrete graph constraints are dynamically fused into the QAG process. Furthermore, given that existing metrics fail to adequately evaluate semantic diversity and factual consistency, we present an evaluation metric, called **SeFa**, which harmonizes the **Se**mantic entropy and the **Fa**ctual consistency score, quantifying the trade-off between diversity and correctness. Our contributions are threefold:

- We propose DiFRa, an effective framework designed to reconcile the inherent tension between semantic diversity and factual consistency in QAG tasks.
- A well-designed diversity and consistency aware mechanism is introduced to unify the concept construction and diffusion module and the factual constraint construction module, fusing continuous embeddings with discrete constraints to guide the QAG process.
- The SeFa metric is first introduced, which harmonizes semantic entropy and the factual consistency score to quantify the trade-off between diversity and correctness.

2 Related Work

Early QAG relied on multi-stage pipelines to construct QA pairs (Yao et al., 2022), but suffered from error propagation. To address this, some studies adopted a one-stop approach to generate QA pairs (Cui et al., 2021), as well as large-scale synthetic QA resources coupled with retrieval (Lewis et al., 2021) and end-to-end meth-

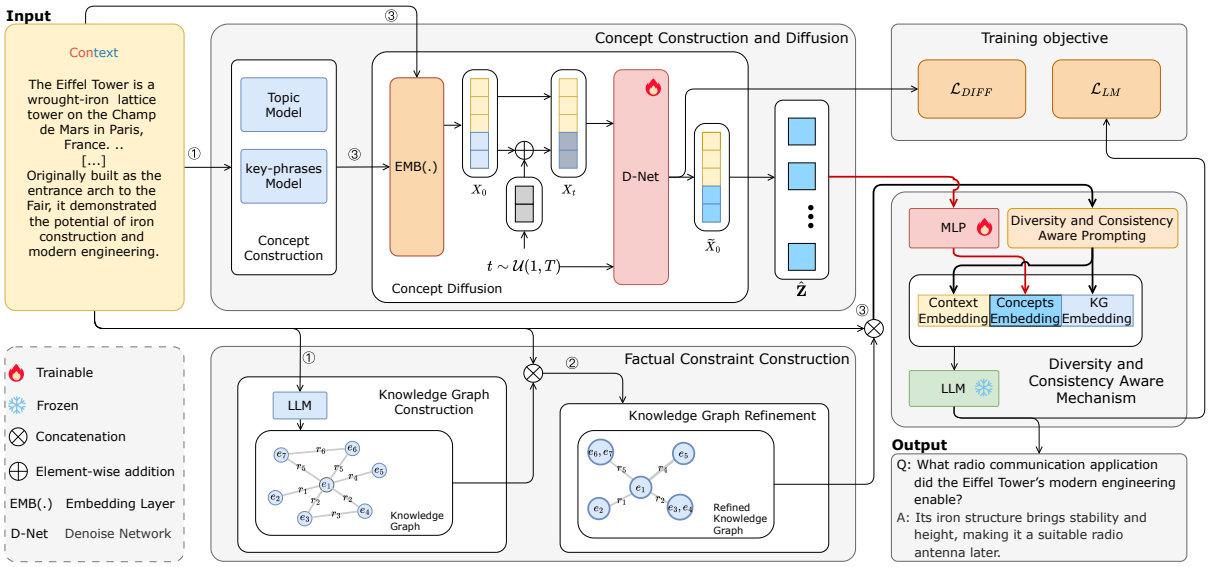

Figure 2: The overall architecture of the DiFRa framework. ① to ③ show the steps of the DiFRa pipeline. Concept Construction and Diffusion incorporates concept formulation and diffusion processes to capture diverse concept embeddings. Factual Constraint Construction utilizes the constructed knowledge graph to improve factual constraints. The Diversity and Consistency Aware Mechanism dynamically integrates concept embeddings and a knowledge graph to guide a frozen LLM in generating QA pairs.

ods (Ushio et al., 2023). Recently, Shahgir et al. (2025) utilized LLM few-shot prompting to generate QA pairs. Additionally, some methods target diversity and consistency. To enhance generation diversity, variational latent variable models introduce randomness via latent sampling, expanding the semantic solution space (Shen et al., 2020; Yang et al., 2021; Ma et al., 2024). Conversely, to improve consistency, prior work enforces answerability via round-trip filtering (Alberti et al., 2019) or constraint-based control, such as procedural graphs (Pham et al., 2024) and difficulty-adaptive frameworks (Tomikawa et al., 2024). However, existing approaches tend to be either diversity-oriented or constraint-driven, while the simultaneous optimization of semantic diversity and factual consistency remains underexplored.

Diffusion models generate high-quality samples by learning to reverse a gradual noising process (Ho et al., 2020; Song et al., 2021a). Kingma et al. (2021) and Ho and Salimans (2021) optimized continuous diffusion models by introducing variational formulations and classifier-free guidance. To adapt these techniques to discrete text, some methods map tokens to embeddings and perform diffusion in latent spaces (Li et al., 2022; Dieleman et al., 2022). Building on this formulation, many studies have further developed specialized architectures and sequence-to-sequence frameworks to optimize text generation (Gong et al., 2023a; He et al., 2023; Yuan et al., 2024; Cheng and Li, 2024). Crucially, by operating in a smooth semantic space, continuous latent diffusion allows for the exploration of diverse concepts for QAG.

Knowledge Graphs (KGs) have been integrated into pre-trained models in early approaches, including K-BERT (Liu et al., 2020), ERNIE (Sun et al., 2019), and DKPLM (Zhang et al., 2022), to enhance language representations. Recently, the focus has shifted towards integrating KGs with LLMs to bolster generation and reasoning capabilities. This includes fine-tuning LLMs for improved knowledge manipulation (Chen et al., 2025) and utilizing KG retrieval to support complex multi-hop QA (Jiang et al., 2023b). Furthermore, graph-based augmentation has been shown to promote faithful reasoning (Sui et al., 2025). By leveraging such knowledge to constrain generation, KGs provide a robust mechanism for ensuring factual consistency.

3 DiFRa

3.1 Problem Formulation and Overview

The goal of Question-Answer Generation (QAG) is to generate a set of Question-Answer (QA) pairs given a context. Formally, an LLM, denoted as a generator $f_G(\cdot)$, processes a prompt constructed by applying $\mathrm{P}(\cdot)$ to the context C. The generation

process can be expressed as:

$$\hat{y} = f_G(\mathrm{P}(C)), \tag{1}$$

where $\hat{y} = \{(q_i, a_i)\}_{i=1}^{n}$ represents the generated set of QA pairs, in which q_i represents the i-th question and a_i is the corresponding answer.

Figure 2 illustrates the architecture of DiFRa, which comprises three key components: Concept Construction and Diffusion (CCD), Factual Constraint Construction (FCC), and the Diversity and Consistency Aware Mechanism (DCAM). The CCD module comprises two stages: Concept Construction and Concept Diffusion. In the Concept Construction stage, a set of concepts, $\mathcal{Z}_i = \{z_j\}_{j=1}^{\ell}$, is derived via topic modeling and keyword extraction, based on the source corpus $\mathcal{C}$ and a specific context $C_i \in \mathcal{C}$. Then, in the Concept Diffusion stage, a diffusion model conditioned on C_i generalizes the concepts $\mathcal{Z}_i$, yielding the diverse concept embeddings denoted as $\hat{\mathbf{Z}}_i$. The FCC module also has two stages: Knowledge Graph Construction and Knowledge Graph Refinement. In the first stage, a fine-grained knowledge graph $\mathcal{G}_i$ is constructed from the context C_i. Subsequently, the refinement stage takes $\mathcal{G}_i$ as input and conditions on C_i to obtain the refined factual knowledge graph $\mathcal{G}_i^*$. The DCAM module injects the diverse concept embeddings $\hat{\mathbf{Z}}_i$ into the LLM's input embedding space to promote diversity, and leverages the refined factual knowledge graph $\mathcal{G}_i^*$ to improve factual consistency. Besides, the backbone LLM is kept frozen, enabling flexible application in low-resource scenarios.

3.2 Concept Construction and Diffusion

To promote semantic diversity during QAG, it is essential to construct a set of semantically rich and diverse concepts. As shown in Figure 2, latent concepts are extracted from the context in the form of topics and key-phrases, and a conditional diffusion model is subsequently employed to generalize and diversify these concepts to guide the QAG process.

Concept Construction. In this stage, topic modeling and key-phrase extraction are primarily performed. Topic modeling is employed to uncover the semantic information in the source corpus $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$, as topic-guided features can effectively enhance language models (Xiao et al., 2023). This process is expressed as a mapping from each context C_i to a set of relevant topics $\mathcal{T}_i \subseteq \mathcal{T}$, where $\mathcal{T}$ represents the set of all topics identified across the entire corpus. For each context C_i, the set of topic words $\mathcal{W}_i$ is defined as the union of the top-k keywords derived from every relevant topic $t \in \mathcal{T}_i$. To further capture fine-grained semantic information, key-phrase extraction is performed on each context C_i, yielding a set of key-phrases denoted as $\mathcal{K}_i$. The concept set of the context is defined as:

$$\mathcal{Z}_i = \mathcal{W}_i \cup \mathcal{K}_i, \tag{2}$$

where $\mathcal{Z}_i$ denotes the concept set of context C_i.

Concept Diffusion. A continuous diffusion model is employed on the concept set $\mathcal{Z}_i$ to enrich semantic diversity. By leveraging conditional generation within the latent space, semantically rich representations are reconstructed from Gaussian noise, yielding diverse concept embeddings. Specifically, the context tokens $\mathbf{W}^C$ and concept tokens $\mathbf{W}^Z$ are jointly projected into a unified embedding $\mathbf{E}_W = \mathrm{EMB}([\mathbf{W}^C; \mathbf{W}^Z])$, which serves as the anchor for the diffusion process. To map the discrete embeddings into the continuous diffusion space (Li et al., 2022), the latent state $\mathbf{x}_0$ is initialized by a adding little Gaussian noise to $\mathbf{E}_W$.

During training, the noisy latent state $\mathbf{x}_t$ is sampled at a random time step $t \sim \mathcal{U}(1, T)$ using the standard reparameterization trick (Ho et al., 2020). Subsequently, the model predicts the denoised concept embeddings $\hat{\mathbf{Z}}$ from $\mathbf{x}_t$, which serve as input for the LLM to generate QA pairs.

In the Concept Diffusion inference, as illustrated in the upper part of Figure 3, the process initiates by combining the context embedding $\mathbf{E}_C$ with sampled Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to construct the initial latent state $\mathbf{x}_T$. Subsequently, the denoising network iteratively recovers the enhanced concept embeddings over T time steps by solving the probability flow ODE (Song et al., 2021b; Li et al., 2025) via DPM-Solver++ (Lu et al., 2023). During this process, the context partition is explicitly anchored to $\mathbf{E}_C$ at each step. The trajectory terminates at $\tilde{\mathbf{x}}_0$, yielding the final output $\hat{\mathbf{Z}}$.

Denoising Network. As illustrated in Figure 3, the denoising network is implemented via a BERT-initialized Transformer (Vaswani et al., 2017) architecture, which enables the processing of continuous latent embeddings. To encode the diffusion timestep information, the discrete timestep t is mapped to a continuous vector $\mathbf{e}_t \in \mathbb{R}^{1\times d}$ via an MLP projection. Distinct from standard diffusion approaches that perform element-wise addition, the network input is constructed by prepending $\mathbf{e}_t$ as

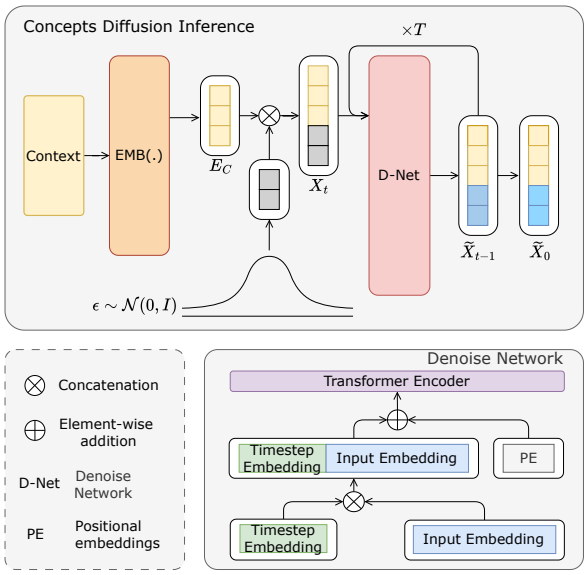

Figure 3: Illustration of the Inference Process and Denoise Network. Top: The iterative sampling process of the diffusion. Bottom: The Transformer-based Denoise Network fusing timestep and positional embeddings.

a prefix embedding to the noisy state $\mathbf{x}_t \in \mathbb{R}^{L\times d}$. Subsequently, positional embeddings are added to maintain sequence awareness. The Transformer encoder f_{enc} processes the input sequence utilizing an extended attention mask, which allows each position to attend to the timestep, context, and latent concepts. The output $\mathbf{H} \in \mathbb{R}^{(L+1)\times d}$ is expressed as:

$$\mathbf{H} = f_{\text{enc}}\left([\mathbf{e}_t; \mathbf{x}_t] + \mathbf{p}\right), \tag{3}$$

where f_{enc} denotes the Transformer encoder, $[\cdot;\cdot]$ denotes concatenation, and $\mathbf{p} \in \mathbb{R}^{(L+1)\times d}$ represents the trainable positional embeddings. Finally, the target output $\hat{\mathbf{Z}}$ is obtained by applying a mask to $\mathbf{H}$ to extract the concept representations.

3.3 Factual Constraint Construction

To improve factual consistency during the QAG process, the Factual Constraint Construction (FCC) module is employed to establish explicit knowledge constraints. As shown at the bottom of Figure 2, the module transforms the unstructured context into a structured knowledge graph through two stages: knowledge graph construction and refinement.

Knowledge Graph Construction. In this stage, a fine-grained knowledge graph $\mathcal{G}_i$ is constructed from the input context C_i. This structured representation provides the foundation for subsequent factual constraints. Specifically, leveraging the instruction-following capabilities of LLMs, entities and relationships from the context C_i are identified. This process is represented as $\mathcal{G}_i = \text{LLM}(C_i)$, where the knowledge graph is defined as $\mathcal{G}_i = \{(h, r, t) \mid h, t \in \mathcal{E}_i, r \in \mathcal{R}_i\}$. Here, h and t denote the head and tail entities from the entity set $\mathcal{E}_i$, and r represents a relation from the relation set $\mathcal{R}_i$.

Knowledge Graph Refinement. Following the construction phase, the initial knowledge graph $\mathcal{G}_i$ often contains redundant and fragmented triples. To consolidate fragmented facts and distill critical information, a concise, high-density factual knowledge graph $\mathcal{G}_i^*$ is constructed. Specifically, the LLM is employed to aggregate and compress redundant triples within $\mathcal{G}_i$. The refinement process is formally defined as $\mathcal{G}_i^* = \text{LLM}(\mathcal{G}_i \mid C_i)$, where $\mathcal{G}_i^*$ represents the refined knowledge graph comprising a set of consolidated facts. This refinement guarantees that $\mathcal{G}_i^*$ retains only the essential facts by merging fragmented details and filtering out superfluous elements.

3.4 Diversity and Consistency Aware Mechanism

In the generation process, it is crucial for the LLM to utilize both the diverse concept embeddings and the refined knowledge graph effectively. As shown in the right of Figure 2, the Diversity and Consistency Aware Mechanism (DCAM) is employed to incorporate both the diverse concept embeddings and the refined knowledge graph into the LLM.

First, a projection layer is employed to map the semantically diverse concept embeddings $\hat{\mathbf{Z}}$ into the LLM's embedding space. Formally, this is represented as:

$$\mathbf{E}^d = f_{\to d_h}(\hat{\mathbf{Z}}) = \{\mathbf{e}_i^d\}_{i=1}^{n_z}, \tag{4}$$

where $f_{\to d_h}$ denotes the projection network employing an MLP, and d_h represents the hidden dimension of the LLM's token embedding layer.

Then, the context and the refined knowledge graph are tokenized and converted into embeddings via the LLM's tokenizer. Specifically, the triples in $\mathcal{G}_i^*$ are linearized into a textual sequence $S_{\mathcal{G}}$ and then tokenized into $T^g = \{t_k^g\}_{k=1}^{n_g}$, where t_k^g is the k-th token and n_g is the number of tokens in the graph sequence. The graph embeddings are formulated as:

$$\mathbf{E}^g = f_{\text{emb}}(T^g) = \{\mathbf{e}_k^g\}_{k=1}^{n_g}, \tag{5}$$

where f_{emb} denotes the token embedding layer of the LLM, and $\mathbf{E}^g \in \mathbb{R}^{n_g \times d_h}$ is the embeddings of the refined knowledge graph. The context is also

tokenized and mapped to its embeddings $\mathbf{E}^c = \{\mathbf{e}_k^c\}_{k=1}^{n_c}$, where n_c is the number of context tokens.

To unify the context and external knowledge, the diverse concept embeddings are interposed between the context and the refined knowledge graph embeddings, which guide the LLM to enhance generative diversity while adhering to factual constraints. Similar to Ye et al. (2024), external features are injected into the embedding space to guide the frozen LLM. The input is formulated as:

$$\mathbf{E} = [\underbrace{\mathbf{e}_1^c, \ldots, \mathbf{e}_{n_c}^c}_{\text{context}}, \underbrace{\mathbf{e}_1^d, \ldots, \mathbf{e}_{n_z}^d}_{\text{concepts}}, \underbrace{\mathbf{e}_1^g, \ldots, \mathbf{e}_{n_g}^g}_{\text{KG}}]. \quad (6)$$

This explicit integration ensures that the generation remains both semantically rich and factually grounded. Finally, the QA pairs are generated by:

$$\hat{y} = f_{\text{G}}(\mathbf{E}), \quad (7)$$

where f_{G} denotes the frozen LLM, and $\hat{y} = \{(q_i, a_i)\}_{i=1}^{n}$ is the set of generated QA pairs.

3.5 Training

To align semantic guidance with text generation, Concept Diffusion training is incorporated into the generative framework. This strategy facilitates the diffusion model in capturing domain-specific semantic features from the context, ensuring that the generated concept embeddings are aligned with the LLM's input space for effective utilization.

In the Concept Diffusion training, a conditional noise prediction task is conducted to estimate the noise residual added to the latent state $\mathbf{x}_t$. Specifically, the denoising network ϵ_θ is employed to predict the noise $\hat{\epsilon} = \epsilon_\theta(\mathbf{x}_t, t, \mathbf{W}^C)$ contained within the noisy state $\mathbf{x}_t$. The model is optimized via the mean squared error (MSE):

$$\mathcal{L}_{\text{DIFF}} = \mathbb{E}_{\mathbf{x}_0, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{W}^C)\|^2 \right], \quad (8)$$

where ϵ denotes the ground truth Gaussian noise sampled during the forward diffusion process.

To align semantic guidance with text generation, the Language Modeling (LM) task is employed to maximize the log-likelihood of target tokens conditioned on the composite embeddings $\mathbf{E}$. The LM loss is calculated as:

$$\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{|\mathcal{Y}|} \log P(y_i \mid y_{<i}, \mathbf{E}), \quad (9)$$

where P denotes the next-token distribution modeled by the backbone LLM, and $\mathcal{Y}$ represents the target token sequence of the QA pair.

To balance semantic reconstruction and generation compatibility, uncertainty-based adaptive weighting (Kendall et al., 2018) is employed to dynamically adjust the task weights. The final objective function is formulated as:

$$\mathcal{L} = \frac{1}{2\sigma_1^2}\mathcal{L}_{\text{LM}} + \frac{1}{2\sigma_2^2}\mathcal{L}_{\text{DIFF}} + \sum_{j=1}^{2} \log \sigma_j, \quad (10)$$

where σ_j denotes the learnable observation noise parameter for the j-th task. Additionally, the overall algorithm is detailed in App. E.

4 Experiments

4.1 Experimental Setup

Datasets. Experiments are conducted on two widely-used datasets featuring multiple QA pairs per context: DROP (Dua et al., 2019) and SQuAD (Rajpurkar et al., 2018). Due to the unavailability of official test sets, the original validation sets are used for testing. To address the uneven distribution of QA pairs across contexts, a specialized data-processing workflow is implemented, with detailed procedures provided in App. D.

Metrics. Following previous studies (Zhao and Li, 2025; Xia et al., 2023), the quality of the generated QA pairs is primarily evaluated using BLEU (B) (Papineni et al., 2002), ROUGE-L (R-L) (Lin, 2004), and BERTScore (BSc) (Zhang* et al., 2020). Detailed evaluation is provide in App. C.1.

Baselines. To comprehensively evaluate the performance of DiFRa, two mainstream open-source instruction-tuned models are selected as backbones: Llama3.1$_{8B}$ ("Llama-3.1-8B-Instruct") (Grattafiori et al., 2024) and Mistral$_{7B}$ ("Mistral-7B-Instruct-v0.3") (Jiang et al., 2023a). Given the frozen-backbone setting, the proposed method is compared against various baselines, including CoT (Wei et al., 2022), Self-QA (Zhang and Yang, 2023), Prompt Tuning (Lester et al., 2021), Prefix Tuning (Li and Liang, 2021), and P-Tuning (Liu et al., 2022). Additionally, GPT-4o (OpenAI et al., 2024) and GPT-5 (OpenAI, 2025) are included as closed-source baselines, while DeepSeek-R1 (DeepSeek-AI et al., 2025a) and DeepSeek-V3 (DeepSeek-AI et al., 2025b) are also incorporated for a comprehensive comparison. All enhanced methods are implemented on the same base LLMs for fair comparison. Detailed descriptions are provided in App. G.

Dataset (→)	DROP						SQuAD					
Method (↓)/Metric (→)	**B-1**	**B-2**	**B-3**	**B-4**	**R-L**	**BSc**	**B-1**	**B-2**	**B-3**	**B-4**	**R-L**	**BSc**
GPT-4o†	24.21	14.78	10.11	7.26	27.46	64.35	27.49	17.52	12.38	9.09	32.44	69.03
GPT-5†	20.76	11.76	7.68	5.34	23.82	61.88	18.49	10.52	7.04	4.97	22.68	63.38
DeepSeek-V3$_{671B}$	25.60	15.75	10.80	7.78	28.25	64.34	28.59	18.47	13.12	9.66	33.12	69.30
DeepSeek-R1$_{671B}$	29.57	18.00	12.23	8.72	30.27	64.62	35.26	22.97	16.45	12.23	37.78	70.35
Llama3.1$_{8B}$	31.61	19.60	13.26	9.46	31.23	65.01	35.83	23.53	17.12	12.88	37.22	70.25
+CoT‡	31.63	19.50	13.27	9.52	31.36	65.04	36.24	23.74	17.16	12.81	37.74	70.13
+Self-QA‡	25.46	15.41	10.36	7.33	27.44	64.65	28.64	18.58	13.35	9.98	32.76	68.24
+Prompt Tuning	33.52	21.32	14.79	10.77	33.47	66.60	40.12	26.56	19.39	14.69	39.99	70.74
+Prefix Tuning	31.07	18.71	12.49	8.81	30.70	64.37	38.91	25.96	19.02	14.36	39.17	69.99
+P-Tuning	34.65	22.12	15.44	11.27	34.17	66.91	39.79	26.60	19.48	14.69	39.81	70.94
+DiFRa (ours)	35.62	23.29	16.47	12.18	35.55	67.63	40.59	27.33	20.09	15.26	41.00	71.42
Mistral$_{7B}$	27.86	17.12	11.64	8.36	29.31	64.50	30.70	20.09	14.53	10.90	34.46	69.36
+CoT‡	28.62	17.92	12.37	8.93	30.00	64.83	31.27	20.65	15.13	11.50	34.80	69.48
+Self-QA‡	18.11	11.01	7.42	5.29	22.48	62.42	14.36	9.17	6.48	4.82	19.69	63.76
+Prompt Tuning	34.07	21.85	15.25	11.17	33.91	66.57	41.07	27.66	20.30	15.36	41.18	71.40
+Prefix Tuning	30.83	18.68	12.52	8.80	30.94	64.91	39.73	26.49	19.60	15.06	40.25	71.17
+P-Tuning	34.03	21.86	15.34	11.20	33.94	66.55	41.02	27.55	20.17	15.25	41.02	71.54
+DiFRa (ours)	**36.92**	**24.45**	**17.42**	**12.92**	**36.64**	**68.15**	**41.74**	**28.36**	**20.92**	**15.90**	**41.98**	**71.75**

Table 1: Conventional automatic evaluation on DROP and SQuAD. † indicates a closed-source LLM, and ‡ means a training-free method. Results marked in deeper blue point to higher scores with the same backbone LLM. The **bold** and underline mark the best and second-best results.

4.2 Main Results

The performance comparison on DROP and SQuAD is summarized in Table 1, from which several insights can be drawn: (1) DiFRa significantly enhances open-source LLMs and demonstrates robust generalization capabilities. Extensive experiments on Llama3.1$_{8B}$ and Mistral$_{7B}$ demonstrate that DiFRa achieves consistent gains across nearly all automatic metrics, effectively boosting performance on the DROP and SQuAD datasets. (2) Compared with large-scale LLMs, DiFRa demonstrates competitive capabilities. Despite the massive parameters and general capabilities of the GPT and DeepSeek series, DiFRa exhibits superior performance on automatic evaluation metrics for QAG tasks. (3) It is clear that DiFRa consistently outperforms other enhanced methods. Compared to CoT and Self-QA, DiFRa achieves superior performance across all benchmarks. Additionally, compared with PEFT approaches, DiFRa shows competitive performance across most metrics. App. B.1 provides detailed analyses.

Dataset (→)	DROP		SQuAD	
Method (↓)/Metric (→)	**SE ↑**	**SeFa ↑**	**SE ↑**	**SeFa ↑**
Llama3.1$_{8B}$	2.2746	0.7859	2.2704	0.7855
+CoT‡	2.2037	0.7705	2.2213	0.7770
+Self-QA‡	2.3142	0.7857	2.2208	0.7710
+Prompt Tuning	2.4016	0.7906	2.3246	0.7785
+Prefix Tuning	2.2120	0.7608	1.9932	0.7173
+P-Tuning	2.3773	0.7914	2.3333	0.7798
+DiFRa (ours)	**2.4693**	**0.8105**	**2.3620**	**0.7900**
Mistral$_{7B}$	2.1163	0.7478	2.1887	0.7605
+CoT‡	2.2004	0.7637	2.1230	0.7510
+Self-QA‡	2.2138	0.7574	1.9932	0.7164
+Prompt Tuning	2.4408	0.7885	2.3106	0.7716
+Prefix Tuning	2.0713	0.7130	1.5086	0.6019
+P-Tuning	2.4071	0.7921	2.2836	0.7663
+DiFRa (ours)	2.4555	0.7977	2.3460	0.7773

Table 2: Comparison of semantic diversity (SE) and SeFa on DROP and SQuAD. ‡ indicates a training-free method. Results marked in deeper blue point to higher scores with the same backbone LLM. The **bold** and underline mark the best and second-best results.

4.3 SeFa

Lexical metrics (BLEU, ROUGE-L) and semantic measures (BERTScore) are limited by their reliance on surface matching and insensitivity to logical nuances. These metrics are collectively insufficient to capture the critical semantic diversity and factual consistency of the generated QA pairs. Inspired by (Farquhar et al., 2024; Kuhn et al., 2023), which utilize semantic uncertainty to detect hallucinations, Semantic Entropy (SE) is repurposed to quantify the semantic diversity of generated QA

Model	B-4	R-L	BSc	SE	FC	SeFa
DiFRa	12.18	35.55	67.63	2.4693	4.2214	0.8105
w/o FCC	11.76 (3.45%↓)	34.99 (1.58%↓)	67.37 (0.38%↓)	2.4300 (1.59%↓)	4.1182 (2.45%↓)	0.7975 (1.60%↓)
w/o CCD	9.28 (23.81%↓)	31.19 (12.26%↓)	64.95 (3.96%↓)	2.2750 (7.87%↓)	4.4066 (4.39%↑)	0.7856 (3.07%↓)
w/o DCAM	11.61 (4.68%↓)	34.65 (2.53%↓)	67.13 (0.74%↓)	2.3910 (3.17%↓)	4.1992 (0.53%↓)	0.7956 (1.84%↓)

Table 3: Ablation results on the DROP dataset for Llama3.1$_{8B}$, evaluated via conventional automatic metrics (B-4, R-L, BSc) and metrics for semantic diversity and factual consistency (SE, FC, SeFa).

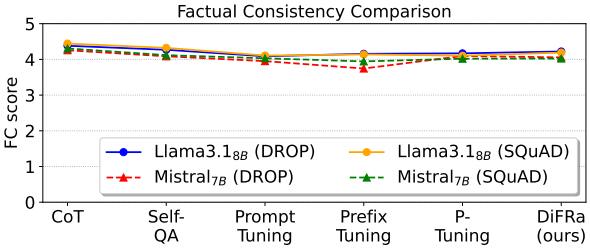

Figure 4: Comparison of factual consistency between DiFRa and baseline methods on DROP and SQuAD datasets across different LLMs.

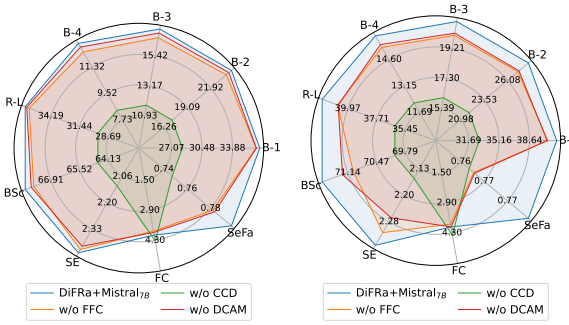

Figure 5: Ablation studies of Mistral$_{7B}$ for DROP on the left and SQuAD on the right.

pairs. Formally, SE is calculated as:

$$SE = -\sum_{c \in \mathcal{C}} \frac{|c|}{N} \log\left(\frac{|c|}{N}\right), \tag{11}$$

where N denotes the total number of QA pairs, which are partitioned into semantic clusters $\mathcal{C}$, and $|c|$ represents the number of pairs within a specific cluster c. Additionally, a Factual Consistency (FC) score is introduced to evaluate the factual consistency of the generated QA pairs, denoted as $FC = \text{LLM}(c, q, a)$, where c, q, a denote the context, generated question, and answer, respectively.

To quantify the trade-off between semantic diversity and factual consistency of generated QA pairs, the SeFa metric is introduced. Formulated analogously to the F1 score, SeFa is defined as:

$$\text{SeFa} = \frac{2 \cdot \overline{SE} \cdot \overline{FC}}{\overline{SE} + \overline{FC}}, \tag{12}$$

where $\overline{SE}$ and $\overline{FC}$ denote the normalized values of SE and FC, respectively. Further details on the metrics can be found in App. C.2.

Experimental results in Table 2 demonstrate that DiFRa consistently outperforms all baseline methods in both SE and SeFa metrics across two advanced open-source backbones. Compared to training-free baselines such as CoT and Self-QA, DiFRa exhibits a more effective exploration of the semantic space. Furthermore, compared with PEFT approaches, DiFRa achieves a superior balance between semantic diversity and factual consistency. As illustrated in Figure 4, the factual consistency scores of all methods remain stable within a specific range. Detailed analyses in App. B.2.

4.4 Ablation Studies

To evaluate the effectiveness of DiFRa, three variants were developed by systematically removing key components: FCC, CCD, and DCAM.

DiFRa w/o FCC. This variant eliminates the KG by removing the factual constraint construction module. The model generates outputs directly based on concept embeddings and the context.

DiFRa w/o CCD. This variant removes the concept construction and diffusion module and omits the MLP used for alignment with the LLM.

DiFRa w/o DCAM. This variant removes the rules related to concepts and the knowledge graph within the DCAM.

Ablation results for Llama3.1$_{8B}$ on the DROP dataset are presented in Table 3, whereas those for Mistral$_{7B}$ on both DROP and SQuAD are shown in Figure 5. Specifically, eliminating FCC leads to a significant decline in SE, FC, and SeFa, confirming the critical role of KG constraints. Then, removing CCD lowers scores across nearly all metrics. The slight gain in consistency likely stems from a shift toward a more conservative generation. Finally, removing DCAM degrades performance on the majority of metrics, validating its effectiveness in integrating concept embeddings with the KG.

5 Conclusion

This work proposes DiFRa, a novel framework designed to address the inherent tension between semantic diversity and factual consistency in QAG. By integrating continuous concept diffusion with discrete knowledge graph constraints, the approach effectively generalizes across the semantic space while reliably maintaining factual integrity. Furthermore, a new metric named SeFa is introduced to provide a unified evaluation of both semantic diversity and factual consistency. Experimental results across both conventional metrics and the proposed SeFa metric demonstrate that DiFRa consistently outperforms baselines, achieving a superior balance between diversity and consistency.

Limitations

This study also has several limitations. Firstly, the experiments were conducted on two public QA datasets. The applicability and performance of DiFRa on domain-specific datasets warrant further exploration. Secondly, the proposed DiFRa incorporates a diffusion model for concept generalization. The iterative denoising process inherently requires multiple sampling steps, which introduces additional inference latency compared to single-pass generation methods. Thirdly, given that the knowledge graph is constructed via an LLM, inherent model hallucinations or extraction errors may propagate to the generated QA pairs, compromising factual consistency.

Ethics Statement

This work aims to trade off semantic diversity against factual consistency in generating QA pairs. This work strictly adheres to the ethical guidelines of the academic community. Publicly available datasets and open-source models were exclusively utilized to ensure reproducibility and data compliance. The authors declare that no conflicts of interest exist regarding this work. The proposed approach aligns with ethical AI practices, prioritizing trust, accountability, and responsible research.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62006083).

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Hanzhu Chen, Xu Shen, Jie Wang, Zehao Wang, Qitan Lv, Junjie He, Rong Wu, Feng Wu, and Jieping Ye. 2025. Knowledge graph finetuning enhances knowledge manipulation in large language models. In *The Thirteenth International Conference on Learning Representations*.

Le Cheng and Shuangyin Li. 2024. DiffusPoll: Conditional text diffusion model for poll generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 925–935, Bangkok, Thailand. Association for Computational Linguistics.

Shaobo Cui, Xintong Bao, Xinxing Zu, Yangyang Guo, Zhongzhou Zhao, Ji Zhang, and Haiqing Chen. 2021. Onestop qamaker: Extract question-answer pairs from text in a one-stop approach. *Preprint*, arXiv:2102.12128.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. 2022. Continuous diffusion for categorical data. *Preprint*, arXiv:2211.15089.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*

(Long and Short Papers), pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Sugyeong Eo, Hyeonseok Moon, Jinsung Kim, Yuna Hur, Jeongwook Kim, SongEun Lee, Changwoo Chun, Sungsoo Park, and Heuiseok Lim. 2023. Towards diverse and effective question-answer pair generation from children storybooks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6100–6115, Toronto, Canada. Association for Computational Linguistics.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Matteo Gabburo, Siddhant Garg, Rik Koncel-Kedziorski, and Alessandro Moschitti. 2023. Learning answer generation using supervision from automatic question answering evaluators. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8389–8403, Toronto, Canada. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023a. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023b. DiffuSeq-v2: Bridging discrete and continuous text spaces for accelerated Seq2Seq diffusion models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9868–9875, Singapore. Association for Computational Linguistics.

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.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *Preprint*, arXiv:2203.05794.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. 2023. DiffusionBERT: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4521–4534, Toronto, Canada. Association for Computational Linguistics.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pages 21696–21707. Curran Associates, Inc.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Shuangyin Li, Jihua Yang, Yixuan Wang, Shimin Di, and Lei Chen. 2025. Escfd: Probabilistic flow diffusion model for accelerated high-quality single-cell rna-seq data synthesis. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 1517–1528, New York, NY, USA. Association for Computing Machinery.

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

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of AAAI 2020*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2023. DPM-solver++: Fast solver for guided sampling of diffusion probabilistic models.

Yueen Ma, DaFeng Chi, Jingjing Li, Kai Song, Yuzheng Zhuang, and Irwin King. 2024. VOLTA: Improving generative diversity by variational mutual information maximizing autoencoder. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 364–378, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2025. GPT-5 System Card.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hai Pham, Isma Hadji, Xinnuo Xu, Ziedune Degutyte, Jay Rainey, Evangelos Kazakos, Afsaneh Fazly, Georgios Tzimiropoulos, and Brais Martinez. 2024. Graph guided question answer generation for procedural question-answering. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2501–2525, St. Julian's, Malta. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Maximilian Schmidt, Andrea Bartezzaghi, and Ngoc Thang Vu. 2024. Prompting-based synthetic data generation for few-shot question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13168–13178, Torino, Italia. ELRA and ICCL.

Haz Sameen Shahgir, Chansong Lim, Jia Chen, Evangelos E. Papalexakis, and Yue Dong. 2025. Expert-GenQA: Open-ended QA generation in specialized domains. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2934–2955, Suzhou, China. Association for Computational Linguistics.

Sheng Shen, Yaliang Li, Nan Du, Xian Wu, Yusheng Xie, Shen Ge, Tao Yang, Kai Wang, Xingzheng Liang, and Wei Fan. 2020. On the Generation of Medical Question-Answer Pairs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8822–8829.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021a. Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole.

2021b. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. 2025. FiDeLiS: Faithful reasoning in large language models for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8315–8330, Vienna, Austria. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *Preprint*, arXiv:1904.09223.

Yuto Tomikawa, Ayaka Suzuki, and Masaki Uto. 2024. Adaptive question–answer generation with difficulty control using item response theory and pretrained transformer models. *IEEE Transactions on Learning Technologies*, 17:2186–2198.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023. An empirical comparison of LM-based question and answer generation methods. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14262–14272, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Zehua Xia, Qi Gou, Bowen Yu, Haiyang Yu, Fei Huang, Yongbin Li, and Cam-Tu Nguyen. 2023. Improving question generation with multi-level content planning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 800–814, Singapore. Association for Computational Linguistics.

Qingfa Xiao, Shuangyin Li, and Lei Chen. 2023. Topic-DPR: Topic-based prompts for dense passage retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7216–7225, Singapore. Association for Computational Linguistics.

Sen Yang, Qingyu Zhou, Dawei Feng, Yang Liu, Chao Li, Yunbo Cao, and Dongsheng Li. 2021. Diversity and consistency: Exploring visual question-answer pair generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1053–1066, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Mo Yu, and Ying Xu. 2022. It is AI's turn to ask humans a question: Question-answer pair generation for children's story books. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–744, Dublin, Ireland. Association for Computational Linguistics.

Fuda Ye, Shuangyin Li, Yongqi Zhang, and Lei Chen. 2024. R^2AG: Incorporating retrieval information into retrieval augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11584–11596, Miami, Florida, USA. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2024. Text diffusion model with encoder-decoder transformers for sequence-to-sequence 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)*, pages 22–39, Mexico City, Mexico. Association for Computational Linguistics.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2022. DKPLM: Decomposable Knowledge-Enhanced Pre-trained Language Model for Natural Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11703–11711.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xuanyu Zhang and Qing Yang. 2023. Self-qa: Unsupervised knowledge guided language model alignment. *Preprint*, arXiv:2305.11952.

Wenzhuo Zhao and Shuangyin Li. 2025. RUBY: An effective framework for multi-constraint multi-hop question generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18164–18188, Vienna, Austria. Association for Computational Linguistics.

A Implementation Details

For concept construction, topic modeling is performed on the entire corpus using BERTopic (Grootendorst, 2022), and keywords are extracted from the context using KeyBERT (Grootendorst, 2020). The concept set is formed by the union of the top-3 topics and top-5 keywords derived from each context. The gpt-4o-2024-11-20 model[2] is employed via the OpenAI API for knowledge graph construction and refinement.

For concept diffusion, an architecture adapted from DiffuSeq (Gong et al., 2023a,b) is employed. It is initialized with pre-trained BERT (Devlin et al., 2019) weights. For DiFRa, a two-layer MLP is employed to align the latent spaces of the diffusion model and the LLM. To minimize trainable parameters, LoRA (Hu et al., 2021) is utilized for feature extraction. Specifically, the low-rank adapters are applied to the query and value projections, utilizing a configuration where the rank r is set to 16, the alpha parameter is set to 32, and the dropout rate is fixed at 0.1. Crucially, the time-conditioning parameters are explicitly unfrozen to achieve better alignment with the specific noise distribution.

All models are trained on a single NVIDIA A800-40G GPU with seed 42. In the implementation, a hybrid precision strategy is employed, where bf16 is used for the MLP alignment module to improve efficiency, and fp32 is retained for the diffusion backbone to preserve numerical stability. Training is conducted using the AdamW (Loshchilov and Hutter, 2019) optimizer for three epochs. A cosine noise schedule with $T = 1,000$ steps is utilized for the diffusion process. A cosine learning rate scheduler is applied with a warmup ratio of 0.1, and gradients are clipped at a maximum norm of 1.0. The learning rate is initialized at 1×10^{-4} for both the diffusion model and the MLP, while the adaptive weights σ_1 and σ_2 are updated with a higher learning rate of 1×10^{-3}. For inference, diffusion sampling uses DPM-Solver++ (Lu et al., 2023), which is configured as a third-order solver with a total of 25 sampling steps. Text generation across all experiments and baselines follows a unified configuration: max_new_tokens=1,024, temperature=0.7, top_p=0.95, and a repetition penalty of 1.10. Finally, ten QA pairs are generated for each input to evaluate diversity.

[2] https://platform.openai.com/docs/models/gpt-4o

B Experimental Analysis

B.1 Main Experimental Results

By consolidating the results from Table 1 in the main text, reveals several insights:

(1) DiFRa brings substantial improvements for open-source LLMs. On two multi-QA datasets (DROP and SQuAD), DiFRa yields significant gains across almost all automatic metrics (BLEU, ROUGE-L, and BERTScore). Although Mistral$_{7B}$ is intrinsically weaker than Llama3.1$_{8B}$, pairing it with DiFRa allows Mistral$_{7B}$ to surpass Llama3.1$_{8B}$. These results indicate that DiFRa helps LLMs more precisely exploit the underlying latent distribution of domain data, yielding more high-quality QA pairs.

(2) Compared with large-scale LLMs, DiFRa demonstrates competitive capabilities. Despite the massive parameters and general capabilities of the GPT and DeepSeek series, DiFRa exhibits superior performance on automatic evaluation metrics for QAG tasks. Notably, irrespective of the backbone employed, DiFRa consistently yields substantial improvements. This indicates that the integration of concept diffusion and knowledge graphs, together with the diversity and consistency aware mechanism, enables DiFRa to generate high-quality text that captures core information and contextual details. This suggests that while large-scale LLMs excel in general scenarios, they often struggle to adapt effectively to domain-specific tasks.

(3) It is clear that DiFRa consistently outperforms other enhanced methods. Compared with training-free approaches, DiFRa achieves significant performance gains. While CoT yields marginal improvements for Llama3.1$_{8B}$ and Mistral$_{7B}$. Similarly, the Self-QA method exhibits consistent performance regression across all backbones. This is likely because training-free methods rely solely on the intrinsic probability distribution of frozen LLMs, which results in a misalignment between the general pre-training distribution and the specific requirements of the target task. Additionally, compared with PEFT approaches, DiFRa still shows competitive performance across most metrics.

B.2 Analysis of Semantic Diversity and Factual Consistency

Table 2 in the main text present the results for the two backbones across different methods, the following findings are derived.

(1) Comparison across the two backbones (Llama3.1$_{8B}$ and Mistral$_{7B}$) shows significant improvements in SE, alongside substantial improvements in SeFa, the metric assessing the trade-off between semantic diversity and factual consistency. This demonstrates that by merging continuous concept diffusion with discrete knowledge graph constraints, DiFRa effectively enhances semantic diversity while optimizing the balance between semantic diversity and factual consistency.

(2) Regarding training-free methods (CoT and Self-QA), the results exhibit significant inconsistency across different backbones. It leads to performance regression in Llama3.1$_{8B}$ regarding both SE and SeFa metrics. Similarly, although Self-QA effectively boosts SE in most cases, it frequently fails to have a high SeFa score, which indicates that the generated diversity often comes at the expense of factual consistency. This suggests that relying solely on prompting strategies without parameter updates is insufficient for reliably navigating the trade-off between diversity and consistency in domain-specific tasks.

(3) Compared with representative PEFT baselines (Prompt Tuning, Prefix Tuning, and P-Tuning), DiFRa demonstrates superior capability in harmonizing the trade-off between diversity and consistency. DiFRa consistently achieves the highest SeFa scores across all two backbones and datasets. This indicates that DiFRa provides a more robust solution for maintaining factual fidelity while expanding semantic breadth than methods relying solely on soft prompt optimization.

(4) Figure 4 in the main text illustrate the factual consistency performance across different methods and datasets for the three backbones. The line charts exhibit a near-linear trend, indicating that the factual consistency of these methods remains within an acceptable range.

B.3 Analysis of Dynamic Weighting

Based on the formulation in Eq. (10), Figure 6 illustrates the evolution of uncertainty and loss weights. The curves reveal an inverse relationship where weights adaptively increase as the model gains certainty in its predictions. This mechanism enables the dynamic prioritization of more reliable training signals across different stages, ensuring a stable and efficient multi-task optimization process.

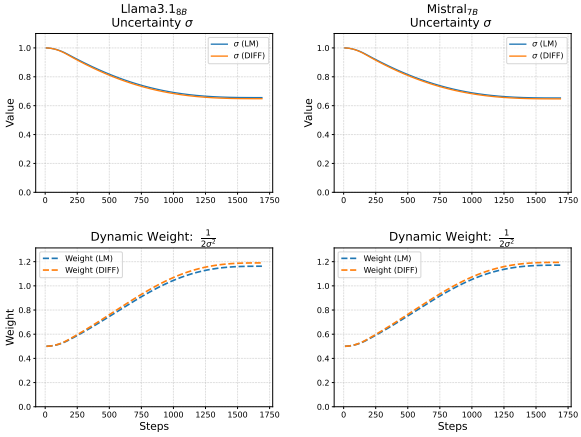

Figure 6: Visualization of the learnable uncertainty parameter σ in the top row and the corresponding dynamic weight $\frac{1}{2\sigma^2}$ in the bottom row during training. The curves compare the LM loss and Diffusion loss components across Llama3.1$_{8B}$, and Mistral$_{7B}$.

C Evaluation Details

C.1 Conventional Metrics

Following previous studies, BLEU-{1, 2, 3, 4}, ROUGE-L, and BERTScore are employed as conventional metrics to evaluate the generated QA pairs. Specifically, for each generated QA pair, the question and answer are concatenated into a single sequence. Subsequently, a comparison is performed between this sequence and every pair in the reference set to determine the maximum similarity score. The final score for the generated QA set is calculated as:

$$\mathrm{S} = \frac{1}{|\mathcal{G}|} \sum_{(q,a)\in\mathcal{G}} \max_{(q',a')\in\mathcal{R}} \mathcal{M}(q \oplus a, q' \oplus a'), \quad (13)$$

where $\mathcal{G}$ and $\mathcal{R}$ denote the sets of generated and reference QA pairs, respectively, and $|\mathcal{G}|$ represents the total number of generated pairs. The symbol $\oplus$ indicates string concatenation, and $\mathcal{M}(\cdot)$ refers to the specific metric function computing the similarity between sequences.

C.2 Detailed Evaluation for Semantic Diversity and Factual Consistency

Conventional metrics such as BLEU, and ROUGE-L rely heavily on literal lexical overlap, primarily emphasizing surface-form matching. While embedding-based metrics like BERTScore offer improved semantic similarity measurement, they often remain insensitive to fine-grained logical discrepancies or subtle factual contradictions. Consequently, these metrics are collectively inadequate

to fully capture the critical semantic diversity and factual consistency required for the evaluation of generated QA pairs.

Semantic Diversity. Inspired by Farquhar et al. (2024) and Kuhn et al. (2023), who utilize semantic uncertainty to detect hallucinations, Semantic Entropy (SE) is repurposed to quantify the semantic diversity of generated QA pairs. While high entropy typically signifies uncertainty in standard QA tasks, within the proposed generation framework, it serves as a metric for information coverage. Specifically, Llama-3.1-70B-Instruct[3] is deployed via the vLLM engine (Kwon et al., 2023) to perform semantic clustering on the generated QA pairs by identifying and grouping semantically equivalent instances. The entropy is then calculated over the distribution of these clusters to measure diversity, as formally defined in Eq. (11).

Factual Consistency. To evaluate the quality of the generated QA pairs, an LLM-based Factual Consistency (FC) score is employed. Specifically, Llama-3.1-70B-Instruct is also utilized to perform this assessment. In this specific setting, factual consistency is rigorously defined as a dual constraint: the generation must be factually verifiable against the source context c and semantically aligned with the gold-standard reference set $\mathcal{R}$. A 5-point scale is adopted to assess this composite consistency. Score 5 (Fully Consistent and Aligned) denotes pairs that are factually flawless regarding the source context c and explicitly semantically equivalent to a reference pair in $\mathcal{R}$. Score 4 (Verifiable but Unaligned) is assigned to pairs factually verifiable from c but absent in $\mathcal{R}$, representing truthful content that nonetheless falls outside the target factual scope. Score 3 (Partially Correct) accounts for pairs containing minor factual imprecisions or those that omit critical constraints found in the context. Score 2 (Unverifiable) corresponds to instances where the information is entirely absent from the context c, whereas Score 1 (Contradiction) is assigned when the content explicitly contradicts the source.

Evaluation of the Proposed SeFa. To ensure a balanced evaluation between semantic diversity and factual consistency, the proposed SeFa metric aggregates semantic entropy and the factual consistency score. However, since these two metrics operate on disparate dimensions and scales, direct combination is infeasible. Therefore, both metrics are normalized to the unit interval $[0, 1]$ prior to calculating their harmonic mean. Analogous to the F_1 score, this aggregation method effectively penalizes extreme disparities between the two components, ensuring that a high SeFa score reflects a robust balance across both dimensions.

Regarding semantic entropy, the theoretical maximum value is determined by the sample size N. Maximum diversity occurs when every generated QA pair forms a unique semantic cluster, yielding an entropy of $\log(N)$. To derive the normalized metric $\overline{SE}$, the raw entropy value calculated in Eq. (11) is divided by this theoretical maximum:

$$\overline{SE} = \frac{SE}{\log(N)}. \tag{14}$$

This results in a ratio representing the degree of diversity achieved relative to the maximum potential diversity for the given sample size.

For the factual consistency score, Min-Max scaling is employed to map the discrete scores to the unit interval $[0, 1]$. Let $FC_{\min}$ and $FC_{\max}$ denote the minimum and maximum scores of the metric (i.e., $FC_{\min} = 1$ and $FC_{\max} = 5$). The normalized score $\overline{FC}$ is calculated as:

$$\overline{FC} = \frac{FC - FC_{\min}}{FC_{\max} - FC_{\min}}. \tag{15}$$

This transformation maps the score range to $[0, 1]$.

Finally, SeFa is defined as the harmonic mean of these normalized metrics, formulated analogously to the F_1 score, as shown in Eq. (12). By adopting the harmonic mean to strictly penalize imbalance, the metric ensures that a high SeFa score necessitates simultaneous excellence in both diversity and factual correctness.

D Dataset Details

Dataset	Split	# Contexts	# QA Pairs	Avg. QA/Ctx
DROP	Training	4,517	52,452	11.6
	Testing	271	2,662	9.8
SQuAD	Training	2,000	14,679	7.3
	Testing	200	1,425	7.1

Table 4: Statistics of the processed DROP and SQuAD datasets used for training and testing.

DROP (Dua et al., 2019) is a reading comprehension benchmark requiring discrete reasoning over paragraphs. To prepare the data, all QA pairs associated with the same context were first aggregated.

[3] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

Then, contexts containing more than 15 QA pairs were filtered out to ensure data quality. Furthermore, the gpt-4o-2024-11-20 model was utilized to correct factual errors across the entire dataset, while contexts containing fewer than 10 QA pairs were specifically augmented. For the test set, instances were selected from the validation split that contained between 5 and 15 QA pairs. Following these processing steps, the final dataset comprises 4,517 training examples and 271 testing examples.

SQuAD (Rajpurkar et al., 2018) is a large-scale question-answering dataset consisting of questions posed on a set of Wikipedia articles. To prepare this dataset, unanswerable QA pairs were first filtered out, and all remaining QA pairs associated with the same context were aggregated. Given that the data distribution is heavily concentrated on contexts with five QA pairs, a random seed of 42 was used to sample 2,000 training contexts containing exactly five QA pairs. Similarly, 200 contexts were selected from the validation set for testing purposes. Table 4 summarizes the statistics of the two datasets used for training and evaluation.

E DiFRa Algorithm

Algorithms 1 and 2 present the pseudocode for the inference and training phases of DiFRa, respectively.

Algorithm 1: Inference

Input : Context C_i, Graph $\mathcal{G}_i^*$
Output : Generated QA pairs $\hat{\mathcal{Y}}_i$.

$S_{\mathcal{G}} \leftarrow \text{Linearize}(\mathcal{G}_i^*)$;
$\mathbf{W}_i^C \leftarrow \text{Enc}_{\text{BERT}}(C_i)$;
Initialize latent noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
for $t = T$ *down to* 1 **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(\mathbf{x}_t, t, \mathbf{W}_i^C)$;
  Update latent state $\mathbf{x}_{t-1}$ via DPM-Solver++:
  $\mathbf{x}_{t-1} \leftarrow \text{DPM-Solver++}(\mathbf{x}_t, \hat{\epsilon}, t)$;
Get concepts embedding $\hat{\mathbf{Z}}$ from $\tilde{\mathbf{x}}_0$;
$\mathbf{E}_i^d \leftarrow f_{\rightarrow d_h}(\hat{\mathbf{Z}}_i)$;
$\mathbf{E}^c \leftarrow f_{\text{emb}}(C_i), \mathbf{E}^g \leftarrow f_{\text{emb}}(S_{\mathcal{G}})$;
Final input: $\mathbf{E}_{\text{i}} \leftarrow [\mathbf{E}^c; \mathbf{E}^d; \mathbf{E}^g]$;
Generate QA pairs: $\hat{\mathcal{Y}}_i \leftarrow f_G(\mathbf{E}_i)$;
return $\hat{\mathcal{Y}}_i$;

Algorithm 2: Training

Input : Dataset $\mathcal{D} = \{(C_i, \mathcal{Z}_i, \mathcal{G}_i^*, \mathcal{Y}_i)\}_{i=1}^N$.
Output : Trained parameters θ, projection $f_{\rightarrow d_h}$, and weights σ_1, σ_2.

Initialize model parameters $\theta, f_{\rightarrow d_h}, \sigma_1, \sigma_2$.
for $i \leftarrow 1$ *to* N **do**
  $S_{\mathcal{G}} \leftarrow \texttt{Linearize}(\mathcal{G}_i^*)$;
  $\mathbf{W}_i^C \leftarrow \texttt{Enc}_{\texttt{Bert}}(C_i)$, $\mathbf{W}_i^Z \leftarrow \texttt{Enc}_{\texttt{Bert}}(\mathcal{Z}_i)$;
  Initialize latent: $\mathbf{x}_0 \leftarrow \texttt{Emb}([\mathbf{W}_i^C; \mathbf{W}_i^Z]) + \alpha\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and α is the scaling factor;
  Sample timestep $t \sim \mathcal{U}(1, T)$ and noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  Get $\mathbf{x}_t$ via reparameterization trick: $\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$;
  $\hat{\epsilon} \leftarrow \epsilon_\theta(\mathbf{x}_t, t, \mathbf{W}_i^C)$;
  $\mathcal{L}_{\text{DIFF}} \leftarrow \|\epsilon - \hat{\epsilon}\|^2$; // Eq. (8)
  $\tilde{\mathbf{x}}_0 \leftarrow \texttt{Denoise}(\mathbf{x}_t, \epsilon_\theta, t)$;
  Get concepts embedding $\hat{\mathbf{Z}}$ from $\tilde{\mathbf{x}}_0$;
  $\mathbf{E}^d \leftarrow f_{\rightarrow d_h}(\hat{\mathbf{Z}})$; // Eq. (4)
  $\mathbf{E}^c \leftarrow f_{\text{emb}}(C_i), \mathbf{E}^g \leftarrow f_{\text{emb}}(S_{\mathcal{G}})$;
  Final input: $\mathbf{E}_{\text{i}} \leftarrow [\mathbf{E}^c; \mathbf{E}^d; \mathbf{E}^g]$;
  $\mathcal{L}_{\text{LM}} \leftarrow -\sum_{j=1}^{|\mathcal{Y}_i|} \log P(y_j \mid y_{<j}, \mathbf{E}_{\text{i}})$; // Eq. (9)
  $\mathcal{L} \leftarrow \frac{1}{2\sigma_1^2}\mathcal{L}_{\text{LM}} + \frac{1}{2\sigma_2^2}\mathcal{L}_{\text{DIFF}} + \sum_{j=1}^2 \log \sigma_j$; // Eq. (10)
  Update $\theta, f_{\rightarrow d_h}, \sigma_1, \sigma_2$ via gradient descent on $\nabla\mathcal{L}$;
end
return $\theta, f_{\rightarrow d_h}, \sigma_1, \sigma_2$.

F Further Discussion

In this section, we further explore the following research questions. **RQ1**: Why were certain baseline methods, including RAG with KG and the specific works listed in Introduction, excluded from the comparison? **RQ2**: How does the framework mitigate data contamination risks regarding backbone models potentially memorizing pre-training datasets? **RQ3**: What are the exact inference latencies, training times, and computational costs of DiFRa compared to the baselines? **RQ4**: Does the heavy reliance on external LLMs for KG construction and evaluation compromise the framework's reproducibility and fairness?

F.1 RQ1: Baseline Method Selection

The exclusion of RAG and several previously published works stems from fundamental differences in task formulation and architectural paradigms. RAG is fundamentally designed for QA, whereas the proposed task is QAG, which requires generating a set of QA pairs from a context. Introducing RAG would conflate these two distinct problem domains. Regarding the specific literature mentioned in Introducion, they were omitted because they are incomparable to the current generative, LLM-based benchmark. Specifically, Eo et al. (2023) and Yao et al. (2022) employ traditional multi-stage pipelines; Shahgir et al. (2025) relies on an extraction-based pipeline where generation quantity is bounded by extracted topics; and Gabburo et al. (2023) fine-tunes relatively small pre-trained models rather than modern LLMs. The prompt-based strategies (Schmidt et al., 2024) are already represented in the current experiments through the comparison with Self-QA (Zhang and Yang, 2023) and foundational LLM backbones.

F.2 RQ2: Analysis of Data Contamination Risks

The potential inflation of factual consistency scores due to dataset memorization (e.g., SQuAD/DROP) during pre-training does not compromise the validity of the experimental comparisons. All baselines evaluated in this study share the identical frozen LLM backbone, ensuring that any potential factual consistency inflation applies uniformly across all compared methods. Furthermore, the generative nature of the QAG task fundamentally differs from standard Question Answering (QA). While QA is highly susceptible to memorization, QAG requires the model to actively generate novel QA pairs conditioned dynamically on a given context, significantly mitigating the impact of static data memorization.

F.3 RQ3: Analysis of Inference Latency

The primary objective of the proposed framework is to balance semantic diversity and factual consistency rather than strict inference acceleration, though the computational overhead remains highly bounded and practical. The construction of the KG via an LLM is a strictly offline, one-time process that introduces zero overhead during actual QA generation. During the generation phase, the application of DPM-Solver++ for diffusion sampling restricts the iterative denoising process to exactly 25 lightweight steps. Therefore, inference latency is limited to these 25 steps plus a single prompting pass. Furthermore, the training cost is minimized because the backbone LLM remains entirely frozen; only the diffusion denoising network and an MLP are optimized. This paradigm drastically lowers GPU memory demands and practical training times compared to full-parameter fine-tuning approaches.

F.4 RQ4: Impact on Reproducibility and Fairness

The reliance on external LLMs for specific framework components does not reduce the reproducibility or fairness of the proposed method. As established, the LLM-driven KG construction is a one-time offline step, serving as a reusable methodological tool rather than an unpredictable runtime variable. For evaluation purposes, the LLM-as-a-judge mechanism replaces subjective human evaluation with a standardized automated metric. To guarantee fairness and reproducibility, all baselines are evaluated against the exact same standards, with the evaluator LLM's temperature strictly set to 0 to eliminate stochasticity and ensure deterministic scoring.

G Baseline Details

G.1 LLMs

GPT-4o and GPT-5. GPT-4o (OpenAI et al., 2024) and GPT-5 (OpenAI, 2025) are prominent closed-source LLMs accessed via the OpenAI API[4]. For GPT-4o, the gpt-4o-2024-11-20 version is utilized, and the generation hyperparameters are kept consistent with those employed in the main experiments. The gpt-5-2025-08-07 version of GPT-5 is used, following the recommended default decoding strategy due to the current absence of support for temperature customization in this specific model.

DeepSeek-V3 and DeepSeek-R1. DeepSeek-V3 (DeepSeek-AI et al., 2025b) and DeepSeek-R1 (DeepSeek-AI et al., 2025a) are 671B open-source LLMs that demonstrate impressive performance. These are accessed via the DeepSeek API platform[5].

[4] https://platform.openai.com/docs
[5] https://platform.deepseek.com

LLaMA3.1. Llama3.1 (Grattafiori et al., 2024) is a popular family of open-source LLMs. The experiments are conducted using Llama3.1$_{8B}$ ("Llama-3.1-8B-Instruct").

Mistral. Mistral (Jiang et al., 2023a) is another efficient and high-performance open-source LLM. The experiments utilize the Mistral$_{7B}$ ("Mistral-7B-Instruct-v0.3") version of the Mistral model family.

G.2 Enhancement Methods

CoT. Chain-of-Thought (CoT) (Wei et al., 2022) is a prompting strategy that elicits intermediate reasoning steps from LLMs. A standard zero-shot CoT setting is adopted by appending the instruction "Let's think step by step" to each prompt.

Self-QA. Self-QA (Zhang and Yang, 2023) is an enhancement strategy that leverages the model's ability to generate its own question-answer pairs or self-critique its reasoning process to improve answer quality. This study adheres to the original implementation, employing the exact prompt templates from the source work for QA pair generation.

Prompt Tuning. Prompt Tuning (Lester et al., 2021) optimizes a set of continuous "soft prompt" tokens while keeping the backbone LLM parameters frozen.

Prefix Tuning. Prefix Tuning (Li and Liang, 2021) prepends a sequence of learnable continuous vectors to each layer of the LLM while keeping the original model parameters frozen.

P-Tuning. P-Tuning (Liu et al., 2022) employs a trainable prompt encoder to generate continuous prompt embeddings, focusing on optimizing the prompt tokens' logical structure.

For all the aforementioned PEFT-based methods, although a maximum token budget is allocated, the actual value of k remains dynamic and strictly adheres to the same setting as the proposed method to ensure a fair comparison.

H Prompt Templates

This section presents the prompt templates employed in the DiFRa framework. Table 5 provides a unified overview of all prompt templates used across the pipeline, covering knowledge graph construction and refinement, controllable question-answer generation, and automated evaluation for semantic diversity and factual consistency.

Task	Prompt Template
KG Construction	You are a high-precision information extraction engine. Extract fact triples from the text and return strict json only. Return exactly one fenced code block starting with "'json and ending with "'; no extra text. Do not invent facts. Use valid json (no trailing commas, no nan/infinity). Extract fact triples from the text. Schema (json): "triples": [{"h": string, "r": string, "t": string}] } Rules: 1) No invention. Each triple must be directly supported by the text. 2) "h" and "t" must use the surface forms from the text (do not create ids, do not append hashes). 3) "r" is a short phrase copied from the text; avoid paraphrasing. 4) Do not output type/label/meta triples (e.g., "is a", "type", "rdfs:label", "note", "alias"). 5) Trim spaces, collapse internal whitespace. Deduplicate (case-insensitive) and sort by (h, r, t). 6) If nothing is extractable, return {"triples": []}. Text: «<text»>
KG Refinement	You are a knowledge-graph engineer. Your task: Convert fine-grained relational triples into a concise set of event-level fact triples that can be used for fact verification. Output requirements: - Return only a json array – no prose, no markdown. - Each element must contain exactly the keys "h", "r", "t". - Produce no more than k triples total. Aggregation rules: A. Merge all fragments that describe the same event (same date, place, participants). B. Compress multiple predicates in one action chain into a single clear predicate. C. Combine related numeric facts (e.g., all casualties) into one triple per fact dimension. D. Preserve exact numbers, main actors, dates, and locations. E. Remove vague, hypothetical, or redundant predicates (e.g., "nearly", "moved to"). F. No duplicate facts; each triple must represent a unique, verifiable statement. Inputs: {context} and {triples} Output: Return the json array only.
DiFRa Generation	Generate {num_qa_pairs} high-quality question-answer pairs and obey the following rules: (1) Concepts (optional): If concepts are provided, integrate them with the context to encourage wider semantic exploration in the generated question-answer pairs. (2) Knowledge graph (optional): If a knowledge graph is provided, use it only to check for conflicts. (3) Output format: Return a json array with {num_qa_pairs} objects; each object has two keys: "question" and "answer". Here is an example of the exact format required (using 2 pairs): {output_example} Output only the json array and do not output any other words. Inputs: {context} {given_concepts} {knowledge_graph} Output: {qa_pairs}
Semantic Diversity Evaluation	You are a careful semantic clustering assistant for qa pairs. Task: Partition the given qa pairs into clusters such that items in the same cluster make the same semantic claim. Strict output contract: - Output only a valid json array of integer lists (list of clusters), e.g. [[0,2,5],[1,4],[3]]. - Do not wrap the whole output in quotes. No markdown fences. No explanations or extra text. - Indices must be integers referring only to the indices shown in the user message (starting at 0). Inputs: Shared context: {context}. QA pairs (each line starts with its index): {qa_block} Output: Return only the clusters in json (list of lists of indices).
Factual Consistency Evaluation	You are a factual consistency evaluator. Your goal is to determine the factual fidelity of candidate qa pairs and their relevance to the provided reference qa pairs, using the context and general knowledge as supporting evidence. Scoring rubric (1-5): - Score 5: Correct and relevant. Factuality: The candidate qa pair as a single unit is factually flawless and explicitly supported by the context. Alignment: The pair mirrors a "twin" in the reference set. The question's intent and the answer's core information must be semantically equivalent to a specific reference qa pair. The "noise" penalty: If the candidate is factually true but includes extra "fluff" or describes a fact from the context that the reference ignored, it must be downgraded to score 4. - Score 4: Correct but irrelevant. Factuality: The candidate qa pair is empirically verifiable. Divergence: If the candidate's question has no semantically equivalent match in the reference qa list, it is considered irrelevant, regardless of its factual accuracy. It is a "correct answer to the wrong question." - Score 3: Partially correct. Incompleteness: The pair captures the core intent but omits critical constraints or nuances found in the context. Minor imprecision: The answer is "directionally" correct but contains slight errors in details (dates, figures) while grounded in context. - Score 2: Unverifiable. Evidence void: The information is absent from the context. - Score 1: Direct contradiction. Conflict: The candidate qa pair contradicts the context. Constraints: - Output format: Must be a raw json list of objects. - Structure: [{"index": <int>, "score": <int>}] - No markdown, no explanations, no deviations. - No postscript: The response must start with "[" and end with "]". Strictly no text after the json array. Inputs: Context: $context Reference qa: $ref_block Candidate qa: $qa_block

Table 5: Unified prompt templates for KG construction, refinement, QA generation, and evaluation.