# ORSO QGen: Odds-Ratio Steerable Optimization for Controlling Question Generation

**Andreea Dutulescu and Stefan Ruseti and Mihai Dascalu**
National University of Science and Technology POLITEHNICA Bucharest
{andreea.dutulescu, stefan.ruseti, mihai.dascalu}@upb.ro

**Danielle S. McNamara**
Arizona State University
danielle.mcnamara@asu.edu

## Abstract

Question generation plays an important role in educational applications, enabling automated assessment and reading comprehension support. Attribute-controlled question generation aims to produce questions that fit predefined characteristics such as difficulty, focus, or coverage. Existing methods predominantly rely on supervised fine-tuning, which often fails to impose a strong adherence to attribute values, resulting in weak coupling between prompt specifications and model outputs. We introduce Odds-Ratio Steerable Optimization (ORSO), a framework designed to enhance attribute sensitivity in question generation models. Building upon preference-based learning techniques without requiring human-curated preference sets, ORSO uses input-level perturbations to create contrastive training signals. Empirical evaluations on both exhaustive and expert-validated attribute configurations indicate that ORSO performs better than SteerLM and ORPO methods in enforcing attribute conformity while maintaining output quality. These results argue for the benefits of explicit attribute-aware optimization in controllable question generation tasks.

## 1 Introduction

Automated question generation represents a scalable approach to educational assessment, enabling rapid feedback and supporting reading comprehension. However, producing questions that align with diverse objectives continues to present technical challenges. Accordingly, the task of attribute-controllable question generation involves producing natural language questions that adhere to predefined attributes, such as difficulty, focus, or type. Such fine-grained control is increasingly important in downstream applications.

Despite growing interest in this domain, the predominant approach remains supervised fine-tuning, where models are trained on annotated datasets with explicit attribute labels. A further obstacle is the scarcity of datasets annotated with multiple fine-grained attributes, which significantly limits the capacity of models to generalize or transfer control to new combinations of attributes. This often necessitates data augmentation strategies or proxy methods. While methods (Dong et al., 2023; Tu et al., 2024) have been proposed to facilitate controllable generation, they are designed under the assumption that only a limited subset of attributes is important at generation time. Consequently, they fall short in scenarios demanding full-spectrum attribute control. To the best of our knowledge, there is a scarcity of end-to-end solutions capable of enforcing comprehensive and simultaneous control across all relevant attributes in question generation.

To address these challenges, we propose Odds-Ratio Steerable Optimization (ORSO), a training paradigm that improves prompt attribute-awareness. Unlike classical supervised fine-tuning, where the model is only encouraged to maximize the likelihood of an output given an attribute-conditioned input, ORSO explicitly penalizes the model for producing the same output under differing, incompatible attribute configurations. This directly targets the issue wherein models tend to only loosely consider attribute values.

We build upon recent advances in preference-based learning, but instead of relying on human-curated preference pairs, ORSO leverages input-level perturbations. Specifically, a positive input containing the correct attribute values is paired for each training instance with a negative input where the attributes are intentionally corrupted, while keeping the target output fixed. The training objective then encourages the model to assign a higher likelihood to the output conditioned on the correct attribute configuration than on the perturbed one.

In this work, we argue that ORSO achieves superior attribute sensitivity and generation fidelity across multiple control variables, outperforming



*Findings of the Association for Computational Linguistics: EACL 2026*, pages 5248–5259

conventional supervised methods. We publicly release the source code and the best model at https://github.com/upb-nlp/EACL-ORSO/.

## 2 Related Work

### 2.1 Controllable Generation

Li and Zhang (2024) introduced a two-step pipeline for controllable question generation, combining answer planning and question synthesis. First, an instruction-tuned LLM generates a structured answer plan from a given context, guided by control signals. The plan consists of key information units or candidate answer spans. In the second stage, the context, plan, and control-aware prompt were used as input to a question generation model to produce aligned question-answer pairs. While the approach achieved strong generation quality, the authors have not released their models for open evaluation, limiting reproducibility.

Tu et al. (2024) introduced a constrained decoding framework for LLMs to guide text generation for satisfying arbitrary constraints (e.g., lexical inclusion, toxicity avoidance). At each decoding step, the next candidate tokens are reranked not only by their standard likelihood scores but also by an auxiliary estimate of constraint satisfaction. Experimental results indicated that their method outperformed standard greedy and beam search decoding, but with a high computational cost.

Dong et al. (2023) introduced SteerLM. Unlike RLHF, which relies on a complex, online training pipeline, SteerLM supports multi-attribute conditioning (e.g., helpfulness, humor, toxicity) at inference time. The training was done by appending discrete attribute values to the input prompt, conditioning the model's generation to explicit signals. This conditioning mechanism is applied during a supervised fine-tuning phase using both human-annotated and automatically predicted attributes. To improve performance, the authors generated additional training data by sampling model outputs. Empirical evaluations argued that this method outperformed RLHF-trained baselines in both automatic and human evaluations. Due to its strong performance and attribute-controllability, SteerLM constitutes a particularly relevant baseline for evaluating alternative alignment strategies.

Guo et al. (2024) proposed a modification to Direct Preference Optimization (DPO; Rafailov et al., 2023) to align LLMs with multiple competing objectives (e.g., helpfulness, harmlessness, honesty). Recognizing that these objectives cannot be maximized simultaneously, the method sought Pareto-optimal solutions by conditioning generation on user-specified preferences, similar to SteerLM. In preference training, rewards were computed based on the deviation between generated outputs and the target preferences, with the higher-reward output treated as the preferred sample. Results showed improved alignment over baseline DPO in multi-objective settings.

### 2.2 Alignment Methods and Self Supervision

Multiple preference alignment techniques have been proposed in the field (Ouyang et al., 2022; Rafailov et al., 2023; Gheshlaghi Azar et al., 2024). ORPO (Odds-Ratio Preference Optimization) was proposed by Hong et al. (2024) for preference alignment to eliminate the need for a reference model. The method combined the standard negative log-likelihood with a log odds ratio penalty of the chosen and rejected responses based on their generation probabilities. Empirical evaluations across diverse model sizes and benchmarks showed that ORPO often surpassed RLHF or DPO paradigms. The authors also provided theoretical evidence for using the odds ratio instead of the probability ratio in preference modeling, highlighting its more stable gradient properties.

These alignment techniques, as well as classical supervised fine-tuning, have served as a base for self-training paradigms when augmenting datasets with synthetic samples generated by the models themselves. Multiple iterative self-supervision studies have been proposed.

Jung et al. (2024) proposed a semi-supervised framework for label bootstrapping using LLMs. Initially, a small set of labeled data was used to prompt an LLM to annotate unlabeled samples. Only high-confidence model predictions were retained and treated as pseudo-gold annotations, which were iteratively added to the training set. To further enhance performance, the model integrated the reasoning for the pseudo-label during both training and inference. Results showed that this strategy outperformed few-shot learning that relied on gold labels.

Liu et al. (2025) used a weak model to produce a large pool of candidate examples, which were then filtered using imposed quality criteria (e.g., likelihood, self-consistency). Only samples that met the standards were retained to form a refined training set. The base model was retrained on this filtered

dataset, and the generation–filtering–retraining cycle was repeated until performance convergence. This iterative framework achieved competitive results in tasks such as paraphrasing and summarization, despite relying on a weaker generation model.

Wang et al. (2024) and Pang et al. (2024) aimed to improve mathematical reasoning and introduced a training pipeline that incorporates Chain-of-Thought (CoT) rationales. The model was initially fine-tuned on CoT-annotated datasets. Subsequently, the model was used to generate intermediate reasoning steps for datasets that contained only (question, answer) pairs. These generated rationales were filtered into good (answer-producing) and bad (non-answer-producing) examples and used to train the model with DPO.

## 3 Method

### 3.1 Dataset

The dataset used in our experiments is FairytaleQA (Xu et al., 2022), a reference dataset in the field of educational question answering. FairytaleQA comprises a collection of children's stories, each accompanied by questions and answers curated and annotated by educational experts. These annotations are particularly relevant to our study, as each question is labeled along two distinct attribute dimensions — i.e., *Focus* and *Coverage*.

The *Focus* attribute indicates the target of the question. Typical categories include character, action, setting, and conflict, among others. This classification helps determine which aspect of the story a model should focus on when generating or answering a question. In contrast, the *Coverage* attribute captures the level of abstraction or textual scope required to answer the question. Specifically, it distinguishes between questions that can be answered based on a local phrase or sentence, versus those that require integrating information across multiple parts of the story or summarizing the narrative as a whole.

We selected FairytaleQA due to its wide usage as a trusted benchmark in the educational community. Its construction ensures high-quality annotations aligned with pedagogical objectives. Importantly, to the best of our knowledge, FairytaleQA is one of the few publicly available question generation datasets that provides annotations along multiple attribute dimensions, which is essential for our work on attribute-conditioned generation and evaluation.

The authors pre-partitioned the dataset into training, validation, and test subsets, consisting of 8548, 1025, and 1007 samples, respectively. However, a detailed analysis of the distribution of attribute values reveals a high degree of imbalance across both *Focus* and *Coverage* categories in the training partition (see Table 1). This imbalance presents challenges for controlled generation and fair evaluation, directly motivating our use of synthetic data generation and attribute balancing techniques, as discussed in later sections.

| **Focus** | **%** | **Coverage** | **%** |
|---|---|---|---|
| Action | 32% | Local | 91% |
| Causal Relation | 28% | Summary | 9% |
| Character | 11% | | |
| Feeling | 10% | | |
| Outcome Resolution | 9% | | |
| Setting | 6% | | |
| Prediction | 4% | | |

Table 1: Distribution of the two FairytaleQA attributes (i.e., Focus & Coverage) on the train partition.

### 3.2 ORSO Training Paradigm

The task at hand involves using a model to generate a question and answer pair based on a structured input. This input consists of: a task description, a text, and a set of attribute values such as *Focus* and *Coverage*. These components are combined into a single prompt provided to the LLM. Given an input $(Ctx, Attr\_Vals)$, a concatenation of the text and attribute values $Attr\_Vals = [attr\_val_1, \ldots, attr\_val_m]$, the objective is to generate an output $(Q, A)$, the question and corresponding answer.

However, in the case of classical supervised fine-tuning, there is a loose connection between different attributes and the generated response. The model generates questions and answers without adapting them to the different attribute configurations, and does not learn the semantics of their values. In supervised fine-tuning, we maximize the probability $P_\theta(Q, A \mid (Ctx, Attr\_Vals)$, but there is no penalty if the model generates a similar output for a different attribute configuration $Attr\_Vals'$. As a result, the model may ignore attribute conditioning and end up with $P_\theta(Q, A \mid Ctx, Attr\_Vals')$ being close to $P_\theta(Q, A \mid (Ctx, Attr\_Vals)$. The model is not penalized explicitly for not respecting the attribute values, as long as the generated context adheres to the provided text and prompt.

Recent progress in LLM alignment introduced methods that optimize over comparative outputs (typically, given the same input, a model is trained to prefer a "chosen" output over a "rejected" one). This training paradigm, adopted by Ouyang et al. (2022), Rafailov et al. (2023), or Hong et al. (2024), is effective but requires the construction or selection of negative samples, which is often non-trivial given the large and diverse output space.

In our approach, we adapt this alignment framework by introducing input-level perturbations instead of sampling from a large space of output negatives. Specifically, we keep the output $y$ fixed and modify the attribute values in the input prompt, creating two input variants: $x_c$: the input with correct attribute values, and $x_r$: the input with incorrect or mismatched attribute values (sampled from the attribute domain, and deliberately inconsistent with $y$). The objective is to encourage the model to attend to attribute variations and penalize inconsistencies between the input attributes and the generated output. This results in better attribute sensitivity during generation.

Our method is inspired by the ORPO framework (Hong et al., 2024), which does not require a reference model and achieved superior empirical performance over DPO and RLHF. The ORPO training objective is defined as:

$$\mathcal{L}_{ORPO}(\theta) = \mathcal{L}_{SFT}(\theta) + \lambda \cdot \mathcal{L}_{OR}(\theta) \quad (1)$$

$$\mathcal{L}_{OR}(\theta) = -\log \sigma \left( \log \frac{odds_\theta(y_c \mid x)}{odds_\theta(y_r \mid x)} \right) \quad (2)$$

where $y_c$ is the chosen output, $y_r$ is the rejected output, $x$ is the input and $odds_\theta(y \mid x) = \frac{P_\theta(y|x)}{1-P_\theta(y|x)}$.

In our proposed variant, ORSO, we redefine the odds-ratio loss (see Equation 2) to operate over perturbed input pairs instead of output pairs:

$$\mathcal{L}_{OR}(\theta) = -\log \sigma \left( \log \frac{odds_\theta(y \mid x_c)}{odds_\theta(y \mid x_r)} \right) \quad (3)$$

where $x_c$ is the input sequence that contains the correct attribute values, $x_r$ is the input with deliberately incorrect attribute values, and $y$ is the reference output corresponding to $x_c$.

A visual comparison illustrating the difference between classical supervised fine-tuning and our ORSO method is provided in Figure 1.

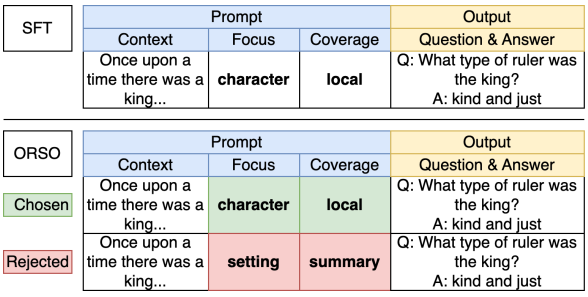



Figure 1: SFT and ORSO prompt and output example

### 3.3 Training Pipeline

We are following the same general framework described by Dong et al. (2023) for SteerLM. It is a common self-supervised approach, also used by Jung et al. (2024), Morimura et al. (2024), and Liu et al. (2025). We describe the steps in the following subsections and provide an overview in Figure 2.

#### 3.3.1 Attribute Prediction Model

This component consists of a set of classifiers, each trained to predict a specific attribute of the generated question, answer, or context (i.e., Focus, Coverage). The purpose of the Attribute Prediction Model is to verify whether the generated outputs conform to the target attribute values specified in the input prompt. A separate classifier is trained for each attribute in the dataset. These classifiers are subsequently used to evaluate the performance of both our proposed method and the baseline approach, in terms of adherence to the prompt.

#### 3.3.2 Training on the Initial Dataset

At this stage, a Large Language Model (LLM) is fine-tuned to generate a question and its corresponding answer, conditioned on a given context and specified attribute values (e.g., Focus or Coverage). The fine-tuning process consists of two sequential phases: an initial epoch of standard supervised learning to adapt the model to the task domain, followed by an epoch of the previously described ORSO-based training. The prompt format is illustrated in Figure 3. In **bold** we denote the expected generated text by the model.

#### 3.3.3 Bootstrapping

Due to the class imbalance in attribute values and the limited number of training instances overall, we introduce a synthetic data augmentation stage. Specifically, for each text in the original training set, we generate synthetic data by oversampling outputs as new question–answer pairs corresponding to all

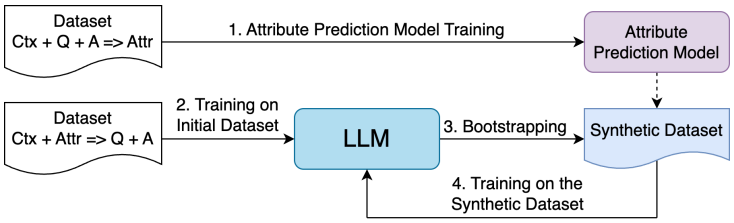



Figure 2: General training pipeline.

Generate a question and an answer based on the following context.
Context: *{{Text}}*
The question must fulfill the following requirements:
- The question must focus on *{{focus_value}}*.
- The question must be answerable based on a *{{coverage_value}}* context.
<start_generation_token>
**Question: {{question}} Answer: {{answer}}**

Figure 3: Prompt format.

possible combinations of attribute values. In our case, we generate 14 new prompts per input text (i.e., Focus with 7 possible values $\times$ Coverage with 2 values). These generated samples expand the training set and are used to enhance the model's generalizability across all attribute configurations.

To improve the effectiveness of the synthetic training data, we apply a filtering pipeline with 4 primary objectives, derived from multiple works in self-supervised frameworks (Ouyang et al., 2022; Rafailov et al., 2023; Hong et al., 2024): *attribute balance* (an equal number of examples for each attribute combination in the synthetic dataset), *generation confidence* (retain only those samples for which the model exhibits high confidence during generation), *output label consistency* (using the previously trained Attribute Prediction Model to infer the actual attributes reflected in each generated output), and *sample diversity* (avoid overly similar samples and reduce redundancy in the training set).

The resulting sample distribution across attribute configurations may still be imbalanced. To construct a balanced synthetic dataset, we aim to retain the top K samples per attribute configuration. Sample selection is guided by generation confidence, where we filter out examples with a negative log-likelihood below an empirical threshold.

To further promote semantic diversity, we apply clustering to the filtered samples. For each attribute configuration, we perform $k$-means clustering, using sequence embeddings obtained via a Sentence Transformer (Wang et al., 2020). From each cluster, we retain the sample with the lowest negative log-likelihood score, ensuring both diversity and quality.

The resulting dataset comprises $K$ high-quality and diverse synthetic samples for each attribute configuration. Note that this process is executed independently for SteerLM and ORSO, yielding two distinct synthetic training datasets, each with its own samples generated using the respective model.

#### 3.3.4 Training on the Synthetic Dataset

Once the filtered synthetic dataset is obtained, we proceed with an additional training phase to further improve model performance. In contrast to the earlier fine-tuning stage (see Section 3.3.2), this phase consists of a single epoch of ORSO-based training. Since the model has already been exposed to the task through supervised fine-tuning, no additional warm-up epoch is required at this stage.

#### 3.3.5 Filtering of Invalid Attribute Configurations

In practice, there are input contexts and attribute configurations for which no valid question can be generated. One way to handle this is to rely on expert supervision: generate questions only for texts and attribute configurations that have been confirmed by experts to support question generation. This can be done by using the human-annotated attributes in the dataset, restricting generation to text–attribute combinations that appear in the annotations, where the possibility of generating a valid question is already established.

However, it is necessary to include a mechanism to detect and communicate this limitation to the user if no expert judgments are available. Specifically, the model should assess whether a requested generation, conditioned on a given context and set of attribute values, is feasible. One approach is to establish a generation confidence threshold, based on the negative log-likelihood of the output. If the model's confidence falls below this threshold, this would indicate that the requested attribute combi-

nation is likely incompatible with the input context. In such cases, the system should abstain from producing an output. This mechanism would help prevent the model from producing incoherent or semantically irrelevant outputs, thereby improving both the reliability and interpretability of the generation process in attribute-controlled settings.

In our case, the threshold is chosen using the validation set by examining the distribution of the negative log-likelihood values for the ground truth examples and considering that values greater than $mean + 1.5 \times std$ are potentially unreliable.

### 3.4 Experimental Setup

The baseline considered in this study is represented by a classical model inspired by the training paradigm of SteerLM (Dong et al., 2023). We use the same training pipeline as described in Section 3.3 for both our approach and SteerLM. The only distinction with SteerLM is that, for Step 3.3.2 (the second epoch) and Step 3.3.4, the training is done with the proposed SteerLM SFT approach. Otherwise, each step remains the same, using the SteerLM baseline instead of our approach for comparison.

This controlled setup allows us to isolate the contribution of our alignment method while leveraging SteerLM as a strong baseline for attribute-aware generation.

In our experiments, we used open-source language models chosen for their computational efficiency and suitability for resource-constrained environments. We ensure that our experiments remain accessible and reproducible without requiring high-end infrastructure. The models and hyperparameters for our experiments are presented in the Appendix C.

We first fine-tuned two independent classifiers, one for each attribute (Focus and Coverage), to implement the Attribute Prediction Model (see Section 3.3.1). We considered ModernBERT (Warner et al., 2024), a recent Transformer-based encoder model, and extended it with a classification head. Each classifier takes as input a concatenation of the context, question, and answer, and predicts the corresponding attribute value. We fine-tune both classifiers for one epoch using the training split of the dataset.

For the task of attribute-conditioned question generation, we use the Instruct version of Llama 3.2 1B (Grattafiori et al., 2024) as the base generative model for both our approach and the SteerLM baseline. During the phase of Training on Initial Dataset (see Section 3.3.2), both models are first fine-tuned using one epoch of standard supervised learning (for task adaptation), followed by an additional epoch using either SteerLM-style SFT or ORSO (our approach).

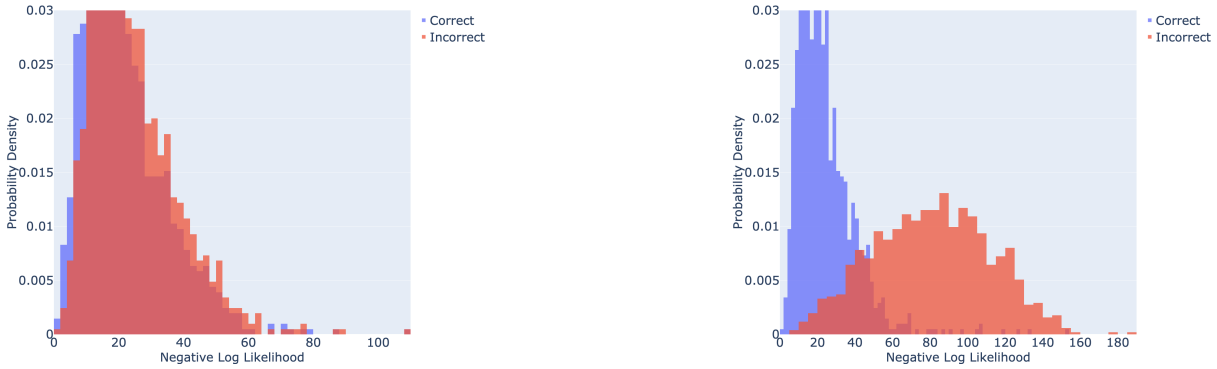
For the bootstrapping stage, we generate 5 samples for each text and attribute configuration to create a large pool of synthetic candidates to select for further training. Based on the analysis in Figure 4, most validation samples exhibit negative log-likelihood scores below a threshold of 100. This threshold is derived from the model's probability of generating the expert-curated validation data and reflects internal model representations. Consequently, during bootstrapping, we discard all generated samples with a negative log-likelihood exceeding 100, as they are likely to be of low quality. Moreover, for each attribute configuration, we keep $K$ = 5000 samples, resulting in a synthetic dataset of 70k examples for each model.

### 3.5 Evaluation Protocol

Initially, we prompt each model on the texts from the test split using all valid attribute combinations to evaluate model behavior. As such, we generate 5 candidate outputs for each input text and attribute pair and select the one with the lowest negative log-likelihood, assuming it represents the most confident and coherent generation.

An important limitation is that some attribute combinations may not be appropriate for the associated input text. For instance, a story that focuses exclusively on a character may not support the generation of meaningful questions about the setting. Since our evaluation initially prompts models to generate outputs for all attribute combinations, it includes cases where the prompt–attribute pairing is unachievable.

To address the limitations of evaluating all attribute combinations, we introduce two new evaluation scenarios. First, we consider an ideal setup in which an expert knows the possible attribute combinations for a given text. This scenario is simulated using the ground truth attributes from the test partition, as these attributes comprise a subset of all valid combinations. Second, we consider a more realistic scenario in which we do not have access to expert knowledge, so we must rely on an automated method for detecting implausible combinations. To do this, we rely on a log-likelihood threshold, as presented in Section 3.3.5.




(a) SteerLM (b) ORSO

Figure 4: Step 3.3.2: Negative log likelihood for Correct versus Incorrect.

### 3.6 Performance Assessment

Question generation is inherently a subjective and open-ended task, where multiple valid outputs may exist for a given input. As such, direct comparison between generated outputs and ground-truth questions is not a reliable or meaningful evaluation strategy. Instead, given our focus on attribute-controlled generation, we evaluate the model's performance based on how well the generated content aligns with the specified attribute values. A prediction is considered correct only if both inferred attributes exactly match the target ones.

### 3.7 Human and LLM-based Evaluations of the Quality of Generated Questions

We conducted an additional experiment to verify that neither model engages in reward hacking, defined as producing outputs that mislead the classifier without generating semantically valid or high-quality questions. We evaluated the quality of the generated questions (in the Ground Truth Attributes setup) using both human raters and LLM-as-a-Judge (GPT-4o). The evaluation was conducted across five criteria, with assigned scores between 1 and 5. Details behind this evaluation are present in Appendix B.

## 4 Results and Discussion

**Attribute Prediction.** The performance of the Attribute Prediction Models, as they are prerequisites in this study, is found in Table 2. These values argue for the adequacy of relying on these classifiers in all subsequent phases.

**SteerLM versus ORSO Evaluations.** We evaluate both SteerLM and ORSO on the initial dataset and after bootstrapping (i.e., on the synthetic dataset). The models were evaluated in three different attribute configurations: all possible combinations, only those present in the dataset, and based on the log probability threshold described in Section 3.3.5. All results are included in Table 3.

| Metric | Focus | Coverage |
|---|---|---|
| F1 | 95.9 | 99.9 |
| Acc. | 96.0 | 99.9 |

Table 2: Classifiers' scores for each attribute.

Initial empirical results indicated that ORSO handles complex attribute interactions better. Analysis of model likelihoods revealed that ORSO learned to distinguish between valid and invalid attribute prompts, whereas SteerLM generated outputs almost agnostically. ORSO, which has been explicitly trained to distinguish between attribute configurations, does not generate plausible outputs when prompted with incompatible attribute values. This is expected, as it has been penalized during training for mismatched attribute adherence. In contrast, SteerLM tends to generate outputs regardless of attribute relevance, as it lacks an explicit mechanism to enforce attribute–output consistency.

This behavior is illustrated in Figure 4, which shows the negative log-likelihood of the model for generating human-curated question–answer pairs (from the validation partition) under both correct and incorrect attribute prompts. For ORSO, the model assigns significantly higher likelihood to generations under the correct attributes (blue) compared to incorrect ones (red), indicating strong attribute sensitivity. SteerLM, on the other hand, displays minimal distinction, suggesting it struggles to differentiate between attribute configurations.

Figure 4 and Table 3 collectively indicate that

| Stage | Attributes combinations | macro-F1 (%) SteerLM | ORSO |
|---|---|---|---|
| Initial - Step 3.3.2 | All Attribute Combinations | 41.6 | **42.8** |
| | Ground Truth Attributes | 85.6 | **97.3** |
| Bootstrapping - Step 3.3.3 | All Attribute Combinations | 43.9 | **53.1** |
| | Ground Truth Attributes | 95.6 | **98.0** |
| | Threshold-filtered Attributes | 44.0 | **83.1** |

Table 3: Model comparison in different configurations

our model outperforms SteerLM. When prompted with expert-curated (plausible) attribute combinations, it approaches near-perfect accuracy. Furthermore, it maintains robust performance across all possible attribute configurations. This improvement stems from the model's enhanced capability to attend to attribute semantics and enforce alignment between prompt attributes and generated content. In contrast, SteerLM fails to reliably distinguish between valid and invalid attribute-output pairings. Moreover, SteerLM potentially overlooks one attribute when it is unable to match both.

The performance of our model, as shown in Table 3, remains robust under automatic filtering of invalid attribute combinations and continues to surpass SteerLM, indicating that this approach provides an effective approximation for identifying implausible generations without requiring human supervision at inference time. This contributes to the development of a more reliable and attribute-aware generation pipeline. The proposed thresholding mechanism proves highly effective for our model but yields limited utility when applied to SteerLM.

The performance gap is even more visible in Figure 5, which shows the performance of the two models in relation to the number of kept examples. ORSO maintains a very high performance while retaining half of the total number of examples, proving the robust performance of the model and that the choice of the threshold value is not critical.

Qualitative examples generated with both methods are available in Appendix A.

**Alternative: Classic ORPO.** We additionally evaluated ORPO (Hong et al., 2024) as an alternative. ORPO optimizes an output-level contrastive objective, whereas ORSO (our method) operates at the input level. Applying ORPO in this setting is not always feasible, as it requires output-level perturbations in the form of reliable rejected answers, which typically need to be human-curated. In contrast, ORSO enables systematic, scalable input-level perturbations by modifying attribute values, without requiring changes to the full textual output.

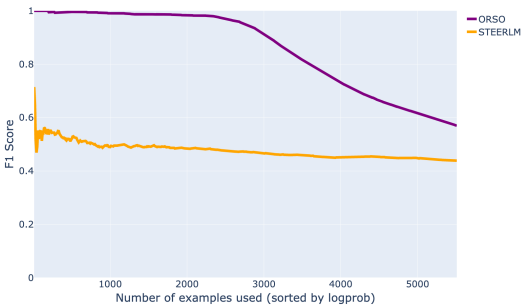



Figure 5: Model performance for different filtering thresholds.

| Attributes Combinations | macro-F1 (%) ORPO | ORSO (ours) |
|---|---|---|
| All Attributes Combinations | 41.4 | **42.8** |
| Ground Truth Attributes | 91.1 | **97.3** |

Table 4: Performance comparison alternative method, Initial Step (3.3.2)

To construct rejected outputs for ORPO, we used the initial (pre-bootstrapped) dataset. For each instance where an alternative question–answer pair from the same context but with a different attribute combination was available, this pair was used as the rejected output, covering approximately 78% of the training data. For the remaining instances, no alternative rejected output was available; in these cases, the rejected sample coincided with the chosen one, and the ORPO objective reduced to its SFT component (the first part of Equation 1). The results (Table 4) show that ORPO yields lower performance than ORSO.

**Evaluating Question Quality.** The human evaluation revealed only a slight difference in the quality of questions generated by the two methods, with

an average overall quality of 4.34 for ORSO, compared with 4.24 for SteerLM. A potential source of bias emerges in the LLM-as-a-Judge evaluations, which favored SteerLM (4.52 versus 4.31 in overall quality), despite the human annotation showing similar results for the two methods. Moreover, it is important to note that the models differ substantially in their training objectives: ORSO was not explicitly optimized for maximizing perceived output quality, but rather for ensuring stronger attribute conformity. This distinction may partly explain the discrepancy between automated and human judgments, reiterating the need for caution when relying on LLM-based evaluators. Nevertheless, the resulting quality metrics are consistently high, accurately reflecting the control attributes.

## 5 Conclusions and Future Work

This study introduces a novel method for training models for attribute-controlled question generation using a combination of human-curated and synthetic data. Our ORSO approach systematically surpassed a strong baseline, SteerLM, when generating question–answer pairs that adhere to specified attributes (i.e., Focus and Coverage). Our experimental protocol evaluated both exhaustive attribute combinations and plausible, expert-validated configurations to identify differences in attribute sensitivity and fidelity of generation. ORSO consistently exhibited high attribute sensitivity and robustness to prompt variance, validating the effectiveness of the proposed supervised alignment mechanism. These findings suggest that explicit attribute-aware training can significantly enhance controllable generation models.

For future work, we plan to explore whether our method can be effectively extended to settings where attribute annotations are synthetically generated, thereby reducing reliance on costly human-curated dataset labels. This direction may enable scalable and domain-adaptive control in question generation models while maintaining alignment with specified content attributes.

## Limitations

The proposed ORSO framework significantly enhances attribute awareness in question generation; however, certain aspects may impact its broader applicability.

ORSO, as well as the evaluated baseline, depends on the availability of an accurate attribute classifier. This classifier plays a central role in both evaluation and generation filtering. In settings where such a classifier is not sufficiently accurate or not available, the control signals used to assess or guide generation may be less reliable. While this is a general requirement for current approaches in attribute-controlled generation, it is worth noting that future work may benefit from exploring approaches that reduce dependence on external classifiers.

Moreover, generative models are inherently inclined to produce outputs even under infeasible or contradictory attribute combinations. To address this, we introduced a thresholding mechanism based on generation confidence, suppressing outputs that are unlikely to satisfy the given constraints. Although this solution proved adequate on the dataset used in our experiments, the threshold may not transfer directly to other instances. Additional experimentation is needed to study the possibility of generalizable strategies or feasibility classifiers.

These factors do not limit the core contributions of ORSO but suggest directions for improving robustness and deployment readiness in more diverse environments.

## Acknowledgments

The research reported here was supported by the project "Romanian Hub for Artificial Intelligence - HRIA", Smart Growth, Digitization and Financial Instruments Program, 2021–2027, MySMIS no. 351416, by the Institute of Education Sciences, U.S. Department of Education, through Grant R305T240035 to Arizona State University, and the grant of the Academy of Romanian Scientists, AOSR-TEAMS-IV Edition 2025-2026 "Digital Transformation in Science". The opinions expressed are those of the authors and do not represent the views of the Institute or the U.S. Department of Education.

## References

Yi Dong, Zhilin Wang, Makesh Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. 2023. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11275–11288.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical

paradigm to understand learning from human preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, and 1 others. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1437–1454.

Kilem L Gwet. 2014. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189.

Jaehun Jung, Peter West, Liwei Jiang, Faeze Brahman, Ximing Lu, Jillian Fisher, Taylor Sorensen, and Yejin Choi. 2024. Impossible distillation for paraphrasing and summarization: How to make high-quality lemonade out of small, low-quality model. 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 4439–4454, Mexico City, Mexico. Association for Computational Linguistics.

Kunze Li and Yu Zhang. 2024. Planning first, question second: An llm-guided method for controllable question generation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4715–4729.

Chaoqun Liu, Qin Chao, Wenxuan Zhang, Xiaobao Wu, Boyang Li, Luu Anh Tuan, and Lidong Bing. 2025. Zero-to-strong generalization: Eliciting strong capabilities of large language models iteratively without gold labels. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3716–3731.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. 2024. Filtered direct preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22729–22770.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Lifu Tu, Semih Yavuz, Jin Qu, Jiacheng Xu, Rui Meng, Caiming Xiong, and Yingbo Zhou. 2024. Unlocking anticipatory text generation: A constrained approach for large language models decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15532–15548.

Tianduo Wang, Shichen Li, and Wei Lu. 2024. Self-training with direct preference optimization improves chain-of-thought reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11917–11928.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, and 1 others. 2022. Fantastic questions and where to find them: Fairytaleqa–an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460.

## A Appendix: Generated Questions Examples

Figure 6 illustrates an example input text along with the corresponding generations produced by SteerLM and our method, ORSO. Only generations whose negative log-likelihood falls below a specified threshold are retained for analysis. The results demonstrate that ORSO more effectively filters out invalid or incompatible attribute combinations. In contrast, SteerLM lacks this discriminative capability, due to limitations in how it attends to attribute information. Consequently, SteerLM exhibits high

confidence even in generations that cannot satisfy the intended attribute constraints. ORSO, by design, incorporates attribute discrimination during training, enabling it to generate outputs that align more accurately with the specified attribute or filter out invalid attribute combinations.

## B Appendix: Quality Evaluation

In the human evaluation stage, four raters, all fluent in English and holding a completed Master's degree, rated a total of 100 questions for each method: ORSO (our approach) and SteerLM (the current state of the art). All annotators independently rated the same subset of 10 questions per method to estimate inter-rater agreement, while the remaining 45 questions per rater were randomly sampled and mixed across methods. In total, this yielded 200 annotated questions (100 for each model). Details regarding the annotation prompts, rating rubrics, and instructions are available in the project's code repository.

Inter-rater agreement was computed for each scoring rubric using the Weighted Brennan–Prediger free-marginal kappa with quadratic weights (Gwet, 2014). This coefficient was selected because it accounts for chance agreement without requiring fixed marginal distributions, and the quadratic weighting penalizes larger discrepancies between raters more strongly than smaller ones, making it appropriate for ordinal rating scales.

| Criteria | Avg. Agrmnt. | Interpretation |
|---|---|---|
| Clarity | 0.83 | Excellent |
| Grammar | 0.98 | Excellent |
| Relevance | 0.35 | Fair |
| Educational value | 0.53 | Moderate |
| Overall quality | 0.63 | Substantial |

Table 5: Raters' agreement for each rubric.

For the LLM-as-a-Judge stage, we evaluated all available questions. Tables 6 and 7 present the average of scores for each rubric, in the ground-truth attribute configurations setup.

## C Appendix: Hyperparameter Setup

The hyperparameters and models used are as follows:

- **Attribute Prediction Model:** ModernBERT-large[1] (Warner et al., 2024); Training: 1 epoch, batch size 64, constant learning rate 1e-5, AdamW optimizer;
- **LLM for attribute-controllable question generation:** Llama-3.2-1B-Instruct[2] (Grattafiori et al., 2024); Training: 1+1 epochs, batch size 64, constant learning rate 5e-6, AdamW-8bit optimizer, ORSO $\lambda = 0.3$; Inference: Regex guided decoding, top-p 0.9, min-p 0.2, 5 samples per prompt;
- **Embedding model:** Sentence Transformers MiniLM-L6-V2[3] (Wang et al., 2020);
- **Hardware:** 1 x NVIDIA A100 80GB.

| Criteria | SteerLM | ORSO |
|---|---|---|
| Clarity | 4.64 | 4.61 |
| Grammar | 4.84 | 4.84 |
| Relevance | 4.49 | 4.34 |
| Educational value | 4.14 | 4.13 |
| Overall quality | 4.28 | 4.34 |

Table 6: Human raters' results for question quality.

| Criteria | SteerLM | ORSO |
|---|---|---|
| Clarity | 4.48 | 4.30 |
| Grammar | 4.83 | 4.77 |
| Relevance | 4.68 | 4.44 |
| Educational value | 3.88 | 3.70 |
| Overall quality | 4.52 | 4.31 |

Table 7: LLM-as-a-Judge results for question quality.

[1] https://huggingface.co/answerdotai/ModernBERT-large

[2] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Nothing would appease the anger of the skillful fisher, for he had a bad disposition, and had always hated his brother because of his virtues, and now with the excuse of the lost fishing hook he planned to kill him and to usurp his place as ruler of Japan. The happy hunter knew all this full well, but he could say nothing, for being the younger he owed his elder brother obedience; so he returned to the seashore and once more began to look for the missing hook. He was much cast down, for he had lost all hope of ever finding his brother's hook now. While he stood on the beach, lost in perplexity and wondering what he had best do next, an old man suddenly appeared carrying a stick in his hand. The happy hunter afterwards remembered that he did not see from whence the old man came, neither did he know how he was there - he happened to look up and saw the old man coming towards him. As soon as the crocodile landed him, the happy hunter hastened to tell the skillful fisher of his safe return. He then gave him back the fishing hook which had been found in the mouth of the tai and which had been the cause of so much trouble between them. He earnestly begged his brother's forgiveness, telling him all that had happened to him in the sea king's palace and what wonderful adventures had led to the finding of the hook.

| **SteerLM** | | | |
|---|---|---|---|
| **Question** | **Answer** | **Focus** | **Coverage** |
| what did the old man do when the crocodile landed him? | he hastened to tell the skillful fisher of his safe return | action | local |
| why did the skillful fisher have a bad disposition? | he had always hated his brother because of his virtues | causal relationship | summary |
| who appeared carrying a stick in his hand? | an old man | character | local |
| who was the happy hunter's brother? | the skillful fisher | character | summary |
| how did the happy hunter feel when he lost the fishing hook? | much cast down | feeling | local |
| how did the happy hunter feel after the old man saved him? | relieved | feeling | summary |
| what happened after the happy hunter returned to the seashore? | he was much cast down , for he had lost all hope of ever finding his brother 's hook now | outcome resolution | local |
| what will the happy hunter do when he returns to the seashore? | begin to look for the missing hook | prediction | local |
| where did the old man appear from? | carrying a stick in his hand | setting | local |
| where did the old man come from? | he happened to look up and saw the old man coming towards him | setting | summary |
| **ORSO** | | | |
| **Question** | **Answer** | **Focus** | **Coverage** |
| what did the skillful fisher plan to do? | kill his brother and to usurp his place as ruler of japan | action | local |
| why did the fisher have a bad disposition? | he had always hated his brother because of his virtues | causal relationship | local |
| why did the skillful fisher hate his brother? | because of his virtues | causal relationship | summary |
| who was the happy hunter's brother? | the skillful fisher | character | summary |
| how did the happy hunter feel when he was cast down? | sad | feeling | summary |
| what happened after the fisher returned to the seashore? | the happy hunter remembered that he did not see from whence the old man came , neither did he know how he was there - he happened to look up and saw the old man coming towards him | outcome resolution | summary |
| where did the old man come from? | he came from the seashore | setting | local |
| where did the fisher return to? | the seashore | setting | summary |

Figure 6: Generated questions examples.