GlossaGen: Making Academic Translation Smarter with Glossing

Zixiao Wang[1], **Duzhen Zhang**[1], **Juntian Zhang**[2], **Yuhan Liu**[2],
Guoming Li[1], **Haolun Wu**[3], **Le Song**[1], **Xiuying Chen**[1]*
[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Renmin University of China
[3]McGill University
{zixiao.wang,duzhen.zhang,xiuying.chen}@mbzuai.ac.ae

Abstract

When reading foreign-language literature, non-native users often face significant challenges. Existing machine translation systems often obscure or mistranslate key terminology, while lay-oriented paraphrasing tends to oversimplify it, hindering readers from acquiring domain-specific technical vocabulary. To address this gap, we define a new task, **Glossing-Oriented Academic Translation (GOAT)**, which aims to produce translations adapted to a reader's academic level. We then propose **GlossaGen**, a comprehensive framework to address this task. GlossaGen combines a multi-agent data synthesis pipeline that generates large-scale, level-specific training data with a training strategy based on **dynamic adapter merging**, which balances task-level generalization and reader-level specialization through a "generalist" adapter and a fine-grained "expert" one. We evaluate GlossaGen on a synthesized benchmark using automatic metrics and large language model (LLM)-based assessments at both reader levels, together with a human evaluation study on the undergraduate setting. Across these evaluations, our approach outperforms strong baselines on most metrics. Overall, GlossaGen provides a practical step toward making scientific literature more accessible to non-native readers through more accurate translations and pedagogically appropriate, level-specific term explanations. We release our code and data to facilitate further research: GlossaGen.

1 Introduction

The dominance of English in global scientific communication presents a significant accessibility barrier for non-native speakers (Meneghini and Packer, 2007; Montgomery, 2013; Ramírez-Castañeda, 2020; Fiorini et al., 2023). They must simultaneously grapple with complex syntax and nuanced, domain-specific terminology, often without adequate support (Hanauer et al., 2019; Amano et al., 2021). This constant struggle not only slows down the process of literature review but also risks leading to critical misinterpretations (Flowerdew, 2001; Tardy, 2004). Thus, there is a pressing need for advanced tools that go beyond simple translation to actively support comprehension and learning.

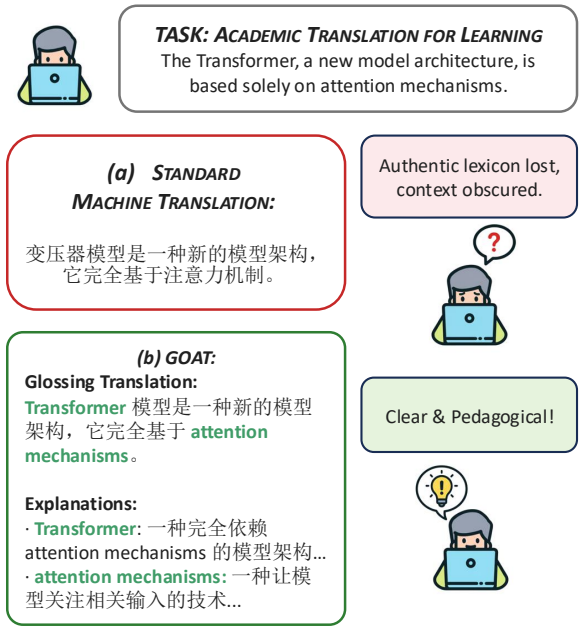

Figure 1: Comparison of Standard MT and GOAT for Academic Translation.

Existing tools are insufficient for supporting academic readers. Standard machine translation systems often produce awkward literal translations that obscure meaning and fail to connect concepts to their widely-used acronyms, a challenge noted in context-aware Machine Translation (MT) research (Jin et al., 2023; Yang et al., 2023). Conversely, lay paraphrasing, which aims to simplify content, is fundamentally misaligned with educational goals. By replacing core technical terms

* Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 19077–19095

(e.g., "myocardial infarction" with "heart attack"), it dismantles the very pedagogical scaffold that learners need to acquire a domain's specific vocabulary and engage with further literature (Tang et al., 2024; Kim and Shin, 2025).

To overcome these weaknesses, we introduce a new task: **Glossing-Oriented Academic Translation (GOAT)**. GOAT aims to translate scholarly text into a user's native language while intelligently preserving key English technical terms and providing clear, context-grounded explanations tailored to the user's academic level (e.g., undergraduate and graduate students). Although the GOAT formulation is not tied to a specific language pair, our empirical validation in this work is limited to English-to-Chinese translation in the computer-science domain; broader cross-lingual and cross-domain validation remains future work. Figure 1 provides a conceptual illustration of the GOAT task, contrasting its pedagogically-oriented output with that of a standard machine translation system.

Cognitive Load Theory suggests that redundant explanations can introduce extraneous cognitive load for knowledgeable readers, and that instructional support beneficial for novices may hinder experts, a phenomenon known as the expertise-reversal effect (Sweller, 1988). Motivated by this theory, we model reader proficiency as an explicit control variable in GOAT. In this initial work, we operationalize reader proficiency using two anchor levels (undergraduate vs. graduate) for controlled benchmarking, while future work may extend the framework to finer-grained or continuous user modeling.

To validate the premise of the GOAT task, we conducted a pilot user study with 15 students (8 undergraduates and 7 graduates). Participants performed a blind A/B test, comparing outputs from a standard MT system against our GOAT format based on comprehensibility and helpfulness. The results revealed a strong preference for the GOAT format, favored by 12 of the 15 participants (80%). A one-sided binomial test confirms this result is statistically significant ($p = 0.0176$), providing robust empirical motivation for our work.

The core challenge of GOAT is twofold. First, it demands that a model masters a difficult, constrained generation format: seamlessly weaving untranslated English technical terms into fluent target-language sentences, a structural requirement for which standard translation models are not trained. Second, and more profoundly, the task requires that the generated explanations be pedagogically sound and precisely tailored to the audience's cognitive level; a definition helpful for a graduate student may be opaque to an undergraduate.

To tackle the GOAT task, we propose **GlossaGen**, a comprehensive framework that addresses both the data scarcity and model adaptation challenges inherent to this new problem. Recognizing the absence of suitable training corpora, the first component of our framework is a novel multi-agent data synthesis pipeline. This pipeline leverages academic personas to automatically generate a large-scale, structured dataset with level-specific explanations, providing the necessary foundation for a model to learn the complex, constrained format of GOAT. With this level-specific data as a foundation, the second component of GlossaGen is an adaptive model architecture based on a dynamic adapter merging strategy. Specifically, we fine-tune the base language model with Low-Rank Adaptation (LoRA) adapters: two "expert" adapters, each on data tailored to a single academic level (undergraduate or graduate), and one "generalist" adapter on a mixture of all data to capture the core task mechanics. At inference time, we dynamically generate a high-capacity composite adapter by computing a weighted linear combination of the generalist adapter and the expert adapter matching the target user's level. This strategy allows a single base model to efficiently serve diverse user needs with high fidelity.

The main contributions of this paper are summarized as follows: (1) We formally define a new task, **Glossing-Oriented Academic Translation (GOAT)**, and construct a corresponding benchmark dataset to foster future research in this area. (2) We propose **GlossaGen**, an end-to-end framework featuring a multi-agent data synthesis pipeline to create level-specific data, and a dynamic adapter merging strategy to achieve nuanced pedagogical adaptation. (3) Through extensive experiments, including both automatic and human evaluations, we demonstrate that our approach achieves stronger performance than strong baselines in both translation quality and the pedagogical value of its explanations.

2 Related Work

Lay Paraphrasing and Scientific Text Simplification. Lay paraphrasing and scientific text simplification aim to enhance the accessibility of technical

documents for a general audience (Al-Thanyyan and Azmi, 2021). These tasks are crucial for bridging the knowledge gap between experts and the public. Research in this area has explored approaches such as evaluating meaning preservation via human comprehension questions (Agrawal and Carpuat, 2024) and leveraging edit-operation-aware training methods in biomedical text simplification (Knappich et al., 2023). To address the challenge of cross-domain generalization, Sci-LoRA (Cheng et al., 2025) proposes a mixture of LoRA modules that dynamically adapt to input domains, outperforming prior models across multiple scientific areas. The predominant strategy in these works involves selectively simplifying complex sentences and controlling transformation operations; however, this remains suboptimal for academic learners who must master, rather than bypass, technical terminology to engage with scholarly discourse.

Complexity-Controlled and Terminology-Constrained Machine Translation. Our GOAT setting is closely related to research on controlling the complexity/readability of MT outputs for different readers, where systems condition generation on a desired reading level or simplification degree (Marchisio et al., 2019; Agrawal and Carpuat, 2019). However, complexity-controlled MT typically produces a monolingual translation and does not explicitly preserve key English technical terms or provide pedagogical definitions that support vocabulary acquisition. Conversely, terminology-aware or terminology-constrained MT focuses on enforcing the presence of domain terms or preferred translations (Dinu et al., 2019; Semenov et al., 2025), but it does not address reader-adaptive explanation depth or the mixed-language "translation+gloss" format required for learning. GOAT complements these lines by treating term retention and level-specific glossing as first-class objectives: the output must remain a fluent target-language translation while embedding English terms and providing concise explanations tailored to a specified reader proficiency.

AI in Educational Technology. Artificial Intelligence in Education focuses on developing computational tools to create more personalized and effective learning environments (Abd-Alrazaq et al., 2023). Recent advancements leveraging LLMs have led to a surge of innovative applications. These range from AI-powered tutors that assist learners in STEM fields by explaining scientific concepts (Wang et al., 2024), to intelligent systems designed for language education that offer personalized vocabulary exercises (Leong et al., 2024). However, existing methods still often lack nuanced mechanisms for academic terminology acquisition in non-native speakers and the ability to dynamically adjust explanatory depth based on learner background.

3 Problem Formulation

We formally define the task of **Glossing-Oriented Academic Translation (GOAT)** as the process of generating a level-specific, glossed translation of a source scholarly text. The core challenge lies in producing outputs that are simultaneously tailored to a reader's academic proficiency while preserving and explaining key English terminology to facilitate learning.

Given a source academic text X_s and a specified reader level $L \in \{\text{undergraduate, graduate}\}$, the task is to train a model M parameterized by θ that learns the mapping to a structured output Y_g. This output is a tuple $Y_g = (K_p, T_g, E_k)$, where $K_p \subset X_s$ is a set of preserved keywords, T_g is the glossing translation of X_s containing the terms in K_p, and E_k is a set of corresponding explanations for each keyword.

The overall task is thus to model the conditional probability $P(Y_g|X_s, L)$ via the mapping:

$$Y_g = M(X_s, L; \theta)$$

A successful model must learn to adapt both the selection of keywords (K_p) and the explanatory style (E_k) based on the discrete input level L, making this a challenging structured prediction problem.

4 Methodology

Our GlossaGen framework is founded on a novel multi-agent data synthesis pipeline, designed to address the critical lack of level-specific training data. This pipeline transforms raw academic texts into structured supervised fine-tuning (SFT) instances through a series of specialized agents. We first describe this data synthesis process, followed by our model architecture and training strategy.

4.1 A Multi-Agent Pipeline for Data Synthesis

To address the critical lack of suitable training corpora for the GOAT task, we developed an automated multi-agent data synthesis pipeline. All agents in the pipeline are instantiated with the

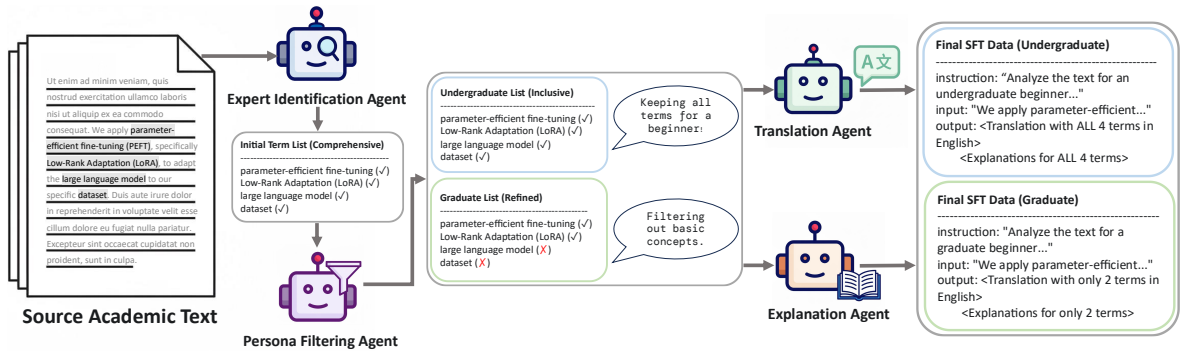

Figure 2: An overview of our multi-agent data synthesis pipeline. An Expert Identification Agent first extracts all potential terms from a Source Academic Text. A Persona Filtering Agent then creates two level-specific term lists, which in turn guide downstream agents to generate tailored SFT data for undergraduate and graduate levels.

GPT-4.1 model, chosen for its strong multilingual and reasoning capabilities. Following principles of modularity and specialization, a sequence of language-model-based agents, each an expert in a single, well-defined task, collaborate to transform raw scholarly texts into structured, level-specific SFT instances. The full workflow is illustrated in Figure 2.

Source Data. Our pipeline is designed to process scholarly texts, taking a corpus of academic abstracts as its primary input. The specific details of the corpus constructed and utilized for our experiments, including its source, size, and partitioning strategy, are described in Section 5.1.

Expert Term Identification Agent. This agent acts as a domain expert. It receives a raw abstract and is prompted to perform a comprehensive identification of all potential technical terms, including concepts, jargon, and model names. The agent ensures that the initial pool of terms is exhaustive, capturing any vocabulary that might require explanation.

Persona-based Term Filtering Agents. This stage is crucial for creating level-specific data. The initial term list from the Expert Term Identification Agent is passed to two parallel agents, each embodying a different academic persona (Duggan et al., 2014): **The Graduate Persona Agent**, which simulates a first-year graduate student and filters out foundational concepts, retaining only advanced terms; and **The Undergraduate Persona Agent**, which simulates a novice and inclusively retains almost all technical terms. This parallel processing results in two distinct, level-specific lists of key terms for each abstract.

Context-Aware Translation Agent. For each academic level, a dedicated translation agent receives the original abstract and the corresponding filtered list of terms. Its task is to translate the abstract into fluent Chinese while strictly preserving the terms on the list in their original English form and capitalization. This process generates the core translated text for our task.

Term Explanation Generation Agent. This agent is responsible for the "glossing" aspect. It takes the final list of English terms and the translated text as input. For each term, it generates a concise and context-aware explanation in Chinese, tailored to be accessible to the target academic level.

Final Data Formatting. The outputs from these agents are aggregated into a single structured data instance, formatted for instruction fine-tuning. This creates a rich dataset where each entry contains the original abstract, a level-specific translation with preserved terms, and corresponding explanations.

4.2 Model Architecture and Training Strategy

With the level-specific data as a foundation, we propose an adaptive model architecture, depicted in Figure 3, that leverages the parameter efficiency of LoRA (Hu et al., 2022) to create a repository of specialized adapters.

4.2.1 Training Specialized Expert and Generalist Adapters

Our training strategy moves beyond creating a single adapter. Instead, we perform three separate fine-tuning processes to generate a set of distinct LoRA adapters: an Undergraduate Expert Adapter ($\Delta W_{\text{undergrad}}$), fine-tuned exclusively on the undergraduate dataset to capture novice-specific nuances;

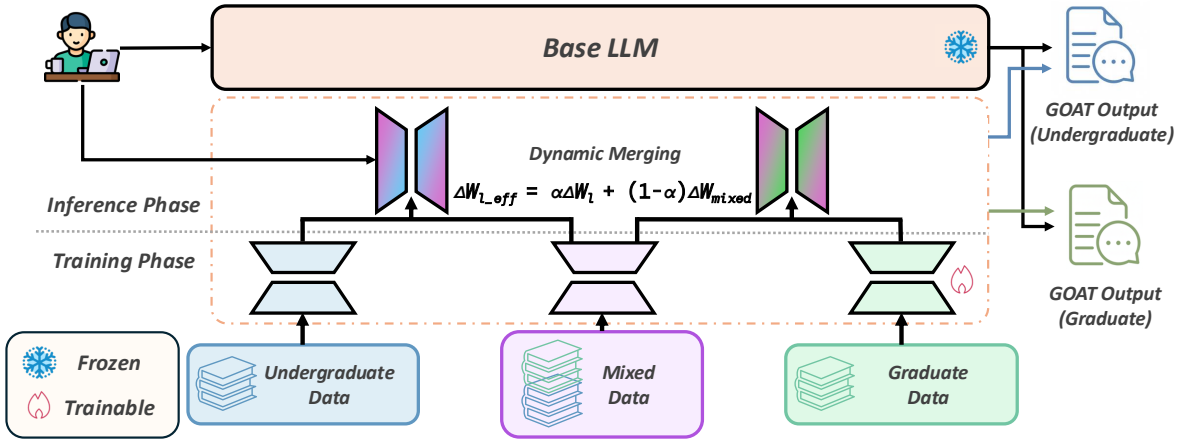

Figure 3: The model framework. **(Bottom) Training Phase:** The framework fine-tunes three distinct LoRA adapters: a "generalist" on the mixed dataset, alongside two "expert" adapters, each specialized for the undergraduate and graduate datasets. **(Top) Inference Phase:** The relevant expert and generalist adapters are dynamically merged via a weighted linear combination to form a composite adapter, which is then applied to the frozen base LLM to generate the level-specific GOAT output.

a Graduate Expert Adapter (ΔW_{grad}), fine-tuned on the graduate dataset to cater to an advanced audience; and a Generalist Adapter (ΔW_{mixed}), trained on a combination of both datasets to learn the core, shared capabilities of the task. This process yields three modular and specialized adapters, forming the foundation of our merging strategy.

4.2.2 Dynamic Adapter Merging for Inference

At inference time, our goal is to leverage the strengths of both the generalist and the relevant expert adapters. We introduce a **dynamic adapter merging** strategy, where we create a powerful composite adapter for each academic level by performing a weighted linear combination of the adapter weights.

The effective LoRA update, ΔW_{l_eff}, for a given user level $l \in \{\text{undergrad, grad}\}$ is computed as:

$$\Delta W_{l\_\text{eff}} = \alpha \Delta W_l + (1-\alpha)\Delta W_{\text{mixed}} \quad (1)$$

Here, α is a hyperparameter that balances the influence of the expert adapter (ΔW_l) and the generalist adapter, which we set to $\alpha = 0.5$ based on our experiments. This linear combination is performed as an offline preprocessing step, resulting in a single, unified adapter for each academic level. This strategy elegantly combines the specialized knowledge of two adapters into a single, efficient module, simplifying the serving architecture while maximizing performance.

5 Experiments

5.1 Datasets

Our benchmark dataset is constructed from the first three sentences of 8,000 abstracts from the arXiv "cs.CL" category (2018-2023) (arXiv.org submitters, 2024). This collection of source texts is first partitioned into training, validation, and test sets using an 85/5/10 split. Our multi-agent data synthesis pipeline (see Section 4.1) is then applied to each partition to generate two parallel datasets: one for the undergraduate persona and one for the graduate. Further details on the data source are available in Appendix A.

5.2 Synthetic Corpus Characterization and Quality Audit

To characterize the dataset beyond downstream model performance, we report both corpus-level statistics and a targeted manual audit.

Corpus statistics. Table 1 summarizes the scale and basic properties of our synthesized benchmark. In particular, the undergraduate subset contains systematically more preserved terms and longer explanations than the graduate subset, reflecting the intended pedagogical design.

Manual audit. We manually audited N=100 randomly sampled instances that are balanced across levels to assess: (i) term identification quality, (ii) gloss factual correctness, and (iii) format compliance with the GOAT constraints. As shown

Statistic	Undergrad	Grad
Avg. source length (tokens)	66.9	66.9
Avg. preserved terms / inst.	9.79	3.45
Avg. explanation length (chars)	428.1	205.1
Unique preserved terms	41,500	22,058
Top-10 term mass (%)	3.46	1.51

Table 1: Corpus statistics of the synthesized benchmark. Numbers are reported separately for the undergraduate and graduate subsets.

Audit metric	Undergrad	Grad
Term selection precision	94.0%	96.0%
Term selection recall	98.3%	93.7%
Gloss accuracy pass rate	100.0%	98.3%

Table 2: Manual audit results on $N = 50$ samples per level.

in Table 2, the majority of samples satisfy the core constraints.

5.3 Baselines

We compare our method against two main categories of strong baselines. The first is **Zero-shot Baselines**, where we evaluate several state-of-the-art, instruction-tuned LLMs without any in-domain training, as detailed in our result tables. The second category comprises Fine-tuned Models. To ensure a fair comparison, both our GlossaGen framework and the fine-tuned baselines are implemented on three powerful 7B/8B-parameter base LLMs: Llama-3.1-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), and Qwen3-8B (Yang et al., 2025). The fine-tuned baselines include: **(1) Standard SFT**, a conventional approach where a single LoRA adapter is trained on the combined undergraduate and graduate datasets; **(2) LoRA Soups** (Prabhakar et al., 2025), for which we average the weights of two adapters trained separately on the undergraduate and graduate data; and **(3) Sci-LoRA** (Cheng et al., 2025), where we treat our two level-specific adapters as experts and apply its dynamic weighting mechanism at runtime.

5.4 Evaluation Metrics

We evaluate model performance using both automatic and LLM-based metrics. For automatic evaluation, we report BLEU (Papineni et al., 2002), ROUGE-1, and ROUGE-L (Lin, 2004).

Recognizing the limitations of automatic metrics for this nuanced task, we also conduct an extensive LLM-based evaluation using GPT-4o (Hurst et al., 2024). We designed five metrics grouped into two key aspects. To assess task fidelity, we measure: Keyword Completeness (K1), ensuring all necessary terms are preserved; Instruction Adherence (K2), for correct format following; and Translation Fluency (K3), for readability of the mixed-language text. To evaluate pedagogical quality, we assess Explanation Simplicity (E1) for accessibility and Accuracy (E2) for factual correctness. The full scoring rubric and the prompt used are detailed in Appendix F.

5.5 Implementation Details

All experiments were conducted on an NVIDIA A100 GPU. For our LoRA-based methods, we set the LoRA rank to $r = 16$ and the balancing hyperparameter for our dynamic adapter merging strategy to $\alpha = 0.5$. A comprehensive description of all training hyperparameters is provided in Appendix B.

5.6 Main Results

Tables 3 and 4 report the full results on the undergraduate and graduate test sets. We discuss the undergraduate setting first because it is also the setting used in our human evaluation, but we now present both reader levels in the main text. Across both settings, GlossaGen consistently outperforms competitive baselines on most automatic and LLM-based metrics. The graduate test set follows the same overall pattern as the undergraduate one. For example, on Mistral-7B-Instruct, GlossaGen improves over Sci-LoRA from 24.72 to 26.53 BLEU and from 74.74 to 77.37 ROUGE-L; on Qwen3-8B, it also improves K2/E2 from 4.61/4.66 to 4.85/4.87. These results show that the gains of GlossaGen are consistent across reader levels rather than being limited to the undergraduate setting.

5.6.1 Performance on Translation Quality Metrics

Our framework achieves consistently stronger performance on traditional translation quality metrics such as BLEU and ROUGE-L. A clear gap remains between zero-shot models and all fine-tuned approaches, confirming that GOAT is a non-trivial task that requires specialized training. While standard LoRA fine-tuning already provides substantial improvements over zero-shot baselines, GlossaGen further improves upon this baseline across both reader levels. For instance, on the undergraduate test set, GlossaGen based on Mistral-7B-Instruct

Model	Automatic Metrics			LLM-based Evaluation				
	BLEU	R-1	R-L	K1	K2	K3	E1	E2
Baselines								
Zero-shot								
DeepSeek-R1-Distill-Llama-8B	0.41	20.38	19.88	3.07	2.28	3.42	3.14	4.09
Mistral-7B-Instruct	0.53	20.22	19.80	2.77	2.05	3.19	3.09	3.96
Llama-3.1-8B-Instruct	0.75	23.06	22.56	3.05	2.17	3.33	3.17	4.07
Qwen3-8B	1.56	30.17	29.33	3.36	2.55	3.51	3.44	4.40
DeepSeek-R1-0528-Qwen3-8B	2.15	33.68	32.91	3.65	2.89	3.71	3.62	4.55
QwQ-32B	2.25	66.18	63.75	4.55	3.50	3.95	3.61	4.69
Gemma-3-27B-IT	3.62	41.13	40.37	3.83	2.88	3.79	3.63	4.57
Qwen3-32B	5.17	67.73	65.36	4.51	3.69	3.88	3.62	4.72
Standard LoRA								
Qwen3-8B (LoRA)	32.26	87.45	84.03	4.78	4.66	4.38	3.98	4.72
Llama-3.1-8B-Instruct (LoRA)	31.51	89.11	85.12	4.70	4.58	4.34	3.93	4.67
Mistral-7B-Instruct (LoRA)	32.62	89.34	85.57	4.78	4.64	4.38	3.95	4.72
LoRA Soups								
LoRA Soups (Qwen3-8B)	31.46	83.69	82.28	4.72	4.39	4.31	4.01	4.67
LoRA Soups (Llama-3.1-8B-Instruct)	32.64	86.70	84.81	4.66	4.39	4.26	3.91	4.60
LoRA Soups (Mistral-7B-Instruct)	32.93	86.17	84.73	4.77	4.33	4.28	3.88	4.61
Sci-LoRA								
Sci-LoRA (Qwen3-8B)	32.83	88.48	84.12	4.85	4.58	4.49	4.46	4.87
Sci-LoRA (Llama-3.1-8B-Instruct)	30.32	88.58	84.11	4.86	4.66	4.47	4.46	4.84
Sci-LoRA (Mistral-7B-Instruct)	33.07	89.35	85.63	4.86	4.65	4.50	4.43	4.86
Ours (GlossaGen)								
GlossaGen (Qwen3-8B)	34.44	89.53	85.82	4.91	4.77	4.68	**4.50**	**4.89**
GlossaGen (Llama-3.1-8B-Instruct)	33.05	**90.27**	86.17	4.90	4.74	4.67	4.45	4.86
GlossaGen (Mistral-7B-Instruct)	**34.68**	90.23	**86.53**	**4.93**	**4.78**	**4.70**	4.47	4.88

Table 3: Overall model performance on the **Undergraduate-Level** test set. Automatic metrics are reported on a 0-100 scale, while LLM-based evaluation uses a 1-5 scale. K1: Completeness, K2: Adherence, K3: Fluency, E1: Simplicity, E2: Accuracy. **Bold** indicates the best performance for each metric.

achieves 34.68 BLEU, compared to 32.62 for standard LoRA; on the graduate test set, the corresponding scores are 26.53 and 24.51.

The same trend also holds against stronger adapter-combination baselines such as LoRA Soups and Sci-LoRA. On the undergraduate test set, GlossaGen achieves 34.68 BLEU, surpassing both LoRA Soups (32.93) and Sci-LoRA (33.07). On the graduate test set, GlossaGen reaches 26.53 BLEU, compared with 23.98 and 24.72, respectively. We attribute this consistent advantage to our merging strategy: instead of relying on uniform averaging or complex runtime weighting, our method combines a generalist adapter with a level-matched expert adapter, yielding a more balanced and effective representation for the GOAT task.

5.6.2 Evaluating Pedagogical Quality via LLM

The LLM-based evaluation provides additional insights into the pedagogical quality of the generated outputs. While all fine-tuned methods perform well on translation-focused metrics (K1–K3), GlossaGen shows consistent advantages in explanation-related metrics (E1 and E2). For example, on the undergraduate test set, the Qwen3-8B model achieves the highest scores in Explanation Simplicity (E1: 4.50) and Accuracy (E2: 4.89), compared to 4.46 and 4.87 for Sci-LoRA. These results suggest that the "generalist" adapter effectively captures the structural properties of high-quality explanations, while the "expert" adapter refines them to the target user level. Taken together, these findings indicate that GlossaGen supports both accurate translation and effective explanation, aligning well with the educational goals of the GOAT task.

6 Analysis and Discussion

6.1 Ablation Study

To assess the contribution of each adapter in GlossaGen, we conduct an ablation study on the undergraduate test set, using Mistral-7B-Instruct-v3 as the backbone. We report four representative metrics: BLEU and ROUGE-L to assess overall translation quality, K2 (Adherence) to evaluate con-

Model	Automatic Metrics			LLM-based Evaluation				
	BLEU	R-1	R-L	K1	K2	K3	E1	E2
Baselines								
Zero-shot								
DeepSeek-R1-Distill-Llama-8B	1.27	31.28	30.63	3.07	2.28	3.37	3.18	4.05
Mistral-7B-Instruct	1.94	31.79	31.00	2.92	2.09	3.19	3.05	3.98
Llama-3.1-8B-Instruct	2.00	35.42	34.68	3.10	2.17	3.33	3.18	4.08
Qwen3-8B	2.48	38.16	37.40	3.42	2.63	3.49	3.39	4.31
DeepSeek-R1-0528-Qwen3-8B	2.98	43.56	42.75	3.63	3.13	3.79	3.57	4.53
QwQ-32B	3.75	62.93	60.93	4.62	3.55	3.98	3.60	4.75
Gemma-3-27B-IT	5.38	47.28	46.59	3.56	2.78	3.69	3.55	4.46
Qwen3-32B	5.76	62.67	60.99	4.50	3.75	3.94	3.57	4.71
Standard LoRA								
Qwen3-8B (LoRA)	22.68	72.26	71.00	4.20	4.28	4.33	3.95	4.64
Llama-3.1-8B-Instruct (LoRA)	24.42	75.08	73.60	4.26	4.18	4.30	3.95	4.63
Mistral-7B-Instruct (LoRA)	24.51	74.28	72.74	4.17	3.93	4.25	3.95	4.64
LoRA Soups								
LoRA Soups (Qwen3-8B)	22.18	72.79	71.46	4.15	4.24	4.29	3.94	4.67
LoRA Soups (Llama-3.1-8B-Instruct)	24.22	76.34	74.46	4.12	4.08	4.26	3.98	4.67
LoRA Soups (Mistral-7B-Instruct)	23.98	76.06	74.49	4.16	4.15	4.17	3.90	4.56
Sci-LoRA								
Sci-LoRA (Qwen3-8B)	23.23	75.39	73.42	4.50	4.61	4.62	4.38	4.66
Sci-LoRA (Llama-3.1-8B-Instruct)	23.21	75.30	73.83	4.38	4.53	4.45	4.28	4.56
Sci-LoRA (Mistral-7B-Instruct)	24.72	76.75	74.74	4.37	4.44	4.47	4.31	4.58
Ours (GlossaGen)								
GlossaGen (Qwen3-8B)	25.97	77.13	75.65	**4.74**	4.85	**4.68**	**4.44**	**4.87**
GlossaGen (Llama-3.1-8B-Instruct)	26.27	78.62	77.02	4.69	**4.88**	4.65	4.43	4.85
GlossaGen (Mistral-7B-Instruct)	**26.53**	**78.88**	**77.37**	4.65	4.86	4.64	4.41	4.85

Table 4: Overall model performance on the **Graduate-Level** test set. Automatic metrics are reported on a 0-100 scale, while LLM-based evaluation uses a 1-5 scale. K1: Completeness, K2: Adherence, K3: Fluency, E1: Simplicity, E2: Accuracy. **Bold** indicates the best performance for each metric.

formity to the glossing format, and E2 (Accuracy) to measure the pedagogical quality of explanations. We compare the full model against two ablated versions: **w/o Generalist**, which removes the generalist adapter and relies solely on specialized knowledge, and **w/o Expert**, which removes the expert adapter and thus lacks task-specific specialization.

Configuration	BLEU	R-L	K2	E2
GlossaGen (Full)	**34.68**	**86.53**	**4.78**	**4.88**
w/o Expert	32.62	85.57	4.64	4.72
w/o Generalist	31.29	83.65	4.65	4.70

Table 5: Ablation study on Mistral-7B-Instruct-v3. Both the generalist and expert adapters are crucial for optimal performance.

Results in Table 5 show that the full GlossaGen model achieves the best performance across all metrics. Removing the generalist adapter leads to a notable drop in BLEU and ROUGE-L, highlighting the importance of generalist knowledge for translation quality. By contrast, removing the expert adapter results in a larger decline on E2, confirming that specialized expertise is critical for producing pedagogically accurate explanations. Both ablations also slightly reduce K2, suggesting that each adapter contributes to format adherence.

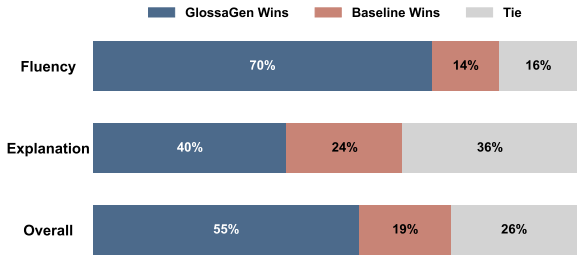

Figure 4: Human evaluation results (% of preference). GlossaGen consistently outperforms the baseline across all dimensions, with a particularly strong advantage in translation fluency.

6.2 Human Evaluation

We recruited two graduate students in computer science, both proficient in English and Chinese, who were not involved in this project. We intentionally

used bilingual CS graduate annotators because this experiment is designed as an expert comparative judgment of system outputs, rather than as a downstream learning study with novice end users. In particular, evaluating translation fluency, factual correctness, and whether explanations are appropriately selective under the GOAT format requires sufficient bilingual domain expertise to judge reliably. At the same time, annotators were explicitly instructed to assess the outputs with undergraduate readers in mind, that is, to judge which output would be more helpful and appropriate for an undergraduate audience.

We therefore focus the human evaluation on the undergraduate test set, which is also the setting foregrounded in the main text, and randomly selected 100 samples for a blind A/B comparison between our best-performing model, GlossaGen based on Mistral-7B, and the strongest baseline, Sci-LoRA based on Mistral-7B.

The evaluators were asked to indicate their preference on two key dimensions that directly reflect the core goals of the GOAT task: **(1) Translation Fluency**, to assess the quality of the core translation, and **(2) Explanation Quality**, to assess the pedagogical value of the generated glosses.

As shown in Figure 4, GlossaGen was strongly preferred by human evaluators, with the most significant advantage on Translation Fluency, while also performing better on Explanation Quality. We measured inter-annotator agreement using Cohen's Kappa, which yielded a score of 0.62, indicating substantial agreement. Notably, this preference pattern is directionally consistent with the GPT-4o scores for the same model pair on the undergraduate set: GPT-4o also favors GlossaGen over Sci-LoRA on translation fluency (K3) and explanation-related quality (E1/E2).

6.3 Case Study

To complement our aggregate metrics, Table 10 in Appendix G presents a qualitative case study comparing our framework against two baselines. The example is selected to test a model's ability to distinguish true domain-specific jargon from general academic vocabulary.

The baselines illustrate common failure modes. The zero-shot model Gemma-3-27B-IT shows fundamental limitations: it incorrectly translates all terms, including "neural auto-regressive language models", thus violating the core GOAT format. The Standard SFT baseline reproduces the format but demonstrates poor judgment. It unnecessarily preserves and explains common academic phrases such as "critical thinking curriculum", treating them as technical jargon. While structurally valid, this behavior introduces noise and reduces its usefulness for learners.

In contrast, our GlossaGen framework produces precise and targeted outputs. It identifies "neural auto-regressive language models" as the only technical term requiring preservation, while appropriately translating "critical thinking curriculum." This outcome reflects the synergy of our training strategy: the "generalist" adapter captures the task structure, whereas the "expert" adapter provides the fine-grained, level-specific judgment needed for focused and pedagogically effective outputs.

7 Conclusion

In this paper, we introduced the task of Glossing-Oriented Academic Translation (GOAT), aimed at supporting non-native speakers in comprehending technical literature. We proposed GlossaGen, a framework with a multi-agent data synthesis pipeline and a dynamic adapter merging strategy. Through automatic and human evaluations, our results show consistent improvements over competitive baselines, particularly in generating accurate translations accompanied by pedagogically useful explanations. These findings highlight GlossaGen as a step towards scalable and accessible scientific communication. Alongside the framework itself, the public release of our benchmark dataset provides a valuable new resource for the community, intended to foster future research in this area. Future work will explore automatic modeling of user proficiency for personalized glossing and extending the framework to other domains where accessibility is critical, such as medicine and law.

Limitations

While our work presents a promising direction for academically oriented translation, we acknowledge several limitations.

First, our empirical validation is restricted to the computer-science domain and English-to-Chinese translation. Although the GOAT task formulation is not tied to a specific language pair, broader cross-lingual and cross-domain validation remains necessary. In particular, because our synthesis and generation pipeline relies on large language models, performance may vary across target languages

and technical domains depending on the underlying model's multilingual and domain-specific capabilities.

Second, our current framework operationalizes reader proficiency using two pre-defined, discrete levels (undergraduate and graduate). While this provides a controlled setting for benchmarking reader-adaptive glossing, real users' background knowledge is more continuous and multidimensional. A more fully adaptive system would benefit from finer-grained or continuous representations of user knowledge, as well as additional personalization factors such as disciplinary background.

We hope future work will address these limitations by extending GOAT to broader domains, additional target languages, and richer models of reader proficiency.

Ethical Considerations

Our work aims to enhance the accessibility of scientific knowledge for non-native speakers, a goal with positive societal benefits. The primary ethical considerations for our framework, GlossaGen, revolve around the data and potential misuse.

The academic abstracts used for data synthesis were sourced from arXiv, a public repository of scholarly articles, in adherence with its terms of use. Our multi-agent data synthesis pipeline is designed to process this public data and does not involve any private or personally identifiable information.

Like any generative model, GlossaGen could potentially produce inaccurate translations or explanations. While our evaluations show high performance, end-users should be encouraged to use the outputs as a supplementary learning aid rather than an infallible authority. Furthermore, the base LLMs used for fine-tuning may carry inherent societal biases. Although our task is focused on technical terminology, which is less susceptible to such biases, future work could involve auditing the model for potential biased language in its translations and explanations, especially when applied to more socially-oriented academic disciplines.

References

Alaa Abd-Alrazaq, Rawan AlSaad, Dari Alhuwail, Arfan Ahmed, Padraig Mark Healy, Syed Latifi, Sarah Aziz, Rafat Damseh, Sadam Alabed Alrazak, and Javaid Sheikh. 2023. Large language models in medical education: opportunities, challenges, and future directions. *JMIR medical education*, 9(1):e48291.

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4545–4550, Hong Kong, China. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2024. Do text simplification systems preserve meaning? a human evaluation via reading comprehension. *Transactions of the Association for Computational Linguistics*, 12:432–448.

Suha S Al-Thanyyan and Aqil M Azmi. 2021. Automated text simplification: a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Tatsuya Amano, Clarissa Rios Rojas, Yap Boum II, Margarita Calvo, and Biswapriya B Misra. 2021. Ten tips for overcoming language barriers in science. *Nature Human Behaviour*, 5(9):1119–1122.

arXiv.org submitters. 2024. arxiv dataset.

Ming Cheng, Jiaying Gong, and Hoda Eldardiry. 2025. Sci-LoRA: Mixture of scientific LoRAs for cross-domain lay paraphrasing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18524–18541, Vienna, Austria. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3073, Florence, Italy. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

EM Duggan, CMP O'Tuathaigh, M Horgan, and S O'Flynn. 2014. Enhanced research assessment performance in graduate vs. undergraduate-entry medical students: implications for recruitment into academic medicine. *QJM: An International Journal of Medicine*, 107(9):735–741.

Susanna Fiorini, Arda Tezcan, Tom Vanallemeersch, Sara Szoc, Kristin Migdisi, Laurens Meeus, and Lieve Macken. 2023. Translations and open science: Exploring how translation technologies can support multilingualism in scholarly communication. In *International Conference on Human-Informed Translation and Interpreting Technology (HiT-IT 2023)*, pages 41–51. INCOMA Ltd.

John Flowerdew. 2001. Attitudes of journal editors to nonnative speaker contributions. *TESOL quarterly*, 35(1):121–150.

David I Hanauer, Cheryl L Sheridan, and Karen Englander. 2019. Linguistic injustice in the writing of research articles in english as a second language: Data from taiwanese and mexican researchers. *Written Communication*, 36(1):136–154.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Linghao Jin, Jacqueline He, Jonathan May, and Xuezhe Ma. 2023. Challenges in context-aware neural machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15246–15263, Singapore. Association for Computational Linguistics.

Euigyum Kim and Hyo Jeong Shin. 2025. Leveraging large language model for automatic translation of educational content: Exploring the effectiveness of curriculum-aware prompt engineering. *Korean Educational Research Association*.

Valentin Knappich, Simon Razniewski, and Annemarie Friedrich. 2023. Boschai @ plaba 2023: Leveraging edit operations in end-to-end neural sentence simplification. *Preprint*, arXiv:2311.01907.

Joanne Leong, Pat Pataranutaporn, Valdemar Danry, Florian Perteneder, Yaoli Mao, and Pattie Maes. 2024. Putting things into context: Generative ai-enabled context personalization for vocabulary learning improves learning motivation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Kelly Marchisio, Jinhua Du, and Kevin Duh. 2019. Controlling the reading level of machine translation output. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 193–203, Dublin, Ireland. European Association for Machine Translation.

Rogerio Meneghini and Abel L Packer. 2007. Is there science beyond english? initiatives to increase the quality and visibility of non-english publications might help to break down language barriers in scientific communication. *EMBO reports*, 8(2):112–116.

Scott L Montgomery. 2013. *Does science need a global language?: English and the future of research*. University of Chicago Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. 2025. Lora soups: Merging loras for practical skill composition tasks. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 644–655.

Valeria Ramírez-Castañeda. 2020. Disadvantages in preparing and publishing scientific papers caused by the dominance of the english language in science: The case of colombian researchers in biological sciences. *PloS one*, 15(9):e0238372.

Kirill Semenov, Xu Huang, Vilém Zouhar, Nathaniel Berger, Dawei Zhu, Arturo Oncevay, and Pinzhen Chen. 2025. Findings of the WMT25 terminology translation task: Terminology is useful especially for good MTs. In *Proceedings of the Tenth Conference on Machine Translation*, pages 554–576, Suzhou, China. Association for Computational Linguistics.

John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285.

Kenan Tang, Peiyang Song, Yao Qin, and Xifeng Yan. 2024. Creative and context-aware translation of east asian idioms with gpt-4. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9285–9305.

Christine Tardy. 2004. The role of english in scientific communication: lingua franca or tyrannosaurus rex? *Journal of English for academic purposes*, 3(3):247–269.

Rose E Wang, Ana T Ribeiro, Carly D Robinson, Susanna Loeb, and Dora Demszky. 2024. Tutor copilot: A human-ai approach for scaling real-time expertise. *arXiv preprint arXiv:2410.03017*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jian Yang, Yuwei Yin, Shuming Ma, Liqun Yang, Hongcheng Guo, Haoyang Huang, Dongdong Zhang, Yutao Zeng, Zhoujun Li, and Furu Wei. 2023. Hanoit: Enhancing context-aware translation via selective context. In *International Conference on Database Systems for Advanced Applications*, pages 471–486. Springer.

Appendix

A Dataset Details

The source abstracts for our dataset were randomly sampled from the publicly available arXiv Dataset on Kaggle: https://www.kaggle.com/datasets/Cornell-University/arxiv. We filtered the dataset to include only papers from the Computation and Language ('cs.CL') category with original publication dates between 2018 and 2023.

B Training Hyperparameters

All LoRA-based fine-tuning experiments were conducted with a consistent set of hyperparameters to ensure fair comparison. We used an NVIDIA A100 GPU with 80GB of memory for all training runs. The fine-tuning was performed using the LoRA method with bfloat16 precision.

The LoRA rank was set to $r = 16$ and was applied to all linear layers of the base models (Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen3-8B). For optimization, we used the AdamW optimizer with a learning rate of 8.0×10^{-5}. The learning rate was managed by a cosine decay scheduler with a warmup ratio of 0.1 over 3 training epochs. The training was configured with a per-device batch size of 1 and 8 gradient accumulation steps, resulting in an effective batch size of 8.

C Pilot Cross-Lingual Transfer Check

To provide a preliminary signal of transferability, we conducted a pilot study on **English–Korean** using the same undergraduate synthesis pipeline and the same GlossaGen training recipe, without architectural changes. The results are shown in Table 6. We observe consistent trends: GlossaGen continues to outperform strong baselines on both task fidelity and explanation quality.

Model	BLEU	ROUGE-L	K2	E2
Qwen3-32B	7.35	70.22	3.80	4.65
Standard LoRA	31.52	84.12	4.57	4.66
GlossaGen	**33.92**	**86.02**	**4.70**	**4.76**

Table 6: Pilot cross-lingual transfer results on English-Korean translation. GlossaGen and Standard LoRA settings are based on the Mistral-7B-Instruct backbone.

α	BLEU	ROUGE-L	K2	E2
0.00	32.62	85.57	4.64	4.72
0.25	33.07	85.63	4.72	4.85
0.50	**34.68**	**86.53**	**4.78**	**4.88**
0.75	31.57	85.89	4.68	4.74
1.00	31.29	83.65	4.65	4.70

Table 7: Sensitivity analysis of α in dynamic adapter merging.

Model	BLEU	R-L	K1	K2	E2
GPT-4.1	12.50	70.77	3.68	3.10	4.59
GlossaGen	**34.68**	**86.53**	**4.93**	**4.78**	**4.88**

Table 8: Comparison with a frontier proprietary LLM (GPT-4.1) on the undergraduate test set. GlossaGen setting is based on the Mistral-7B-Instruct backbone.

D Sensitivity Analysis of the Adapter Merging Weight

We sweep $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ on the development set, and evaluate on the undergraduate test set using the same backbone and metrics as in Section 5. Table 7 reports representative metrics.

E Comparison with a Frontier Proprietary LLM

To estimate the performance ceiling of GOAT-style generation under prompt-only control, we evaluate GPT-4.1 as a strong proprietary baseline on the undergraduate test set. GPT-4.1 is prompted with the same task description and target audience specification as other zero-shot baselines. Table 8 reports the results.

F Details of LLM-based Evaluation

To ensure a consistent and replicable evaluation, all model outputs were assessed by GPT-4o using a standardized scoring rubric and a detailed prompt. We first define the five core metrics of our rubric and then present the complete prompt provided to the judge.

F.1 Scoring Rubric

To standardize the evaluation, we established a detailed scoring rubric with scores ranging from 1.0 (very poor) to 5.0 (excellent). The rubric is organized into two main categories, containing five fine-grained metrics designed to assess distinct aspects of the model's performance:

A. Keyword and Translation Dimensions This category evaluates the model's core ability to perform the mixed-language translation task correctly and fluently.

- **K1 - Keyword Identification Completeness:** This metric evaluates whether the model successfully identified and preserved all necessary technical terms from the source text. It serves as a foundational check on the model's attention to key concepts.

- **K2 - Instruction Adherence:** This assesses the model's ability to strictly follow the core instruction of embedding English keywords directly into the translated sentences, measuring its control over the output format.

- **K3 - Translation Fluency with Mixed Terms:** This metric measures the linguistic quality and readability of the final translation, focusing on how naturally the English terms are integrated into the Chinese sentences.

B. Term Explanation Dimensions This category evaluates the pedagogical quality of the generated explanations, which is a central goal of the GOAT task.

- **E1 - Explanation Simplicity:** This evaluates whether the explanations are presented in a clear and accessible manner suitable for the specified academic level (e.g., undergraduate), avoiding overly technical jargon.

- **E2 - Explanation Accuracy:** This metric ensures that in the process of simplification, the explanations remain factually correct and do not misrepresent the core meaning of the technical concepts.

The full prompt in Table 11 contains the detailed scoring criteria for each of these metrics that were provided to the judge.

F.2 Full Evaluation Prompt

The complete prompt provided to GPT-4o is detailed in Table 11. Placeholders were populated programmatically for each sample.

G Additional Qualitative Case Studies

This appendix complements the qualitative discussion in Section 6.3 by providing representative, color-coded examples. We select cases that simultaneously contain (i) true domain-specific technical terms that should be preserved in English, and (ii) general academic phrases that should be translated normally. Red highlights indicate constraint violations or unhelpful behaviors, while green highlights indicate correct and pedagogically desirable behaviors.

Metric (test outputs)	Undergrad	Grad
Avg. # preserved terms / inst.	9.49	3.40
Avg. explanation length (chars)	424.1	202.7
Term overlap rate (%)	96.14	96.73

Table 9: Quantifying reader-level differences in generated outputs.

H Reader-Level Adaptation Analysis

A central goal of GOAT is that the system should adjust its glossing behavior to the target reader level. To directly quantify level adaptation, we compare the outputs generated under the undergraduate and graduate conditions. Table 9 shows that graduate outputs contain substantially fewer preserved terms and shorter (more specialized) glosses, indicating that the model filters out foundational concepts for expert readers.

Positioning of discrete levels. While real proficiency is continuous, we treat undergraduate/graduate as two anchor points for controlled benchmarking and to connect with prior work on controlling output complexity in MT and generation. We discuss future extensions to finer-grained or continuous modeling in Section 7.

I Prompts for Multi-Agent Data Synthesis

This section details the system prompts provided to the language models for each agent in our multi-agent data synthesis pipeline. The pipeline begins with an Expert Term Identification agent (Table 12), which performs an initial, comprehensive extraction of all potential technical terms. This initial list is then processed by two parallel Persona-based Filtering agents, one embodying an undergraduate student (Table 13) and the other a graduate student (Table 14), to create level-specific term sets.

The two level-specific term lists then guide the downstream content generation. This process begins with an essential quality control step, Orthographic Verification (Table 15), to ensure term capitalization is consistent with the source text. Sub-

Case	Generated Output & Analysis
Source Text	This paper takes a first step towards a critical thinking curriculum for neural auto-regressive language models.
Zero-shot (Gemma-3-27B-IT)	**Translation:** 本文为建立一个面向神经自回归语言模型的批判性思维课程迈出了第一步。 **Explanations:** (1) Neural Auto-Regressive Language Models: 一类利用神经网络预测序列中下一个词语或符号的语言模型，例如GPT系列。 ***Analysis:*** *Fails to adhere to the core task constraint. The model translates all terms, violating the instruction to preserve keywords. It then explains the original English term, creating a self-contradictory output and demonstrating a fundamental failure to follow the GOAT format.*
Standard SFT (Mistral-7B-Instruct)	**Translation:** 本文首次探讨了针对 neural auto-regressive language models 的 critical thinking curriculum。 **Explanations:** (1) Neural Auto-Regressive Language Models: 神经自回归语言模型是一类利用神经网络根据已有文本逐步生成下一个词或句子的模型，广泛应用于自然语言处理。 (2) Critical Thinking: 批判性思维是指有系统、逻辑和客观地分析信息并做出判断的能力，常用于教育和学习领域。 (3) Curriculum: 课程设计指的是为学习或训练过程设计的系统性计划，通常包括学习内容、顺序和目标。 ***Analysis:*** *Reproduces the format but lacks discerning judgment. It correctly preserves the main technical term but fails to distinguish general academic language from domain-specific jargon. Explaining common phrases like "critical thinking curriculum" adds unnecessary noise and reduces the pedagogical value for the learner.*
Ours (GlossaGen) (Mistral-7B-Instruct)	**Translation:** 本文首次探索了针对 neural auto-regressive language models 的批判性思维课程。 **Explanations:** (1) Neural Auto-Regressive Language Models: 神经自回归语言模型是一类通过逐步预测下一个词来生成文本的人工智能模型，常用于自然语言处理领域。 ***Analysis:*** *Exhibits precise and targeted judgment. It correctly identifies the single, core technical term ('neural auto-regressive...') while correctly treating 'critical thinking curriculum' as general language to be translated. This demonstrates that our model has learned not just the format, but the semantic criteria for term identification, producing a focused and pedagogically effective output.*

Table 10: A comparative case study on a challenging example from our undergraduate test set. Red text highlights errors or weaknesses, while green text indicates strengths and correct behavior. The table showcases GlossaGen's superior discerning judgment in identifying only the true domain-specific technical term, unlike the inconsistent zero-shot model and the noisy SFT baseline.

sequently, the two primary generation agents, the Context-Aware Translation Agent (Table 16) and the Term Explanation Agent (Table 17), use these verified lists to produce the core content for each level. Finally, to prepare the data for supervised fine-tuning, a concluding synthesis step assembles these outputs and generates a first-person reasoning narrative, with distinct prompts for the undergraduate (Table 18) and graduate (Table 19) personas.

Full Prompt for LLM-based

You are a professional AI evaluation expert, tasked with assessing the performance of an academic translation support system. Your evaluation must focus on **5 core metrics**.

Target Audience: {target_audience}
Core Task: Translate an academic abstract into Chinese, keep technical terms in their original English, and provide simple explanations for them.

INPUT DATA

Source English Text:
`{original_text}`
Model Output:
- **Translation:** `{translation}`
- **Term Explanations:** `{terminology_explanation}`

EVALUATION METRICS AND SCORING RUBRIC (1.0-5.0 Scale)

Scoring Precision: Please provide scores from 1.0 to 5.0, with one decimal place allowed (e.g., 3.2, 4.1) for fine-grained assessment.

A. Keyword and Translation Dimensions (3 metrics)

K1 - Keyword Identification Completeness:
- *Task:* Assess if the model identified and extracted **all** relevant technical terms.
- *Focus on:* Term coverage, identification accuracy, any omissions.
- *Scoring:* 5.0=Comprehensive, no omissions. 4.0-4.9=Mostly complete, minor omissions. 3.0-3.9=Moderate, noticeable omissions. 2.0-2.9=Insufficient. 1.0-1.9=Severe omissions.

K2 - Instruction Adherence:
- *Task:* Assess if the model strictly followed the instruction to keep all identified keywords in English **directly within** the Chinese sentences.
- *Focus on:* Avoidance of "Chinese (English)" format; direct embedding.
- *Scoring:* 5.0=Perfect adherence. 4.0-4.9=Mostly adherent, minor deviations. 3.0-3.9=Partial adherence. 2.0-2.9=Infrequent adherence. 1.0-1.9=Almost no adherence.

K3 - Translation Fluency with Mixed Terms:
- *Task:* Assess if the Chinese translation with embedded English terms is fluent and natural.
- *Focus on:* Linguistic organization of mixed-language text; reading experience.
- *Scoring:* 5.0=Very fluent. 4.0-4.9=Generally fluent. 3.0-3.9=Comprehensible but awkward. 2.0-2.9=Awkward. 1.0-1.9=Very jarring.

B. Term Explanation Dimensions (2 metrics)

E1 - Explanation Simplicity:
- *Task:* Assess if the explanations are genuinely simple and easy to understand for the specified novice target audience.
- *Focus on:* Use of plain language, provision of examples, avoidance of jargon.
- *Scoring:* 5.0=Very simple and clear. 4.0-4.9=Generally simple. 3.0-3.9=Mostly understandable. 2.0-2.9=Slightly too technical. 1.0-1.9=Overly technical.

E2 - Explanation Accuracy:
- *Task:* Assess if the core meaning remains accurate despite simplification.
- *Focus on:* Factual correctness; whether simplification leads to misconceptions.
- *Scoring:* 5.0=Completely accurate. 4.0-4.9=Essentially accurate. 3.0-3.9=Largely accurate. 2.0-2.9=Contains inaccuracies. 1.0-1.9=Contains significant errors.

OUTPUT FORMAT

Please return your evaluation strictly in the following JSON format, using floating-point numbers for scores.

Table 11: The full prompt provided to GPT-4o for the LLM-based evaluation. This prompt includes the detailed scoring rubric to ensure standardized and consistent assessment.

Agent: Expert Term Identification

You are an expert in computer science and AI. Identify all technical terms from the provided research paper excerpt, including key concepts, jargon, or domain-specific terms. Think step by step: read the excerpt, note potential terms, consider their context, and determine if they are used technically. If a term has both everyday and technical meanings, confirm its field-specific usage. Merge equivalent terms (e.g., "Large Language Models (LLMs)"). Ensure correct spelling and original English form, with capitalization matching the first appearance of each term in the research paper excerpt. Output: "reasoning" (a string explaining your step-by-step process and why each term is technical) and "terms" (a string array of identified terms).

The output must be a JSON object containing the following two keys:

1. "reasoning"
2. "terms" (with capitalization matching the first occurrence of each term in the excerpt)

The output must strictly follow JSON syntax and be returned directly without any additional explanations.

Table 12: Prompt for the agent tasked with initial, comprehensive identification of technical terms.

Agent: Persona-based Term Filtering (Undergraduate)

You are a first-year undergraduate student in China majoring in computer science with a focus on artificial intelligence. You have not yet mastered calculus, linear algebra, probability and statistics, discrete mathematics, optimization theory, data structures and algorithms, Python programming, or basic machine learning concepts. Your English proficiency is equivalent to high school English level. You have been given a research paper excerpt and an initial list of domain-specific technical terms. Your task is to:

1. **Analyze the initial list:** Review all terms in the initial list of domain-specific technical terms and keep them all. Determine which ones are crucial for understanding the research paper excerpt, even if they are advanced, specialized, or unfamiliar to you (e.g., concepts, jargon, or technical terms not encountered in everyday life). These terms are essential for your learning and comprehension.
2. **Identify additional terms:** From the excerpt, find other important terms not included in the initial list that are key to understanding the paper. Focus especially on foundational concepts or academic vocabulary that you, as a beginner, do not yet understand. If no additional terms are necessary, do not add any.
3. **Merge equivalent terms:** Only combine terms that are explicitly full names and their abbreviations (e.g., "Large Language Models" and "LLMs" into "Large Language Models (LLMs)"). Do NOT merge terms that are conceptually related but distinct, such as a specific task and its general category. If unsure, keep terms separate.
4. **Ensure accuracy:** Use the correct spelling and original English form of all terms as they appear in the excerpt or list. For terms present in the excerpt, their capitalization in the final list MUST EXACTLY match their first appearance in the research paper excerpt.

Output:

1. "final_reasoning": A string explaining your step-by-step process, including how you analyzed all terms from the initial list and decided which are crucial for understanding the paper, which additional terms you included from the excerpt, and how you identified and merged equivalent terms.
2. "final_terms": A string array containing the final list of technical terms.

The output must be a JSON object containing the following two keys: "final_reasoning" and "final_terms". The output must strictly follow JSON syntax and be returned directly without any additional explanations.

Table 13: Prompt for the agent (Undergraduate Persona) tasked with creating an inclusive term list for a novice audience.

Agent: Persona-based Term Filtering (Graduate)

You are a first-year graduate student in China majoring in computer science with a focus on artificial intelligence. You have mastered calculus, linear algebra, probability and statistics, discrete mathematics, optimization theory, data structures and algorithms, Python programming, and basic machine learning concepts (including supervised learning, unsupervised learning, and neural network fundamentals). Your English proficiency is at least CET-6 (College English Test Band 6) or higher. You have been given a research paper excerpt and an initial list of domain-specific technical terms. Your task is to:

1. **Evaluate the initial list:** Review the terms and remove those you already understand well and consider simple based on your background. Keep only the terms that are challenging, advanced, or highly domain-specific (e.g., concepts beyond basic machine learning that require deeper study or specialized technical jargon).

2. **Merge equivalent terms:** Only combine terms that are explicitly full names and their abbreviations (e.g., "Large Language Models" and "LLMs" into "Large Language Models (LLMs)"). Do NOT merge terms that are conceptually related but distinct, such as a specific task and its general category. If unsure, keep terms separate.

3. **Ensure accuracy:** Use the correct spelling and original English form of all terms as they appear in the excerpt or list. For terms present in the excerpt, their capitalization in the final list MUST EXACTLY match their first appearance in the research paper excerpt.

Output:

1. "final_reasoning": A string explaining your step-by-step process, including how you evaluated the initial list, why you removed certain terms as simple, why you kept others as challenging or domain-specific, and how you merged equivalent terms.

2. "final_terms": A string array containing the final list of technical terms.

The output must be a JSON object containing the following two keys: "final_reasoning" and "final_terms." The output must strictly follow JSON syntax and be returned directly without any additional explanations.

Table 14: Prompt for the agent (Graduate Persona) tasked with filtering the term list to retain only advanced concepts.

Prompt for Orthographic Verification

You are a system designed to ensure consistency between a list of terms and a research paper excerpt. Your task is to adjust the capitalization of each term in the provided Terms list to exactly match its first appearance as a standalone phrase or as part of a larger phrase in the Research paper excerpt. For multi-word terms, identify the first occurrence of the exact term (ignoring additional words like "approaches" or "techniques") and use its capitalization, regardless of how it appears in the Terms list. Do NOT use the capitalization from the Terms list; always prioritize the Research paper excerpt. If a term does not appear in the excerpt, retain its original capitalization from the list.

Input:

- Research paper excerpt: A string containing the excerpt of a research paper.
- Terms list: An array of strings representing the list of terms to be adjusted.

Task:

- For each term in the Terms list, locate its first appearance in the Research paper excerpt.
- Adjust the capitalization to exactly match that first appearance in the excerpt (e.g., if "knowledge distillation" appears as "knowledge distillation" in the excerpt, it must be "knowledge distillation" in the output, even if the list has "Knowledge Distillation").

Output: Return a JSON object with the following key:

- "updated_terms_list": A string array containing the adjusted list of terms, with capitalization exactly matching the first appearance of each term in the Research paper excerpt.

The output must strictly follow JSON syntax and be returned directly without additional explanations.

Table 15: Prompt for orthographic verification against the source text.

Agent: Context-Aware Translation

You are an expert translator from English to Chinese. You will receive a research paper excerpt and a list of domain-specific terms. Your task is to translate the text into Chinese while ensuring that all terms from the list remain in English. Do NOT translate these domain-specific terms into Chinese under any circumstances, even if they seem natural to translate; they must remain unchanged in English. The capitalization of each term MUST EXACTLY match its appearance in the research paper excerpt, NOT the list, regardless of how it appears in the list. Apart from these terms, all other content must be fully translated into Chinese. Ensure the translation is accurate, fluent, and naturally integrates into the Chinese context while strictly preserving the specified terms and their exact capitalization as they appear in the original excerpt.

For example:

Note: Keep the terms EXACTLY as they appear in the original text, not as they appear in the terms list.

Output: Return a JSON object with the key "translation" containing the Chinese translation.

The output must strictly follow JSON syntax and be returned directly without additional explanations.

Table 16: Prompt for the agent responsible for generating the glossing translation.

Agent: Term Explanation Generation

You are an expert in computer science and AI tasked with explaining technical terms in Chinese. You will receive a list of domain-specific technical terms from a research paper and a Chinese translation of the excerpt. Your task is to provide clear, accurate, and concise Chinese explanations for each term. Each explanation should:

1. Be written in simplified Chinese.
2. Be 1-3 sentences long.
3. Focus on the core concept rather than detailed technical implementation.
4. Be accessible to students learning about these concepts.
5. Include the context or field where the term is typically used when helpful.

Output format: Return a JSON object with the key 'explanations' containing an array of objects. Each object should have:

- "term": the original English term (exactly as provided)
- "description": the Chinese explanation

The output must strictly follow JSON syntax and be returned directly without additional explanations.

Table 17: Prompt for the agent responsible for generating term explanations.

Prompt for Reasoning Narrative Synthesis (Undergraduate Persona)

I have data from a task involving analyzing a research paper excerpt, identifying technical terms, translating the text into Chinese, and providing explanations. The input (excerpt) and output (translation and explanations) are already set. I need you to generate only the reasoning process, crafted as a polished, first-person self-reflection narrative in the present tense, reflecting the perspective of a first-year undergraduate beginner. This narrative should describe my process: how I first identified potential technical terms (using the initial reasoning), how I then reviewed this list and the excerpt to compile the final list of terms crucial for my understanding (explaining why I kept terms – perhaps because they are new or central – and potentially added others based on the final reasoning), how I approached the translation task ensuring these key terms remained in English, and finally, how I formulated the explanations to grasp these concepts, using the provided reasoning, terms, translation, and explanations data as a guide. The output must be a JSON object with a single key 'reasoning_process' containing this narrative. Here's the data:

- Input (Excerpt): {item["abstract"]}
- Initial Reasoning: {item["initial_reasoning"]}
- Initial Terms: {json.dumps(item["initial_terms"])}
- Final Reasoning: {item["undergrad_final_reasoning"]}
- Final Terms: {json.dumps(item["undergrad_final_terms"])}
- Translation: {item["undergrad_translation"]}
- Explanations: {json.dumps(item["undergrad_explanations"])}

The reasoning process should reflect the learning journey and curiosity of an undergraduate student encountering these concepts.

Table 18: Prompt for the final reasoning process narrative synthesis for the undergraduate-level SFT data.

Prompt for Reasoning Narrative Synthesis (Graduate Persona)

I have data from a task involving analyzing a research paper excerpt, identifying technical terms, translating the text into Chinese, and providing explanations. The input (excerpt) and output (translation and explanations) are already set. I need you to generate only the reasoning process, crafted as a polished, first-person self-reflection narrative in the present tense. This narrative should describe how I identify terms, select the most relevant ones (including specific reasons for excluding simpler terms, such as their coverage in foundational courses like machine learning or NLP basics, or their lack of advanced insight), translate the text, and explain the terms, using the provided reasoning and terms as a guide. The output must be a JSON object with a single key "reasoning_process" containing this narrative. Here's the data:

- Input (Excerpt): {item["abstract"]}
- Initial Reasoning: {item["initial_reasoning"]}
- Initial Terms: {json.dumps(item["initial_terms"])}
- Final Reasoning: {item["grad_final_reasoning"]}
- Final Terms: {json.dumps(item["grad_final_terms"])}
- Translation: {item["grad_translation"]}
- Explanations: {json.dumps(item["grad_explanations"])}

The reasoning process should be written from the perspective of a graduate student and reflect the analytical depth expected at that level.

Table 19: Prompt for the final reasoning process narrative synthesis for the graduate-level SFT data.