

LLP: LLM-Based Product Pricing in E-commerce

Hairu Wang^{1,*} Sheng You^{1,*} Qingheng Zhang¹ Xike Xie²
Shuguang Han^{1,†} Yuchen Wu¹ Fei Huang¹ Jufeng Chen¹

¹Xianyu of Alibaba, Hangzhou, China

²University of Science and Technology of China, Suzhou, China

{wanghairu.whr, yousheng.ys, qingheng.zqh, shuguang.sh}@taobao.com
{wuyuchen.wyc, huangfei.hf, jufeng.cjf}@taobao.com xkxie@ustc.edu.cn

Abstract

Setting appropriate prices for second-hand products remains challenging for inexperienced individual sellers on Consumer-to-Consumer (C2C) e-commerce platforms. Existing regression-based approaches rely on static product-price mappings, making it difficult to capture subtle condition differences expressed in colloquial product descriptions. Inspired by recent breakthroughs in Large Language Models (LLMs), we present LLP, a *retrieval-then-generation* framework that generates price suggestions based on dynamically retrieved comparable products. LLP employs bidirectional reasoning to generate high-quality rationales as supervision, and subsequently applies retrieval-aware policy optimization to further improve both reasoning faithfulness and pricing accuracy. At inference time, LLP employs entropy-guided filtering to discard low-confidence price predictions. Extensive evaluations demonstrate the effectiveness of our method. It raises the static adoption rate (SAR) from 40% to 72%, while increasing GMV by 0.7% and CTCVR by 1.4%. LLP is now fully deployed in Xianyu¹, serving millions of individual sellers daily.

1 Introduction

Driven by the sharing economy and sustainable consumption, C2C e-commerce platforms (e.g., eBay, Xianyu and Mercari) have flourished globally in recent years (Guiot and Roux, 2010; Moriuchi and Takahashi, 2022; Chen et al., 2024; Wu et al., 2025). Whereas business-to-consumer (B2C) platforms (e.g., Amazon) maintain deep inventories of standardized new products, C2C marketplaces aggregate highly heterogeneous second-hand items from individual sellers. Consequently, due to the pronounced disparities in product condition and

*Equal contribution.

†Corresponding author.

¹Xianyu is China’s largest C2C e-commerce platform.

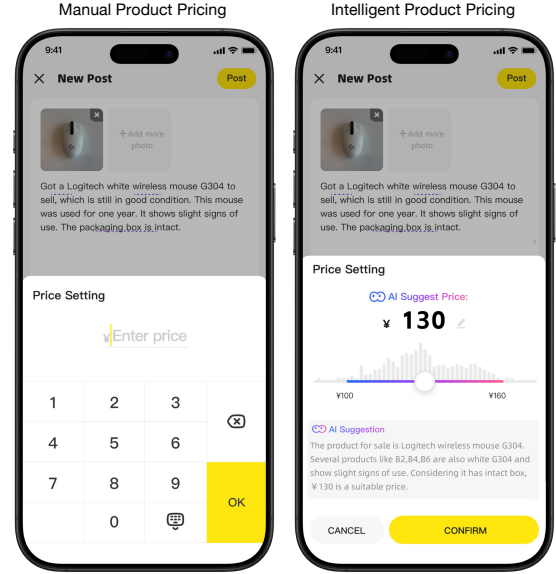


Figure 1: Intelligent Product Pricing on C2C Platforms.

limited market knowledge, these sellers often struggle to set appropriate prices for their products (Raykhel and Ventura, 2009). This not only affects product listing efficiency, but also influences online transactions.

To this end, it is imperative to assist individual sellers in setting reasonable prices for their second-hand items. If we could automatically generate price suggestions by leveraging similar products from C2C platforms, it would significantly streamline the listing process. As shown in Figure 1, after uploading product photos and composing content descriptions, individual sellers can adopt the price with a single click or adjust it as needed, effectively eliminating the burden of manual price entry.

Leading C2C e-commerce platforms are increasingly exploring automated pricing methods. Early works (Raykhel and Ventura, 2009; Sun et al., 2017; Varshitha et al., 2022) employ traditional machine learning algorithms, such as K-Nearest Neighbors (KNN) and decision trees, to perform category-specific price prediction. Subsequent studies (You

et al., 2017; Law et al., 2019; Han et al., 2019, 2020; Vedula et al., 2025) shift towards deep learning models to learn the mapping between product representations and prices, progressively extending their scope to more categories. In our previous industrial deployment, we utilized a Category Property Value (CPV)-based pricing method (Sabehe et al., 2024; Su et al., 2025; Shinzato et al., 2023), which maps target products to pre-computed clusters based on CPV attributes, followed by a Gaussian Mixture Model (GMM) for price estimation within the matched cluster. Although effective in certain tasks, existing methods still fall short in C2C transactions, due to two primary constraints:

Difficulty in Capturing Fine-Grained Semantics: Unlike standardized B2C products, second-hand items exhibit high heterogeneity (e.g., varying degrees of wear and tear), which lies in colloquial product descriptions. Conventional regression models, lacking rich text understanding, struggle to capture these subtle yet critical price determinants (e.g., 85% battery health), thereby compromising pricing accuracy.

Limited Adaptability and Generalization: Prices in the second-hand market are highly sensitive to product iterations and evolving market sentiment (e.g., the rapid depreciation of the iPhone 16 following the release of the iPhone 17). Furthermore, product attributes diverge significantly across categories (e.g., battery health for phones vs. remaining volume for perfumes). However, existing regression models, relying on static parameters and unified feature spaces, fail to capture such temporal or category-specific dynamics, resulting in poor adaptability and generalization.

These challenges raise a critical question: *Can we design a system that captures fine-grained semantics and adapts to market dynamics for accurate product pricing?*

C2C platforms host a vast and diverse range of second-hand products. The Law of One Price (Froot and Rogoff, 1995; Isard, 1977; Goldberg and Knetter, 1997) in economics posits that homogeneous goods trade at a uniform price in efficient markets, implying that items of similar condition should command comparable prices. Moreover, autoregressive LLMs have demonstrated remarkable success across diverse domains (Gu, 2023; Laban et al., 2023; Yi et al., 2025). Therefore, we can adopt a "retrieval-then-generation" paradigm (Wang et al., 2025; Asai et al., 2024; Yan et al., 2024): retrieving similar products as dynamic mar-

ket references and leveraging the nuanced semantic understanding of unstructured text to generate accurate price suggestions. Inspired by the efficacy of Group Relative Policy Optimization (GRPO) (DeepSeek-AI, 2025; Hou et al., 2025; Jiang et al., 2025; Li et al., 2025; Huang et al., 2025) on verifiable tasks, we can further align the LLMs' generation with retrieved candidates, guiding them to produce more reliable, evidence-grounded price suggestions.

Specifically, our contributions are fourfold:

- We introduce LLP for second-hand product pricing using a retrieval-then-generation paradigm, which integrates similar product retrieval with LLM-based price estimation.
- We propose *bidirectional reasoning* to automatically construct high-quality rationale supervision. Building on this, *retrieval-aware policy optimization* encourages the model to leverage retrieved products for more accurate price estimation.
- During inference, we develop an *entropy-guided filtering* mechanism that effectively prunes low-confidence price suggestions, enabling a flexible precision–coverage trade-off in the practical application.
- Extensive offline and online evaluations demonstrate the effectiveness of our method. LLP has been fully deployed in production, achieving a substantial increase in the static adoption rate (SAR) from 40% to 72%.

2 Preliminaries

The abundance of listings on C2C marketplaces provides valuable references for estimating the price of a target product. Formally, let $\mathcal{M}$ denote an LLM and b_q be the target product awaiting pricing. The knowledge base is defined as $\mathcal{B} = \{(b_i, p_i)\}_{i=1}^N$, containing N product-price pairs, where b_i represents the details of a product (including images, textual description, and condition) and p_i is its corresponding price. Our objective is to find relevant market references $\mathcal{B}_q$ from $\mathcal{B}$ by a retrieval algorithm $\mathcal{G}$: $\mathcal{B}_q = \mathcal{G}(b_q, \mathcal{B})$. Then, a pricing algorithm $\mathcal{F}$ derives a rationale $\mathcal{R}$ and the final price estimate $\hat{p}_q$. Given the powerful language comprehension capabilities of LLMs, we instantiate $\mathcal{F}$ with $\mathcal{M}$ here: $\mathcal{R}, \hat{p}_q = \mathcal{F}(b_q, \mathcal{B}_q)$.

To further enhance the reasoning capabilities of off-the-shelf LLMs for pricing tasks, we conduct domain-adaptive post-training. This process, denoted by $\mathcal{T}$, aims to produce an optimized model

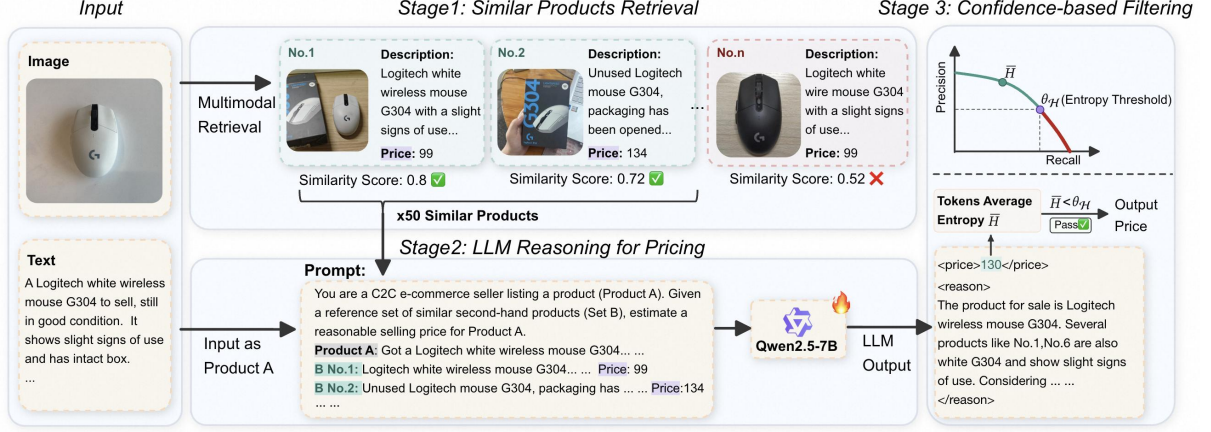


Figure 2: LLP: LLM-based Pricing Framework for C2C E-commerce Platforms.

$\mathcal{M}'$ whose error $\mathcal{L}$ is lower than that of $\mathcal{M}$.

$$\mathcal{M}' = \mathcal{T}(\mathcal{M}, \mathcal{B}), s.t., \mathcal{L}_{\mathcal{M}'} < \mathcal{L}_{\mathcal{M}} \quad (1)$$

Moreover, we introduce an entropy-based filtering mechanism that retains a price prediction only when its average token entropy $\bar{\mathcal{H}}$ does not exceed a predefined threshold $\theta_{\mathcal{H}}$:

$$\hat{p}_q = \hat{p}_q \cdot \mathbb{I}(\bar{\mathcal{H}} \leq \theta_{\mathcal{H}}) \quad (2)$$

3 Methodology

3.1 Overview

In this section, we introduce the framework of our intelligent pricing system LLP, as illustrated in Figure 2. LLP comprises two primary components: similar product retrieval (Section 3.2) and LLM-driven price estimation (Section 3.3).

3.2 Similar Product Retrieval

Candidate Product Pool Construction. To provide timely and reliable market references, we construct a product pool from historical listings. First, we select products listed within the t -day window preceding the target product’s listing date. The window is set to 90 days to preserve market recency while retaining sufficient candidates for subsequent retrieval. Beyond pool size, candidate quality is also important, as unreliable listings can degrade both retrieval relevance and downstream pricing accuracy. We therefore exclude anomalous listings, including off-platform trades, clickbait, and counterfeits. To further ensure that the candidate products reflect genuine market interest, we retain only high-engagement items whose click-through rates exceed the c -th percentile (e.g., $c = 70$), yielding the final product pool $\mathcal{B}$.

Offline Product Representation Extraction. To efficiently retrieve similar products from $\mathcal{B}$ as dynamic market references, we employ Semantic ID (Yang et al., 2025b; Zheng et al., 2025; Pang et al., 2025) for representation extraction. This approach leverages a coarse-to-fine hierarchical structure that facilitates knowledge transfer and ensures robust generalization. Concretely, a T5-based encoder-decoder autoregressively generates a sequence of discrete codes via step-specific codebooks, and the decoder hidden state at the final step is taken as the product representation (see Appendix A.10 for details). For all products in the candidate product pool $\mathcal{B} = \{(b_i, p_i)\}_{i=1}^N$, we compute their representations offline and build an index repository $\mathbb{E} = \{e_i\}_{i=1}^N = \{\mathcal{E}(b_i)\}_{i=1}^N$.

Online Product Retrieval. For a newly listed product b_q , we map it into the same representation space to obtain the query embedding $e_q = \mathcal{E}(b_q)$. The semantic relevance between b_q and a candidate b_i is quantified by cosine similarity: $\text{sim}(b_q, b_i) = \frac{\mathcal{E}(b_q)^\top \mathcal{E}(b_i)}{\|\mathcal{E}(b_q)\| \|\mathcal{E}(b_i)\|}$. To enable efficient vector search, we employ Proxima (Alibaba DAMO Academy, 2021), an approximate nearest-neighbor search (ANNS) engine. Finally, for the target product b_q , the system returns the top- k relevant second-hand products $\mathcal{B}_q = \{(b_j, p_j)\}_{j=1}^k$ as market references for the subsequent reasoning.

3.3 LLM-Driven Price Inference

Prompt Engineering. Since retrieved products may differ from the target in key attributes, we leverage the LLM’s strong semantic understanding capabilities to discern fine-grained differences and generate a well-informed price suggestion. Our prompt incorporates three essential parts: (1) def-

inition and criteria of similar products, (2) target product description, and (3) retrieved product descriptions and prices. The prompt templates are provided in Appendix A.11.

Bidirectional Reasoning-based Fine-Tuning.

To construct training data for supervised fine-tuning, we require both a price label and a corresponding pricing rationale for each target product. Specifically, we treat the seller-provided listing price p_q as ground truth. To obtain the rationale $\mathcal{R}$, a straightforward approach is to prompt a teacher model to generate it based on the retrieved products. However, the generated $\mathcal{R}$ may refer to inappropriate products or fail to support the ground-truth price, resulting in inconsistencies among the retrieved products, reasoning process and price label. To construct a more coherent and evidence-grounded rationale as supervision, we introduce *Bidirectional Reasoning*, a two-stage pipeline consisting of backward comparable-product selection and forward rationale synthesis.

Backward comparable-product selection. Specifically, in the backward stage, the teacher LLM uses the target product–price pair (b_q, p_q) to select products from the retrieved set $\mathcal{B}_q$ that match b_q on key attributes (e.g., brand, model, and condition) and have prices close to p_q . These products constitute a refined subset $\mathcal{B}_q^* \subseteq \mathcal{B}_q$. We filter samples for which the backward reasoning returns an empty subset. Such cases typically arise when the target item’s description is incomplete, too few similar products are retrieved, or the retrieved prices vary substantially, making it difficult to infer the target item’s price from the retrieved evidence.

$$\mathcal{B}_q^* = \mathcal{M}_{\text{backward}}(\mathcal{B}_q, b_q, p_q) \quad (3)$$

Forward rationale synthesis. In the subsequent forward stage, the teacher synthesizes an evidence-grounded rationale $\mathcal{R}$ based on the target product b_q and the retrieved set $\mathcal{B}_q$, drawing primarily on the comparable products in $\mathcal{B}_q^*$. The ground-truth price p_q serves only as a reference to ensure that $\mathcal{R}$ remains consistent with that price. To prevent label leakage and post hoc rationalization, we instruct the teacher not to use p_q as a given premise or reason backward from it. Instead, every claim in $\mathcal{R}$ must be supported by evidence from the target product or the retrieved set. The resulting training data helps the student model learn to analyze product information and derive accurate price estimates.

$$\mathcal{R} = \mathcal{M}_{\text{forward}}(\mathcal{B}_q, \mathcal{B}_q^*, b_q, p_q) \quad (4)$$

To enhance second-hand e-commerce domain knowledge and retrieval-grounded reasoning capability, we fine-tune (Chen et al., 2022; Wang et al., 2020; Zhang et al., 2024; Clemedtson and Shi, 2025) a small student model on the constructed dataset. The generated rationale $\mathcal{R}$ and the target product’s listing price p_q are used as supervision for standard next-token prediction:

$$\mathcal{L}_{\text{SFT}}(\theta) = - \sum_{j=1}^T \log P(y_j | X, y_{<j}; \theta) \quad (5)$$

where $X = (b_q, \mathcal{B}_q)$ denotes the input, $y = (\mathcal{R}, p_q)$ represents the output sequence, and T is its length.

Retrieval-aware Policy Optimization. Building on the supervised fine-tuned model, we apply GRPO to further enhance the model’s reasoning capability and pricing accuracy. In the design of the reward function, we evaluate both the generation process and the final outcome. The *outcome-based reward* assigns higher scores to more accurate price estimates, decreasing as the relative prediction error increases. However, optimizing for price accuracy alone is insufficient, as the LLM may overlook similar products or hallucinate unsupported details, resulting in unfaithful rationales. To mitigate this issue, we introduce a *process-based reward* that measures how well the products referenced in the rationale $\hat{\mathcal{B}}_q^*$ cover the comparable subset $\mathcal{B}_q^*$ selected during backward reasoning.

$$\text{reward} = \underbrace{\frac{1}{1 + \alpha \cdot \left(\frac{\hat{p}_q - p_q}{p_q} \right)^2}}_{\text{price accuracy}} \times \underbrace{\frac{|\hat{\mathcal{B}}_q^* \cap \mathcal{B}_q^*|}{|\mathcal{B}_q^*|}}_{\text{similar product recall}} \quad (6)$$

Using the above reward, we optimize the policy model π_θ via GRPO for each target product b_q and its retrieved set $\mathcal{B}_q$. The full objective and notation are provided in Appendix A.7.

Entropy-guided Price Filtering. Due to sellers’ subjective pricing preferences and substantial variation in the condition of retrieved products, the LLM often exhibits considerable uncertainty when generating price suggestions. To quantify this uncertainty, following recent studies (Geng et al., 2024; Fu et al., 2025; Kang et al., 2025), we use next-token entropy as a proxy for model confidence,

with higher entropy indicating lower confidence. Formally, at generation step i , the entropy H_i of the model’s next-token distribution P_i is defined as:

$$H_i = - \sum_{j \in \mathcal{V}} P_i(j) \log P_i(j), \quad (7)$$

where $\mathcal{V}$ is the model’s vocabulary, and $P_i(j)$ represents the probability of the j -th token in $\mathcal{V}$ at generation step i . Since the predicted price $\hat{p}_q$ may span multiple tokens, we estimate its confidence by averaging the entropy across the corresponding generation steps. Then, an entropy threshold $\theta_{\mathcal{H}}$ is applied to filter out unreliable price suggestions, retaining only those $\bar{H} \leq \theta_{\mathcal{H}}$. In practice, $\theta_{\mathcal{H}}$ is configured according to application requirements, allowing a flexible trade-off between precision and recall.

4 Experiments

4.1 Settings

Datasets. Our experimental data are collected from real-world e-commerce scenarios. For the test set, we sample 80,000 second-hand products from the top 55 categories, ranked by Gross Merchandise Volume (GMV). After manually filtering out anomalous items, the final test set comprises 72,561 products. Sourced from the same 55 categories, all products in the training set strictly pre-date the test samples. The training data comprises approximately 620,000 samples, with 580,000 allocated to SFT and the remainder for GRPO. To evaluate generalization, we construct an out-of-distribution (OOD) test set from 611 distinct categories unseen during training (Table 1). Standardized products (Li, 2024) feature uniform brands and model identifiers (e.g., smartphones), whereas non-standardized products are characterized by high variability and a lack of universal specifications. Further details are provided in Appendix A.8.

Table 1: Statistics of Dataset. “Cats.” means Categories.

Split	Subset	TOP 55 Cats.	Other 611 Cats.
Train	Standardized	177,547	-
	Non-standardized	438,885	-
	Total	616,432	-
Test	Standardized	27,947	35,670
	Non-standardized	44,614	174,597
	Total	72,561	210,267

Metrics. We evaluate accuracy of the price estimation using four metrics: Root Mean Square Log

Error (RMSLE) (Han et al., 2019, 2021), Mean Absolute Log Error (MALE) (Han et al., 2019, 2021), Static Adoption Rate (SAR), and Dynamic Adoption Rate (DAR). SAR measures the fraction of predictions whose relative error is within $\pm 20\%$ of the ground-truth price, whereas DAR employs a dynamic threshold, becoming progressively stricter as the product price increases. This property makes it more tolerant of errors for low-value products while imposing tighter accuracy requirements on high-value ones.

$$\text{RMSLE} = \sqrt{\frac{1}{M} \sum_{i=1}^M (\log(\hat{p}_i) - \log(p_i))^2} \quad (8)$$

$$\text{MALE} = \frac{1}{M} \sum_{i=1}^M |\log(\hat{p}_i) - \log(p_i)| \quad (9)$$

$$\text{SAR} = \frac{1}{M} \sum_{i=1}^M \mathbb{I} \left(\frac{|\hat{p}_i - p_i|}{p_i} \leq \tau \right), \tau = 0.2 \quad (10)$$

$$\text{DAR} = \frac{1}{M} \sum_{i=1}^M \mathbb{I} \left(\frac{|\hat{p}_i - p_i|}{p_i} \leq \tau \right), \tau = \frac{a}{\ln(p_i + b)} \quad (11)$$

Here, M denotes the total number of test samples, while $\hat{p}_i$ and p_i represent the predicted and ground-truth prices for the i -th product, respectively. For the DAR metric, we set the hyperparameters $a = 1.4$ and $b = 10$.

Baselines. We compare LLP with the following baselines: (a) *Text-based K-Nearest Neighbors (KNN)*: retrieving products using textual representations and estimating the target price as the arithmetic mean of their prices (Raykhel and Ventura, 2009); (b) *Semantic ID (SID)-based KNN*: following the same retrieval-and-averaging procedure using Semantic ID representations; (c) *Vision-based DNN*: predicting prices using only visual representations (Han et al., 2019); (d) *Multimodal-based DNN*: leveraging both visual and textual representations (Han et al., 2020); (e) *Previous CPV-based Pricing Method*: first identifying the CPV of the target product using TACLR (Su et al., 2025), and then estimating its price via a Gaussian Mixture Model over the cluster of products sharing the same CPVs (Figure 4). For a fair comparison, we retrieve the top-50 products for all retrieval-based methods.

Models. Our experiments employ a diverse selection of LLMs, encompassing both closed-source models like GPT and Claude, as well as leading open-source families, such as the Qwen2.5 (Yang

Table 2: Overall Performance. The best performance is highlighted in **bold**. "Stan." denotes Standardized Products, while "Non-stan." refers to Non-standardized Products.

Method	RMSLE ↓			MALE ↓			SAR ↑			DAR ↑		
	Stan.	Non-stan.	Overall	Stan.	Non-stan.	Overall	Stan.	Non-stan.	Overall	Stan.	Non-stan.	Overall
Text-based KNN	1.125	1.655	1.471	0.661	1.118	0.939	0.367	0.190	0.259	0.370	0.253	0.299
SID-based KNN	0.918	1.609	1.380	0.597	1.095	0.900	0.317	0.169	0.227	0.313	0.221	0.257
Vision-based DNN	1.244	1.415	1.343	0.918	1.035	0.984	0.162	0.143	0.151	0.177	0.209	0.195
Multimodal-based DNN	0.968	1.194	1.101	0.693	0.856	0.785	0.218	0.167	0.190	0.232	0.245	0.239
CPV-based Pricing	0.982	1.177	1.090	0.575	0.733	0.659	0.433	0.306	0.366	0.443	0.392	0.416
Vanilla LLM	1.244	1.589	1.463	0.932	1.227	1.111	0.197	0.158	0.174	0.198	0.192	0.195
LLM (w/ SFT)	1.749	1.288	1.486	1.341	0.916	1.082	0.173	0.242	0.215	0.184	0.298	0.253
LLM (w/ Retrieval)	0.487	0.841	0.723	0.272	0.510	0.417	0.610	0.417	0.492	0.624	0.513	0.557
LLM (w/ Ret.+SFT)	0.442	0.744	0.643	0.244	0.459	0.375	0.649	0.459	0.534	0.664	0.550	0.594
LLP (Ours)	0.459	0.714	0.627	0.239	0.429	0.355	0.652	0.478	0.546	0.669	0.572	0.610

et al., 2025c) and Qwen3 (Yang et al., 2025a) series. We evaluate performance across model scales in Appendix A.9.

Parameters. We use Qwen2.5-32B-Instruct as the teacher model to construct the training data and Qwen2.5-7B-Instruct as the student model. Our training pipeline performs full-parameter supervised fine-tuning (SFT) on the student model, followed by GRPO, which samples 8 trajectories per prompt. See Appendix A.8 for more details.

4.2 Results

As demonstrated in Table 2, our proposed LLP significantly outperforms all baselines on the second-hand price prediction task, highlighting its superior performance. Specifically, benefiting from the robust semantic comprehension of LLMs, LLP significantly outperforms the simplistic mean-based approach, KNN, given the same retrieved items. Both the vanilla LLM and its SFT variant exhibit poor performance due to a lack of domain knowledge and overfitting, respectively. In contrast, incorporating retrieved products as supplementary context leads to marked performance gains. For instance, the SAR on standardized products increases more than threefold (from 19.7% to 61.0%), underscoring the importance of incorporating market references. Building on retrieval augmentation, SFT and GRPO further improve pricing performance, yielding a nearly 20% gain in DAR over the strongest CPV-based baseline. These results validate the effectiveness of our LLP.

4.3 Analysis

4.3.1 Generalization Performance

To assess generalizability, we evaluate LLP on another 611 categories. As shown in Table 3, LLP significantly outperforms all baselines. Both the

Table 3: Analysis on 611 OOD Categories.

Method	RMSLE	MALE	SAR	DAR
Multimodal-based DNN	1.439	1.103	0.123	0.181
LLM (w/ SFT)	5.014	5.014	0.060	0.061
LLM (w/ Retrieval)	0.938	0.581	0.346	0.438
LLM (w/ Retrieval+SFT)	0.854	0.539	0.394	0.484
LLP (w/ Retrieval+SFT+GRPO)	0.808	0.487	0.429	0.523

multimodal-based DNN and fine-tuned LLMs fail to generalize to unseen categories, with the latter achieving a SAR of merely 6%. In contrast, integrating product retrieval yields a substantial boost, elevating the SAR to 34.6%. Building on this improvement, SFT raises SAR to 39.4%, and GRPO further improves it to 42.9%. These consistent gains demonstrate that LLP generalizes well to categories unseen during training.

4.3.2 Entropy-guided Filtering Analysis

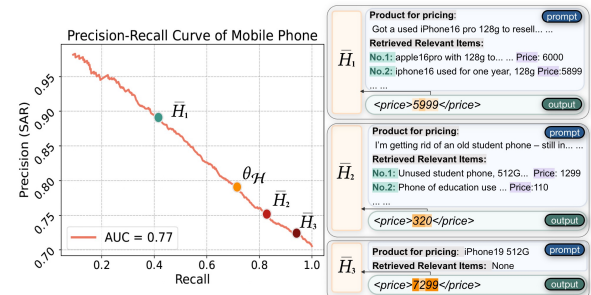


Figure 3: Entropy-based PR Curve of the Mobile Phone Category. Darker orange shading signifies higher token entropy. $\bar{H}_1$: well-described product (low entropy); $\bar{H}_2$: ambiguous input lacking key attributes; $\bar{H}_3$: unreleased product with no retrieved references (high entropy).

To validate the effectiveness of our entropy-based price filtering, we conducted a case study on the mobile phone category. Figure 3 illustrates

the Precision-Recall (PR) curve obtained by varying the entropy threshold. At the threshold θ_H , LLP achieves a high precision of 78% at 70% product coverage. Ambiguous inputs (e.g., $\bar{H}_2$) that lack key attributes (e.g., brand and model), result in higher output entropy, thereby automatically triggering the price filtering. For unreleased products, the failure to retrieve similar products leads to maximal pricing uncertainty (e.g., $\bar{H}_3$). With an AUC of 0.77, the entropy score provides a reliable confidence signal, enabling a flexible precision-recall trade-off to meet different requirements in real-world deployment. Further analyses are provided in Appendix A.9.

4.4 Case Study

Table 4: Case study on *Apple iPhone 16, 256 GB* under different battery conditions.

Product	Battery Condition	CPV (\$)	LLP (\$)
iPhone 16, 256GB	Battery Health 95%	870	840
iPhone 16, 256GB	Battery Health 70%	870	780
iPhone 16, 256GB	Battery dead	870	653

To provide an intuitive illustration of LLP’s advantage over the CPV-based baseline, we present a case study of three second-hand listings of the same brand, model and storage (*Apple iPhone 16, 256 GB*) that differ only in their battery condition, as shown in Table 4. The CPV-based method maps the three products to a single cluster and assigns them a uniform price of 870, ignoring fine-grained condition differences. In contrast, LLP captures variations in battery health and predicts progressively lower prices as it deteriorates. For a device with a dead battery, LLP predicts 653, which is 217 lower than the CPV-based method’s unchanged estimate of 870. This case demonstrates that LLP provides more condition-aware and appropriate price suggestions than the CPV-based baseline.

4.5 Online Deployment

To evaluate the performance of our pricing system, we conduct a 10-day online A/B test, with the CPV-based method and LLP each receiving 20% of platform traffic. Compared with the CPV-based method, LLP raises *direct adoption*—the proportion of sellers who submit the suggested price without modification—from 6% to 17%. Consistent with this improvement, online SAR, which measures the proportion of suggestions falling within

$\pm 20\%$ of the seller’s final submitted price, also increases from 40% to 72%. By helping sellers set more reasonable prices, LLP delivers a 0.7% lift in GMV and a 1.4% gain in Click-Through Conversion Rate (CTCVR) during the A/B test. LLP has been fully deployed in production on 20 NVIDIA L20 GPUs, supporting 25 QPS with an end-to-end latency of approximately 790 ms. The detailed latency breakdown is provided in Appendix A.1.

5 Related Works

C2C e-commerce has flourished in recent years. However, setting appropriate prices for second-hand products remains challenging for inexperienced individual sellers. Numerous studies have explored intelligent pricing methods to improve listing efficiency, a key objective of our work. Early research focuses on traditional machine learning algorithms (Raykhel and Ventura, 2009; Sun et al., 2017; Varshitha et al., 2022). However, these methods are tailored to specific products, limiting their generalization across diverse categories. Subsequent research shifts to deep neural networks, modeling complex item-price mappings. While You et al. (2017) and Han et al. (2019) rely on single-modal features, Han et al. (2020) integrate both visual and textual information to improve the accuracy of price estimation. More recently, Vedula et al. (2025) incorporate quantile regression into LLMs to predict price distributions. In contrast, LLP recasts price prediction as a generative task that captures market dynamics and analyzes fine-grained differences in product condition, effectively addressing key limitations of prior works.

6 Conclusion

In this paper, we present LLP, an LLM-based generative pricing system for C2C e-commerce platforms. Inspired by the Law of One Price, LLP retrieves similar products as dynamic market references and leverages LLMs’ fine-grained semantic understanding of colloquial product descriptions, effectively overcoming the poor generalization of prior regression-based models. During training, it enhances reasoning faithfulness and pricing accuracy through bidirectional reasoning-based supervision and retrieval-aware policy optimization. During inference, it uses entropy as a confidence signal to filter out unreliable price suggestions. Extensive offline and online evaluations demonstrate the effectiveness of our method.

Limitations

LLP establishes a novel generative paradigm for intelligent pricing of second-hand products on C2C e-commerce platforms. Although LLP employs multimodal Semantic IDs for product representation during retrieval, its reasoning stage currently relies solely on textual descriptions. Product images provide rich visual cues for identifying key attributes such as category, brand, model, and condition. Consequently, extending LLP with multimodal large language models (MLLMs) to leverage this visual information is a promising direction. We acknowledge this limitation. Given the substantial inference cost and latency of MLLMs, future work will explore image-token pruning to enable efficient multimodal reasoning and deployment.

References

- Alibaba DAMO Academy. 2021. Proxima: An approximate nearest neighbor search engine. <https://github.com/alibaba/proxima>. GitHub repository.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pages 9112–9141.
- Kang Chen, Qing Heng Zhang, Chengbao Lian, Yixin Ji, Xuwei Liu, Shuguang Han, Guoqiang Wu, Fei Huang, and Jufeng Chen. 2024. IPL: Leveraging multimodal large language models for intelligent product listing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 697–711, Miami, Florida, US. Association for Computational Linguistics.
- Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022, WWW '22*, page 2778–2788, New York, NY, USA. Association for Computing Machinery.
- Alfred Clemedtson and Borun Shi. 2025. GraphRAFT: Retrieval augmented fine-tuning for knowledge graphs on graph databases. *Preprint*, arXiv:2504.05478.
- DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.
- Kenneth A. Froot and Kenneth Rogoff. 1995. Chapter 32 perspectives on ppp and long-run real exchange rates. volume 3 of *Handbook of International Economics*, pages 1647–1688. Elsevier.
- Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *Preprint*, arXiv:2508.15260.
- Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koepl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.
- Pinelopi Koujianou Goldberg and Michael M. Knetter. 1997. Goods prices and exchange rates: What have we learned? *Journal of Economic Literature*, 35(3):1243–1272.
- Qiuhan Gu. 2023. Llm-based code generation method for golang compiler testing. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 2201–2203.
- Denis Guiot and Dominique Roux. 2010. A second-hand shoppers’ motivation scale: Antecedents, consequences, and implications for retailers. *Journal of retailing*, 86(4):355–371.
- Liang Han, Zhaozheng Yin, Zhurong Xia, Li Guo, Mingqian Tang, and Rong Jin. 2019. Vision-based price suggestion for online second-hand items. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1988–1996.
- Liang Han, Zhaozheng Yin, Zhurong Xia, Li Guo, Mingqian Tang, and Rong Jin. 2021. Price suggestion for online second-hand items. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5920–5927.
- Liang Han, Zhaozheng Yin, Zhurong Xia, Mingqian Tang, and Rong Jin. 2020. Price suggestion for online second-hand items with texts and images. *Preprint*, arXiv:2012.06008.
- Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. T1: Advancing language model reasoning through reinforcement learning and inference scaling. *Preprint*, arXiv:2501.11651.
- Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Hao Peng, Julia Hockenmaier, and Tong Zhang. 2025. RAG-RL: Advancing retrieval-augmented generation via rl and curriculum learning. *Preprint*, arXiv:2503.12759.
- Peter Isard. 1977. How far can we push the "law of one price"? *The American Economic Review*, 67(5):942–948.
- Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. s3: You don’t need that much data to train a search agent via rl. *Preprint*, arXiv:2505.14146.

- Bowen Jin, Jinsung Yoon, Jiawei Han, and Serkan O Arik. 2025. [Long-context LLMs meet RAG: Overcoming challenges for long inputs in RAG](#). In *The Thirteenth International Conference on Learning Representations*.
- Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. [Scalable best-of-n selection for large language models via self-certainty](#). *Preprint*, arXiv:2502.18581.
- Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, Alexander Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. [SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9662–9676, Singapore. Association for Computational Linguistics.
- Stephen Law, Brooks Paige, and Chris Russell. 2019. [Take a look around: Using street view and satellite images to estimate house prices](#). *ACM Trans. Intell. Syst. Technol.*, 10(5).
- Elmer Zongyang Li. 2024. E-commerce and regional inequality: A trade framework and evidence from amazon’s expansion. *URL* <https://www.china-ces.org/Files/3058abstract/202501120104402392.pdf>.
- Yuan Li, Qi Luo, Xiaonan Li, Bufan Li, Qinyuan Cheng, Bo Wang, Yining Zheng, Yuxin Wang, Zhangyue Yin, and Xipeng Qiu. 2025. [R3-RAG: Learning step-by-step reasoning and retrieval for llms via reinforcement learning](#). *Preprint*, arXiv:2505.23794.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Emi Moriuchi and Ikuo Takahashi. 2022. The role of perceived value, trust and engagement in the c2c online secondary marketplace. *Journal of Business Research*, 148:76–88.
- Ming Pang, Chunyuan Yuan, Xiaoyu He, Zheng Fang, Donghao Xie, Fanyi Qu, Xue Jiang, Changping Peng, Zhangang Lin, Zheng Luo, and Jingping Shao. 2025. [Generative retrieval and alignment model: A new paradigm for e-commerce retrieval](#). *WWW ’25*, page 413–421, New York, NY, USA. Association for Computing Machinery.
- Alexander Peysakhovich and Adam Lerer. 2023. Attention sorting combats recency bias in long context language models. *arXiv preprint arXiv:2310.01427*.
- Guanghui Qin, Yukun Feng, and Benjamin Van Durme. 2023. [The NLP task effectiveness of long-range transformers](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3774–3790, Dubrovnik, Croatia. Association for Computational Linguistics.
- Ilya Raykhel and Dan Ventura. 2009. Real-time automatic price prediction for ebay online trading. In *IAAI*.
- Kassem Sabeh, Robert Litschko, Mouna Kacimi, Barbara Plank, and Johann Gamper. 2024. An empirical comparison of generative approaches for product attribute-value identification. *arXiv preprint arXiv:2407.01137*.
- Keiji Shinzato, Naoki Yoshinaga, Yandi Xia, and Wei-Te Chen. 2023. [A unified generative approach to product attribute-value identification](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6599–6612, Toronto, Canada. Association for Computational Linguistics.
- Yindu Su, Huike Zou, Lin Sun, Ting Zhang, Haiyang Yang, Liyu Chen, David Lo, Qingheng Zhang, Shuguang Han, and Jufeng Chen. 2025. [TACLR: A scalable and efficient retrieval-based method for industrial product attribute value identification](#). *Preprint*, arXiv:2501.03835.
- Ning Sun, Hongxi Bai, Yuxia Geng, and Huizhu Shi. 2017. [Price evaluation model in second-hand car system based on bp neural network theory](#). In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pages 431–436.
- Janke Varshitha, K Jahnavi, and C. Lakshmi. 2022. [Prediction of used car prices using artificial neural networks and machine learning](#). In *2022 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–4.
- Nikhita Vedula, Dushyanta Dhyani, Laleh Jalali, Boris Oreshkin, Mohsen Bayati, and Shervin Malmasi. 2025. [Quantile regression with large language models for price prediction](#). *Preprint*, arXiv:2506.06657.
- Hairu Wang, Yuan Feng, Xike Xie, and S Kevin Zhou. 2025. [Path pooling: Training-free structure enhancement for efficient knowledge graph retrieval-augmented generation](#). *Preprint*, arXiv:2503.05203.
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. [K-adapter: Infusing knowledge into pre-trained models with adapters](#). *Preprint*, arXiv:2002.01808.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.
- Wenhao Wu, Xiaojie Li, Lin Wang, Jialiang Zhou, Di Wu, Qinye Xie, Qingheng Zhang, Yin Zhang, Shuguang Han, Fei Huang, and Jufeng Chen. 2025. [IU4Rec: Interest unit-based product organization and](#)

- [recommendation for e-commerce platform](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 5039–5048, New York, NY, USA. Association for Computing Machinery.
- Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. [Retrieval meets long context large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. [Corrective retrieval augmented generation](#). *Preprint*, arXiv:2401.15884.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Haiyang Yang, Qinye Xie, Qingheng Zhang, Liyu Chen, Huike Zou, Chengbao Lian, Shuguang Han, Fei Huang, Jufeng Chen, and Bo Zheng. 2025b. [GSID: Generative semantic indexing for e-commerce product understanding](#). *Preprint*, arXiv:2509.23860.
- Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025c. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. 2025. [A survey on recent advances in llm-based multi-turn dialogue systems](#). *Preprint*, arXiv:2402.18013.
- Quanzeng You, Ran Pang, Liangliang Cao, and Jiebo Luo. 2017. [Image-based appraisal of real estate properties](#). *IEEE Transactions on Multimedia*, 19(12):2751–2759.
- Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. [RAFT: Adapting language model to domain specific RAG](#). In *First Conference on Language Modeling*.
- Carolina Zheng, Minhui Huang, Dmitrii Pedchenko, Kaushik Rangadurai, Siyu Wang, Fan Xia, Gaby Nahum, Jie Lei, Yang Yang, Tao Liu, Zutian Luo, Xiaohan Wei, Dinesh Ramasamy, Jiyan Yang, Yiping Han, Lin Yang, Hangjun Xu, Rong Jin, and Shuang Yang. 2025. [Enhancing embedding representation stability in recommendation systems with semantic id](#). *RecSys '25*, page 954–957, New York, NY, USA. Association for Computing Machinery.

A Appendix

A.1 Online Serving

We further analyze the online latency of LLP in comparison with the previous CPV-based pricing method. Table 5 reports a detailed latency breakdown, including the retrieval stage and the inference stage of LLMs. The CPV-based method first predicts the Category Property Values (CPVs) of the target product to map it to a pre-computed cluster, and then estimates the price via a Gaussian Mixture Model (GMM), with an end-to-end latency of about 450 ms. Although LLP introduces additional computation compared with the CPV-based method, its average end-to-end latency remains below 800 ms, satisfying the latency requirements of our online service.

Table 5: Latency comparison (ms) between LLP and the CPV-based pricing method.

Method	Retrieval	Inference	Total
CPV-based Pricing	–	–	450
LLP	40	750	790

A.2 Prompt Sensitivity

To evaluate the robustness of LLP to prompt design, we conduct a sensitivity analysis across four prompt variants. Specifically, we consider: (A) the original prompt used in our paper; (B) a rephrased version that substitutes instructions with synonymous expressions; (C) an ordering variant that swaps the chain-of-thought and the predicted price in the output; and (D) a variant that randomly shuffles the order of the retrieved products. The results are reported in Table 6.

Variant B performs on par with the original, suggesting that LLP is insensitive to paraphrasing of the instructions. Variant C yields a larger drop, as chain-of-thought can improve reasoning by generating intermediate steps prior to the final answer (Wei et al., 2022). Variant D, shuffling the retrieved products, incurs the largest drop. The retrieval ranking itself encodes similarity to the target product, but random shuffling discards this signal. Despite these effects, the overall SAR fluctuation remains within 1.2%, demonstrating that LLP is robust to the prompt formulation.

A.3 Statistical Significance

To assess LLP’s SAR improvement over the CPV-based baseline, we conduct a significance test on

Table 6: Prompt sensitivity analysis of LLP. Δ denotes the absolute change in SAR relative to Variant A.

Variant	Description	SAR	Δ
A	Original (paper)	0.546	–
B	Rephrased instructions	0.545	–0.001
C	Swapped reasoning and price	0.538	–0.008
D	Shuffled retrieved products	0.535	–0.011

72,561 samples. As shown in Table 7, the 95% CIs are narrow (approximately one percentage point wide) and non-overlapping. Together with a p -value below 0.001, these results indicate that LLP’s SAR gain is statistically significant and unlikely to be caused by random sampling variation.

Table 7: Statistical significance analysis of LLP against the CPV-based baseline on 72,561 test samples.

Method	SAR	95% CI	p -value
CPV-based	0.3656	[0.3607, 0.3705]	–
LLP (Ours)	0.5462	[0.5414, 0.5510]	$p < 0.001$

A.4 Error Analysis

To better understand the limitations of LLP in real-world deployment, we collect online failure cases and manually categorize each case into one of four types, as summarized in Table 8: (1) **Pricing Subjectivity** accounts for roughly 40% of the failure cases, stemming from the inherent subjectivity of individual sellers (e.g., urgent sellers tend to price lower, whereas patient sellers tend to price higher than the market reference); (2) **Incomplete Information** contributes about 20%, where user-submitted product descriptions lack key attributes such as brand, model, or condition, making accurate pricing infeasible; (3) **Retrieval Quality** is responsible for another 20%, caused by either failing to retrieve relevant products or retrieving similar products with excessively large price variance; (4) **Reasoning Errors** account for the remaining 20%, where LLP incorrectly analyzes the retrieved products or generates hallucinated conclusions. These observations suggest that future improvements could focus on richer user-side information elicitation, more robust retrieval, and more faithful LLM-based reasoning.

A.5 Human Evaluation

To assess the quality and reliability of rationales synthesized during the forward-reasoning stage, we manually audit 1,000 randomly sampled trajec-

Table 8: Error analysis of LLP on online failure cases.

Error Type	Prop.	Description
Pricing Subjectivity	40%	Inherent seller subjectivity, leading to divergent price expectations for similar items.
Incomplete Information	20%	User-submitted description lacking key attributes, such as brand, model, and condition.
Retrieval Quality	20%	No relevant product retrieved, or retrieved products exhibit large price variance.
Reasoning Errors	20%	LLP incorrectly analyzes the retrieved products or generates hallucinations.

ries. As shown in Table 9, all six evaluation criteria achieve high pass rates. In particular, 98% of the trajectories show no observable post-hoc rationalization, providing evidence for the faithfulness of the synthesized rationale supervision. The results also indicate that the vast majority of forward-reasoning trajectories do not leak privileged information, such as the ground-truth price. Accounting for overlap across issue categories, only 9% of the trajectories are identified as problematic. The low rate supports the use of synthesized rationales as reliable supervision for student post-training. During real-world inference, the student has access to neither the ground-truth price nor any other privileged information. Moreover, GRPO goes beyond direct teacher imitation by sampling multiple trajectories per prompt and reinforcing the best-performing ones, thereby further mitigating the issues described above.

A.6 Discussion on Pricing Bias

We further discuss potential biases in LLP from two perspectives: the user side and the category side. (1) **From the user perspective**, LLP generates price suggestions solely based on product-side attributes (e.g., brand, model, condition, and specifications), without relying on any user-related information. Moreover, all suggested prices are advisory only. Individual sellers may freely adopt, modify, or reject the suggested price. (2) **From the category perspective**, we find that overestimations and underestimations are roughly symmetric across all categories, with an imbalance within 5% per category, indicating that LLP does not exhibit systematic category-level pricing bias. Together, these findings confirm that it delivers user-agnostic and category-balanced price suggestions.

A.7 GRPO Objective

For each target product b_q and its retrieved context $\mathcal{B}_q$, GRPO samples a group of outputs $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{old}}$. Based on the rewards defined in Section 3.3, the policy model π_θ is optimized by maximizing the following objective:

$$J_{GRPO}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^G \left(\min (w_i(\theta) A_i, \text{clip}(w_i(\theta), 1 - \epsilon, 1 + \epsilon) A_i) - \beta \mathbb{D}_{KL} \right) \right] \quad (12)$$

Here, $w_i(\theta) = \frac{\pi_\theta(o_i|x)}{\pi_{\theta_{old}}(o_i|x)}$ is the importance weight, with ϵ and β as hyperparameters. A_i is the advantage computed by group-wise normalization of the rewards $\{r_1, r_2, \dots, r_G\}$:

$$A_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)} \quad (13)$$

$\mathbb{D}_{KL}$ is the KL divergence that penalizes excessive deviations of the current policy from the reference model π_{ref} .

A.8 Experimental Setup

Dataset In e-commerce, products can be broadly classified into two categories based on their degree of standardization: standardized and non-standardized products. Standardized products refer to items with clear specifications, models, and functions that typically adhere to recognized industry standards. Non-standardized products, conversely, are defined by an emphasis on individual attributes and subjective appeal rather than uniform standards. Consequently, consumer decisions are driven more by factors such as style, design, and brand identity, leading to lower price sensitivity. Specific categories for both standardized and non-standardized items are listed in Table 10.

Implementation Details The detailed experimental settings are as follows: **1) Bidirectional Reasoning-based Fine-Tuning.** We first synthesize the training data using Qwen2.5-32B-Instruct served by vLLM, with greedy decoding (temperature = 0) and a maximum context length of 8,192 tokens. We then fine-tune Qwen2.5-7B-Instruct for 4,000 steps with a learning rate of 1×10^{-5} , a per-device batch size of 2, and 4 gradient accumulation steps. The training set is a 1:1 mixture of samples with and without the reasoning process

Table 9: Human evaluation of 1,000 forward-reasoning trajectories randomly sampled from the training data. The first three criteria assess rationale quality, while the remaining three assess rationale reliability.

Metric	Criterion	Result
Evidence Fidelity $\uparrow$	The rationale is accurate and supported by the product information and retrieved evidence.	96%
Comparable Relevance $\uparrow$	The referenced products are appropriate comparisons for the target product on key attributes.	94%
Price Justification $\uparrow$	The suggested price is reasonably justified by the retrieved products.	99%
No Explicit Target-price Leakage $\uparrow$	The rationale does not explicitly reveal or refer to the known price of the target product.	92%
No Backward Reasoning Disclosure $\uparrow$	The rationale does not mention the backward-reasoning process or the selected subset.	96%
No Observable Post-hoc Rationalization $\uparrow$	The rationale does not justify an exact price without sufficient support from the retrieved products.	98%

Table 10: Examples of Standardized and Non-standardized Categories.

Classification	Category
Standardized	Graphics Cards, Motorcycles, Game Consoles, Automobiles, Mobile Phones, Tablet PCs, Billiard Cues, Digital Cameras, Laptops, Smartwatches, Electric Bicycles, Custom-built Desktops, Air Conditioners, Bluetooth Earphones, Fishing Rods, Keyboards
Non-standardized	Dresses, Handbags, T-shirts, Watches, Action Figures (GKs), Coffee/Tea, Low-top Shoes, Necklaces, Running Shoes, Dolls, Bracelets, Lolita Fashion, Skate Shoes, Cotton Dolls, Concert Tickets, Video Memberships, Phone Components, Gimbals, E-bike Batteries, Motorcycle Brake Systems, E-bike Controllers, Hotel Booking Services, Attraction Tickets, Buffet Vouchers, Proxy Shopping Services, Food Coupons, Fast Food Vouchers, Mobile Top-ups

R , distinguished via system prompts. **2) Retrieval-aware Policy Optimization.** Following supervised fine-tuning, we employ Group Relative Policy Optimization (GRPO) to further optimize the model’s output distribution. To facilitate exploration and ensure diversity within the sampled groups, we set the sampling temperature to 0.8 and top-p to 0.95. We train for 1,000 steps, sampling $G = 8$ outputs per prompt, with a maximum generation length of 512 tokens. The optimization is constrained by a KL coefficient of 0.04 and a clip range of $\epsilon = 0.2$. In the outcome reward, the sensitivity coefficient is set to $\alpha = 25$, which assigns a reward of 0.5 at a relative price error of 20%. For evaluation, we use greedy decoding to guarantee reproducibility. All experiments are conducted on 64 NVIDIA H20 GPUs.

A.9 Additional Analysis

Analysis on Various LLMs Table 11 provides a detailed performance comparison of various open-source and closed-source LLMs on the task of price prediction for second-hand products. All models reported in Table 11 are *off-the-shelf* LLMs without any fine-tuning, and use exactly the same retrieved products as in Table 2. The goal of this experiment is to examine how pricing performance varies across LLMs of different scales, thereby informing the choice of a base model that balances accuracy against inference cost in deployment. Since this serves as a supplementary analysis rather than a core comparison, we present it here.

We observe a clear Scaling Law: as the number of model parameters increases, the accuracy of the LLMs’ price prediction also improves. The performance improvement slows down when the model



Figure 4: Previous CPV-based Pricing Framework for C2C E-commerce Platforms.

size grows from 14B to 72B, with diminishing marginal returns. For models of comparable scale, the Qwen2.5 series consistently outperform Qwen3 series, for which the thinking mode is disabled to mitigate inference latency. Furthermore, the experimental results indicate that the open-source Qwen2.5-72B-Instruct outperforms GPT-4.1-mini and Claude 3.5-Haiku across all metrics, and remains competitive with the latest proprietary models such as GPT-5.2 and Gemini-2.5-Pro. Consequently, we ultimately select the Qwen2.5 series for pricing tasks.

We note that scaling up the base model can yield additional accuracy gains, but the associated inference overhead grows substantially. In real-world deployment, our system serves millions of individual sellers per day, imposing stringent constraints on serving cost and latency. Balancing these practical considerations with pricing accuracy, we adopt Qwen2.5-7B-Instruct as the base model of LLP. Notably, after our proposed post-training, the resulting 7B model achieves lower pricing error than the vanilla Qwen2.5-72B-Instruct and even closed-source counterparts such as GPT-5.2 and Gemini-2.5-Pro, while being orders of magnitude cheaper to serve.

Table 11: Performance Comparison of Various Models. The best performance is highlighted in **bold**.

Model	RMSLE ↓	MALE ↓	SAR ↑	DAR ↑
<i>Open-source Models</i>				
Qwen2.5-3B-Instruct	0.881	0.503	0.434	0.489
Qwen2.5-7B-Instruct	0.723	0.417	0.492	0.557
Qwen2.5-14B-Instruct	0.659	0.381	0.511	0.573
Qwen2.5-32B-Instruct	0.666	0.382	0.503	0.565
Qwen2.5-72B-Instruct	0.627	0.364	0.523	0.586
Qwen3-4B	0.805	0.470	0.417	0.471
Qwen3-8B	0.718	0.412	0.483	0.542
Qwen3-14B	0.667	0.386	0.487	0.546
Qwen3-32B	0.639	0.387	0.476	0.533
Qwen3-30B-A3B	0.739	0.405	0.490	0.547
<i>Closed-source Models</i>				
Claude 3.5-Haiku	0.812	0.449	0.469	0.524
GPT-4.1-mini	0.676	0.378	0.521	0.582
GPT-5.2	0.662	0.375	0.534	0.599
Gemini-2.5-Pro	0.649	0.360	0.526	0.598

Analysis on Different Product Representation

In this section, we investigate the impact of different product representations on the LLMs’ price prediction. We conduct this analysis using the vanilla Qwen2.5-7B-Instruct for all experiments. Table 12 shows that retrieval based on product image

Table 12: Performance Comparison of Different Product Representations During the Retrieval Phase.

Method	RMSLE ↓	MALE ↓	SAR ↑	DAR ↑
Text	0.9982	0.5557	0.4166	0.4719
Image	0.8817	0.5085	0.4276	0.4802
Semantic ID (Ours)	0.7234	0.4167	0.4923	0.5569

representations leads to better performance compared to text representations. This suggests that product images carry essential information as well. More importantly, the multimodal Generative Semantic ID representation consistently outperforms both unimodal alternatives across all metrics, reducing RMSLE by 18.0% and 27.5% compared with image- and text-based retrieval, respectively.

Table 13: Performance Comparison with a Varying Number of Retrieved Products (k). The best performance is highlighted in **bold**.

Number (k)	MALE ↓	RMSLE ↓	SAR ↑	DAR ↑
0	1.0912	1.5156	0.1923	0.2368
1	0.4872	0.8792	0.4569	0.5160
3	0.4175	0.7277	0.4945	0.5568
5	0.3972	0.6952	0.5086	0.5725
15	0.3702	0.6502	0.5327	0.5964
30	0.3598	0.6348	0.5419	0.6068
50	0.3549	0.6263	0.5462	0.6110
100	0.3584	0.6328	0.5414	0.6099

Analysis on the Number of Retrieved Products

To study the effect of the number of retrieved products on price prediction, we perform an experiment with the results shown in Table 13. Overall, as the number of retrieved products varies from 0 to 100, the performance of our LLP first increases and then declines. Nevertheless, leveraging retrieved similar products as a basis for reasoning consistently outperforms LLMs without external knowledge, which validates the effectiveness of the retrieval module. Initially, with a small number of retrieved products, the LLMs’ price prediction accuracy is constrained by a lack of relevant information, so it rises as more products are added. The performance drops when retrieving a large quantity of products (e.g., 100). This can be attributed to interference from irrelevant products, and the LLMs’ sensitivity to token positions (Liu et al., 2024; Xu et al., 2024; Jin et al., 2025; Peysakhovich and Lerer, 2023; Qin et al., 2023), a phenomenon known as position bias, which arises from the position encoding. Accordingly, we select 50 products as the market references in our practical deployment, which yields the

best pricing performance while maintaining acceptable inference overhead.

Analysis on Entropy-based Price Filtering Figure 5 further shows the Precision-Recall (PR) curve for handbags and the top 10 categories, obtained by dynamically tuning the entropy threshold. The trends are similar to those observed in the mobile phone category, demonstrating the approach’s generality and robustness across diverse product categories. On the top 10 categories, it achieves over 80% pricing accuracy at 60% product coverage, demonstrating strong reliability in real-world deployment. Even under full coverage, where every product can be priced, LLP still sustains a pricing accuracy near 70%. In the single handbag category, at 20% product coverage, SAR approaches 100%, demonstrating high pricing accuracy. At the expense of partial product coverage, the system delivers a dramatically improved user experience, while enabling a dynamic trade-off between precision and recall.

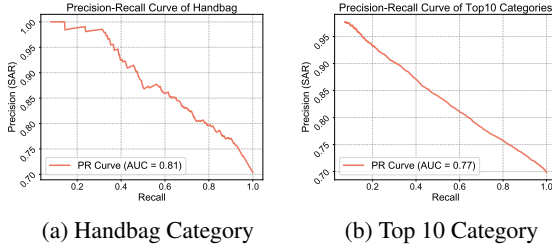


Figure 5: Entropy-based PR Curve of the Handbag and Top 10 Categories.

A.10 Semantic ID

Following the generative tokenization learning paradigm, we adopt a T5-based model that autoregressively transforms products into hierarchical Semantic IDs via coarse-to-fine progressive quantization. Built upon step-specific codebooks, we construct a generative tree structure that integrates multimodal features and collaborative signals, facilitating knowledge transfer and ensuring robust generalization (as illustrated in Figure 6).

Encoder-Decoder Architecture. We maintain M step-specific codebooks $\{E_t\}_{t=1}^M$, each $E_t \in \mathbb{R}^{K \times D}$, where each row of E_t stores the embedding of a discrete code at level t . Given a product description b , the encoder extracts its semantic representation, while the decoder autoregressively generates Semantic IDs conditioned on the encoder

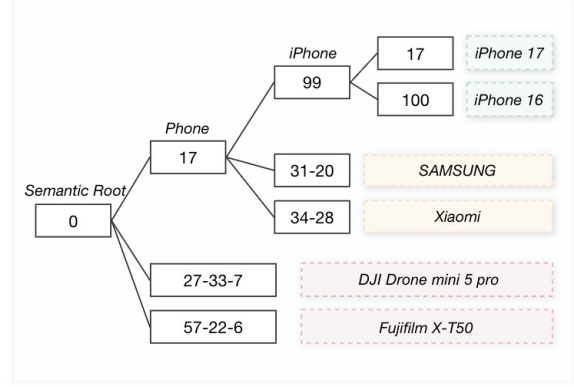


Figure 6: Examples of Semantic IDs.

output and the embeddings of previously produced codes. At step t , the decoder yields a hidden state

$$d_t = \text{Decoder}(\text{Encoder}(b), (E_\tau[z_\tau])_{\tau < t}) \in \mathbb{R}^D. \quad (14)$$

Hierarchical Code Generation. The discrete code z_t is selected via inner-product matching against E_t :

$$Q(z_t = j \mid z_{<t}, b) = \text{softmax}_j(d_t \cdot E_t^\top), j \in [K]. \quad (15)$$

This sequential decoding produces a coarse-to-fine code sequence $(z_1, \dots, z_M)$ as the product’s Semantic IDs. As shown in Figure 7, semantically proximate items (e.g., two mice) converge at the root index (e.g., 32) and follow an identical refinement path “32-45-21”, whereas distinct items (e.g., drones) diverge immediately at the root (e.g., 27) to form a separate sequence “27-33-7”. This contrast between intra- and inter-category cases visualizes how the codebooks encode intrinsic semantic similarity in a data-driven manner.

Final Product Representation. Rather than directly using raw encoder features, we take the decoder hidden state at the last step as the product representation. The codebooks impose a coarse-to-fine semantic prior that routes items with shared prefix codes into a common subspace, so that long-tail products can benefit from the semantic structure learned over popular categories.

Training Objective. The model is optimized in two stages. The *domain-adaptive pre-training* stage injects e-commerce semantics into the T5 backbone via three sequence-to-sequence tasks: *query generation*, which predicts a search query from a product description; *item cloze*, which reconstructs a product description from its masked version; and *suffix completion*, which generates

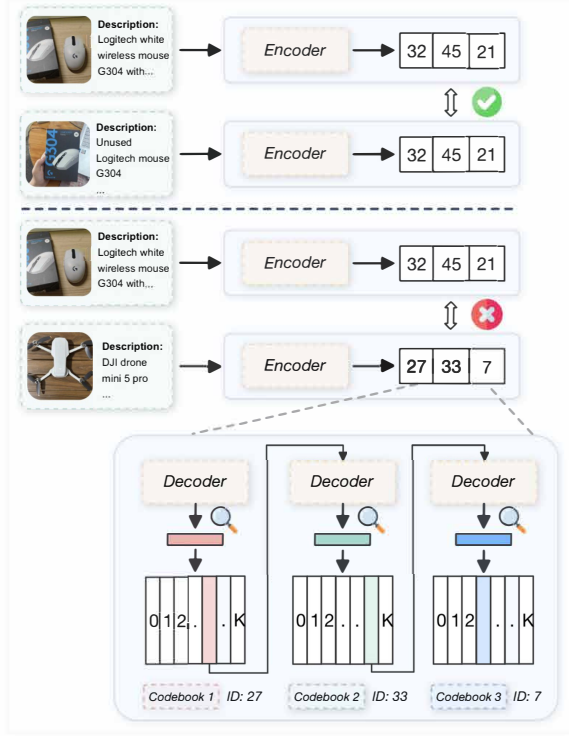


Figure 7: Comparison of Intra- and Inter-Category Semantic IDs.

the suffix of a product description given its prefix. Each task is trained with a standard negative log-likelihood loss, jointly forming

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{qry_gen}} + \mathcal{L}_{\text{item_cloze}} + \mathcal{L}_{\text{suffix_comp}}. \quad (16)$$

The *progressive training* stage learns Semantic IDs layer by layer. At step T , codes $z_{<T}$ are frozen, and the model is trained on (q, b) pairs from search logs, where q is a search query associated with product b . The total loss combines a query-item alignment term and a commitment term, $\mathcal{L} = \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{com}}$:

$$\begin{aligned} \mathcal{L}_{\text{align}} = & -\log \frac{\exp(q_T \cdot d_T)}{\sum_{d^* \sim \mathcal{B}} \exp(q_T \cdot d_T^*)} \\ & + \sum_{t=1}^T \mathbb{D}_{\text{KL}}(P(z_t | z_{<t}, q) \parallel P(z_t | z_{<t}, b)), \end{aligned} \quad (17)$$

$$\mathcal{L}_{\text{com}} = -\sum_{t=1}^T \log Q(z_t | z_{<t}, \text{Encoder}(b)). \quad (18)$$

Codebooks are updated by exponential moving averages for stability. Hard negatives sharing the same code prefix are sampled within each batch to force fine-grained discrimination among semantically similar items.

Three-Stage Multimodal Training. So far we have only considered text inputs. To bring images

into the same semantic space, we add a separate vision encoder in front of the T5 backbone. We then train the model in three stages. Stage 1 trains the T5 backbone on text alone to establish its textual semantic space. Stage 2 trains a ViT-based vision encoder to align image features with the textual semantic space, while keeping the T5 encoder frozen. Stage 3 then jointly fine-tunes the full model on text, image, and multimodal inputs, producing the final multimodal Semantic IDs. This staged design decouples modalities early to avoid unstable gradient interactions, while enabling efficient end-to-end inference fully compatible with the T5 architecture.

A.11 Prompt

In this section, we provide the prompt templates.

Bidirectional Reasoning for Training Data Construction In this section, we present the task templates for backward reasoning and forward reasoning in training data construction, as detailed below.

Product Retrieval for Price Prediction Given a target product and a set of retrieved products with their prices, we prompt the LLM to predict the price of the target product. The full prompt consists of a system prompt and a user prompt, as shown below.

Prompt for Backward Reasoning

You are an expert in second-hand product analysis. Your task is to use the information and ground-truth price of Product A to identify strictly comparable products from the retrieved product set that provide a sound basis for pricing.

[Product A]: {Description of Product A.}

[Price of Product A]: {Price of Product A.}

[Retrieved Products]: {Details of Products B1-B50.}

Prompt for Forward Reasoning

You are an expert in second-hand product pricing. Your task is to use the information about Product A, the complete retrieved product set, and the strictly comparable product subset identified during backward reasoning to construct a coherent and evidence-grounded pricing rationale.

The price of Product A is provided only as a reference to help align the rationale with the target price. Do not treat this price as a known fact or state that it is provided as input. Instead, derive the rationale naturally from the key attributes of Product A and the retrieved products. Do not work backward from the target price or construct post hoc justifications for it. Every claim, comparison, and pricing consideration in the rationale must be supported by the available product information.

[Product A]: {Description of Product A.}

[Price of Product A — For Alignment Only]: {Price of Product A.}

[Retrieved Products]: {Details of Products B1-B50.}

[Comparable Product Subset]: {Product IDs selected by the backward teacher.}

System Prompt for Price Prediction

You are an expert in predicting the prices of second-hand products. Please provide your pricing rationale and the final predicted price. Wrap the rationale in <reason></reason> and the price in <price></price>.

User Prompt for Price Prediction

Please generate a price suggestion for Product A based on the retrieved products (set B). Carefully assess the retrieved products, as not all of them are similar to Product A. Your final price suggestion should be derived from a careful analysis of primary pricing factors, such as brand, model, condition, specifications, version and ...

[Product A]

iPhone 16 Pro, White, 256GB, Nearly new, Fully functional.

[Retrieved Products]

- Product B1: Apple iPhone 16 Pro, 256GB, Black, Excellent Condition (90% Battery). Price: 6088.
- Product B2: Apple iPhone 16 Pro, 256GB, White, With a cracked screen. Price: 4999.
- ...
- Product B50: Apple iPhone 15 Pro, 512GB, White, Purchased 1 year ago but barely used (kept idle). In perfect condition. Free charger included. Price: 3999.