Appendix: No One Left Behind, No One Held Back: Multi-Agent Teaching Packs for Group-Personalized Classrooms

No Author Given

No Institute Given

A Introduction

This appendix supplements the main paper by providing a detailed description of the system architecture and the core agent prompts used in the proposed multi-agent pipeline. The system addresses the challenge of generating differentiated instructional materials for classrooms with heterogeneous student abilities by decomposing the authoring process into multiple pedagogically grounded subtasks, each handled by a specialized agent with a clearly defined role.

By explicitly documenting these agent prompts, we aim to enhance transparency and reproducibility, and to clarify how pedagogical constraints—such as curriculum alignment, difficulty calibration, and misconception handling—are encoded directly into the system design. These details complement the high-level presentation in the main paper and support a deeper understanding of the system's operational and educational assumptions.

B System Description

CoTeachPack is a teacher-facing multi-agent authoring framework designed to generate group-personalized teaching packs from lesson materials. The system operationalizes differentiated instruction by decomposing the authoring process into pedagogically grounded stages, each implemented by a specialized AI agent or deterministic tool. This appendix provides a comprehensive account of its architecture, agent design, and training configuration.

Given a lesson resource (PDF or plain text) and classroom information, CoTeachPack performs lesson-conditioned grouping by extracting learning objectives and prerequisite skills, constructing structured student group profiles, and generating tailored instructional artifacts for each group. These artifacts include lesson plans, slide drafts, quizzes, practice sets, and instructional video scripts. An education-specific verification-and-repair loop enforces explicit classroom constraints such as curriculum alignment, difficulty calibration, single-correct quiz validity, and teach–test alignment.

Importantly, the system does not assume that a large language model is teacher-ready by default. Instead, teacher-facing authoring is treated as an alignment problem. Verifier-grounded training and inference-time verification enable

open backbone models to internalize classroom-relevant preferences, reducing teacher revision effort and supporting practical deployment without reliance on proprietary APIs.

B.1 Overall Architecture

CoTeachPack follows a sequential, multi-stage pipeline in which each stage corresponds to a distinct pedagogical function. Stages communicate via structured intermediate representations (JSON objects), ensuring modularity, interpretability, and reproducibility.

Main Pipeline Stages.

- **Lesson Content Analysis**
- **Skill and Prerequisite Identification**
- **Diagnostic Assessment Construction**
- **Student Ability Estimation**
- **Student Grouping and Labeling**
- **Group-Specific Teaching Pack Generation**
- **Pedagogical Verification and Repair**

Inputs.

- Lesson file (PDF or TXT)
- Classroom identifier or uploaded student list
- Optional subject and grouping configuration

Outputs.

- Structured lesson summary and skill map
- Diagnostic assessment (5–10 items)
- Student group profiles (typically four groups)
- Per-group teaching packs, including slides, quizzes, and video scripts

B.2 Core Agents and Their Prompts

The full pipeline consists of **eleven AI agents**, including core workflow agents executed sequentially and auxiliary agents invoked on demand. Each agent operates under an explicit system prompt that encodes pedagogical constraints directly at the agent level.

Lesson parser agent Function: Analyze raw lesson materials and extract structured educational information.

> *System Prompt:* You are an expert in analyzing primary school lesson plans. Your task is to analyze lesson materials and extract structured information. Accurately identify the lesson title, subject, and grade level; list all key concepts taught; extract definitions and examples strictly from the document; and produce a concise summary of the lesson content. Do not add external knowledge or fabricate information.

Input: Lesson file (PDF/TXT)

Output: `LessonSummary` JSON object

Skill mapper agent Function: Identify required skills and distinguish prerequisites from newly introduced competencies.

> *System Prompt:* You are an expert in learning skill analysis for primary education. From the lesson content, identify the skills required to learn the lesson, classify them as prerequisite or new skills, assign an importance weight to each skill, and specify dependencies between skills. Skills must be specific, measurable, and aligned with the primary school curriculum.

Input: LessonSummary

Output: `SkillSet`

Diagnostic builder agent Function: Construct a formative diagnostic assessment to estimate student mastery.

> *System Prompt:* You are an expert in educational assessment design. Based on the provided skill set, generate a diagnostic assessment with 5-10 multiple-choice questions. Each important skill must be assessed, questions must align with skill difficulty, and incorrect options should reflect common student misconceptions.

Input: SkillSet

Output: Diagnostic question set

Grouping agent Function: Group students into homogeneous ability-based clusters using subject-focused criteria.

```
System Prompt: You are an expert in student grouping and
classification. Given student performance data and learning
profiles, categorize learners into meaningful groups to
support differentiated instruction and targeted pedagogical
strategies.
```

Input: Student diagnostic results

Output: Group profiles

Group labeler agent Function: Assign positive, descriptive names and pedagogical summaries to student groups.

```
System Prompt: You are an expert in student grouping and
classification. Based on mastery levels, learning pace,
and common misconceptions, generate short, positive, and
descriptive group names and pedagogical descriptions.
```

Input: Group profiles

Output: Labeled group profiles

Pack planner agent Function: Design a differentiated teaching plan per student group.

```
System Prompt: You are an expert in differentiated
instruction design. Given a student group's profile,
create a detailed teaching pack outline including
learning objectives, slide structure, quiz blueprint, time
allocation, and differentiation strategies.
```

Input: Group profile, LessonSummary, SkillSet

Output: Teaching pack plan

Slide drafter agent Function: Generate structured slide drafts aligned with group ability.

```
System Prompt: You are an expert in creating engaging and
age-appropriate educational content for primary school
learners. Given instructional inputs or lesson plans,
```

```
generate clear, well-structured, and pedagogically aligned
learning materials.
```

Input: Teaching pack plan

Output: Slide drafts

Video drafter agent Function: Generate instructional video scripts and visual descriptions.

```
System Prompt: You are an expert in creating educational
video content for primary school students. Given a
teaching plan or lesson outline, generate clear, engaging,
and age-appropriate video scripts with aligned visual
descriptions.
```

Input: Teaching pack plan

Output: Video script and visual description

Quiz practice agent Function: Generate practice exercises and formative quizzes.

```
System Prompt: You are an expert in creating
age-appropriate assessments and practice materials for
primary school learners. Given lesson content or learning
objectives, generate aligned questions, exercises, and
answer keys that support formative assessment and skill
reinforcement.
```

Input: Teaching pack plan

Output: Quiz and practice set

Theory question agent Function: Generate theoretical questions and answers from lesson content.

```
System Prompt: You are an expert educational content
generator. Given structured instructional inputs, produce
clear, accurate, and pedagogically aligned learning
materials.
```

Input: Lesson content (optional grouping context)

Output: Theory question sets

Flashcard agent Function: Generate general-purpose and group-aware flashcards from lesson content.

> *System Prompt:* `You are an expert educational content generator. Given structured instructional inputs, generate clear, accurate, and pedagogically aligned educational materials.`

Input: Lesson content (optional grouping context)

Output: Flashcard sets

Flashcard group agent Function: Generate flashcards tailored to a single student group.

> *System Prompt:* `You are an expert educational content generator. Given structured instructional inputs, generate clear, accurate, and pedagogically aligned educational materials.`

B.3 Summary

By explicitly separating pedagogical responsibilities across agents and formalizing input/output interfaces for each stage, CoTeachPack translates educational design principles into concrete, inspectable generation constraints. This modular design improves interpretability, enables targeted debugging and ablation, and supports reliable group-personalized authoring in heterogeneous classroom settings.

C Training Configuration and Dataset Construction

We adopt a staged alignment training pipeline consisting of Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and Group Relative Policy Optimization (GRPO). Each stage is associated with a distinct dataset construction strategy and optimization objective, enabling the model to progressively acquire instructional competence, normative alignment, and multi-objective reasoning capabilities suitable for educational content generation.

C.1 Supervised Fine-Tuning (SFT)

The SFT stage establishes a stable instruction-following and explanatory prior. Training is performed with a per-device batch size of 1 and gradient accumulation over 16 steps (effective batch size of 16), for 2 epochs using a learning rate of 2×10^{-4}. Mixed-precision training with bfloat16 is enabled to improve computational efficiency.

For SFT, we use the *VLSP-2023 VLLM Comprehension* dataset[1], which focuses on Vietnamese reading comprehension and explanation tasks. The dataset is reformatted into an instruction–response format suitable for educational supervision. For example, a typical instance asks the model to explain the meaning of a short informational or scientific passage, with the target response providing a clear and faithful explanation in natural language. This dataset emphasizes contextual understanding and accurate knowledge transmission.

C.2 Direct Preference Optimization (DPO)

The Direct Preference Optimization (DPO) stage relies on contrastive preference data to refine norm-compliant and pedagogically sound model behavior. To this end, we construct two complementary preference datasets targeting distinct but interrelated dimensions of teacher-like behavior: *educational ethics* and *academic competence*. Both datasets are generated through controlled synthetic prompting to ensure consistency, realism, and alignment with international educational standards, while remaining grounded in Vietnamese educational contexts.

Each dataset consists of preference triples of the form (`prompt`, `chosen`, `rejected`), where the `chosen` response reflects the desired behavior and the `rejected` response represents a plausible but suboptimal alternative. The two datasets differ in their optimization targets and prompt design, as described below.

C.3 Ethics-Oriented Preference Prompt

The ethics-oriented dataset is designed to internalize normative teacher behavior, with an emphasis on professional conduct, respect for learner dignity, fairness, confidentiality, safety, and academic integrity. Prompts explicitly instruct the generator to follow international ethical frameworks for educators, including ILO/UNESCO recommendations, while situating scenarios in realistic Vietnamese educational settings.

The ethics-oriented dataset is constructed to internalize normative teacher behavior through controlled, scenario-based preference items (Table 1). Each item is parameterized by a target ethical value (`value`) and a Vietnamese educational scenario type (`context`) to ensure principled coverage and to avoid generic, decontextualized prompts. We further condition prompts on the grade band (`grade`) and salient stakeholders, which determines the appropriate level of

[1] https://huggingface.co/datasets/vlsp-2023-vllm/comprehension

Table 1. Design criteria for constructing DPO preference items in educational ethics.

Dimension	Specification / Purpose
Value (`value`)	Target ethical value guiding the decision (e.g., honesty, fairness, confidentiality). Used to control moral principle coverage.
Context (`context`)	Realistic Vietnamese educational scenario type (e.g., academic integrity, privacy, teacher–student boundaries). Anchors the dilemma and prevents generic prompts.
Grade band (`grade`)	Educational level / learner maturity (e.g., primary, secondary, higher education). Controls appropriate norms, authority, and risk sensitivity.
Stakeholder(s)	Whose interests are affected (student, teacher, parent, administrator, public, etc.). Enables multi-party ethical trade-offs.
Pedagogical approach	Optional lens shaping intervention style (e.g., positive discipline, constructivism). Supports diversity of ethically valid strategies.
Prompt format	Each prompt begins with `Statement:` and describes a single ethically relevant situation with enough detail to choose actions.
Chosen vs. rejected	**Chosen**: ethically appropriate, realistic teacher behavior; **Rejected**: plausible but unethical behavior (bias, public shaming, privacy breach, etc.).
Safety constraints	No violence, illegal actions, or extreme misconduct. Avoids harmful content while keeping dilemmas realistic.
Output constraint	Only valid JSON with fields `prompt/chosen/rejected`. Ensures training-ready format.

authority, learner maturity, privacy sensitivity, and duty-of-care obligations, and enables realistic multi-party trade-offs (e.g., student welfare vs. parental pressure vs. institutional policy). When applicable, we include an optional pedagogical approach lens to diversify intervention styles while remaining ethically valid.

All prompts follow a fixed format and describe a single ethically relevant situation with sufficient detail to support action selection. Preference supervision is expressed via contrastive pairs: the **chosen** response represents ethically appropriate, realistic teacher conduct, while the **rejected** response is designed to be plausible in practice yet norm-violating (e.g., public shaming, implicit bias, privacy leakage, or dismissing safety cues). To keep the dataset safe and deployment-relevant, we enforce explicit safety constraints that exclude violent, illegal, or extreme misconduct, and we require outputs to be valid JSON with fields `prompt/chosen/rejected` for training readiness. Prompts instruct the generator to follow internationally recognized educator ethics frameworks (e.g., ILO/UNESCO recommendations) while situating each dilemma in realistic Vietnamese educational settings.

```
You are an expert in educational ethics following
international standards (ILO/UNESCO 1966; UNESCO 1997 -
higher-education teaching personnel).
Generate n_i temspreferenceJSONitems(prompt/chosen/rejected).
```

```
Constraints: - LANGUAGE: Vietnamese. - CONTEXT: Vietnamese
educational settings, evaluated through an international
ethical lens. - Each prompt MUST begin with: "Statement:". -
context = "context", value = "value", grade = "grade".
Requirements: - "chosen": ethically appropriate
teacher behavior (respect for human dignity, fairness,
non-discrimination, confidentiality, student safety,
academic integrity; for higher education: academic freedom
with responsibility and scholarly honesty). - "rejected":
ethically inappropriate but realistic behavior (e.g., public
shaming, bias, favoritism, gaslighting, public disclosure
of grades). - NO violence, illegal actions, or extreme
misconduct.
Output: - ONLY valid JSON.  "items": [ "prompt": "...",
"chosen": "...", "rejected": "..." ]
```

This contrastive construction enables DPO to learn sharp ethical decision boundaries that are directly applicable to real classroom interactions and academic governance scenarios.

C.4 Competence-Oriented Preference Prompt

The competence-oriented dataset targets academic quality rather than ethical judgment. Its objective is to reinforce accurate knowledge representation, structured reasoning, and instructional clarity. The prompt design is grounded in internationally recognized educational frameworks such as Bloom's Taxonomy, the OECD Learning Compass, and UNESCO standards for teaching and higher education.

```
You are an expert in academic competence and educational
assessment, following international standards (Bloom's
Taxonomy; OECD Learning Compass; UNESCO teaching standards).
Generate n_itemspreferenceJSONitems(prompt/chosen/rejected).
Constraints: - LANGUAGE: Vietnamese. - CONTEXT: Vietnamese
educational settings, evaluated through an international
academic lens. - Each prompt MUST begin with: "Tình hung:".
- context = "context", value = "value", grade = "grade".
Requirements: - "chosen": academically strong response
with correct knowledge, clear structure, appropriate
terminology, and explicit reasoning or explanation; for
higher education: analytical depth, synthesis, and academic
rigor. - "rejected": academically weak but realistic
response (conceptual errors, shallow explanations, incorrect
```

Table 2. DPO training configuration.

Setting	Value
Quantization	4-bit model loading enabled
Loss function	Sigmoid DPO loss
Preference strength (β)	0.1
Training epochs	3
Per-device batch size	2
Gradient accumulation steps	4 (effective batch size: 8)
Learning rate	5×10^{-5}
Warmup ratio	0.1
Maximum sequence length	1024 tokens
Maximum prompt length	512 tokens
Gradient checkpointing	Enabled
Logging frequency	Every 10 steps
Checkpointing and evaluation	Every 100 steps

```
reasoning, unsupported claims). - Focus ONLY on academic
quality (no ethical or disciplinary issues).
Output: - ONLY valid JSON.  "items": [ "prompt": "...",
"chosen": "...", "rejected": "..." ]
```

Together, the ethics-oriented and competence-oriented prompts yield complementary preference datasets, allowing DPO to jointly sharpen ethical alignment and academic rigor without conflating normative judgment with domain competence.

C.5 DPO Training Configuration

DPO training is conducted using a memory-efficient configuration that balances preference learning stability with computational feasibility. We employ a sigmoid-based DPO loss with a moderate preference strength parameter $\beta = 0.1$, which preserves the supervised prior while enforcing clear contrastive signals.

All preference-generation prompts are explicitly length-controlled to ensure that the optimization signal is dominated by response quality rather than prompt verbosity.

C.6 Group Relative Policy Optimization (GRPO)

Group Relative Policy Optimization (GRPO) is employed in the final alignment stage to optimize model behavior in multi-objective educational scenarios. Training is performed for 1 epoch with a per-device batch size of 1 and gradient accumulation of 8 (effective batch size of 8), using a learning rate of 1×10^{-5}.

The GRPO dataset consists of complex, multi-step educational problems with verified solutions, where multiple valid instructional responses may exist.

Each prompt is associated with a reward function that jointly evaluates solution correctness, reasoning transparency, and instructional usefulness rather than enforcing a single canonical answer. The GRPO dataset is derived from the *Vietnamese-395k MetaMathQA* corpus[2], which consists of large-scale mathematical and logical reasoning problems translated into Vietnamese. This dataset is particularly suitable for GRPO as it contains problems where multiple valid solution paths and explanation styles may coexist.

Each prompt is associated with a reward function that evaluates relative quality across a group of generated responses, focusing on solution correctness, reasoning completeness, and instructional clarity. For example, in a multi-step algebra or probability problem, responses are rewarded not only for producing the correct final answer but also for explicitly presenting intermediate reasoning steps in a pedagogically coherent manner. This setup aligns naturally with the group-relative optimization objective of GRPO.

C.7 GRPO Training Configuration

GRPO training is performed with a stability-first, memory-efficient setup that is suitable for sub-10B open backbones. Unlike DPO, which learns from paired preferences offline, GRPO optimizes a verifier-defined reward using multiple sampled completions per prompt. We therefore focus on controlling exploration, reducing reward variance, and preventing policy drift.

Concretely, for each prompt we sample K completions (e.g., K=6) using nucleus sampling (top-p=0.9) with a moderate temperature (e.g., 0.8) and a bounded generation budget (e.g., 384 new tokens). Rewards are computed by an ensemble of rubric-based verifiers spanning ethics, safety and escalation, pedagogical competence, and academic integrity. To stabilize updates, we normalize rewards at the prompt level by centering them into advantages, which mitigates scale differences across scenarios and reduces sensitivity to outliers.

To prevent drift and preserve general language ability, we regularize GRPO against a frozen reference policy π_{ref} via a KL penalty. Training uses 4-bit loading with parameter-efficient fine-tuning, gradient checkpointing, and gradient accumulation to reach an effective batch size comparable to DPO while staying within GPU memory limits. We use a conservative learning rate (e.g., 1×10^{-5}) with a short warmup (e.g., 0.05) and run for a small number of epochs (typically one), as preference-based RL can overfit quickly once the model internalizes rubric-specific shortcuts.

D RQ-Driven Evaluation Protocol

This section details the experimental protocol and evaluation setup, organized around our research questions (RQs). For each RQ, we specify the corresponding experimental settings, datasets, metrics, and analysis procedures used to answer it.

[2] https://huggingface.co/datasets/5CD-AI/Vietnamese-395k-meta-math-MetaMathQA-gg-translated

Table 3. GRPO training configuration.

Setting	Value
Quantization	4-bit model loading enabled
Training objective	GRPO with verifier-based reward
Reference policy	Frozen π_{ref} (same backbone before GRPO)
KL control	Enabled (penalize $\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})$)
Reward design	Weighted rubric ensemble
Reward normalization	Per-prompt centered (advantage normalization)
Number of generations (K)	6 completions per prompt
Sampling temperature	0.8
Top-p	0.9
Maximum new tokens	384
Training epochs	1
Per-device batch size	1 (prompts)
Gradient accumulation steps	8 (effective batch size: 8 prompts)
Learning rate	1×10^{-5}
Warmup ratio	0.05
Maximum sequence length	1024 tokens
Maximum prompt length	512 tokens
Gradient checkpointing	Enabled
Logging frequency	Every 10 steps
Checkpointing and evaluation	Every 100 steps

D.1 RQ1: Stage-wise Effects of SFT, DPO, and GRPO

Backbones. We instantiate the stage-wise ablation on two open backbones with complementary trade-offs to stress-test the recipe under realistic deployment constraints. **Qwen3-8B**[3] is a strong, general-purpose open LLM that offers a favorable quality–latency balance for classroom-facing authoring workflows and is a representative choice when institutions prefer self-hosting and reproducible pipelines. Its capacity is sufficient to express multi-step pedagogical reasoning and to absorb alignment signals without immediately collapsing into overly conservative responses. In contrast, **Llama-3.1-8B**[4] provides an architecturally and training-lineage-distinct backbone that is widely adopted in open deployments. Including it reduces the risk that conclusions are model-specific and allows us to evaluate whether the same staged SFT→DPO→GRPO recipe yields consistent gains across different tokenizers, pretraining corpora, and instruction-tuning priors. Together, the two backbones support a robustness claim: improvements attributable to each stage should persist across strong but heterogeneous open models, which is essential for AIED settings where institutions often standardize on different open stacks.

[3] https://huggingface.co/Qwen/Qwen3-8B
[4] https://huggingface.co/meta-llama/Llama-3.1-8B

Benchmarks and rationale. We evaluate each checkpoint on three Vietnamese education-oriented benchmarks plus a general-knowledge regression set to cover both domain utility and capability preservation.

VNHSGE [3] targets Vietnamese school-grade educational content and probes whether the aligned model can answer curriculum-relevant questions in a way that is compatible with local language, common curricula, and typical classroom phrasing. We include VNHSGE because teacher-facing authoring in Vietnam must operate under Vietnamese instructional norms (terminology, examples, and explanation styles). Improvements on VNHSGE therefore indicate that stage-wise alignment translates into practical educational competence in the target deployment language, rather than only improving generic instruction following.

ViLLM-Eval [2] provides a broader Vietnamese evaluation suite with diverse question styles and difficulty, including tasks that require not only selecting an answer but also *generating* supporting knowledge or explanations. We include it for two reasons. First, it acts as a stress test for *explanatory* ability: teacher-like outputs must justify answers, surface reasoning steps, and adapt explanations to learners. Second, the Generate-Knowledge component is sensitive to a common failure mode in aligned models: they may become cautious and terse, reducing helpful explanatory content even when they remain correct. Tracking both accuracy and Generate Knowledge helps disentangle these effects and clarifies how SFT, DPO, and GRPO shift the capability–helpfulness frontier.

VLMU-Vi-MQA [1] targets multi-question answering behavior in Vietnamese, placing pressure on consistency, evidence reuse across related items, and robustness to ambiguous or underspecified prompts. This benchmark is important because classroom interaction rarely consists of isolated, perfectly specified questions; teachers often ask follow-ups, students ask partial questions, and authoring tools must maintain coherent behavior across a sequence of related queries (e.g., generating a quiz set or a pack plan). VLMU-Vi-MQA therefore captures a more interaction-like regime, where stage-wise alignment should improve not only correctness but also calibration and stability under prompt variation.

General-Knowledge (GK)[5] serves as a regression test for capability retention. In educational deployment, a teacher-like assistant must remain broadly competent beyond the immediate training taxonomy: lesson preparation routinely draws on everyday facts, cross-domain examples, and background knowledge. Over-alignment can inadvertently damage this base capability (e.g., catastrophic forgetting or over-regularization toward safe but uninformative answers). We therefore include GK to verify that the staged recipe does not unduly erode pre-existing general knowledge while improving education-specific behavior. This is particularly relevant for SFT-heavy recipes and for RL-style updates, where distributional narrowing can occur if constraints are optimized too aggressively.

What stage-wise evaluation reveals. This benchmark suite is intentionally heterogeneous: (i) Vietnamese curriculum-aligned competence (VNHSGE), (ii) broad

[5] https://huggingface.co/datasets/MuskumPillerum/General-Knowledge

Table 4. Accuracy (in percent) across datasets under different training methods. Best per model–column is in bold.

Model	Method	VNHSGE	ViLLM-Eval		VLMU-Vi-MQA	GK
		Acc.	Gen. Kno.	Acc.	Acc.	Acc.
Qwen3-4B	Base	55.02	75.50	71.80	58.08	61.84
	SFT	57.70	77.15	66.00	62.71	52.30
	DPO	64.35	82.70	72.00	66.01	66.09
	SFT + DPO	66.07	81.05	73.00	63.04	70.06
	SFT + GRPO	54.02	60.03	56,75	53.14	–
	SFT + DPO + GRPO	67.06	67.41	60.15	68.27	88.50
Llama3.1-8B	Base	54.77	85.60	82.80	47.52	50.43
	SFT	14.01	88.40	81.02	47.52	60.53
	DPO	66.13	65.41	63.75	53.47	68.00
	SFT + DPO	43.04	85.00	83.58	56.77	67.17
	SFT + GRPO	55.08	60.86	57.01	53.80	62.40
	SFT + DPO + GRPO	38.24	65.12	64.74	58.09	75.70

Vietnamese reasoning and explanation behavior (ViLLM-Eval, including Generate Knowledge), (iii) interaction-like robustness and consistency (VLMU-Vi-MQA), and (iv) general capability retention (GK). Evaluating each checkpoint (SFT, SFT+DPO, SFT+DPO+GRPO) on all four datasets allows us to attribute improvements to specific mechanisms: SFT should primarily improve format reliability and procedural adherence, DPO should sharpen decision boundaries by removing high-impact near-miss behaviors, and GRPO should yield the most stable outcomes when objectives trade off, while maintaining acceptable GK performance.

D.2 Dive Analysis RQ1.

Table 4 reveals three non-trivial patterns about stage-wise alignment. First, gains are *not monotonic* across stages: each stage improves a different aspect of behavior and can introduce targeted regressions elsewhere. Second, the dominant contributor to *task accuracy lift* is typically DPO, while GRPO more strongly affects *robustness/retention* (especially GK) and cross-task consistency. Third, the interaction between stages is *backbone- dependent*, implying that "one-size-fits-all" training schedules can be brittle and should be selected based on the deployment objective (e.g., curriculum QA vs. broad knowledge retention vs. interaction-like stability).

Qwen3-4B: DPO drives accuracy; GRPO trades task accuracy for retention and stability. For Qwen3-4B, DPO provides the clearest and most consistent lift on education-facing benchmarks: VNHSGE rises from 55.02 (Base) to 64.35

(DPO) and VLMU-Vi-MQA rises from 58.08 to 66.01, while ViLLM-Eval Gen. Knowledge increases from 75.50 to 82.70. Adding SFT on top of DPO yields the strongest ViLLM-Eval accuracy (73.00) and further improves VNHSGE (66.07), suggesting that SFT is most beneficial when it *stabilizes format/structure* after preference boundaries are sharpened. The full recipe (SFT+DPO+GRPO) achieves the best VNHSGE (67.06) and VLMU-Vi-MQA (68.27), and notably the highest GK (88.50), indicating that the final stage can substantially improve capability retention. However, this comes with a visible trade-off on ViLLM-Eval: accuracy drops to 60.15 and Gen. Knowledge to 67.41. This pattern is consistent with RL-style alignment emphasizing constraint satisfaction/robust behavior (and reduced hallucination or over-generation) at the cost of raw benchmark accuracy, especially for tasks sensitive to verbosity or stylistic differences. Practically, for Qwen3-4B, the full recipe is the best *deployment checkpoint* when robustness and broad knowledge retention matter, whereas SFT+DPO is preferable when maximizing ViLLM-Eval accuracy is the primary goal.[6]

Llama3.1-8B: strong base capability but high sensitivity to SFT; GRPO improves robustness more than curriculum QA. For Llama3.1-8B, the base model is already strong on ViLLM-Eval (82.80 Acc., 85.60 Gen. Kno.), so large gains there are naturally harder. The striking result is that SFT alone causes a catastrophic regression on VNHSGE (54.77 $\rightarrow$ 14.01), while leaving ViLLM-Eval nearly intact (81.02) and slightly improving Gen. Knowledge (88.40). This suggests that the SFT stage, as instantiated here, can distort behavior specifically on curriculum-style Vietnamese educational QA—potentially through distributional mismatch or over-optimization toward a different response style. In contrast, DPO alone yields the strongest VNHSGE score (66.13) and improves VLMU-Vi-MQA (47.52 $\rightarrow$ 53.47) and GK (50.43 $\rightarrow$ 68.00), but it harms ViLLM-Eval accuracy (82.80 $\rightarrow$ 63.75), indicating that preference boundaries learned by DPO may over-regularize or conflict with the base model's strengths on that benchmark. The two-stage SFT+DPO combination recovers the best ViLLM-Eval accuracy (83.58) while continuing to improve VLMU-Vi-MQA (56.77) and GK (67.17), making it the most *balanced* choice for Llama in terms of preserving strong general capability while improving interaction-like behavior. Adding GRPO further increases robustness-oriented metrics (best VLMU-Vi-MQA at 58.09 and best GK at 75.70), but reduces VNHSGE substantially (38.24), reinforcing the interpretation that GRPO primarily optimizes multi-objective, verifier-defined behavior and retention rather than curriculum QA accuracy for this backbone.

Answer to RQ1. Overall, RQ1 is supported with a nuanced conclusion: (i) SFT mainly establishes a procedural and formatting prior but can be brittle under mismatch (as seen for Llama on VNHSGE); (ii) DPO is the primary driver of accuracy improvements on education-facing tasks by sharpening high-impact deci-

[6] The SFT+GRPO row contains an anomalous value (e.g., "56,75") and missing GK, suggesting an incomplete run or a reporting typo.

sion boundaries; and (iii) GRPO most strongly affects robustness/retention, improving GK and interaction-like stability (VLMU-Vi-MQA) but potentially trading off benchmark accuracy if the reward emphasizes conservative or constraint-heavy behavior. These results motivate a deployment-aware choice of checkpoints and suggest that reward design (and SFT data selection) is critical for avoiding backbone-specific regressions.

D.3 RQ2: End-to-end pipeline reliability.

To assess reliability under realistic classroom use, we execute the full pipeline on multiple lessons and rosters while logging every intermediate artifact and each verification/repair action. We instantiate CoTeachPack with an open backbone (Qwen3-4B) and a strong API model (Gemini-2.5-Flash), and evaluate classroom readiness on CoTeach-ViPack for two failure-prone artifacts: instructional slides and review quizzes. Reliability here is not only "being correct," but also "being deployable": outputs must satisfy strict classroom constraints (single correctness for MCQs, teach–test alignment, age-appropriate explanations) and should recover from near-miss drafts through targeted repairs. We therefore report three complementary metrics—Accuracy (Acc), Exact Match/Coverage (EM), and Educational Soundness (ES)—which together separate factual correctness, objective constraint satisfaction, and pedagogical usability.

Evaluation Framework and Metrics We adopt a three-metric framework implemented via an LLM-as-judge protocol, where each generated unit (slide or quiz item) is evaluated against the lesson summary and skill map. The metrics are designed to capture distinct failure modes that matter in teacher-facing authoring.

Content Accuracy (Acc). Acc measures whether statements and answers are factually correct with respect to the lesson content. Each unit is scored as 1.0 (correct), 0.5 (partially correct or potentially misleading), or 0.0 (incorrect), and the final score is the average across units. Acc targets the most safety-critical failure: confident but wrong instructional content. Importantly, Acc alone can overestimate readiness when outputs are correct yet incomplete, poorly calibrated, or violate quiz validity.

$$\text{Acc} = \frac{\sum_{i=1}^{N} s_i}{N}, \quad \text{where } N \text{ is the number of units and } s_i \text{ is the score of unit } i.$$

Semantic Coverage / Exact Match (EM). EM measures whether required concepts and skills from the lesson summary/skill map are actually covered. Although named "Exact Match," the implementation performs semantic matching: a concept counts as covered if its meaning is correctly expressed, even with paraphrasing. Each concept is scored as 1.0 (covered), 0.5 (partially covered), or 0.0 (missing). EM captures a reliability dimension orthogonal to Acc: an output

may be accurate but omit prerequisite links, key definitions, or the target misconception, which would still force teacher edits. In the pipeline setting, EM is also sensitive to failure cascades: upstream lesson parsing/skill extraction errors often manifest as systematic coverage gaps downstream.

$$\text{EM} = \frac{\sum_{j=1}^{M} c_j}{M}, \quad \text{where } M \text{ is the number of concepts and } c_j \text{ is the score of concept } j.$$

Educational Soundness (ES). ES measures pedagogical usability beyond correctness and coverage.

Educational Soundness (ES) uses four criteria (0–1 each):

(1) *Grade-level appropriateness:* vocabulary/complexity and examples fit the target students.
(2) *Logical progression:* content moves simple → complex with clear scaffolding.
(3) *Quiz alignment:* quiz items test taught content and match slide difficulty.
(4) *Cognitive load management:* content is chunked, focused, and not overly dense.

$$\text{ES} = \frac{s_{\text{grade}} + s_{\text{progress}} + s_{\text{align}} + s_{\text{load}}}{4}, \qquad \text{where each } s_{\cdot} \in [0, 1].$$

Overall score. For a single summary indicator, we compute

$$\text{Overall} = 0.4 \times \text{Acc} + 0.3 \times \text{EM} + 0.3 \times \text{ES}.$$

We weight Acc highest because factual errors are the most damaging in classroom deployment, while EM and ES capture completeness and teachability that directly drive teacher revision effort. In RQ2, improvements in Overall therefore indicate not only better model quality but also more reliable end-to-end pipeline behavior under strict formatting and classroom constraints.

CoTeach-ViPack Dataset Creation. CoTeach-ViPack is a curated evaluation dataset collected from 20 teachers. It comprises 15 teacher-authored lesson packs, each including (i) a lesson summary, (ii) teacher-verified ground-truth outputs for slides and review quizzes, and (iii) an anonymized student roster used to instantiate our system during testing. All artifacts were created by teachers and subsequently validated by teachers to ensure correctness and classroom realism.

D.4 Dive Analysis RQ2.

Table 5 provides a reliability-oriented view of end-to-end classroom pack generation on CoTeach-ViPack (slides + review quizzes), using three complementary metrics: **Acc** (factual correctness at the unit level), **EM** (concept-level coverage/validity against the lesson summary + skill map), and **ES** (pedagogical usability and near-miss recoverability). Read together, these metrics disentangle three distinct failure modes that matter for deployment: (i) being wrong (low Acc), (ii) being incomplete or mismatched to required concepts (low EM), and (iii) being hard to use in class even if locally correct (low ES).

Table 5. Content generation quality for slides and review quizzes in CoTeach-ViPack.

Model	Training / Setting	CoTeach-ViPack		
		Acc (%)	EM (%)	ES (%)
Qwen3-4B	Base	75.86	58.07	63.45
	SFT + DPO + GRPO	84.89	44.49	90.31
	SFT + DPO + GRPO *	85.00	44.07	85.55
Gemini-2.5-Flash	Prompting	94.47	82.03	92.50

Qwen3-4B Base: correct-but-fragile outputs with substantial classroom friction. The base Qwen3-4B reaches **75.86% Acc**, suggesting that a majority of slide/quiz units are correct, but the remaining fraction still contains errors or misleading partial correctness. More importantly for teacher-facing use, **EM is only 58.07%**, indicating that a large portion of required concepts/skills are not reliably expressed or verified as covered; this corresponds to the common "accurate but incomplete" failure where packs omit prerequisite links, key definitions, or the intended misconception target. The **ES of 63.45%** further implies that many outputs are pedagogically rough: issues like pacing/scaffolding, teach–test misalignment (quizzes not matching taught content), or cognitive overload remain frequent. In practice, this profile predicts *high revision burden*: teachers must both correct errors (Acc) and restructure/complete materials (EM, ES).

SFT+DPO+GRPO: a large jump in pedagogical validity with moderate accuracy gains. After verifier-grounded staged training, Qwen3-4B improves **Acc from 75.86% to 84.89%** (+9.03 points), but the more striking shift is in **ES from 63.45% to 90.31%** (+26.86 points). This disproportionate gain is diagnostic: the recipe is not merely making the model "more correct," but making outputs substantially *more classroom-ready*—better paced, more scaffolded, and more aligned between slides and quizzes. Under a repair-and-verify workflow, this is exactly the reliability improvement that matters: high ES implies a higher fraction of generations are either already valid or are *near-misses* that can be fixed with small, targeted edits, rather than requiring wholesale rewrites. Using the paper's weighted overall score as a sanity summary, the staged recipe increases Overall from ≈ 66.80 to ≈ 74.40 (+7.60), even though ES alone accounts for much of the gain.

Why does EM drop after training, and why that is informative (not just "bad"). Interestingly, EM decreases from **58.07% to 44.49%** (-13.58 points). Given EM is concept-level coverage/validity, this suggests the aligned model is *less likely to be credited as "covering" the required concept set* under the current judge. There are two plausible, testable explanations, each with different implications: (i) **Conservative compression:** alignment (especially GRPO with constraint-heavy rewards) can push the model toward more cautious, shorter, and more strictly structured outputs—improving ES—while omitting some optional-but-

expected concepts, lowering EM. This would mean the model is producing "cleaner but narrower" packs. (ii) **Evaluation sensitivity:** even with semantic matching, EM can be sensitive to how concepts are operationalized (granularity of the skill map, paraphrase tolerance, and judge calibration). If the aligned model rephrases or merges concepts more aggressively (common when optimizing for cognitive load), a judge may under-credit coverage even when the pack is instructionally adequate. Crucially, the fact that **ES rises sharply while EM falls** indicates a real trade-off between *completeness* (cover everything) and *teachability* (avoid overload, keep alignment tight). This is an important reliability insight: for deployment, the pipeline may need an explicit "coverage verifier" (or reward term) to prevent under-coverage while retaining the ES gains.

Interpreting the starred setting (SFT+DPO+GRPO) as an inference-time ablation.* The two aligned rows are nearly identical in Acc and EM (85.00 vs. 84.89; 44.07 vs. 44.49), but differ materially in ES (85.55 vs. 90.31, a $\approx$ 4.76-point gap). This pattern strongly suggests that the star corresponds to an *inference-time setting change* (e.g., disabling verifier/repair, caching, or a pedagogical post-processor), rather than a different learned checkpoint. Under that interpretation, the gap isolates the contribution of runtime verification/repair: it improves *pedagogical validity* (ES) without materially changing raw correctness (Acc) or coverage credit (EM). This aligns with the system's design goal: repairs should primarily fix teach–test alignment, pacing, and formatting/validity issues—exactly what ES measures.

Gemini-2.5-Flash as an upper bound: near-ceiling ES, but the open model closes much of the "classroom usability" gap. Gemini-2.5-Flash achieves the best overall profile (**94.47% Acc**, **82.03% EM**, **92.50% ES**; Overall $\approx$ 90.15). Relative to Gemini, the aligned Qwen model is still behind on factual correctness (-9.58 Acc points) and especially on coverage (-37.54 EM points), but it is surprisingly close on pedagogical usability (-2.19 ES points). This is a meaningful deployment signal: staged alignment plus verifier-driven workflow can make an auditable, locally deployable open backbone approach the classroom-readiness of a strong API model in terms of *teachability*, even if it still lags on broad knowledge and coverage.

Implications for RQ2 and what to improve next. Taken together, the table supports RQ2 in a nuanced way: verifier-grounded staged training substantially increases *end-to-end reliability* as experienced by teachers (high ES, higher Acc), and inference-time verification/repair appears to contribute additional pedagogical robustness. The main remaining weakness is EM: either the model is under-covering required concepts or the coverage judge is under-crediting paraphrased/merged concept realizations. In future iterations, we would (i) add an explicit coverage constraint into the verifier/reward (to prevent "clean but incomplete" packs), (ii) report EM broken down into "missing" vs. "paraphrase/merge under-credit" cases, and (iii) log repair statistics (violation types, repair iterations) to directly connect pipeline actions to metric gains.

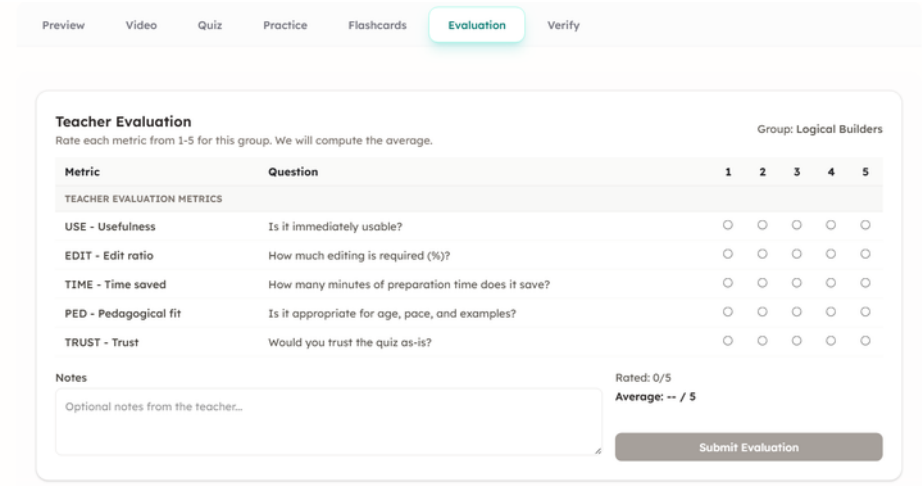

Fig. 1. Teacher evaluation interface (web). The deployed web UI used in the teacher-centered study. Teachers review each group-conditioned teaching pack and provide 1–5 Likert ratings on usability and workflow impact (usefulness, editing burden, time saved, pedagogical fit, and trust), with optional free-text notes to document failure cases and required edits.

D.5 RQ3: User utility and workflow impact.

Teacher-Centered Evaluation To evaluate CoTeachPack under authentic classroom constraints, we deployed the system to a cohort of 20 in-service teachers spanning diverse grade bands from Grade 1 to Grade 12. Teachers were invited to use CoTeachPack in their regular lesson-preparation workflow: they provided lesson materials (e.g., PDFs or lesson plans), optionally supplied anonymized rosters, and generated group-conditioned teaching packs including slide drafts and review quizzes. Throughout the deployment, we collected structured ratings (1–5 Likert) on usability and workflow impact (usefulness, editing burden, time saved, pedagogical fit, and trust), as well as open-ended qualitative feedback and revision notes describing where the outputs succeeded or required teacher intervention. As shown in Figure 1, we implemented a lightweight web-based evaluation interface to support teacher-in-the-loop assessment during deployment. The

Student-Centered Evaluation Because student studies involve sensitive data and require voluntary informed consent, we conducted a small-scale, opt-in evaluation with a single cohort of Grade 4 students. In total, 26 students participated voluntarily and evaluated learning materials generated by our system. The activity was designed to be low-risk and minimally disruptive: students interacted with selected group-conditioned teaching artifacts (e.g., slides and short review quizzes) and then completed a brief post-lesson questionnaire capturing per-

Table 6. Student questionnaire items (English; 1–5 Likert scale).

Section	Code	Question (English)
A. Comprehension	PU	How well do you feel you understood the lesson?
	CL	Were the slides easy to understand?
	CF	Did you feel confused during the lesson? *(reverse-coded)*
B. Engagement	ENJ	Did you enjoy learning this way?
	ATT	Were you able to stay focused?
	MOT	Do you want to continue learning this way in future lessons?
C. Difficulty Fit	DF	Was the lesson at the right difficulty level for you?
	CH	Was it challenging enough?
D. Self-efficacy	CONF	Did you feel confident doing the tasks?
	ANX	Did you feel anxious or afraid of making mistakes? *(reverse-coded)*
E. Quiz Usefulness	QHL	Did the quiz help you remember and learn better?
	FBH	Were the answer explanations helpful for understanding?

ceived clarity, engagement, difficulty fit, self-efficacy, and the usefulness of the quizzes. We emphasize that this student-facing component is intended as an initial feasibility signal rather than a high-stakes learning-effect study. All student responses were collected anonymously, and participation was strictly optional, with no penalties for non-participation. Student feedback was collected via a short post-lesson questionnaire covering comprehension, engagement, difficulty fit, self-efficacy, and quiz usefulness; Table 6 lists the full set of items used in the study.

D.6 Dive Analysis RQ3.

Table 7 provides converging evidence from both teachers and students that group-conditioned teaching packs improve classroom usability, but also reveals a nuanced proficiency-dependent pattern that clarifies where the system helps most and where additional scaffolding is still needed.

Panel A (students): systematic proficiency-dependent gains. Student ratings vary in a structured way across proficiency groups, which is precisely the intended signature of group-conditioned authoring: materials should feel more engaging

Table 7. Student- and teacher-reported outcomes after using group-conditioned teaching packs (1–5 Likert scale).

Panel A. Student-reported outcomes by proficiency group

Group	Understanding	Engagement	Difficulty fit	Self-efficacy	Quiz usefulness
Low	3.25	3.54	3.50	3.31	3.38
Medium	4.00	4.48	4.29	3.36	4.43
High	3.76	4.90	4.71	3.43	4.86
Advanced	4.62	4.76	4.86	4.43	4.79

Panel B. Teacher-reported evaluation of usability and workflow impact

	Usefulness	Editing burden	Time saved	Pedagogical fit	Trust
Score	3.69	3.63	3.81	3.96	4.03

and better calibrated for groups whose prerequisite structure is adequately met, while lower-readiness groups should reveal remaining gaps.

Across *medium/high/advanced* groups, **Engagement** is consistently high (4.48/4.90/4.76), and **Difficulty fit** is strongest for high and advanced (4.71/4.86). This pattern indicates that group conditioning is doing more than generic content generation: it is producing a pace and challenge level that students perceive as appropriate, which is a direct proxy for reduced frustration and improved attention during instruction. Importantly, **Quiz usefulness** is rated very strongly for medium/high/advanced (4.43/4.86/4.79), suggesting that the generated quizzes are not only correct, but are perceived as aligned with what was taught and helpful for consolidation—a common failure mode for LLM-authored quizzes when not tightly constrained.

At the same time, the *low* group reports substantially more moderate ratings across all student-facing dimensions (Understanding 3.25; Engagement 3.54; Difficulty fit 3.50; Self-efficacy 3.31; Quiz usefulness 3.38). The key insight is **not** simply that the low group likes the system less, but **why**: the low group's weakest dimension is **self-efficacy** (3.31), which reflects confidence and perceived ability to succeed. This is consistent with a known pedagogical requirement: learners with larger prerequisite gaps often need heavier scaffolding (worked examples, smaller conceptual steps, more guided practice, and confidence-building feedback) rather than only simplified explanations. In other words, group conditioning helps, but the low group likely requires additional support mechanisms beyond difficulty calibration—e.g., prerequisite repair micro-lessons, misconception-targeted hints, and slower pacing with more formative checkpoints.

A second nuance is that **Understanding** is not strictly monotonic with proficiency (high: 3.76 is below medium: 4.00 and advanced: 4.62). This can occur when high-proficiency packs intentionally increase challenge: students may feel high engagement and fit, yet still report slightly lower "understanding" because the material is more demanding. This is a desirable trade-off if the goal is to keep stronger students appropriately challenged without slipping into triviality.

Panel B (teachers): adoption-relevant signals and residual friction. Teacher ratings are uniformly positive, with the strongest scores on **Trust** (4.03) and **Pedagogical fit** (3.96). This matters for deployment: even if a system is helpful, teachers will not adopt it if they cannot trust its correctness or if it violates classroom norms. The high trust score suggests that the verifier/repair design and staged alignment translate into a teacher-facing perception of reliability, not only offline benchmark gains.

Teachers also report positive workflow impact: **Time saved** is 3.81 and **Usefulness** is 3.69, indicating that the packs provide a meaningful starting point for lesson preparation. However, **Editing burden** is also 3.63, which is informative: teachers still expect some non-trivial editing even when the system is useful. Interpreting this jointly with the student results suggests a concrete actionable hypothesis: remaining edits likely concentrate on (i) strengthening scaffolds and pacing for the low group, and (ii) tightening teach–test alignment and terminology consistency across groups. This aligns with typical teacher-in-the-loop workflows, where AI-generated artifacts are adopted when they reduce blank-page effort but still require professional tailoring.

Implications. Taken together, these results support the practical claim that group-conditioned authoring yields measurable benefits in authentic use, but the benefit is not uniform. The system is most successful at producing engaging, well-calibrated materials for medium-to-advanced learners, while low-readiness learners require more explicit prerequisite repair and confidence-building supports. On the adoption side, teachers' high trust and pedagogical fit scores indicate that the system's reliability mechanisms are effective, but the non-trivial editing burden highlights the next engineering target: reduce edits by improving low-group scaffolding, making coverage more explicit, and surfacing editable pedagogical parameters (pace, worked-example ratio, and misconception emphasis).

D.7 [External] RQ4: Competence-only vs. competence+ethics.

Using the same staged recipe and backbones as RQ1, we compare two objective variants: competence-only and joint competence+ethics. Both are evaluated on the same benchmark suite and verifier rubrics, enabling a direct view of capability gains versus norm compliance.

D.8 Results: RQ4. Competence-only vs. Competence+Ethics Training

Table 8 shows that adding ethics constraints does not simply "regularize away" capability; instead, competence+ethics (CE) can match or exceed competence-only (C) on several downstream tasks while improving norm-grounded behavior. The clearest signal is that the full CE recipe (SFT+DPO+GRPO) yields a strong and often *more stable* profile across datasets, indicating that ethical constraints can function as structured inductive bias rather than a pure trade-off.

Table 8. Comparison of training objectives: competence-ethics vs competence-only/ All scores are reported as percentages (%).

Type	Model	Method	VNHSGE	ViLLM-Eval		VLMU-Vi-MQA	GK
			Acc.	Gen. Kno.	Acc.	Acc.	Acc.
CE	Qwen3-4B	DPO	62.66	57.40	54.42	66.01	65.17
		SFT + DPO	64.26	57.42	**60.52**	57.96	75.78
		SFT + DPO + GRPO	**68.87**	**58.75**	53.00	**66.34**	**86.87**
	Llama3.1-8B	DPO	35.84	60.52	57.42	51.49	67.13
		SFT + DPO	64.00	82.00	80.80	60.73	**72.00**
		SFT + DPO + GRPO	**70.56**	**84.00**	**83.00**	**62.05**	71.67
C	Qwen3-4B	DPO	64.35	82.70	72.00	66.01	66.09
		SFT + DPO	66.07	81.05	73.00	63.04	70.06
		SFT + DPO + GRPO	67.06	67.41	60.15	68.27	88.50
	Llama3.1-8B	DPO	66.13	65.41	63.75	53.47	68.00
		SFT + DPO	43.04	85.00	83.58	56.77	67.17
		SFT + DPO + GRPO	38.24	65.12	64.74	58.09	75.70

For Qwen3-4B, CE under SFT+DPO+GRPO achieves higher VNHSGE accuracy (68.87 vs. 67.06) and improves Vi-MQA, while also producing very strong GK retention (86.87), suggesting that ethics grounding does not incur knowledge collapse. For Llama3.1-8B, CE with SFT+DPO+GRPO performs strongly across VNHSGE and ViLLM-Eval (best-in-column for CE), whereas the competence-only variant shows larger regressions under later stages, indicating a less reliable optimization trajectory. RQ2 is answered as follows: competence-only training can yield higher scores on some capability-heavy settings, but competence+ethics training provides a better capability–norm balance and tends to be *more deployable* due to its improved consistency and stronger retention under the same staged recipe.

E System Demonstration Video

A supplementary video demonstration is provided to visually illustrate the end-to-end workflow of the proposed multi-agent teaching pack framework. This material is intended to complement the textual system description presented in earlier sections and to support reader understanding of the system's operational behavior in realistic classroom-oriented scenarios.

Content Overview. The demonstration presents a teacher-facing workflow that includes lesson submission and classroom configuration, automated lesson analysis and skill extraction, diagnostic construction and ability-based student grouping, and the generation of group-personalized teaching packs. The generated outputs include structured lesson plans, slide drafts, quizzes, and video scripts tailored to different student groups.

Scope. The video emphasizes interaction flow and system orchestration rather than internal model architectures, training procedures, or optimization details,

which are documented separately in Appendices B and C. It is provided solely for explanatory and reproducibility purposes and should not be interpreted as an empirical evaluation or performance comparison.

Privacy Statement. All content shown in the demonstration is fully anonymized. Any student profiles, classroom data, or instructional materials displayed in the video are either synthetically generated or illustrative examples created exclusively for demonstration purposes. No personally identifiable, institutional, or sensitive information is included.

Technical Details. The video is approximately [9] minutes in duration and is provided in MP4 format as part of the supplementary materials accompanying this paper.

F Supplementary Code and Artifacts

To support transparency and reproducibility, we provide a compressed archive containing the core artifacts required to understand, inspect, and reproduce the proposed multi-agent teaching pack framework. The archive is structured to facilitate independent verification while adhering to anonymity and data protection constraints.

Archive Contents. The supplementary materials include the following components:

- **Source Code.** Implementation of the multi-agent pipeline, encompassing agent orchestration logic, data schemas, and deterministic tools for lesson parsing, student grouping, and teaching pack generation.
- **Configuration Files.** YAML and JSON files specifying model settings, training hyperparameters, and pipeline options corresponding to the experimental setup described in Section 5.
- **Prompt Definitions.** System prompts for all core agents (cf. Appendix B), provided as plain-text files to enable inspection and reproducibility.
- **Sample Data.** A curated set of anonymized and synthetic inputs (e.g., lesson files and student profiles) and corresponding outputs (generated teaching packs) illustrating expected system behavior.
- **Documentation.** A README file detailing the project structure, execution instructions for the main pipeline, and guidelines for reproducing the reported experiments.

Ethical Compliance. All personally identifiable information has been removed from the archive. Any student data included are either synthetically generated or manually anonymized. Dataset references point exclusively to publicly available resources, and no proprietary or sensitive data are redistributed.

Scope. The archive is provided as supplementary material and is not required to understand the main contributions of this work. Rather, it serves as an auxiliary resource for readers who wish to examine the system implementation in greater detail or reproduce the reported results under comparable experimental conditions.

Access. The archive is available code in https://anonymous.4open.science/r/Teaching-pack-generator-completed-C912/Teaching-pack-generator-completed/README.md.

Limitations and Future Work

This work is a first step toward practical, group-conditioned authoring for heterogeneous classrooms. Our user study was conducted in one national context and across a limited number of classes; evaluating CoTeachPack across additional regions, curricula, and grade bands would strengthen evidence for generality. While the verifier suite enables scalable and consistent feedback, it may not fully capture local norms or rare edge cases; future work can refine these verifiers with larger expert panels and incorporate uncertainty-aware behaviors when the appropriate action is ambiguous. On the systems side, the current pipeline is largely sequential and relies on simple default grouping; we plan to improve efficiency with parallel execution and caching, and to explore adaptive grouping policies that better balance learning outcomes with teacher constraints. Finally, we primarily studied text-first teaching packs; extending the framework to richer multimodal assets (e.g., interactive activities and diagram-grounded materials) and running longer in-the-wild deployments will help assess sustained learning impact and teacher adoption over semesters.

G Conclusion

This appendix has presented a comprehensive account of the proposed multi-agent teaching pack framework, complementing the high-level description in the main paper. The documentation encompasses system architecture, agent-level design, training configurations, and supplementary artifacts, collectively clarifying how the framework operationalizes group-personalized instruction for heterogeneous classrooms.

Central to the proposed approach is the integration of pedagogically grounded task decomposition with explicit agent prompting, enabling the system to encode educational constraints—including curriculum alignment, difficulty calibration, and misconception handling—directly into the generation process. The staged training strategy, combining supervised fine-tuning with preference-based and group-relative optimization, further ensures reliable and teacher-aligned content generation across diverse instructional scenarios.

The supplementary materials—comprising a system demonstration video and a curated archive of source code and artifacts—are provided to support reproducibility and independent verification while adhering to ethical and anonymity requirements.

We hope this appendix serves as a useful reference for researchers investigating agentic systems for personalized and scalable educational support, and that it facilitates deeper engagement with the design choices and practical considerations underlying the proposed framework.

References

1. Bui, C.T., Son, N.T., Trang, T.V., Phung, L.V., Huy, P.N., Le, H.A., Van, Q.H., Do, P.N.T., Truc, V.L.T., Chau, D.T., Nguyen, L.M.: VMLU benchmarks: A comprehensive benchmark toolkit for Vietnamese LLMs. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 11495–11515. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.acl-long.563, https://aclanthology.org/2025.acl-long.563/
2. Nguyen, T.H., Le, A.C., Nguyen, V.C.: Villm-eval: A comprehensive evaluation suite for vietnamese large language models (2024), https://arxiv.org/abs/2404.11086
3. Xuan-Quy, D., Le, N.B., Vo, T.D., Phan, X.D., Ngo, B.B., Nguyen, V.T., Nguyen, T.M.T., Nguyen, H.P.: Vnhsge: Vietnamese high school graduation examination dataset for large language models (05 2023). https://doi.org/10.48550/arXiv.2305.12199