AutoSynth: Automated Workflow Optimization for High-Quality Synthetic Dataset Generation via Monte Carlo Tree Search

Anonymous ACL submission

Abstract

Supervised fine-tuning (SFT) of large language models (LLMs) for specialized tasks requires high-quality training datasets, yet manual curation remains prohibitively expensive and time-consuming. While synthetic data generation offers a scalable alternative, its effectiveness depends on carefully designed generation workflows—complex, multi-stage processes integrating prompt engineering, programmatic logic, and model orchestration. Existing automated workflow optimization methods fail to address the cold start problem, as they require labeled datasets for reward modeling, particularly problematic for subjective, open-ended tasks where objective ground truth is unavailable. We introduce AutoSynth, a framework that automates workflow discovery and optimization without reference datasets by reframing the problem as a Monte Carlo Tree Search guided by a novel dataset-free hybrid reward signal. This signal enables meta-learning through two components: an LLM-as-judge evaluating sample quality against dynamically regenerated task-specific metrics, and another LLM-as-judge assessing workflow code and prompt quality. Experiments on subjective educational tasks reveal that while expert-designed workflows achieve higher human preference rates (96-99% win rates compared to AutoSynth's 40-51%), models trained on AutoSynth-generated data dramatically outperform baseline models (40-51% vs. 2-5% win rates) and surpass expert-designed workflows on certain metrics, suggesting the discovery of quality dimensions not fully captured by human evaluators. These results are achieved while reducing human effort from 5-7 hours to just 30 minutes (>90% reduction). AutoSynth addresses the critical cold start challenge in data-centric AI, providing a scalable, cost-effective method for developing specialized LLMs for subjective tasks lacking objective ground truth.

1 Introduction

The rapid advancement of large language models (LLMs) has ushered in a new era where model specialization through supervised fine-tuning (SFT) is the dominant paradigm for domain-specific applications (Ouyang et al., 2022; Zhou et al., 2023). However, SFT fundamentally depends on high-quality, task-aligned training data (Xu et al., 2024). Manually curating such datasets requires substantial domain expertise and significant resources, creating a critical bottleneck in the democratization of specialized AI (Levin et al., 2024).

Synthetic data generation has emerged as a promising solution (Wang et al., 2022; Honovich et al., 2022). Yet, the quality of synthetic data critically depends on the design of the *generation workflow*—a complex, multi-stage process integrating prompt engineering, programmatic logic, and model orchestration. A poorly designed workflow can yield data lacking diversity (Chen et al., 2024) or failing to align with task requirements (Long et al., 2024).

Recent frameworks like AFlow (Zhang et al., 2024) and AutoFlow (Li et al., 2024) reformulate workflow design as a search problem. However, they share a fundamental limitation: their optimization presupposes a pre-existing labeled dataset to provide a reward signal. This dependency renders them inapplicable to creating datasets for novel tasks where no reference data exists.

This cold start problem is particularly severe for subjective, open-ended tasks (e.g., educational content, creative writing), where "correct" answers are ill-defined and no validation benchmarks exist. We address this gap by introducing **AutoSynth**, the first framework to automate synthetic data workflow optimization *without requiring any reference datasets*. Our key insight is to replace dataset-dependent rewards with a novel *dataset-*

free hybrid reward signal computed entirely by LLMs, combining:

1. **Sample Quality Assessment**: An evaluator LLM assesses generated samples against task-specific quality metrics that are dynamically regenerated at each iteration.

2. **Workflow Quality Assessment**: An optimizer LLM evaluates the workflow's code structure, prompt design, and logical coherence.

We validate AutoSynth on two challenging educational tasks. Our experiments reveal that while expert-designed workflows achieve higher human preference rates (96–99%), models trained on AutoSynth data dramatically outperform baseline models (40–51% vs. 2–5% win rates) and surpass expert-designed workflows on certain metrics. These results are achieved while reducing human effort by over 90%.

Our contributions are: (1) the first workflow optimization framework for synthetic data that operates without reference datasets; (2) a dataset-free hybrid reward signal combining sample-level and workflow-level assessment; (3) a dynamic metric regeneration strategy enabling meta-learning; and (4) strong empirical evidence validating the framework's effectiveness.

2 Related Work

2.1 Automated Workflow Optimization for LLMs

AFlow (Zhang et al., 2024) pioneered workflow optimization as a Monte Carlo Tree Search (MCTS) problem over code-represented execution graphs. AutoFlow (Li et al., 2024) extends this with reinforcement learning. However, all existing frameworks require *pre-existing labeled validation datasets* to compute reward signals, rendering them inapplicable when no ground-truth data exists. Our work directly addresses this constraint.

2.2 Synthetic Data Generation for SFT

Self-instruct (Wang et al., 2022) established bootstrapping instruction-following datasets by prompting LLMs. Distillation approaches (Gu et al., 2023) leverage teacher models to generate training data. Self-critique methods (Madaan et al., 2023) enable iterative refinement. However, these techniques operate within *static, manually designed pipelines*. The literature identifies persistent challenges: faithfulness (Ji et al., 2023), diversity (Chang et al., 2024), and subjectivity (Yu et al., 2023)—all of which our framework addresses.

2.3 Bridging the Gap

Workflow optimization systems excel at process optimization but cannot define the evaluation metric itself. Synthetic data research focuses on output quality but relies on static pipelines. AutoSynth bridges this gap by enabling simultaneous optimization of the generation process and quality criteria, fundamentally solving the cold start problem through meta-learning that bootstraps both generation capabilities and quality understanding from minimal human guidance.

3 The AutoSynth Framework

Figure 1 presents the architectural overview of AutoSynth.

3.1 Problem Formulation

Given a natural language task description $\mathcal{T}$, our objective is to discover an optimal workflow W^* maximizing a quality function:

$$W^* = \arg\max_{W \in \mathcal{W}} Q(D_W) \tag{1}$$

where D_W is the dataset produced by workflow W. The challenge is that Q is neither predefined nor computable via reference data. We reformulate this as a search problem where Q is dynamically constructed through LLM-based evaluation.

3.2 Workflow Representation and Initialization

A workflow $W = \langle P, C \rangle$ comprises structured prompts P and executable Python code C. AutoSynth begins with human-in-the-loop initialization: (1) an Optimizer LLM (Claude Sonnet 3.7) generates initial workflow W_{init}; (2) W_{init} produces sample data; (3) a human provides feedback; (4) steps repeat for 1-3 iterations until functional baseline W_0 is obtained. This ensures the search begins in a promising region and captures user intentions.

3.3 MCTS-Driven Optimization

The MCTS loop iteratively refines workflows through four stages:

Selection. We identify top-3 workflows by hybrid reward $R(W)$ and sample one according to

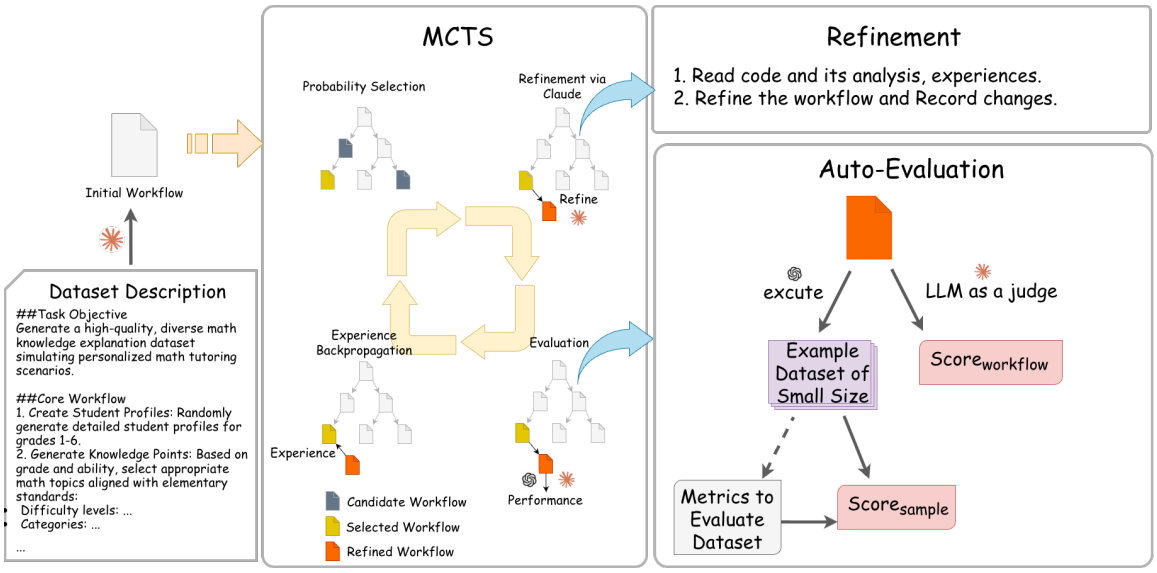

Figure 1: Overview of AutoSynth. **(Left)** Human-in-the-loop initialization generates baseline workflow W_0 through 1-3 iterations; **(Middle & Right)** MCTS-driven optimization refines workflows guided by hybrid reward combining sample quality ($Score_{sample}$) and workflow quality ($Score_{workflow}$).

$p_i = R(W_i)/\sum_{j=1}^{3} R(W_j)$, balancing exploitation and exploration.

Refinement. The Optimizer LLM receives the selected workflow, accumulated experiences, and improvement suggestions, then proposes modifications to produce child node W_{new}.

Evaluation. W_{new} generates 5-10 samples for hybrid reward computation.

Backpropagation. The reward and experience record propagate to ancestors, enriching context for future decisions.

The loop continues until convergence ($\epsilon = 0.05$) or maximum 30 iterations.

3.4 Dataset-Free Hybrid Reward Signal

The hybrid reward combines product and process assessment:

$$R(W) = 0.5 \cdot Score_{sample}(W) + 0.5 \cdot Score_{workflow}(W) \quad (2)$$

Dynamic Sample Quality Score. Instead of fixed metrics, $Score_{sample}$ uses criteria dynamically generated at each iteration. An Evaluator LLM (GPT-5) proposes metrics $M_t = \{m_1, \ldots, m_k\}$ with definitions and exemplars. This dynamic regeneration enables meta-learning—the system refines its conception of quality as workflows improve. The Evaluator scores each sample d_i against each metric m_j:

$$Score_{sample}(W) = \frac{1}{|D_W| \cdot |M_t|} \sum_{i,j} s_{ij} \quad (3)$$

Workflow Quality Score. The Optimizer LLM evaluates W_{new} on code quality (clarity, efficiency, robustness) and prompt quality (clarity, specificity, effectiveness), acting as a regularizer ensuring maintainable implementations.

4 Experiments

4.1 Setup

Tasks. We evaluate on two educational tasks where quality is subjective: (1) **Math Concept Explanation**—generating pedagogically sound explanations; (2) **Interdisciplinary Lesson Plan Generation**—creating comprehensive plans integrating multiple subjects.

Implementation. Claude Sonnet 3.7 serves as Optimizer LLM; GPT-5 as Evaluator LLM. MCTS runs for maximum 30 iterations or until convergence.

Baselines. (1) **Expert-Designed Workflow**: manually designed by a senior researcher (5-7 hours per task); (2) **Base Model**: Qwen-Instruct-32B without fine-tuning.

Fine-Tuning. We generate 1,000 examples per method and perform SFT on Qwen-Instruct-32B (learning rate 2×10^{-5}, batch size 16, 3 epochs).

Evaluation. We use the ELMES framework (Wei et al., 2025) with 10 held-out queries per task.

4.2 Main Results

Table 1 presents results. Key findings:

Table 1: Comparison across educational tasks.

Task	Method	Win%	Score	Effort
Math	AutoSynth	40.5	4.25	30m
	Expert	98.5	4.18	5h
	Base	4.8	3.67	0
Lesson	AutoSynth	51.1	4.82	30m
	Expert	95.5	4.75	7h
	Base	2.0	3.75	0

Table 2: Ablation on reward components.

Config	Task1	Task2	Task3	Task4
Full Hybrid	**3.78**	**4.81**	4.20	4.49
Code-Only	3.40	4.54	**4.36**	**4.62**
Sample-Only	3.41	4.55	4.35	4.46

Substantial Improvement Over Baseline. AutoSynth achieves 40-51% win rates, dramatically outperforming base model (2-5%) with only 30 minutes of human involvement—over 90% reduction from expert workflow engineering.

Superior Metric Performance. AutoSynth achieves higher metric scores than expert-designed workflows (4.25 vs 4.18; 4.82 vs 4.75), revealing divergence between metrical and human evaluation. This suggests AutoSynth discovers quality dimensions not fully captured by human evaluators.

4.3 Ablation Studies

Table 2 shows task-dependent sensitivity to reward components. The full hybrid reward achieves superior performance for most tasks, validating our balanced design. Additional ablations confirm: (1) human-in-the-loop initialization provides 4-5% relative improvement; (2) iterative metric regeneration consistently outperforms static metrics (0.03-0.25 point gains), validating the meta-learning framework.

5 Conclusion

We introduced AutoSynth, the first framework to automate synthetic dataset generation without reference data. By formulating workflow optimization as dataset-free search guided by dynamically evolving metrics, AutoSynth reduces human effort by over 90% while achieving competitive performance.

Our experiments reveal both promise and limitations. AutoSynth substantially outperforms baselines and achieves superior metric scores compared to expert-designed workflows. However, a gap remains in human preference evaluations (40-51% vs 96-99%), indicating room for capturing pedagogical sophistication.

Future directions include enhanced search strategies incorporating domain principles, transfer learning across tasks, and hybrid human-AI collaboration. Beyond education, AutoSynth's paradigm has broad implications for data-centric AI in domains with subjective quality criteria.

Limitations

Our work has several limitations that warrant discussion:

Scope of Evaluation. We evaluated AutoSynth on two educational tasks (math concept explanation and interdisciplinary lesson planning). While these represent challenging subjective domains, the generalizability to other domains such as creative writing, legal document generation, or healthcare applications remains to be validated.

Human Preference Gap. Despite achieving competitive metric scores, AutoSynth-trained models show a significant gap in human preference evaluations (40-51% vs 96-99% for expert-designed workflows). This suggests that our current reward signal may not fully capture all dimensions of quality that human evaluators value, particularly stylistic and pedagogical nuances.

Computational Cost. The MCTS optimization process requires multiple LLM API calls (GPT-5 and Claude Sonnet 3.7), which incurs non-trivial computational costs. We did not conduct detailed cost-benefit analyses comparing AutoSynth to manual workflow design.

Dynamic Metric Stability. While iterative metric regeneration enables meta-learning, it may introduce instability in optimization objectives. Further research is needed to understand when and how metrics should evolve versus remain fixed.

Single Language Focus. All experiments were conducted in Chinese educational contexts. Cross-lingual applicability and performance in other languages were not evaluated.

Ethical Considerations

This work focuses on automating synthetic data generation for educational applications. While our framework reduces human effort, we acknowledge that automated evaluation metrics may not capture all quality dimensions important to educators and students. We recommend that practitioners val-

idate AutoSynth-generated data with domain experts before deployment in real educational settings.

The use of LLMs (GPT-5, Claude Sonnet 3.7) as core components raises considerations about potential biases inherited from these models. Generated educational content should be reviewed for factual accuracy and pedagogical appropriateness before use with students.

References

Allen Chang, Matthew C Fontaine, Serena Booth, Maja J Matarić, and Stefanos Nikolaidis. 2024. Quality-diversity generative sampling for learning with synthetic data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19805–19812.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abdin. 2024. On the diversity of synthetic data and its impact on training large language models. *arXiv preprint arXiv:2410.15226*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Gabriel Levin, Nir Horesh, Yoav Brezinov, and Raanan Meyer. 2024. Performance of chatgpt in medical examinations: A systematic review and a meta-analysis. *BJOG: An International Journal of Obstetrics & Gynaecology*, 131(3).

Zelong Li, Shuyuan Xu, Kai Mei, Wenyue Hua, Balaji Rama, Om Raheja, Hao Wang, He Zhu, and Yongfeng Zhang. 2024. Autoflow: Automated workflow generation for large language model agents. *arXiv preprint arXiv:2407.12821*.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback, 2023. *URL https://arxiv.org/abs/2303.17651*.

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.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Shou'ang Wei, Xinyun Wang, Shuzhen Bi, Jian Chen, Ruijia Li, Bo Jiang, Xin Lin, Min Zhang, Yu Song, BingDong Li, and 1 others. 2025. Elmes: An automated framework for evaluating large language models in educational scenarios. *arXiv preprint arXiv:2507.22947*.

Xinyi Xu, Zhaoxuan Wu, Rui Qiao, Arun Verma, Yao Shu, Jingtan Wang, Xinyuan Niu, Zhenfeng He, Jiangwei Chen, Zijian Zhou, and 1 others. 2024. Data-centric ai in the age of large language models. *arXiv preprint arXiv:2406.14473*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2024. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.

A Example Analysis

To provide a qualitative understanding of model performance, we present detailed comparisons of generated samples from the AutoSynth-trained model, Expert-Designed model, and Base Model across our two main educational tasks.

A.1 Task 1: Math Concept Explanation

Query

> **Student Background**: Third-grade elementary student struggling with mathematics
> **Question**: Please help me review and consolidate my understanding of conversions between millimeters and decimeters

Model Responses

AutoSynth-Trained Model (Translated)
Great! You've taken the initiative to review millimeter and decimeter conversions, which shows you're motivated to improve! What parts do you find unclear? For example, what differences do you see between millimeters and decimeters? How do they convert? Try explaining it.
Let's think together: Have you seen a ruler? It has many small marks—some are centimeters, some are millimeters. Which do you think is smaller? If you measure a pencil's length, would you use millimeters or centimeters? Why?
If I tell you 1 centimeter equals 10 millimeters, can you calculate how many millimeters are in 1 decimeter? Try converting 1 decimeter to centimeters, then to millimeters.
Actually, the relationship between decimeters and centimeters is simple: 1 decimeter equals 10 centimeters. Can you use this method to convert 1 decimeter to centimeters, then to millimeters? How many millimeters do you think are in 1 decimeter? Take your time calculating.
Sometimes students confuse millimeters and centimeters, thinking 1 centimeter equals 100 millimeters—but it's actually 10 millimeters. Do you find this easier to remember? What methods could help you remember this conversion?
Let's practice: If you have a ribbon 2 decimeters long, can you convert it to millimeters? First think about how many centimeters 2 decimeters is, then convert to millimeters. What's your answer?
Here's a challenge: If a book's thickness is 5 millimeters, can you express this in decimeters? Convert 5 millimeters to centimeters first, then to decimeters. What's the result?
Now let's review: What's the conversion pattern between millimeters, centimeters, and decimeters? Can you explain it in your own words? If I give you 70 millimeters, can you convert it to centimeters or decimeters? Try it!
Finally, what was the hardest part of today's lesson? How will you quickly determine what to do when you encounter conversion problems in the future? Feel free to ask if anything is still unclear!

Expert-Designed Model (Translated, Contains Critical Error)
Hello, student!
Today we'll review a very important math concept—**conversions between millimeters and decimeters**! Are you ready? Let's start our adventure!
Millimeters and Decimeters, Our Good Friends
Let's meet our old friends—**millimeters** and **decimeters**.

- **Millimeter (mm):** The smallest marking on our ruler. Look, a sharpened pencil tip is about **1 millimeter** wide. It's so tiny!
- **Decimeter (dm):** Bigger than millimeters but smaller than meters. If you fold an A4 paper in half, the short side is about **1 decimeter**. Much longer than a millimeter, right?

The Magic of Conversion
Now let's learn a super cool magic trick—transforming **millimeters** into **decimeters**, or vice versa!
From Millimeters to Decimeters: 1000 millimeters = 1 decimeter
Imagine we have a very long noodle, a full **1000 millimeters** long. Wow! So long! If we want to express it in decimeters, what would it be?
It's simple: divide **1000 millimeters** by **100**, and you get **1 decimeter**!
So, remember our first magic formula:

$$1000 \text{ millimeters} = 1 \text{ decimeter}$$

[CRITICAL FACTUAL ERROR: This is incorrect! The correct conversion is 1 decimeter = 100 millimeters, NOT 1000 millimeters. This represents a 10x error that would severely mislead students.]
From Decimeters to Millimeters: 1 decimeter = 1000 millimeters

This is even simpler! Just remember **1 decimeter** equals **1000 millimeters**.
[ERROR REPEATED: Again stating the incorrect 1000mm conversion.]
[Additional pedagogically rich content with examples from daily life omitted for brevity...]

Base Model (Qwen-32B-Instruct, Translated)
Alright, today we'll review millimeter and decimeter conversions—a very practical concept since we often use these units to measure object lengths in daily life.
First, remember an important conversion: 1 decimeter equals 10 centimeters, and 1 centimeter equals 10 millimeters. So, if we want to know how many millimeters are in 1 decimeter, we calculate: 1 decimeter = 10 centimeters = 10 Œ 10 millimeters = 100 millimeters. That is, **1 decimeter equals 100 millimeters**.
Now, think about this: if you have a pen holder 5 decimeters tall, how many millimeters is that? Try calculating yourself first, then tell me your answer, okay?
(Waiting for student response)
Great! Did you figure it out? If the pen holder is 5 decimeters tall, its height is 5 Œ 100 millimeters = 500 millimeters. Pretty simple, right?
Next, another example: suppose you have a rope 300 millimeters long. How many decimeters is this? This is slightly harder, but follows the same principle. We know 1 decimeter equals 100 millimeters, so 300 millimeters equals 300 œ 100 decimeters = 3 decimeters. Can you work through this reasoning yourself?
Think about it, then we'll continue to see if you've mastered this concept.

Qualitative Analysis

This comparison starkly illustrates a critical advantage of automated workflow optimization: **factual reliability**. The Expert-Designed response, while pedagogically elaborate with engaging formatting, emojis, and creative metaphors, contains a **catastrophic factual error**—systematically stating that 1 decimeter equals 1000 millimeters throughout the entire lesson. The correct conversion is **1 decimeter = 100 millimeters**, making this a 10-fold error that would fundamentally mislead students.

In contrast, both the **AutoSynth-trained model** and the **Base Model** provide **factually correct** explanations. The Base Model offers a clear, straightforward approach with step-by-step reasoning. The AutoSynth model takes a more **Socratic, inquiry-based approach**, using guided questions to help students discover the conversion relationships themselves, identify common misconceptions, and develop metacognitive strategies.

Notably, the AutoSynth model explicitly anticipates and addresses a common student error ("thinking 1 centimeter equals 100 millimeters"), demonstrating pedagogical sophistication likely learned through iterative refinement.

This example validates a key finding: while human experts bring pedagogical creativity and engaging presentation styles, they are not immune to fundamental factual errors. AutoSynth's iterative refinement process helps systematically avoid such critical pitfalls while maintaining high instructional quality.

A.2 Task 2: Interdisciplinary Lesson Planning

Query

Target Students: Grade 8 students (ages 13-14) familiar with important technological events in history and able to analyze social factors, but lacking systematic understanding of physics' role in historical progress.
Core Subject: Physics **Related Subject**: History
Duration: 4 class periods
Prior Knowledge: Students have mastered friction analysis, pulley systems, and energy transformation examples.

Model Responses

AutoSynth Model: Comprehensive Structure
1. Learning Objectives
Knowledge and Skills:

- **Physics:** Apply lever principles, work concepts (W=Fs) to analyze ancient

tools; identify energy transformations in simple machines; understand buoyancy principles.

- **History:** Master background and significance of landmark engineering achievements from ancient civilizations; analyze their social, economic, and cultural impacts.

2. Key Teaching Points
Integration Focus: Using physics principles as keys to explain historical phenomena...
Learning Difficulties and Strategies:

- **Difficulty 1:** Connecting abstract physics with concrete history
- **Strategies:** Simplified modeling, scenario immersion methods...

3. Materials and Preparation
[Detailed lists of experimental materials, digital resources, pre-class assignments...]
4. Four-Period Lesson Plan
Period 1 (45 min): Introduction and Inquiry Launch

- Opening (15 min): Show video of pyramids, Great Wall, etc.
- Inquiry (30 min): Focus on Egyptian pyramids; decompose problems

Periods 2-3: Hands-on Experiments

- Groups design experiments with inclined planes, pulley systems
- Build physical models demonstrating principles

Period 4: Presentations and Synthesis
5. Assessment Scheme
Process Evaluation (40 percent): Participation, experimental operations, research records, teamwork
Summative Evaluation (60 percent): Group research report; Individual reflection journal

Expert Model: Excellent Detail and Rubrics
I. Theme Design
Project Name: "The Lever of History"—Exploring How Technology Propels Civilization
Driving Question: How have simple machine principles become critical levers for civilizational progress across different historical periods?
II. Assessment Design
Five-Dimensional Evaluation Framework:

1. Physics knowledge mastery
2. Interdisciplinary thinking
3. Evidence-based historical reasoning
4. Practical innovation capability
5. Teamwork competencies

Detailed Rubric: 4-point scales (Excellent/Good/Adequate/Needs Improvement) across all dimensions with specific descriptors.
III. Four-Period Detailed Design
Period 1: Topic Selection and Initial Exploration
Period 2: Principle Analysis and Model Design
Period 3: Model Construction and Refinement
Period 4: "Historical Technology Expo" with presentations
[Includes detailed student reflection prompts, extension activities, and troubleshooting guidance...]

Base Model: Generic Outline
Interdisciplinary Lesson Plan Design
Learning Objectives:

- Help students understand how physics influences historical progress...
- Cultivate ability to analyze social factors behind technological development...

Class Schedule: Four periods covering physics review, historical technologies, case study, and modern applications.
Assessment: Class participation, group reports, reflection journals
[Subsequent periods contain only high-level bullet points without actionable details...]

Qualitative Analysis

This comparison reveals dramatic capability differences for complex generation tasks. The **Base Model** produces only a generic outline with vague bullet points—entirely insufficient for classroom use.

Both **AutoSynth** and **Expert-Designed** models generate outstanding, comprehensive lesson plans with detailed objectives, granular activities, and sophisticated assessment schemes. The subtle differences reflect their optimization processes: AutoSynth emphasizes systematic difficulty coverage (likely from iterative refinement), while Expert-Designed showcases deep assessment design expertise. Both represent quantum leaps over baseline, validating that specialized workflows—automated or manual—are essential for professional-grade content.

This supports our quantitative findings (Table 1): while AutoSynth achieves lower human preference (51% vs 96%), it produces genuinely high-quality outputs far superior to baseline capabilities.