Enhancing Question Generation through Diversity-Seeking Reinforcement Learning with Bilevel Policy Decomposition

Tianyu Ren, Hui Wang*, Karen Rafferty

School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, United Kingdom
{tren01, h.wang, k.rafferty}@qub.ac.uk

Abstract

Recent advancements in question generation (QG) have been significantly propelled by reinforcement learning (RL). Although extensive reward models have been designed to capture the attributes of ideal questions, their associated learning challenges, particularly in sample efficiency and diversity, remain underexplored. This paper introduces a bilevel policy decomposition (BPD) framework and a diversity-seeking RL (DSRL) objective to address these issues. The BPD framework utilizes two cascading policies to divide QG into two more manageable sub-tasks: answer-centric summary generation and summary-augmented QG, facilitating exploration and accelerating policy learning. Concurrently, the DSRL objective preserves the inherent diversity of QG by ensuring the bilevel policies align probabilistically with their reward models rather than merely maximizing returns. Our integrated approach, named BPD-DSRL, demonstrates superior performance over existing baselines on multiple question quality and diversity metrics across various QG benchmarks.

Code and other supplementary material —
https://github.com/Tianyu-Ren/BPD-DSRL

Introduction

Question Generation (QG) aims to generate questions from a given reading passage and answer pair. As a core task within question answering (QA), QG offers a wide range of practical benefits, such as data augmentation for QA (Shakeri et al. 2020; Yu et al. 2024), dialogue system development (Ling et al. 2020) and assessment generation for educational purposes (Zhao et al. 2022; Yoon and Bak 2023).

QG has advanced significantly with the advent of pre-trained language models (PLMs) (Radford et al. 2019) and robust QG datasets (Rajpurkar et al. 2016). Although these PLMs can be effectively adapted for QG through supervised fine-tuning (SFT), they are still prone to generating unfaithful or hallucinated content (Gou et al. 2023; Xia et al. 2023) (see Figure 1 (a), left part). This issue primarily arises from their use of maximum likelihood estimation, which penalizes deviations from the ground truth but does not always promote alignment or faithfulness (Christiano et al. 2017; Xie et al. 2020; Ouyang et al. 2022; Tian et al. 2024). To address this, some studies (Chen, Wu, and Zaki 2023; Hong and Liu 2024) have employed reinforcement learning (RL) (Sutton and Barto 2018) to optimize QG through reward models that explicitly capture desirable question attributes. We refer to this line of research as **RLQG**.

Ideal questions should possess a range of desirable attributes, such as relevance to context, fluency, and answerability. Previous RLQG work has focused on crafting novel QG-specific reward models to ensure these qualities (Xie et al. 2020; Gaur et al. 2022; Chen, Wu, and Zaki 2023). While various robust reward models have been discovered, effectively aligning the QG policy with them remains an

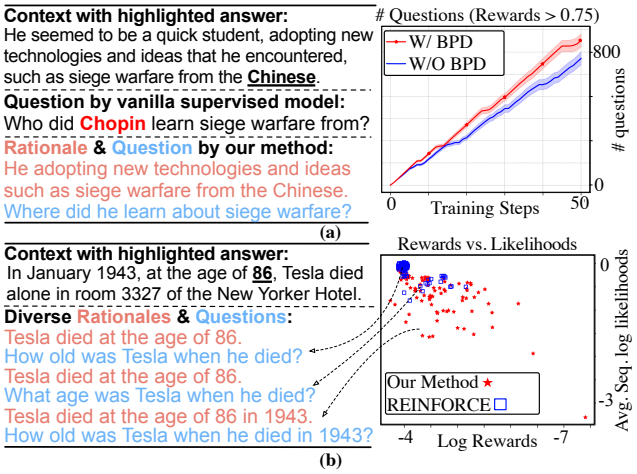

Figure 1: Our motivations. **(a) Left**: The inconsistency problem prevalent in supervised QG models. **Right**: The sample efficiency issue in RLQG. We compare our method with its ablated version (without decomposition) on SQuAD by sampling 16 questions after each training step. The graph shows the cumulative number of questions with rewards exceeding 0.75, indicating improved sample efficiency. **(b)** The diversity issue in RLQG. Compared to the REINFORCE algorithm commonly used in RLQG, polices trained with our RL objective produce more diverse yet acceptable questions, as illustrated by a scatter plot of 128 samples comparing averaged sequence log-likelihoods to the corresponding rewards.

*Corresponding author

underexplored challenge. Existing RLQG methods typically frame QG as a **one-step** generation process (Hong and Liu 2024). However, like many other intricate tasks such as mathematical problem solving (Wei et al. 2022), the source-target mapping in QG is highly implicit and requires deductive reasoning steps (Zhao et al. 2022; Xia et al. 2023). Directly modeling this complex mapping within one single policy presents substantial exploration challenges in pursuit of high-return outcomes, resulting in low sample efficiency and finally a sub-optimal QG policy (Yan et al. 2024; Zhou et al. 2024), as shown in the right part of Figure 1 (a).

Additionally, an inherent property of QG, namely diversity, is frequently neglected in existing RLQG research. Most prior efforts aim for a QG policy that can generate the optimal question with maximum expected returns (Chen, Wu, and Zaki 2023; Hong and Liu 2024). However, QG inherently has a **one-to-many** nature (Shen et al. 2019), where many questions with high returns are all acceptable for one passage-answer pair (Figure 1 (b)). Focusing solely on the single best question while discarding the full range of acceptable alternatives conflicts with this inherent property, compromising the effectiveness of the resulting QG policies in downstream applications such as data augmentation for QA (Sultan et al. 2020; Yu et al. 2024).

In light of the above challenges, this study aims to advance RLQG by enhancing sample efficiency and promoting diversity. To improve sample efficiency during RLQG, we first formulate RLQG as a bilevel optimization problem (Colson, Marcotte, and Savard 2007) and introduce a bilevel policy decomposition framework that hierarchically segments the overall QG policy into a high-level *rationale* policy and a low-level *action* policy. The rationale policy directs the agent to generate answer-centric summaries from the input, maintaining a working memory to provide clues for the action policy to generate questions. This proposed method divides QG into two more manageable and coherent sub-tasks, i.e., answer-centric summary generation and summary-augmented QG, thereby simplifying the exploration process and accelerating policy learning.

To maintain the one-to-many nature of RLQG, we present a diversity-seeking RL objective inspired by Generative Flow Networks (GFlowNets) (Bengio et al. 2021, 2023) and Soft Q-learning (Haarnoja et al. 2017). This diversity-seeking RL objective shifts attention from the maximum-reward state to the entire reward distribution, guiding the policy to match the reward model in probability rather than to find a configuration that maximizes rewards or returns (Malkin et al. 2022; Hu et al. 2024). In other words, the optimal policy under this objective is defined to generate sequences with probabilities proportional to their rewards, thereby balancing optimality and diversity.

In summary, our main contributions are the following: (i) We identify the sample efficiency issue in existing RLQG work. To handle this, we propose a bilevel policy decomposition framework to facilitate the exploration of high-return outcomes, thus improving sample efficiency. (ii) We identify the diversity issue in current RLQG studies and introduce a new RL objective that models the full diversity of the reward distribution rather than the maximal-reward state to maintain QG diversity. (iii) Our method sets a new state-of-the-art on multiple QG benchmarks, surpassing previous RLQG methods in both question quality and diversity metrics.

Related Work

Reinforcement Learning for Question Generation. Previous RLQG research has primarily focused on reward engineering to enhance QG quality. Some investigations emphasize the similarity between the generated questions and the references, using lexical metrics (Chen, Wu, and Zaki 2023) or semantic metrics (Gaur et al. 2022; Zhang and Bansal 2019) as the reward model. However, these reference-based rewards may restrict exploration in policy gradient methods as they actually encourage agents to mimic the reference questions. To this end, some studies have begun to explore reference-free rewards (Ramnath et al. 2024) to reflect QG quality, such as answerability, fluency, and context relevance (Xie et al. 2020; Hong and Liu 2024). While numerous reward signals for QG quality have been studied, their effective deployment in RL has received considerably less attention. This paper aims to advance this research area by refining the policy learning process and ensuring the developed RL policy can maintain the one-to-many nature of QG.

Diverse Question Generation Modeling. Diversity is inherent to QG and holds value for many downstream tasks (Sultan et al. 2020). Early attempts formulate diverse QG modeling as a variational inference problem (Shen et al. 2019; Cho, Seo, and Hajishirzi 2019; Wang et al. 2020). They incorporate a latent variable for content selection and maximize the evidence lower bound to find the best candidate distribution. Similarly, Narayan et al. (2022) introduce a diverse sampling method which first utilizes nucleus sampling (Holtzman et al. 2020) to sample a chain-of-entities and uses beam search (Freitag and Al-Onaizan 2017) to find questions with high likelihood conditioned on the entities. Some other work also seeks to boost question diversity by recursive generation (Yoon and Bak 2023) and retrieval-augmented style transfer (Gou et al. 2023). Different from previous work, we focus on extracting complex and diverse behaviors from the language model itself using RL, with no additional efforts for crafting proposal distributions or retrieving external question templates.

Methodology

Preliminaries and Notions

We follow previous RLQG studies to consider the policy gradient framework of RL and adopt the Markov Decision Process (MDP) as the mathematical model (Ramamurthy et al. 2023). A standard RLQG procedure can be viewed as an MDP $\langle \mathcal{S}, \mathcal{Q}, \mathcal{P}, \mathcal{R}, \mathcal{T} \rangle$ using a finite vocabulary $\mathcal{V}$ in a sparse reward environment. An episode in the MDP starts with a reading passage and answer pair (x, a) which is used as the initial state $s_0 \in \mathcal{S}$, where $\mathcal{S}$ is the state space. At each time step t, the agent follows a policy $\pi : \mathcal{S} \times \mathcal{Q} \mapsto [0, 1]$, which is generally represented by a parameterized function (e.g., a neural network) π_θ, to sample a question token $q_t \in \mathcal{Q}$ from vocabulary $\mathcal{V}$. The transition

function $\mathcal{P} : \mathcal{S} \times \mathcal{Q} \mapsto \triangle(\mathcal{S})$ deterministically appends q_t to the end of the state $s_{t-1} = (x, a, q_{<t})$. This process continues until t exceeds the time horizon $\mathcal{T}$ or an end-of-sentence (EOS) token is generated, yielding a question sequence $q = (q_0, \ldots, q_T)$. At the end of an episode, a reward model $\mathcal{R} : \mathcal{S} \times \mathcal{Q} \mapsto \mathbb{R}^1$ assigns a scalar reward value to the last generated token. A typical goal in previous RLQG work is to find the parameters θ of the policy that maximizes the expected rewards $J(\pi_\theta) = \mathbb{E}_{s\sim\rho^\pi, q\sim\pi_\theta}[\mathcal{R}(s, q)]$.

Bilevel Policy Decomposition

Directly learning a QG policy $\pi_\theta(q|x, a)$ through reward optimization is challenging due to the implicit mapping $(x, a) \mapsto q$, which hinders the exploration of high-return questions and impedes sample efficiency. To address this, we introduce a bilevel policy decomposition (BPD) framework, which segments $\pi_\theta(q|x, a)$ into a high-level *rationale* policy $\pi_\theta(d|x, a)$ and a low-level *action* policy $\pi_\theta(q|d, x, a)$:

$$\pi_\theta(q|x, a) = \sum_d \pi_\theta(q, d|x, a) = \underbrace{\pi_\theta(d|x, a)}_{\text{rationale policy}} \underbrace{\pi_\theta(q|x, a, d)}_{\text{action policy}}. \tag{1}$$

Both policies are optimized with policy gradients to achieve a simpler sub-task and finally work synergistically for QG. Specifically, the rationale policy $\pi_\theta(d|x, a)$ directs the agent to generate a rationale d (i.e., an answer-centric summary; see Figure 1 (a)). It maintains a working memory that captures process reward signals r_d and provides clues for future decision-making; the action policy $\pi_\theta(q|d, x, a)$ subsequently leverages the rationale and the initial input to generate questions, aiming to align these questions with the outcome reward signals r_q.

The idea of BPD is: by breaking down an intricate task into several coherent and simpler sub-tasks, the overall problem becomes more approachable, allowing for more efficient exploration of high-return outcomes (Yan et al. 2024; Zhou et al. 2024; Yao et al. 2023; Zhou 2023). First, we use a rationale policy to solve the task $(x, a) \mapsto d$ for the essential source-target transformation, which intuitively presents less ambiguity compared to $(x, a) \mapsto q$. Subsequently, d serves as an intermediate representation to augment the context to the action policy, which makes the exploration of good questions more explicit and easier, thereby improving the overall sample efficiency during RL.

Diversity-Seeking RL Objective

Building on our BPD framework, we aim to develop rationale and action policies that can sample a diverse set of high-return solutions. This objective recognizes the inherent one-to-many nature of QG, which contrasts previous RLQG work that focuses on a single return-maximizing question. To achieve this, our approach is to develop a learning objective that directly transforms the reward model into a generative policy, such that the likelihood of generating sequences becomes proportional to their rewards. Formally, given the process reward model $r_d(x, a, d)$ and the outcome reward model $r_q(x, a, q)$, the optimal rationale policy and the optimal action policy can be defined as:

$$\pi^*_\theta(d|x, a) \propto r_d(x, a, d) = \frac{r_d(x, a, d)}{\sum_d r_d(x, a, d)}, \tag{2}$$

$$\pi^*_\theta(q|x, a, d) \propto r_q(x, a, q) = \frac{r_q(x, a, q)}{\sum_q r_q(x, a, q)}, \tag{3}$$

where $\sum_d r_d(x, a, d)$ and $\sum_q r_q(x, a, q)$ are the partition functions that turn the unnormalized measures of reward models into probability distributions.

Using the multiplication rule, we can further derive the overall optimal policy from Eq. (2) and (3):

$$\pi^*_\theta(q, d|x, a) = \frac{r_d(x, a, d) r_q(x, a, q)}{\sum_d r_d(x, a, d) \sum_q r_q(x, a, q)}. \tag{4}$$

We cast the optimal policy search problem to a minimization problem. Suppose there is a bilevel decomposed policy $\pi_\theta(q, d|x, a)$ with parameters θ and a regressor $Z_\mu(x, a)$ with parameters μ which takes a (x, a) pair as input and estimates $\sum_d r_d(x, a, d) \sum_q r_q(x, a, q)$, the diversity-seeking RL (DSRL) objective is defined using L_2 loss on a log-scale:

$$\mathcal{L}_{\text{DSRL}}(d, q) = [\log \pi_\theta(q, d|x, a) - \log \frac{r_d(x, a, d) r_q(x, a, q)}{Z_\mu(x, a)}]^2. \tag{5}$$

Eq. (5) reveals an intriguing connection to the Trajectory Balance (TB) objective (Malkin et al. 2022) for GFlowNets training. It can be treated as a special case of TB where the policy is conditional on an input variable and samples actions autoregressively. If one policy can satisfy the constraint in Eq. (4), it is easy to find $\mathcal{L}_{\text{DSRL}}(d, q) = 0$ for any trajectory (d, q) it samples. Conversely, if all trajectories can lead Eq. (5) to zero, the regarding policy then satisfies Eq. (4). Detailed proof can be found in (Malkin et al. 2022), which guarantees the correctness of the DSRL objective.

Implementation

Integrating the above two components, we develop the BPD-DSRL method. The implementation of BPD-DSRL is outlined in Figure 2, which will be detailed as follows.

Bilevel Policy Warm-up with SFT

Given the vast action space during language modeling, directly applying the proposed on-policy RL to PLMs is notably inefficient (Guo et al. 2022). Therefore, we first use SFT to initialize the rationale and action policy from PLMs, with the aim to refine the search domain during RL:

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(d,q,x,a)\sim D}[-\log \pi_\theta(d, q|x, a)], \tag{6}$$

where D is the QG dataset which contains quadruples (d, q, x, a), with d denoting the answer-centric summaries.

A major challenge in this setup is the effective labeling of d. Rather than generating d from (x, a) in a prior manner by text summarization models or manual annotation, we employ a question conversion model (Chen, Choi, and Durrett 2021) to inversely synthesize d from (q, a). This model takes a (q, a) pair as input and converts it to a declarative statement, serving as the summary of the reading passage. Annotating d in this way not only maintains its semantic integrity as the answer-centric summary but also better ensures the coherence between it and the corresponding question.

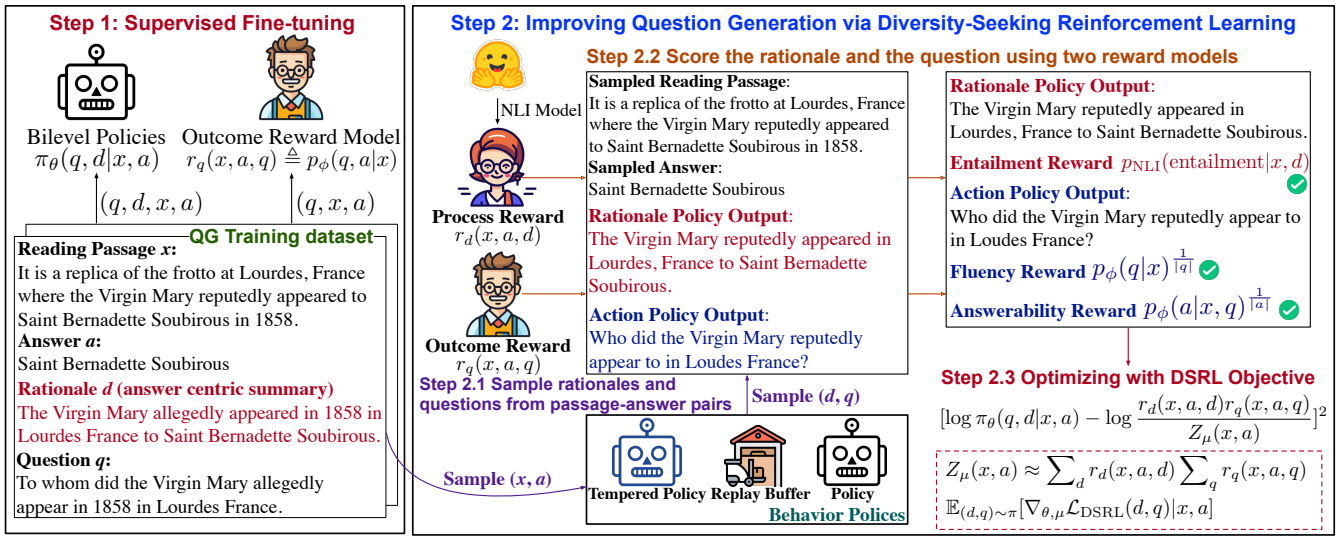

Figure 2: Implementation overview of BPD-DSRL. In the first step, we use SFT to warm up the rationale and action policies, and learn an outcome reward model from the QG dataset augmented by answer-centric summaries. Then, we put the supervised bilevel policies and the reward models into a bandit environment and use the proposed DSRL objective to refine QG.

Reward Model

Process Reward Model. The process reward model offers supervisory signals to guide the rationale policy in generating effective intermediate representations to support the action policy. Consider ideal answer-centric summaries should be textually entailed by the corresponding reading passages to ensure the consistency of subsequent QG, we use natural language inference (NLI) models for the textual entailment task (Bowman et al. 2015) as the process reward model:

$$r_d(x,a,d) \triangleq p_{\text{NLI}}(\text{entailment}|x,d), \tag{7}$$

where p_{NLI} is the NLI model that takes x as premise and d as hypothesis, producing a entailment probability for d which we treat as the process reward score. In practice, we use one off-the-shelf NLI model from (Nie et al. 2020) as the process reward model due to its superior performance.

Outcome Reward Model. The outcome reward model guides the action policy toward generating high-quality questions. Following practices in (Xie et al. 2020), we assess question quality using two key metrics: fluency and answerability. Specifically, given two language models $p_{\text{flu}}(q|x)$ and $p_{\text{ans}}(a|q,x)$, the fluency and answerability scores of an observed question are determined by the geometric mean of the likelihoods produced by these models. Combining these two measures, the outcome reward model is defined as:

$$r_q(x,a,q) \triangleq p_{\text{flu}}(q|x)^{\frac{1}{|q|}} p_{\text{ans}}(a|q,x)^{\frac{1}{|a|}}, \tag{8}$$

where $|\cdot|$ indicates the valid number of tokens.

By observing the left-to-right nature of autoregressive language models, we can exploit both fluency and answerability scores from a unified model $p_\phi(q,a|x) = p_\phi(q|x)p_\phi(a|q,x)$ to lower implementation cost. In practice, we learn this reward model with the following multi-task SFT objective:

$$\begin{aligned}\mathcal{L}^q_{\text{reward}} = -\mathbb{E}_{(q,x,a)\sim D}[&\frac{\beta}{|q|}\sum_{i=1}^{|q|}\log p_\phi(q_i|x,q_{<i}) \\ &+ \frac{1-\beta}{|a|}\sum_{m=1}^{|a|}\log p_\phi(a_m|x,q,a_{<m})],\end{aligned} \tag{9}$$

where β is the coefficient which controls the loss contribution. Note that when β equals to $\frac{|q|}{|q|+|a|}$, the above objective degenerates to the single-task one which maximizes the log-likelihood of the continual sequence (q,a).

Optimize the Policy Against the Reward

Putting the supervised bilevel policies and two reward models in a bandit environment which presents a random (x,a) pair from the training dataset and expects (d,q) as responses, we further fine-tune the bilevel policies using the following RL objective based on Eq. (5):

$$\begin{aligned}\mathcal{L}_{\text{DSRL}}(d,q) = [&\log \pi_\theta(d|x,a) + \log \pi_\theta(q|d,x,a) + \log Z_\mu(x,a) \\ &- \log p_\phi(q|x)^{\frac{1}{|q|}} p_\phi(a|q,x)^{\frac{1}{|a|}} - \log p_{\text{NLI}}(\text{entailment}|x,d)]^2,\end{aligned} \tag{10}$$

with stochastic gradient:

$$\mathbb{E}_{(d,q)\sim\pi}[\nabla_{\theta,\mu}\mathcal{L}_{\text{DSRL}}(d,q)|x,a], \tag{11}$$

where $Z_\mu(x,a)$ is an encoder-only regressor initialized from the supervised bilevel policies $\pi^{\text{SFT}}_\theta(q,d|x,a)$.

As the state space for language modeling is combinatorially large, it is important to have a training policy that can efficiently explore it. We follow the same settings described in (Hu et al. 2024), using trajectories sampled from three different sources to optimize Eq. (10): (1) the current policy π_θ, (2) a tempered version of π_θ, and (3) a replay buffer which stores past experiences credited with high rewards. The pseudo-code for BPD-DSRL training and further technical specifics are provided in the supplementary material.

Dataset	Train	Validation	Test
SQuAD 1.1 / 1	86635	8965	8964
SQuAD 1.1 / 2	75722	10570	11877
NewsQA	92549	5166	5126

Table 1: Statistics of the selected benchmarks. SQuAD 1.1 / 1 and SQuAD 1.1 / 2 are two different splits of SQuAD 1.1 from (Zhou et al. 2017) and (Du, Shao, and Cardie 2017).

Experiments

Experimental Setup

Benchmarks. Following previous work (Gou et al. 2023; Narayan et al. 2022; Wang et al. 2020), we conduct experiments on two QG datasets: SQuAD 1.1 (Rajpurkar et al. 2016) and NewsQA (Trischler et al. 2017). Due to the inaccessibility of SQuAD 1.1's test set, we use two popular splits of it from (Zhou et al. 2017) and (Du, Shao, and Cardie 2017), which we will further refer to SQuAD 1.1 / 1 and SQuAD 1.1 / 2 respectively. As for NewsQA, we apply necessary truncation of its reading passage to comply with the input length constraints of PLMs (e.g., 512 tokens). Statistics about these benchmarks are presented in Table 1.

Metrics. For every testing example $(q_n, x_n, a_n)_{n=1}^N$, we generate K questions from QG models and adopt the following three BLEU-based metrics (Papineni et al. 2002) to evaluate the quality and the diversity of them:

- **Top-1 metric** (↑): This metric evaluates the consistency between model-generated (candidate) questions with the highest confidence and the reference questions q_n. For every test example, it keeps the candidate question $\hat{q}_n^*$ that has the highest likelihood, i.e., $\hat{q}_n^* = \arg\max_{\hat{q}_n^k} p_{\text{QG}}(\hat{q}_n^k, d_n^k | x_n, a_n)$, and calculate the corpus-level BLEU score using $\{(q_n, \hat{q}_n^*)\}_{n=1}^N$.
- **Oracle metric** (↑): This metric evaluates the upper bound of the consistency between the reference and candidate questions. The only difference between this metric and the Top-1 metric lies in the selection strategy of $\hat{q}_n^*$. Here, $\hat{q}_n^*$ is the candidate question that achieves the highest sentence-level BLEU score with q_n.
- **Self metric** (↓): This metric evaluates the average diversity of candidate questions. It pair-wisely calculates the sentence-level BLEU score among the K candidate questions for every testing example. Finally, it averages the N self-metric value to indicate the capability of QG models to generate diverse questions.

In our experiments, we follow (Gou et al. 2023) to set $K = 5$ and report BLEU-4 calculated by SacreBLEU (Post 2018).

Baselines. We compare the proposed BPD-DSRL method with state-of-the-art QG approaches to validate its effectiveness. The baselines can be grouped into three categories:

- **Cross entropy based QG (CEQG)**. We include Composition (Narayan et al. 2022) and RAST (Gou et al. 2023) as our CEQG baselines. They can be treated as extensions of traditional supervised QG methods, where Composition adds a prefix entity sampling task before QG and RAST combines QG with retrieval augmented generation (RAG) (Lewis et al. 2020) to boost diversity.
- **Large language models (LLMs)**. We include two open-sourced LLMs: LLaMA3-8B-instruct and Mistral-7B-instruct-v0.3 (Jiang et al. 2023) as our LLM baselines. We report their zero-shot and three-shot performance in the main experiments. Prompts and demonstrations are presented in the supplementary material.
- **RLQG**. REINFORCE (Williams 1992) and Proximal Policy Optimization (PPO) (Schulman et al. 2017) are two representative RL algorithms used by previous RLQG work (Hong and Liu 2024; Chen, Wu, and Zaki 2023; Gaur et al. 2022). We implement them with BPD as the RLQG baselines. Specific implementation details are presented in the supplementary material.

Implementation Details. All of our QG models and outcome reward models start from the pre-trained checkpoints of T5-large (Raffel et al. 2020). We use consistent hyperparameter configurations across all three datasets during training (SFT warm-up and RL) and inference. The implementation of them is detailed in the supplementary material.

Main Results and Analysis

Table 2 presents the comparative results on three widely-used QG benchmarks. Regarding QG quality (consistency with the reference), our BPD-DSRL consistently surpasses all the baselines on the **Top-1** metric, and secures the best average rank (1.13 compared to the runner-up's 1.67) on the **Oracle** metric. As for diversity, our method outperforms all non-RAG approaches on the **Self** metric, including Composition (Narayan et al. 2022), LLMs, and RLQG baselines.

More specifically, it is clear that the RLQG group generally achieves better performance on Top-1 and Oracle compared to the CEQG baselines (Narayan et al. 2022; Gou et al. 2023). This finding verifies our motivation for using RL to improve QG quality. While RAST (Gou et al. 2023) offers enhanced diversity, its reliance on RAG may incur significant costs regarding memory usage and inference latency. In fact, RAST provides a stand-alone RAG framework that could potentially be integrated into our method to further enrich QG diversity. However, consider the primary focus of this paper is on RLQG and the high implementation demands of RAST, we reserve its integration for future work.

Within the RLQG group, our method generally outperforms the baselines on QG quality and diversity metrics across all test datasets, yielding the most favorable overall performance. We conduct further analysis in the next section using advanced metrics for QG quality and diversity to better understand their model behaviors.

As for LLMs, our method significantly outperforms zero-shot and few-shot configurations of LLaMA-3-8B-Instruct and Mistral-7B-Instruct-v0.3, even with about 10% parameters. This makes our method a more resource-efficient choice for QG-specific scenarios such as data augmentation.

Ablation Studies

To further validate our approach, we conduct a series of ablation studies on our BPD framework and DSRL objective.

Group	Model	Year	SQuAD 1.1 / 1			SQuAD 1.1 / 2			NewsQA		
			Top-1 ↑	Oracle ↑	Self ↓	Top-1 ↑	Oracle ↑	Self ↓	Top-1 ↑	Oracle ↑	Self ↓
CEQG	Composition	2022	16.50	25.70	58.99	15.94	24.90	60.05	-	-	-
	RAST	2023	19.25	23.23	**48.91**	19.36	22.59	**56.42**	11.02	16.26	**23.16**
LLMs	LLaMA3† 3-shot	2024	12.19	20.45	75.14	12.20	20.55	77.87	4.53	8.57	74.36
	LLaMA3† 0-shot	2024	10.83	19.19	73.16	10.42	18.86	73.87	3.91	8.41	69.23
	Mistral† 3-shot	2023	11.02	16.47	71.50	11.80	17.27	74.92	4.41	7.31	71.11
	Mistral† 0-shot	2023	10.74	16.55	69.88	10.33	15.99	69.92	4.38	7.49	68.83
RLQG	BPD-REINFORCE	-	19.82	25.65	67.79	18.93	25.16	71.71	14.11	20.30	63.63
	BPD-PPO	-	18.99	26.31	56.85	18.47	26.72	61.91	13.92	**21.09**	55.44
	BPD-DSRL	**Ours**	**19.92**	**26.58**	55.28	**19.66**	**27.02**	58.63	**14.13**	20.69	49.19

Table 2: Comparison of different QG methods. Here, LLaMA3† and Mistral† respectively indicate the 8B-Instruct version and the 7B-Instruct-v0.3 version of them. The up-arrow ↑ means higher value is better and the down-arrow ↓ means lower value is better. Experimental results of Composition (Narayan et al. 2021) is reevaluated by (Gou et al. 2023).

BPD	DSRL		SQuAD 1.1 / 1			SQuAD 1.1 / 2			NewsQA		
	r_d	r_q	Top-1 ↑	Oracle ↑	Self ↓	Top-1 ↑	Oracle ↑	Self ↓	Top-1 ↑	Oracle ↑	Self ↓
-	-	-	18.52	25.12	48.44	18.45	25.89	51.17	10.20	16.79	31.89
-	✓	-	18.55	25.19	48.40	18.53	26.01	51.84	11.73	18.00	41.87
-	-	✓	17.97	24.71	43.04	17.86	25.36	45.42	10.27	15.73	30.89
-	✓	✓	18.56	25.30	48.51	18.59	25.98	51.57	12.47	18.74	43.43
✓	-	-	19.41	25.91	49.85	18.85	26.52	52.88	12.27	18.52	39.74
✓	✓	-	19.45	25.99	53.40	19.05	26.43	57.35	13.72	19.90	47.04
✓	-	✓	19.64	26.08	51.18	19.30	27.02	53.77	13.18	20.25	44.49
✓	✓	✓	19.92	26.58	55.28	19.66	27.02	58.63	14.13	20.69	49.19

Table 3: Ablation studies on the effectiveness of BPD and DSRL. In the DSRL column, r_d and r_q respectively indicate whether we use the process or outcome reward models during training. Note that for baselines trained without BPD, the process reward model is defined by $p_{\text{NLI}}(\text{entailment}|x, f(q, a)))$, where $f(\cdot)$ is the question conversion model (Chen, Choi, and Durrett 2021).

Effectiveness of BPD for QG. We first perform an ablation study on the BPD framework to see its effectiveness for the task of QG. To achieve this, we train an SFT baseline $\pi^{\text{SFT}}(q|x, a)$, which directly models $(x, a) \mapsto q$, and compare it with our SFT initialization $\pi^{\text{SFT}}(q, d|x, a)$. Presented in the **first** and the **fifth** row in Table 3, $\pi^{\text{SFT}}(q, d|x, a)$ can outperform its ablated version $\pi^{\text{SFT}}(q|x, a)$ on QG quality metrics across all three datasets. These empirical results are consistent with our previous hypothesis that BPD can make QG more tractable. Since the prefix rationales may constrain the searching space for questions, the QG diversity may get influenced. However, on the two test splits of SQuAD 1.1, we find the diversity cost is generally on an acceptable level.

Effectiveness of BPD for RLQG. We proceed to evaluate how BPD improves sample efficiency in DSRL. Specifically, we train an RLQG baseline from the above supervised model $\pi^{\text{SFT}}(q|x, a)$ using DSRL (referred to as DSRL-only) under the same training configurations as BPD-DSRL. The evaluation results for BPD-DSRL and DSRL-only are presented in Table 3. Additionally, after each RL training step, we sample 16 questions (rationales) from the current passage-answer pair using both BPD-DSRL and DSRL-only. The cumulative averages of the process and outcome rewards are documented in Figure 3. As shown in the **fourth** row and the **last** row in Table 3, within the same time horizon, BPD can bring substantial absolute improvement in quality metrics compared to its ablated version. By respectively comparing BPD-DSRL and DSRL-only with their SFT initialization, we also notice a great relative enhancement. Taking SQuAD 1.1 / 1 for example, BPD-DSRL gains 2.61% and 2.58% improvement on Top-1 and Oracle metrics compared to its SFT start point, while the values for DSRL-only are 0.21% and 0.72%. Figure 3 more vividly illustrates the enhanced sample efficiency achieved by BPD, where BPD-DSRL achieves higher cumulative values of both process and outcome rewards compared to DSRL-only.

Effectiveness of DSRL and Reward Models. We further ablate the reward models to validate their effectiveness and explore whether the proposed DSRL can well leverage them. Starting from $\pi^{\text{SFT}}(q|x, a)$ and $\pi^{\text{SFT}}(q, d|x, a)$, we use DSRL to train four RLQG baselines with either the process reward model or the outcome reward model. As shown in the **second** to **third** and the **sixth** to **seventh** rows in Table 3, DSRL with both separate reward models can generally bring improvement to the SFT initialization. Combining these two reward models together (the **fourth** row and the **last** row), the performance on quality metrics can be further improved. Such experimental results demonstrate the effectiveness of DSRL and indicate the compatibility of the employed process and outcome reward models.

Further Analysis

In the previous section, we follow prior work to report experimental results by BLEU-based metrics. While these metrics provide initial insights, they may not adequately capture

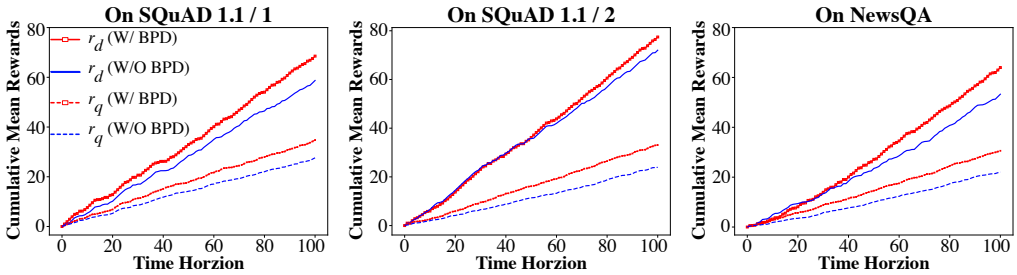

Figure 3: Ablation studies on BPD's sample efficiency for DSRL. At each training step, we sample 16 questions (and corresponding rationales, when applicable) using both BPD-DSRL and the ablated DSRL-only variant. We then report the cumulative average process and outcome rewards to illustrate their relative sample efficiency in aligning with the reward models.

RLQG	SQuAD 1.1 /1				SQuAD 1.1 / 2				NewsQA			
	HR ↓	ANS ↑	ACC ↑	SBS ↓	HR ↓	ANS ↑	ACC ↑	SBS ↓	HR ↓	ANS ↑	ACC ↑	SBS ↓
BPD-REINFORCE	**4.48**	**90.12**	**82.66**	94.84	7.94	94.34	**86.86**	95.41	2.64	91.34	**65.04**	94.41
BPD-PPO	7.46	84.04	80.17	93.84	6.62	93.89	84.51	94.83	**1.97**	90.30	61.22	93.94
BPD-DSRL (Ours)	7.09	86.39	80.93	**93.48**	**6.05**	**94.99**	85.98	**94.07**	2.62	**91.55**	64.63	**92.65**

Table 4: Further analysis on RLQG methods. **HR**: Hallucination Rate based on Spacy. **ANS**: Answerability Rate based on GPT-3.5. **ACC**: Accuracy of GPT-3.5 among questions it judges as answerable. **SBS**: Self metric based on BertScore.

more nuanced aspects of the generated questions. Focusing on RLQG methods, we conduct a further analysis using metrics beyond the lexical level to gain a more holistic understanding of QG quality and diversity.

Moreover, we also enlist three annotators to critically evaluate the performance of all RLQG methods on QG quality and QG diversity, following similar human evaluation procedures in (Xia et al. 2023; Gou et al. 2023). Detailed results can be found in the supplementary material.

Hallucination and Answerability

Hallucination and unanswerability have significant impacts on user experience and are crucial for assessing the quality of generated questions (Xie et al. 2020; Narayan et al. 2022). Traditional corpus-level metrics such as BLEU, however, may not be able to effectively measure these attributes (Dale et al. 2023). To this end, we here quantify both indicators to provide deeper insights into QG quality.

To assess hallucination, we employ SpaCy to extract named entities from generated questions and verify their presence in the corresponding reading passages. We calculate and report the average proportion of questions containing hallucinated entities (HR%). As for answerability, we follow a common practice (Liu, Huang, and Chang 2023; Mohammadshahi et al. 2023) to deploy a QA model to determine if a generated question is answerable. For precision and cost-effectiveness, we utilize GPT-3.5 (Turbo-0125) in a zero-shot setting as the QA model. We present both the average proportion of answerable questions (ANS%) and the model accuracy (ACC%) of these questions.

Table 4 presents the experimental results for the above three metrics. Generally, the RLQG approaches exhibit comparable performance in terms of faithfulness (HR%) and answerability (ANS%). A notable observation is that questions generated by the BPD-REINFORCE baseline are most often solvable by GPT-3.5 (highest ACC%). Our human evaluations indicate this could result from the highly extractive nature of questions generated by BPD-REINFORCE, which may suggest the policy is over-optimized.

Semantic Diversity

Given that the Self-BLEU metric in the main experiments only measures n-gram diversity, we here replace BLEU with BERTScore (Zhang et al. 2020) in the Self metric to provide a more comprehensive analysis of QG diversity. BERTScore evaluates the cosine similarity between embeddings generated by PLMs, offering deeper insights into the semantic diversity of the generated questions. For this analysis, we employ the official BERTScore package in its default configuration and report the resulting BERTScore F1 values.

Table 4 outlines the relevant experimental results. Generally, the outcomes from Self-BERTScore (SBS) are consistent with those obtained from Self-BLEU. Our BPD-DSRL continues to achieve the best performance in diversity among the RLQG group while the BPD-REINFORCE baseline tends to yield the most deterministic policy.

Conclusion

In this paper, we present BPD-DSRL, a bilevel policy decomposition framework and a diversity-seeking reinforcement learning objective, to improve RLQG sample efficiency and preserve QG diversity. Compared to existing RLQG methods, policies developed by BPD-DSRL yield more diverse questions while achieving superior performance across different quality metrics within the same time horizon, setting a new state-of-the-art on three widely-used QG benchmarks. Comprehensive ablation studies, as well as supplementary quantitative and qualitative analyses, further validate the effectiveness of BPD-DSRL in improving sample efficiency and fostering QG diversity.

Acknowledgments

This research is supported by the PwC Research and Development Center (R5212ECS). The opinions presented in this work are not necessarily reflective of the views of the funding organizations. We extend our heartfelt gratitude to Zhaoyu Zhang and Stanley Simoes for their invaluable comments and feedback. Their contributions were instrumental in making this work possible.

References

Bengio, E.; Jain, M.; Korablyov, M.; Precup, D.; and Bengio, Y. 2021. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. In *Neurips 2021*.

Bengio, Y.; Lahlou, S.; Deleu, T.; Hu, E. J.; Tiwari, M.; and Bengio, E. 2023. GFlowNet Foundations. *Journal of Machine Learning Research*, 24(210): 1–55.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *ACL 2015*.

Chen, J.; Choi, E.; and Durrett, G. 2021. Can NLI Models Verify QA Systems' Predictions? In *EMNLP 2021 Findings*.

Chen, Y.; Wu, L.; and Zaki, M. J. 2023. Toward subgraph-guided knowledge graph question generation with graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9): 12706–12717.

Cho, J.; Seo, M.; and Hajishirzi, H. 2019. Mixture Content Selection for Diverse Sequence Generation. In *EMNLP-IJCNLP 2019*.

Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. In *Neurips 2017*.

Colson, B.; Marcotte, P.; and Savard, G. 2007. An overview of bilevel optimization. *Annals of operations research*, 153: 235–256.

Dale, D.; Voita, E.; Barrault, L.; and Costa-jussà, M. R. 2023. Detecting and Mitigating Hallucinations in Machine Translation: Model Internal Workings Alone Do Well, Sentence Similarity Even Better. In *ACL 2023*.

Du, X.; Shao, J.; and Cardie, C. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *ACL 2017*.

Freitag, M.; and Al-Onaizan, Y. 2017. Beam Search Strategies for Neural Machine Translation. In *ACL 2017 Workshop: Neural Machine Translation*.

Gaur, M.; Gunaratna, K.; Srinivasan, V.; and Jin, H. 2022. ISEEQ: Information Seeking Question Generation Using Dynamic Meta-Information Retrieval and Knowledge Graphs. In *AAAI 2022*.

Gou, Q.; Xia, Z.; Yu, B.; Yu, H.; Huang, F.; Li, Y.; and Cam-Tu, N. 2023. Diversify Question Generation with Retrieval-Augmented Style Transfer. In *EMNLP 2023*.

Guo, H.; Tan, B.; Liu, Z.; Xing, E.; and Hu, Z. 2022. Efficient (Soft) Q-Learning for Text Generation with Limited Good Data. In *EMNLP 2022 Findings*.

Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. In *ICML 2017*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. In *ICLR 2020*.

Hong, Z.; and Liu, J. 2024. Towards Better Question Generation in QA-based Event Extraction. arXiv:2405.10517.

Hu, E. J.; Jain, M.; Elmoznino, E.; Kaddar, Y.; Lajoie, G.; Bengio, Y.; and Malkin, N. 2024. Amortizing intractable inference in large language models. In *ICLR 2024*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Neurips 2020*.

Ling, Y.; Cai, F.; Chen, H.; and de Rijke, M. 2020. Leveraging Context for Neural Question Generation in Open-domain Dialogue Systems. In *WWW 2020*.

Liu, Y.; Huang, J.; and Chang, K. 2023. Ask To The Point: Open-Domain Entity-Centric Question Generation. In *EMNLP 2023 Findings*.

Malkin, N.; Jain, M.; Bengio, E.; Sun, C.; and Bengio, Y. 2022. Trajectory balance: improved credit assignment in GFlowNets. In *Neurips 2022*.

Mohammadshahi, A.; Scialom, T.; Yazdani, M.; Yanki, P.; Fan, A.; Henderson, J.; and Saeidi, M. 2023. RQUGE: Reference-Free Metric for Evaluating Question Generation by Answering the Question. In *ACL 2023 Findings*.

Narayan, S.; Simões, G.; Zhao, Y.; Maynez, J.; Das, D.; Collins, M.; and Lapata, M. 2022. A Well-Composed Text is Half Done! Composition Sampling for Diverse Conditional Generation. In *ACL 2022*.

Narayan, S.; Zhao, Y.; Maynez, J.; Simões, G.; Nikolaev, V.; and McDonald, R. 2021. Planning with Learned Entity Prompts for Abstractive Summarization. *Transactions of the Association for Computational Linguistics*, 9: 1475–1492.

Nie, Y.; Williams, A.; Dinan, E.; Bansal, M.; Weston, J.; and Kiela, D. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *ACL 2020*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *Neurips 2022*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002*.

Post, M. 2018. A Call for Clarity in Reporting BLEU Scores. In *WMT 2018*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP 2016*.

Ramamurthy, R.; Ammanabrolu, P.; Brantley, K.; Hessel, J.; Sifa, R.; Bauckhage, C.; Hajishirzi, H.; and Choi, Y. 2023. Is Reinforcement Learning (Not) for Natural Language Processing: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization. In *ICLR 2023*.

Ramnath, S.; Joshi, B.; Hallinan, S.; Lu, X.; Li, L. H.; Chan, A.; Hessel, J.; Choi, Y.; and Ren, X. 2024. Tailoring Self-Rationalizers with Multi-Reward Distillation. In *ICLR 2024*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Shakeri, S.; Nogueira dos Santos, C.; Zhu, H.; Ng, P.; Nan, F.; Wang, Z.; Nallapati, R.; and Xiang, B. 2020. End-to-End Synthetic Data Generation for Domain Adaptation of Question Answering Systems. In *EMNLP 2020*.

Shen, T.; Ott, M.; Auli, M.; and Ranzato, M. 2019. Mixture Models for Diverse Machine Translation: Tricks of the Trade. In *ICML 2019*.

Sultan, M. A.; Chandel, S.; Fernandez Astudillo, R.; and Castelli, V. 2020. On the Importance of Diversity in Question Generation for QA. In *ACL 2020*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book.

Tian, K.; Mitchell, E.; Yao, H.; Manning, C. D.; and Finn, C. 2024. Fine-Tuning Language Models for Factuality. In *ICLR 2024*.

Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; and Suleman, K. 2017. NewsQA: A Machine Comprehension Dataset. arXiv:1611.09830.

Wang, Z.; Rao, S.; Zhang, J.; Qin, Z.; Tian, G.; and Wang, J. 2020. Diversify Question Generation with Continuous Content Selectors and Question Type Modeling. In *EMNLP 2020 Findings*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; ichter, b.; Xia, F.; Chi, E.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Neurips 2022*.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.*, 8(3–4).

Xia, Z.; Gou, Q.; Yu, B.; Yu, H.; Huang, F.; Li, Y.; and Cam-Tu, N. 2023. Improving Question Generation with Multi-level Content Planning. In *EMNLP 2023 Findings*.

Xie, Y.; Pan, L.; Wang, D.; Kan, M.-Y.; and Feng, Y. 2020. Exploring Question-Specific Rewards for Generating Deep Questions. In *COLING 2020*.

Yan, X.; Song, Y.; Cui, X.; Christianos, F.; Zhang, H.; Mguni, D. H.; and Wang, J. 2024. Ask more, know better: Reinforce-Learned Prompt Questions for Decision Making with Large Language Models. arXiv:2310.18127.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *ICLR 2023*.

Yoon, H.; and Bak, J. 2023. Diversity Enhanced Narrative Question Generation for Storybooks. In *EMNLP 2023*.

Yu, L.; Jiang, W.; Shi, H.; YU, J.; Liu, Z.; Zhang, Y.; Kwok, J.; Li, Z.; Weller, A.; and Liu, W. 2024. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. In *ICLR 2024*.

Zhang, S.; and Bansal, M. 2019. Addressing Semantic Drift in Question Generation for Semi-Supervised Question Answering. In *EMNLP-IJCNLP 2019*.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *ICLR 2020*.

Zhao, Z.; Hou, Y.; Wang, D.; Yu, M.; Liu, C.; and Ma, X. 2022. Educational Question Generation of Children Storybooks via Question Type Distribution Learning and Event-centric Summarization. In *ACL 2022*.

Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; and Zhou, M. 2017. Neural Question Generation from Text: A Preliminary Study. arXiv:1704.01792.

Zhou, W. 2023. Bi-Level Offline Policy Optimization with Limited Exploration. In *Neurips 2023*.

Zhou, X.; Yuan, Y.; Yang, S.; and Hao, J. 2024. MENTOR: Guiding Hierarchical Reinforcement Learning with Human Feedback and Dynamic Distance Constraint. arXiv:2402.14244.