User-Preference-Aligned Automatic Question Generation

Nicholas X. Wang
Stellar Learning Technologies
Santa Clara, CA
nicholas@stellarlearning.app

Aggelos K. Katsaggelos
Northwestern University
Evanston, IL
a-katsaggelos@northwestern.edu

***Abstract*— We explore a lightweight preference-alignment framework for Automatic Question Generation (AQG) that enables rapid personalization from a very small amount of feedback. Starting from an instruction-tuned question generator, we collect a short "preference diagnostic" dataset consisting of pairwise comparisons that capture user expectations such as clarity, usefulness, appropriate difficulty, and stylistic presentation. We then apply a simple two-stage training process: (1) supervised fine-tuning (SFT) on the preferred outputs as a warm start, followed by (2) Direct Preference Optimization (DPO) on the same pairwise data to sharpen the model's distinction between preferred and rejected generations, without training an explicit reward model. To evaluate this approach in a controlled and reproducible setting, we instantiate the preference signal with a concrete target style inspired by standardized exam and textbook conventions as a proxy preference persona. We compare three systems (the base generator, SFT-only adaptation, and SFT+DPO) using a rubric-based evaluation protocol that measures pedagogically relevant dimensions, including difficulty calibration, conceptual breadth, contextual dependence when applicable, and option/distractor quality for multiple-choice formats. Across diverse prompts and topics, SFT delivers clear gains over the base model, and DPO adds substantial improvements beyond SFT under the same small feedback budget. Overall, the DPO-tuned model outperforms the reference base solution by up to 13%. These results suggest that pairwise preference optimization offers a practical and sample-efficient path toward controllable, user-aligned AQG, supporting applications such as matching standardized exam formatting and aligning with teachers' styles while remaining extensible to other curricula and users' preference profiles.**

Keywords*— *User Preference, Automatic Question Generation (AQG), Large Language Models (LLMs), Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), Educational AI

I. Introduction

Automatic Question Generation (AQG) is increasingly used to support intuitive learning workflows, such as practice, review, formative assessment, and large-scale content authoring [1–13]. The emergence of instruction-tuned large language models (LLMs) [14-18] has made it practical to generate questions on demand across a wide range of subjects and curricula. However, high-quality educational questions are rarely "one-size-fits-all". Teachers, students, tutors, and content teams often have distinct, experience-driven preferences for what makes a question valuable [19-20]: wording, tone, the level of conceptual integration, targeted difficulty, the construction of multiple-choice distractors, and compliance with specific formats, rubrics, or assessment formats.

This preference sensitivity matters in at least two common settings. First, learners frequently benefit from practice materials that match the formats of standardized exams (e.g., AP, SAT, IB, etc.), where question structure, distractor design, and phrasing conventions shape both what is assessed and how students prepare. Second, many students study more effectively when practice questions resemble their teacher's style, such as, emphasizing certain wording patterns, question types, or expectations around evidence and reasoning. Beyond these format-driven needs, AQG systems also face a broader personalization challenge: even within the same curriculum, users may prefer different levels of hints, conceptual breadth, or stylistic presentation. In practice, a model that produces technically correct questions may still be rejected because it does not match the user's preferred assessment intent or style.

Aligning AQG to user preferences is challenging because preference signals are expensive and scarce. Collecting large-scale human annotations or running extensive interactive studies is often unrealistic for classroom and product settings. Meanwhile, purely supervised fine-tuning (SFT) from a handful of "good examples" can be brittle: it encourages imitation of preferred outputs, but provides limited information about what should be avoided, and may fail to produce consistent improvements under extremely small data budgets. These constraints motivate a practical question: can we align an AQG model to user preferences using only a lightweight amount of feedback, in a way that is simple, reproducible, and effective?

In this work, we explore a minimal-feedback preference-alignment procedure for AQG based on pairwise comparisons. We introduce a short "preference diagnostic" procedure in which a user provides a small number of binary choices between potential questions (e.g., preferred vs. non-preferred pairs). These pairwise judgments directly encode the user's implicit criteria, such as clarity, pedagogical usefulness, appropriate difficulty, and stylistic fit, without requiring the user to write questions from scratch or assign absolute scores. We then adapt the generator in two stages: (1) a supervised fine-tuning warm start on the preferred outputs to stabilize generation toward the target preference profile, followed by (2) Direct Preference Optimization (DPO) [21] using the same pairwise data to sharpen the model's separation between preferred and non-preferred generations, avoiding reward-model training while leveraging contrastive feedback that SFT alone does not cover.

979-8-3315-7111-5/26/$31.00 ©2026 IEEE

We evaluate this approach by comparing three systems: a base instruction-tuned AQG generator, an SFT-only model trained on the preferred outputs, and an SFT+DPO model trained with the same small set of pairwise preferences. To keep the study controlled and reproducible, we instantiate the preference signal using a concrete format and style target inspired by standardized exam and textbook conventions as a proxy preference persona, and assess outputs using a rubric-based protocol that captures pedagogically relevant qualities such as difficulty preferred, conceptual breadth, and option/distractor quality for multiple-choice questions. Across diverse prompts and topics, we consistently observe that SFT improves over the base model and that DPO yields further significant gains beyond SFT, even under an extremely limited feedback budget.

While SFT and DPO are established alignment techniques, our focus is on their sample-efficient use as a short diagnostic mechanism for AQG and on quantifying the marginal benefit of preference optimization beyond SFT in a very small feedback regime. Concretely, we (1) study lightweight preference diagnostic for AQG via a small number of pairwise comparisons, (2) instantiate a practical training procedure (SFT warm start followed by DPO refinement) to adapt an instruction-tuned generator to a target preference profile without reward-model training, and (3) provide controlled experimental evidence that this procedure produces consistent improvements over both a non-personalized baseline and SFT-only adaptation. Overall, our findings suggest that pairwise preference optimization offers a practical path toward rapid personalization of AQG, supporting applications such as standardized-exam format matching and teacher-style alignment while remaining extensible to other preference personas, curricula, and alternative sources of preference signals.

II. Related works

Automatic Question Generation

AQG has a long history in educational NLP, spanning early rule-based and template-driven systems to neural sequence-to-sequence approaches that generate questions from passages, summaries, or knowledge representations. With the adoption of pretrained language models, AQG systems have improved fluency and coverage, enabling question generation conditioned on text, concepts, learning objectives, or skill tags. More recently, instruction-tuned LLMs have made zero-shot and few-shot AQG practical for real deployments, including multiple-choice generation with distractors and rubric-constrained formulations. Despite these advances, LLM-based AQG often struggles with controllability: generated questions may drift in style, difficulty, or assessment intent even when prompts specify desired properties.

Controllability, Style, and Personalization

A central challenge in educational content generation is aligning outputs to a particular audience or assessment format. Prior work has explored conditioning AQG on attributes such as difficulty, Bloom's taxonomy level, question type, or domain concepts, as well as controlling style through prompts, exemplars, or structured constraints. In practice, "style" can reflect standardized-exam conventions (e.g., predictable stems and distractor patterns), teacher-specific preferences, or product guidelines (clarity, concision, scaffolding). However, these approaches typically rely on substantial labeled data, manually curated templates, or repeated prompt engineering. Our work targets a complementary method: rapid preference alignment from a tiny feedback budget, where the goal is not to learn a universal notion of "good questions," but to adapt generation to a particular preference profile.

Preference Learning for Language Models

Preference-based alignment has become a widely adopted strategy for shaping LLM behavior beyond supervised learning. Reinforcement Learning from Human Feedback (RLHF) commonly uses a learned reward model trained on pairwise comparisons, followed by policy optimization. While effective, this pipeline can be complex and data-hungry, and it introduces additional training instability through reward modeling and reinforcement learning. Direct Preference Optimization (DPO) and related methods provide a simpler alternative by directly optimizing the model from preference pairs without training an explicit reward model. DPO-style training has been applied broadly to improve helpfulness, safety, or instruction following, and is especially attractive when feedback data is limited and engineering simplicity matters. Our work adapts this paradigm to AQG and emphasizes a practical "warm start + preference refinement" procedure under extremely small feedback budgets.

Learning from Small Feedback and Rapid Adaptation

Several lines of work study adaptation from small datasets, including parameter-efficient fine-tuning, few-shot instruction tuning, and preference optimization with limited comparisons. In educational settings, collecting large-scale preference annotations is often impractical, motivating approaches that minimize user burden. Our design aligns with this goal: a short diagnostic interaction (e.g., 25 pairwise judgments) yields a compact preference dataset that can steer the generator toward a target preference profile. We view this as a proof-of-concept for rapid personalization rather than a replacement for large-scale alignment.

Quality Evaluation and LLM-as-a-Judge

Evaluating AQG remains challenging because quality is multidimensional (faithfulness/answerability, conceptual coverage, difficulty calibration, distractor plausibility, and style compliance). Traditional evaluations combine automatic proxies with human judgments, but human evaluation is costly and often underpowered. Recent work has explored using LLMs as evaluators under carefully defined rubrics to obtain scalable, consistent assessments. While LLM-based judging can be sensitive to prompt phrasing and may introduce bias, it is increasingly used for comparative evaluation, particularly when combined with clear scoring criteria and paired experimental design. In our experiments, we adopt a rubric-based evaluation protocol to compare base, SFT-only, and

SFT+DPO models, focusing on relative improvements under a fixed and reproducible scoring setup.

Positioning of This Work

In contrast to prior AQG work that primarily emphasizes better generation architectures or large supervised datasets, we focus on preference alignment with minimal feedback, leveraging pairwise comparisons and DPO to achieve controllable AQG. Unlike full RLHF pipelines, our approach avoids reward model training and is designed to be lightweight enough for realistic diagnostic settings, while still supporting applications such as standardized-format practice generation and teacher-style matching.

III. Problem Formulation

In this section, we define the task of preference-aligned Automatic Question Generation (AQG) and describe how lightweight user feedback is represented and used for training.

Task Definition

Let $x \in X$ denote the input provided to the AQG system. Depending on the application, x can be a topic, a learning objective, a short passage (stimulus), or a combination of these along with constraints such as grade level or question format. Let $q \in Q$ denote a generated question. Our goal is to learn a model that produces questions that better match a user's preferences, such as clarity, pedagogical usefulness, appropriate difficulty, and stylistic fit.

We start from an instruction-tuned base generator M_0. For any input x, the model defines a likelihood $M(q|x)$, which can be interpreted as how likely the model is to generate question q given input x. Higher values correspond to questions the model is more inclined to generate.

Lightweight Pairwise Preference Feedback

Instead of requiring users to write questions or assign absolute scores, we collect feedback using pairwise comparisons. For a given input x, we show the user two potential questions and ask which one is preferred. This produces a preferred question q^+ and a non-preferred question q^- forming a labeled triple: (x, q^+, q^-).

After a short diagnostic interaction, we obtain a preference dataset: $D = \bigcup_{i=1}^{N}\{(x_i,\ q_i^+, q_i^-)\}$, where N is small to reflect a realistic, low-burden feedback budget. For controlled testing, the "user" preference can also be instantiated by a fixed preference persona (e.g., a proxy that favors standardized-exam-like question style), while the learning and evaluation protocol remain the same.

Supervised Warm Start (SFT)

We first warm start the model using supervised fine-tuning (SFT) on the preferred questions only. From each preference triple, we keep the preferred output as a supervised target and form: $D_{sft} = \bigcup_{i=1}^{N}\{(x,\ q_i^+)\}$. We train an SFT model M_{sft} by maximizing the log-likelihood of the preferred questions:

$$\max \sum_{i=1}^{N} \log M(q_i^+|x_i). \quad (1)$$

Intuitively, this step teaches the model to imitate the user-preferred style and stabilizes generation in the desired direction, which is especially helpful when only a handful of feedback examples are available.

Direct Preference Optimization (DPO)

SFT alone does not explicitly encode what the user dislikes. To better leverage each feedback pair (q^+, q^-)., we further apply Direct Preference Optimization (DPO), which trains the model to prefer q^+ over q^- for the same input x.

For any triple (x, q^+, q^-), we define a preference gap:

$$g(x, q^+, q^-) = \log M(q^+|x) - \log M(q^-|x). \quad (2)$$

A larger gap means the model more strongly favors the preferred question.

To prevent overly large updates, DPO compares the model to a fixed reference model M_{ref}. In our setting, we set $M_{ref} = M_{sft}$. We then define the improvement in preference gap relative to the reference:

$$d(x, q^+, q^-) = [\log M(q^+|x) - \log M(q^-|x)] - [\log M_{ref}(q^+|x) - \log M_{ref}(q^-|x)]. \quad (3)$$

Finally, DPO maximizes the probability that the preferred output "wins" using a logistic objective:

$$\max \sum_{i=1}^{N} \log \sigma(\beta \cdot d(x, q_i^+, q_i^-)), \quad (4)$$

where $\sigma(\cdot)$ is the sigmoid function and $\beta > 0$ controls how strongly the model is pushed away from the reference.

Goal

Given a small preference dataset D, our overall objective is to produce a final model M_{dpo} that, for typical inputs x, generates questions that are more likely to satisfy the user's preference relation. The two-stage procedure, SFT warm-start followed by DPO refinement, leverages both positive examples (what the user prefers) and contrastive signals (what the user rejects), enabling effective preference alignment under extremely limited feedback.

IV. Solution

We are able to create an end-to-end training pipeline for user-preference-aligned AQG despite being under a small feedback budget. The overall workflow consists of (i) generating potential questions from a base instruction-tuned model, (ii) collecting a lightweight set of pairwise preferences during a diagnostic, (iii) warm-starting the generator with supervised fine-tuning (SFT) on preferred outputs, and (iv) refining alignment using Direct Preference Optimization (DPO) on the same preference pairs.

Given an input x (topic, learning objective, or stimulus with constraints), our goal is to learn a generator that produces questions aligned with a target user's preferences. We assume only a small number of preference comparisons are available.

We compare three models throughout:

- Base: the original instruction-tuned AQG model M_0.
- SFT: a warm-start model M_{sft} fine-tuned using preferred outputs only.

- SFT+DPO: the final user-preference-aligned model M_{dpo} refined by DPO using pairwise preferences.

Potential Question Generation and Preference Diagnostic

For each diagnostic query, we first sample potential questions from the base model M_0. In the simplest instantiation, we use the same input x to generate multiple potential questions $\{q^{(1)}, \ldots, q^{(K)}\}$ via stochastic decoding (e.g., temperature sampling). We then form a pair (q^a, q^b) and present it to the user as a forced choice. The user selects the preferred option based on implicit criteria such as clarity, usefulness, difficulty appropriateness, and stylistic fit. This produces a labeled triple (x, q^+, q^-). Repeating this procedure yields the diagnostic preference dataset D of size N. In addition to real user diagnostic, the same protocol can be evaluated using a fixed preference persona for controlled experiments, where a deterministic or model-based rule chooses q^+ to reflect a consistent target preference profile. It is important to realize that pairwise comparisons reduce user burden relative to open-ended authoring: users need only choose between alternatives rather than craft questions from scratch, making the process feasible as a lightweight diagnostic interaction.

Supervised Fine-Tuning Warm Start (SFT)

We first adapt the base model using supervised fine-tuning on the preferred outputs. Specifically, we create the SFT dataset $D_{sft} = \bigcup_{i=1}^{N}\{(x_i,\ q_i^+)\}$. from the diagnostic pairs and train M_{sft} by maximizing the conditional log-likelihood of preferred questions as indicated in Eq. (1). This warm start serves two purposes. First, it quickly shifts the model toward the desired preference direction under a tiny dataset. Second, it provides a stable initialization for subsequent preference optimization, which empirically improves robustness compared to applying preference optimization directly from the base model in small-data regimes.

Direct Preference Optimization Refinement (DPO)

While SFT encourages the model to imitate preferred outputs, it does not explicitly use negative signals. We therefore further refine the model using DPO on the full pairwise dataset D. DPO trains the model to assign higher relative likelihood to preferred outputs q^+ than to non-preferred outputs q^- for the same input x, while regularizing updates with a reference model. We set the reference model M_{ref} to a frozen copy of the SFT model M_{sft}. The final model M_{dpo} is trained by maximizing the probability that the preferred output "wins" as indicated in Eq. (4). Intuitively, DPO uses each pair to learn not only what the user prefers, but also what the user rejects. This contrastive signal is especially valuable when N is small, because every feedback instance provides both a positive and a negative training signal.

Implementation Details

We instantiate M_0 using an instruction-tuned LLM for question generation. We train two adapted models:

- M_{sft}: fine-tuned on the N preferred outputs from diagnostic;
- M_{dpo}: DPO-trained starting from M_{sft} using the same N preference pairs.

For evaluation and analysis, we generate questions across a diverse set of inputs and compare the three models (Base, SFT, SFT+DPO) under a paired experimental design, ensuring that each model is tested on the same inputs.

V. Experimental Results

This section describes our experimental setup, including the diagnostic data used for alignment, the compared systems, the evaluation protocol, and the implementation details used to ensure a fair comparison.

Diagnostic Preference Data

To emulate a lightweight personalization workflow, we collect a small preference dataset through a short diagnostic quiz. For each diagnostic query, we generate potential questions for the same input x and present the user with a forced-choice comparison. The user selects the preferred question based on a consistent preference profile (e.g., clarity, usefulness, appropriate difficulty, and stylistic fit). Each interaction yields a labeled triple (x, q^+, q^-), and repeating this process produces: $D = \bigcup_{i=1}^{N}\{(x_i,\ q_i^+, q_i^-)\}$ with $N=25$ in our main experiments.

To keep evaluation controlled and reproducible, we instantiate the preference signal using a fixed preference persona inspired by standardized exam and textbook conventions. This serves as a concrete, testable proxy for user preference while preserving the same data collection and training pipeline that would be used with real users.

Compared Systems

We compare three models under identical test inputs:

- Base: the original instruction-tuned AQG model M_0, with no personalization.
- SFT: M_{sft} obtained by supervised fine-tuning on the N preferred outputs (x_i, q_i^+).
- SFT+DPO: M_{dpo} obtained by continuing training from M_{sft} using DPO on the same N preference pairs $\{(x_i, q_i^+, q_i^-)\}$.

This design isolates the contribution of DPO beyond a straightforward supervised warm start, holding the feedback budget fixed.

Test Inputs and Trial Design

We evaluate the models on a set of diverse prompts that represent typical AQG usage. Each trial samples an input x (e.g., a topic or learning objective) from a predefined pool spanning multiple concepts within a curriculum. In our implementation, we perform $T=500$ trials. For each trial t, we sample an input x_t and generate one question from each model:

$$q_t^{base} \sim M_0(\cdot \mid x_t), \quad (5)$$

$$q_t^{sft} \sim M_{sft}(\cdot \mid x_t), \quad (6)$$

$$q_t^{dpo} \sim M_{dpo}(\cdot \mid x_t). \quad (7)$$

All models are evaluated on the same sampled input x_t, yielding a paired experimental design that improves statistical power and reduces variance due to input difficulty.

Rubric-Based Evaluation Metrics

Because question quality is multi-dimensional, we evaluate each generated question using a rubric covering pedagogically relevant dimensions. In our experiments we score each question on a 1–5 scale for the following criteria:

- **Difficulty:** whether the question is appropriately challenging and non-trivial.
- **Conceptual Breadth:** the number of distinct concepts or reasoning steps required.
- **Stimulus/Context Dependence:** whether the question appropriately relies on the provided context (when context is present) and uses it in a meaningful way.
- **Distractor Quality** (for multiple-choice settings): whether incorrect options are plausible and diagnostically meaningful rather than obviously wrong.

For reporting, we compute both per-metric averages and a composite score defined as the mean of the rubric dimensions for each question. We emphasize relative comparisons between models under the same evaluation protocol. If the generation input does not include an explicit stimulus passage, "stimulus dependence" is interpreted as dependence on the provided prompt/context. When a passage is available, the same rubric naturally extends to measuring whether the question is answerable only by using the passage (and whether the stimulus meaningfully contributes to difficulty).

LLM-Based Judge and Scoring Procedure

To scale evaluation while maintaining consistency, we use an independent LLM judge to score each generated question according to the rubric. The judge is prompted with a fixed scoring template and returns structured JSON scores for each metric. We set the judge temperature to *0* to reduce randomness and enforce deterministic scoring.

This approach is increasingly used for comparative evaluation in text generation when large-scale human evaluation is infeasible. While LLM-as-a-judge can introduce biases, our focus is on paired, relative improvements under a fixed rubric and identical judge settings across all methods.

Implementation Details

We implement the evaluation loop as follows:

- For each of T = 500 trials, sample an input x_t.
- Generate one question from each model with identical decoding settings (e.g., temperature sampling).
- Evaluate each output with the same judge model and rubric prompt.
- Aggregate results across trials to obtain mean scores per metric and composite score for each method.

We also compute running averages across trials to assess stability and reduce sensitivity to any single input. Experiments were conducted across diverse subjects, including U.S. History, Java Programming, and Macroeconomics.

As shown in Fig. 1, we report the equal-weight composite score over 500 evaluation trials for an AQG task in U.S. History. Each point on the curve represents the average composite score computed over the first t trials. The results show a clear, consistent trend: SFT+DPO surpasses SFT by up to 9%, and SFT exceeds the base (reference) model by up to 6% across the evaluation. Overall, SFT+DPO outperforms the base by as much as 13%. Fig. 2 provides a qualitative comparison using one representative example. The base model produces a largely recall-style question (focused on the Stamp Act) with vague framing and distractors that do not strongly require evidence-based reasoning. SFT is a clear improvement: it anchors the question in a primary-source excerpt and shifts the task toward interpreting a specific grievance, increasing authenticity, and encouraging contextual reasoning. DPO performs best by further sharpening the targeted concept, namely, the colonial argument behind "taxation without consent," emphasizing the legitimacy of internal taxation and the role of elected assemblies. It also yields more parallel and historically plausible distractors, reducing ambiguity and test-wise elimination while keeping the stem tightly aligned to the excerpt.

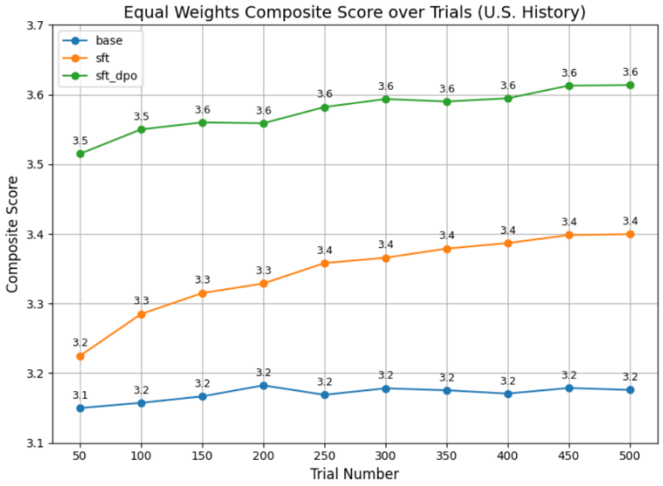

Figure 1. Experimental Results for U.S. History AQG with Equal-Weight Scoring

Base	***SFT Only***	***SFT & DPO***
Which of the following acts passed by the British Parliament was primarily intended to assert the right of the British government to tax the American colonies, leading to widespread protest and contributing to the causes of the American Revolution? A) The Quartering Act of 1765 B) The Stamp Act of 1765 C) The Tea Act of 1773 D) The Intolerable Acts of 1774	*For imposing Taxes on us without our Consent . . .* ***Declaration of Independence, 1776*** The grievance in the excerpt most directly reflected colonial opposition to which of the following? (A) Parliamentary claims of virtual representation (B) Expansion of imperial trade restrictions (C) Limits on westward settlement in the Proclamation of 1763 (D) Increased royal control of colonial legislatures	*For imposing Taxes on us without our Consent . . .* ***Declaration of Independence, 1776*** The grievance in the excerpt most directly reflected which of the following colonial arguments? (A) Only their elected assemblies had the legitimate authority to levy internal taxes (B) The British Parliament should regulate trade but not restrict westward expansion (C) Royal officials in the colonies should be independent of parliamentary oversight (D) Greater imperial military spending in North America required direct colonial representation

Figure 2. Example of U.S. History Questions Generated by Methods in Comparison

The results in Fig. 3 indicate that DPO gains a stronger advantage over SFT when Conceptual Breadth is assigned a higher weight in the composite score. In this setting, the

evaluation favors questions that integrate multiple concepts and penalizes single-fact recall. DPO is particularly effective because it learns from paired preferences, directly contrasting broader, reasoning-oriented questions with narrower alternatives. This contrastive reference helps the model consistently avoid recall-style templates and instead produce stems and distractors that reflect competing concepts or interpretations (e.g., alternative causes, motivations, or consequences), leading to cleaner discrimination and higher breadth scores. By comparison, SFT learns only from preferred examples; while it can imitate the target style, it receives weaker guidance about which narrow patterns to suppress, making its gains less consistent when breadth is emphasized.

Fig. 4 illustrates this effect. In the example, DPO more tightly anchors the correct option to the intended context and refines the stem accordingly, specifically, "the acquisition of overseas territories following the Spanish–American War." This phrasing is broad yet precise, testing multiple relevant historical events (e.g., the Philippines, Puerto Rico, and Guam) without becoming confusing. Meanwhile, the distractors are framed as clearly unrelated events or implausible "developments," making the item cleaner, more discriminative, and less ambiguous.

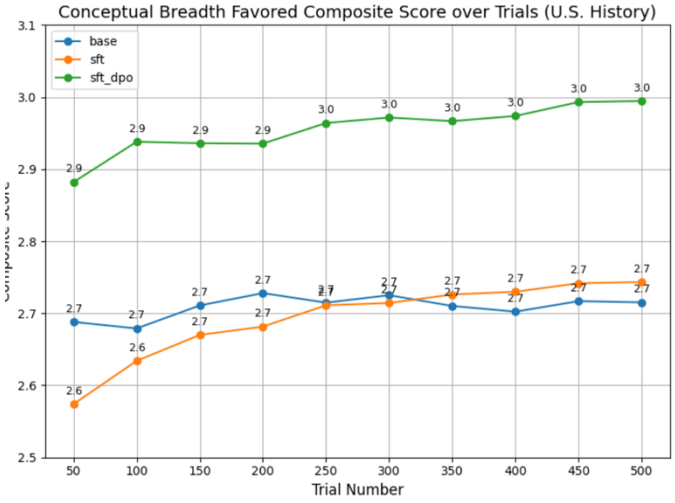

Figure 3. Experimental Results for U.S. History AQG Under Conceptual-Breadth–Weighted Preference Settings

Base	*SFT Only*	*SFT & DPO*
Which of the following best explains a primary motivation for the United States' imperialist actions at the turn of the 20th century? A) The United States sought to establish a global alliance system to counter European powers. B) There was a desire to spread democratic ideals and institutions to other countries. C) The United States aimed to secure new markets and raw materials to fuel industrial growth. D) There was an effort to maintain neutrality and avoid foreign entanglements.	"We hold that the policy known as imperialism . . . is hostile to liberty and tends toward militarism." ***—Platform of the Anti-Imperialist League*, 1899** The argument in the excerpt was most directly a response to which of the following developments? (A) The annexation of territories following the Spanish-American War (B) The purchase of Alaska from Russia (C) The negotiation of the Clayton-Bulwer Treaty (D) The implementation of the Embargo Act of 1807	"We hold that the policy known as imperialism . . . is hostile to liberty and tends toward militarism." ***—Platform of the Anti-Imperialist League*, 1899** The argument in the excerpt most directly responded to which of the following developments? (A) The acquisition of overseas territories following the Spanish-American War (B) The reduction of U.S. naval power through disarmament agreements (C) Congressional rejection of all commercial expansion in Asia (D) A decline in popular support for the idea of American cultural superiority

Figure 4. Example of U.S. History Questions Generated by Methods in Comparison

When we apply the same methods to Java Programming, the performance trend remains consistent. As shown in Fig. 5, DPO demonstrates advantage over the other approaches. The qualitative examples in Fig. 6 further illustrate why. The Base question mainly tests recall (e.g., array bounds and the fact that finally executes), making its distractors easy to rule out. SFT improves the item by emphasizing exception specificity and requiring a brief explanation. DPO goes further by testing precise control-flow reasoning in nested `try/catch/finally` with rethrow: students must trace the print order inside `divide()` `(catch → finally)`, followed by the caller's catch and final finally. This streamlined structure increases discrimination and aligns each option with a distinct execution trace.

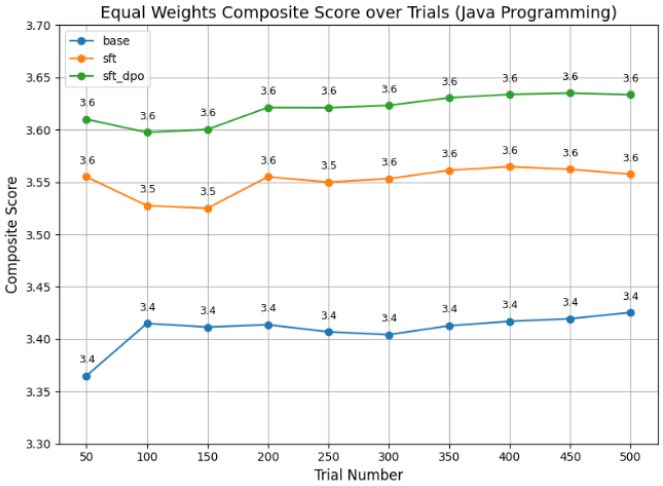

Figure 5. Experimental Results for Java Programming AQG with Equal-Weight Scoring

Base	*SFT Only*	*SFT & DPO*
Consider the following Java code snippet: `public class ExceptionExample { public static void main(String[] args) { try { int[] numbers = {1, 2, 3}; System.out.println(numbers[3]); } catch (ArrayIndexOutOfBoundsException e) { System.out.println("Array index is out of bounds."); } catch (Exception e) { System.out.println("A general exception occurred."); } finally { System.out.println("This block always executes."); } } }`  What will be the output of the above code when executed?  A) `Array index is out of bounds.` `This block always executes.`  B) `A general exception occurred.` `This block always executes.`  C) `This block always executes.`  D) `Array index is out of bounds.`	`public class ExceptionTest { public static void main(String[] args) { try { System.out.println(divide(10, 0)); } catch (ArithmeticException e) { System.out.println("Caught ArithmeticException"); } catch (Exception e) { System.out.println("Caught Exception"); } finally { System.out.println("Finally block"); } } static int divide(int a, int b) { return a / b; } }`  Which statements are printed and why? **(A)** `Caught ArithmeticException, Finally block` because integer division by zero throws ArithmeticException **(B)** `Caught Exception, Finally block` because Exception is more general **(C)** `Finally block only` because exceptions in Java aren't caught **(D)** `Exception at runtime, nothing printed` finally doesn't execute if exception isn't caught	`public class ExceptionTest { public static void main(String[] args) { try { System.out.println(divide(10, 0)); } catch (ArithmeticException e) { System.out.println("Caught: " + e); } finally { System.out.println("Finally block"); } } static int divide(int a, int b) { try { return a / b; } catch (ArithmeticException e) { System.out.println("Divide catch"); throw e; } finally { System.out.println("Divide finally"); } } }`  What is the output? **(A)** `Divide catch, Divide finally, Caught: java.lang.ArithmeticException, Finally block` **(B)** `Divide finally, Caught: java.lang.ArithmeticException, Finally block` **(C)** `Divide finally, Finally block, Uncaught exception terminates` **(D)** `Caught: java.lang.ArithmeticException, Divide finally, Finally block`

Figure 6. Example of Java Programming Questions Generated by Methods in Comparison

Figure 7 presents an additional example from Macroeconomics. The Base item mainly checks a shallow understanding of GDP, with distractors that are easy to eliminate through test-taking heuristics (e.g., used goods and transfer payments). The SFT version improves clarity by grounding the concept in a concrete

household purchase and correctly targeting the consumption component, although some distractors are less parallel (e.g., unemployment or the GDP deflator). The SFT+DPO version is strongest because it assesses a higher-value skill: using nominal GDP growth and the GDP deflator to infer real GDP growth. This requires quantitative reasoning, reduces guessing, and provides tightly structured, mutually exclusive answer choices, resulting in a more discriminative, assessment-grade question.

Base	***SFT Only***	***SFT & DPO***
Which of the following transactions would be included in the calculation of a country's Gross Domestic Product (GDP) for a given year? A) A consumer purchases a used car from a private seller. B) A company manufactures and sells new furniture to a local retailer. C) An individual receives a cash gift from a relative. D) A retiree receives a pension payment from the government.	**A family purchases a new refrigerator manufactured in the United States.** Which of the following will increase as a direct result of this transaction? (A) Consumption component of GDP (B) Government spending component of GDP (C) Net exports in the GDP calculation (D) Structural unemployment rate (E) Real GDP deflator	**A country's nominal GDP increased by 6% in one year, while its GDP deflator rose by 4%.** Based on the information above, what happened to the country's real GDP? (A) Decreased by 2% (B) Increased by 2% (C) Increased by 6% (D) Remained unchanged (E) Increased at the same rate as the GDP deflator

Figure 7. Example of Macroeconomics Questions Generated by Methods in Comparison

VI. Conclusion

This paper explored a lightweight, sample-efficient path to preference-aligned Automatic Question Generation (AQG). We studied a practical "preference diagnostic" setting in which only a small number of pairwise comparisons (e.g., 25) are available, and evaluated a simple two-stage adaptation procedure: supervised fine-tuning (SFT) on preferred outputs followed by Direct Preference Optimization (DPO) on the same preference pairs. Across diverse prompts and topics, we consistently observed that SFT improves over the base instruction-tuned generator and that DPO provides additional, significant gains beyond SFT under the same tiny feedback budget. Qualitative examples further suggest that DPO tends to produce cleaner, more discriminative questions with more parallel and plausible distractors, and its advantage becomes especially pronounced when the evaluation emphasizes higher-level qualities such as conceptual breadth.

While our experiments use a controlled proxy preference persona for reproducibility, the proposed pipeline is designed to generalize to real user diagnostic and other preference profiles, including standardized-exam format matching and teacher-style alignment. Future work will extend this study in three directions: (i) validating the approach with real users and measuring downstream learning outcomes, (ii) expanding to multiple personas and preference dimensions (e.g., scaffolding level, reasoning depth, and question type), and (iii) improving evaluation with stronger grounding checks and complementary human judgments. Overall, our findings support pairwise preference optimization as a practical mechanism for rapid personalization and controllable AQG, offering a simple alternative to more complex reward-modeling and reinforcement-learning pipelines in low-feedback educational settings.

References

[1] N. X. Wang, "GraphGPT: A Self Supervised-Learning for Intuitive, Logical, and Visual Education", in Proc. *IEEE ICNC 2025*, Honolulu, Feb. 2025.

[2] N. X. Wang, A. K. Katsaggelos, Leveraging Interactive Generative AI for Enhancing Intuitive Learning, in Proc. *ACM CHI 2025 workshop on Generative AI and HCI*, Yokohama, April 2025.

[3] N. X. Wang, A. K. Katsaggelos, Scaling Intuitive Education with Multiple LLM-based Agents, in Proc. *ACM CHI 2025 workshop on Augmented Educators and AI*, Yokohama, April 2025.

[4] N. X. Wang, A. K. Katsaggelos, Enhancing Education with Automatic Generation of Engaging Visual Explanations, in Proc. *ACM CHI 2025 workshop on Augmented Educators and AI,* Yokohama, April 2025.

[5] N. X. Wang, N. V. Parpia, A. D. Parikh, A. K. Katsaggelos, "Automatic Question Generation for Intuitive Learning Utilizing Causal Graph Guided Chain of Thought Reasoning", in Proc. *IEEE MIPR 2025*, San Jose, August 2025.

[6] N. X. Wang, A. K. Katsaggelos, Hallucination-Free Automatic Question & Answer Generation for Intuitive Learning, in Proc. *IEEE ICIP 2025 workshop on Generative AI for World Simulations and Communications,* Anchorage, September 2025.

[7] N. X. Wang, A. K. Katsaggelos, Hallucination-Free Causal Graph Guided AI Framework for Intuitive Question and Answer Generation*,* to appear in *Intl. J. on Multimedia Data Engineering and Management,* 2026.

[8] N. Scaria, S. Dharani Chenna, and D. Subramani, July. Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation, In *International Conference on Artificial Intelligence in Education* (pp. 165-179). Cham: Springer Nature Switzerland, 2024.

[9] J. Doughty, Z. Wan, A. Bompelli, J. Qayum, T. Wang, J. Zhang, Y. Zheng, A. Doyle, P. Sridhar, A. Agarwal, and C. Bogart, A comparative study of AI-generated (GPT-4) and human-crafted MCQs in programming education. In *Proceedings of the 26th Australasian Computing Education Conference* (pp. 114-123), 2024.

[10] N. Mulla and P. Gharpure, Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, 12(1), pp.1-32, 2023.

[11] H. A. Nguyen, S. Bhat, S. Moore, N. Bier, and J. Stamper, Towards generalized methods for automatic question generation in educational domains. In *European conference on technology enhanced learning* (pp. 272-284). Cham: Springer International Publishing, 2022.

[12] "Stellar", *Stellar Learning Technologies*, https://stellarlearning.app.

[13] "Khanmigo", Khan Academy, www.khanmigo.ai/.

[14] "ChatGPT", *OpenAI*, https://chat.openai.com/.

[15] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, et al., "The rise and potential of large language model based agents: A survey", *arXiv preprint arXiv:2309.07864*, 2023.

[16] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R.Wang, Y . Yang, Z. Tu, and S. Shi, "Encouraging divergent thinking in large language models through multi- agent debate" , arXiv preprint arXiv:2305.19118, 2023.

[17] Y. Du, S. Li, A. Torralba, J. B Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate", arXiv preprint arXiv:2305.14325, 2023.

[18] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, and D. Zhou, Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, *35*, pp.24824-24837, 2022.

[19] J. Woodrow, Sanmi Koyejo, and Chris Piech. "Improving Generative AI Student Feedback: Direct Preference Optimization with Teachers in the Loop." *International Educational Data Mining Society,* 2025.

[20] Y. Lee, Kim, S. and Jo, Y., 2025. Generating plausible distractors for multiple-choice questions via student choice prediction. *arXiv preprint arXiv:2501.13125*.

[21] R. Rafailov, Sharma, A., Mitchell, E., Manning, C.D., Ermon, S. and Finn, C., 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, *36*, pp.53728-53741.