Text-to-Distribution Prediction with Quantile Tokens and Neighbor Context

Anonymous ACL submission

Abstract

Many applications of LLM-based text regression require predicting a full conditional distribution rather than a single point value. We study *distributional regression* under empirical-quantile supervision, where each input is paired with multiple observed quantile outcomes, and the target distribution is represented by a dense grid of quantiles. We address two key limitations of current approaches: the lack of local grounding for distribution estimates, and the reliance on shared representations that create an indirect bottleneck between inputs and quantile outputs. In this paper, we introduce *Quantile Token Regression*, which, to our knowledge, is the first work to insert dedicated quantile tokens into the input sequence, enabling direct input-output pathways for each quantile through self-attention. We further augment these quantile tokens with retrieval, incorporating semantically similar *neighbor* instances and their empirical distributions to ground predictions with local evidence from similar instances. We also provide the first theoretical analysis of loss functions for quantile regression, clarifying which distributional objectives each optimizes. Experiments on the Inside Airbnb and StackSample benchmark datasets with LLMs ranging from 1.7B to 14B parameters show that quantile tokens with neighbors consistently outperform baselines ($\sim$4 points lower MAPE and 2$\times$ narrower prediction), with especially large gains on smaller and more challenging datasets where quantile tokens produce substantially sharper and more accurate distributions.

1 Introduction

Large Language Models (LLMs) have shown remarkable capabilities beyond text generation, extending to structured prediction tasks such as time series forecasting (Gruver et al., 2023) and regression (Vacareanu et al., 2024; Jacobs et al., 2024). Recent work has shown that LLMs can approximate numerical mappings with strong accuracy when fine-tuned or prompted with in-context examples, making them attractive for text regression tasks where crucial information lies in unstructured text (Bitvai and Cohn, 2015; Chen and Si, 2024).

While most LLM-based regression work focuses on point estimation, many real-world use cases require predicting *full probability distributions* rather than single values. Price prediction, demand forecasting, and risk assessment all benefit from understanding not just central tendencies but also dispersion and tail behavior (Arora et al., 2023; Kneib et al., 2023). Quantile regression (Koenker and Bassett, 1978) provides a natural framework for distribution prediction by estimating conditional quantiles at different probability levels, offering robustness to outliers and the ability to capture heterogeneous effects across the distribution.

Recent work by Vedula et al. (2025) takes an important first step towards LLM-based distributional prediction by attaching multiple linear regression heads to a shared final hidden state, each predicting a different quantile. However, this architecture has three key limitations. First, all quantile predictions derive from the same representation bottleneck, creating only an *indirect* connection between input features and quantile-specific outputs. The model must compress everything relevant about the distribution into a single vector, from which separate heads attempt to extract different quantiles. Second, the method predicts distributions based only on the query text, which may contain limited information about the target distribution. This contrasts with how humans reason about distributions, which naturally relies on comparison with similar instances. For example, when estimating the price of a query product, one searches for similar products and builds an understanding by comparing features and observed price ranges. The model can benefit from a richer local context obtained by finding semantically similar items to inform the distribution estimate, but it does not have explicit ac-

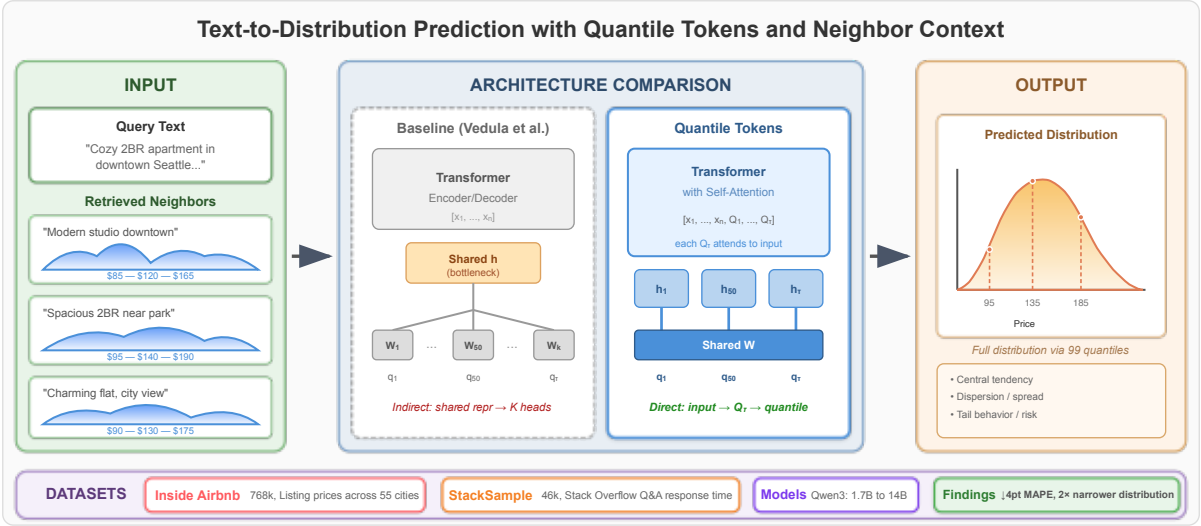

Figure 1: Overview of our approach. *Left:* Input includes query text and retrieved neighbors with their full empirical distributions (as quantiles). *Center:* The baseline (Vedula et al., 2025) uses only query text without neighbors and computes all quantiles from a shared hidden state via separate linear heads. Our Quantile Token approach augments the input with neighbors and inserts dedicated $\langle Q_\tau \rangle$ tokens that attend directly to the input, creating direct input-output pathways for each quantile. *Right:* Output is a complete predicted distribution via τ quantiles. *Bottom:* We evaluate on two diverse datasets (Airbnb, Stack Overflow) with Qwen3 1.7B–14B models.

cess to such relevant reference points. Third, prior retrieval-augmented method (Wang et al., 2025) rely on training with single point labels for each instance, which provides limited distributional supervision compared to training with full empirical distributions.

We address these limitations with a quantile token architecture augmented by neighbor context (Figure 1). Our approach makes two key contributions:

Quantile Token Regression. We propose a novel architecture that inserts learnable quantile tokens ($\langle Q_{\tau_1} \rangle, \ldots, \langle Q_{\tau_K} \rangle$) directly into the input sequence, which allows each quantile token to attend to different parts of the input and accumulate quantile-specific information. This creates a *direct* input-output pathway for each quantile level, rather than relying on separate linear heads over a shared representation. The architecture enables more coherent quantile predictions since all quantile tokens are produced jointly within the same attention computation, and offers interpretability by revealing which input features each quantile attends to.

Retrieval-Augmented Distribution Estimation. We augment quantile regression with retrieved *neighbor* instances, which are semantically similar examples from a candidate pool. Crucially, while prior retrieval-augmented approaches attach only a *single point label* to each neighbor (Wang et al., 2025) and only predict a single price output, we equip each neighbor with its *full empirical distribution* represented as quantiles. This provides the model with richer supervision. By grounding predictions in distributional evidence from neighbors, the model can better estimate not only central tendency but also dispersion and tail behavior of the target distribution.

To evaluate our approach, we construct two text-to-distribution datasets from Inside Airbnb (Inside Airbnb, 2025) and StackSample (Stack Overflow, 2019, 2025). Experiments with Qwen3 models (Yang et al., 2025) spanning 1.7B–14B parameters show that (1) retrieval-augmented inputs consistently improve quantile regression across model scales (8% relative reduction in avg MAPE on Airbnb, 63% on StackSample), (2) quantile tokens outperform the shared-representation baseline (14% relative reduction in avg MAPE on StackSample, 6× narrower intervals), and (3) the combination of both techniques yields the best performance. We also provide a mathematical analysis of different loss functions for quantile regression, clarifying the distributional objectives each optimizes.

2 Related Work

LLM-Based Regression. LLMs have been applied to regression through three main paradigms. First, in-context learning performs regression with-

out fine-tuning by providing numeric examples in the prompt (Garg et al., 2022; Vacareanu et al., 2024). Second, LLM embeddings can be used as features for conventional regressors (Imperial, 2021; Tang et al., 2025). Third, fine-tuning directly optimizes LLMs for numeric prediction, either by treating numbers as text tokens or by adding regression heads (Yang et al., 2020; Jacobs et al., 2024; Song et al., 2024). Recent work also explores regression-specific objectives, such as decision-theoretic fine-tuning (RAFT) (Lukasik et al., 2025) and coupling chain-of-thought with regression losses (TRACT) (Chiang et al., 2025). Our work follows the fine-tuning paradigm but targets *distributional* rather than point prediction.

Quantile Regression and Distributional Prediction. Quantile regression (Koenker and Bassett, 1978) estimates conditional quantiles via the pinball loss, enabling nonparametric characterization of predictive uncertainty. Distributional prediction is widely used in applications such as forecasting and risk-sensitive decision making (Arora et al., 2023; Gürlek et al., 2024; Gu et al., 2024), and recent work has begun integrating these ideas with LLMs (Gruver et al., 2023; Gillman et al., 2025). LLM-based quantile regression often attaches multiple quantile heads to a shared representation (Vedula et al., 2025; Dorka, 2024), which can create a bottleneck between the input and quantile-specific outputs. In contrast, our quantile token architecture inserts dedicated tokens that participate in attention throughout the transformer, yielding more direct input–output pathways for each quantile.

Retrieval-Augmented Prediction. Retrieval augmentation has been effective for grounding LLM outputs in external evidence (Lewis et al., 2020; Asai et al., 2023). For regression, retrieved neighbors can provide context that improves calibration; for example, retrieval-augmented pricing leverages similar items to support numeric estimates (Wang et al., 2025). Existing work largely targets point prediction. We extend retrieval augmentation to *distributional* prediction, leveraging the intuition that similar instances often exhibit similar outcome distributions, which is informative for estimating dispersion and tail behavior.

Text Regression Applications. Text regression maps unstructured language to numeric targets in domains including finance, real estate, product pricing, and content scoring (Gu et al., 2024; Chen and Si, 2024; Vedula et al., 2025; Wang et al., 2025; Chiang et al., 2025). A common challenge is that key signals are embedded in free-form text and are difficult to capture with manual features. We formulate text-to-distribution prediction as a general problem and evaluate across multiple domains (e.g., Airbnb listings and community Q&A) to demonstrate breadth beyond pricing-centric settings.

3 Method

3.1 Quantile Distribution Regression Task

Let X denote an input instance, where each input $X^{(i)}$ is a text sequence $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)})$, and let $Y \in \mathbb{R}$ be a continuous outcome. The objective is to learn a model that maps an input X to a conditional distribution $F_{Y|X}(\cdot \mid X)$, represented via its conditional quantiles. Specifically, for a set of quantile levels $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_Q) \in (0,1)^Q$, we predict the quantile vector $q_{\boldsymbol{\tau}}(X) = \big(q_{\tau_1}(X), \ldots, q_{\tau_Q}(X)\big)$, where $q_\tau(X)$ is the τ-th quantile of $F_{Y|X}(\cdot \mid X)$. This formulation follows the standard view of quantile regression as distribution learning through conditional quantiles.

Empirical Quantile Supervision. Each input X_i is paired with multiple observed outcomes $\mathcal{Y}_i = \{y_{i1}, \ldots, y_{iM_i}\}$, which we treat as realizations of $Y \mid X_i$. We construct the empirical CDF $\widehat{F}_i(t) = \frac{1}{M_i}\sum_{m=1}^{M_i} \mathbf{1}[y_{im} \le t]$ with empirical quantile function $\widehat{Q}_i(\tau) = \widehat{F}_i^{-1}(\tau)$. Since each instance has a variable number of outcomes M_i, we interpolate $\widehat{Q}_i$ to a fixed grid of $Q = 99$ quantile levels $\boldsymbol{\tau} = \{0.01, 0.02, \ldots, 0.99\}$ via linear interpolation (Appendix B), producing target vectors $\widehat{\mathbf{q}}_i \in \mathbb{R}^{99}$.

Learning Objective. Given a dataset $\mathcal{D} = \{(X_i, \mathcal{Y}_i)\}_{i=1}^N$, we train a model f_θ to predict $\widehat{\mathbf{q}}_i$ from X_i, i.e., $f_\theta(X_i) \approx \widehat{\mathbf{q}}_i$. This turns text-to-distribution prediction into structured regression over quantile levels, where the model learns not only a central tendency but also dispersion and tail behavior.

3.2 Quantile Token Regression

We introduce *Quantile Token Regression*, a simple architectural change that makes each quantile prediction depend on a dedicated representation. Given an input sequence $X =$

Figure 2: Quantile tokens enable specialized representations and direct input-output pathways for each quantile level. For a Stack Overflow question about asyncio exception handling, different features signal different response times: the popular "python" tag suggests fast answers, while the niche "asyncio" topic and code complexity suggest slower responses. *Center:* Each quantile token (Q_{10}, Q_{50}, Q_{90}) learns to attend to the features most predictive of its target quantile—Q_{10} focuses on popularity signals while Q_{90} focuses on complexity indicators. *Right:* The resulting time-to-answer distribution captures both the possibility of a quick response (10% within 15 minutes) and the long tail (90% within 18 hours).

$(x_1, \ldots, x_n)$[1], we append Q special quantile tokens $\langle Q_{\tau_1} \rangle, \ldots, \langle Q_{\tau_Q} \rangle$ to the end of the sequence. The resulting sequence is

$$\widetilde{X} = (x_1, \ldots, x_n, \langle Q_{\tau_1} \rangle, \ldots, \langle Q_{\tau_Q} \rangle). \quad (1)$$

We feed $\widetilde{X}$ into a pretrained transformer g_θ and obtain final-layer hidden states $H = g_\theta(\widetilde{X})$. Let $h_{\tau_k} \in \mathbb{R}^d$ denote the hidden state at the position of token $\langle Q_{\tau_k} \rangle$. We then predict the k-th quantile using a shared linear regressor applied to the corresponding quantile-token representation.

$$\hat{q}_{\tau_k}(X) = w^\top h_{\tau_k} + b. \quad (2)$$

Quantile token regression architecture has two advantages. First, it creates a direct input-output relation for each quantile level by allowing $\langle Q_{\tau_k} \rangle$ to collect information across all transformer layers, rather than relying on separate linear heads over a shared final representation. This improves the alignment between the conditioning evidence and quantile-specific predictions, which pays more attention to extreme quantiles. Figure 2 illustrates this mechanism on a Stack Overflow question, where the range of response time to a question is predicted. The $\langle Q_{10} \rangle$ token learns to attend to popularity signals (e.g., "python" tag) that predict fast answers, while $\langle Q_{90} \rangle$ attends to complexity indicators that predict slower responses. Second, quantile tokens result in more coherent quantile representations, since all quantile tokens are produced jointly within the same attention computation, and they enable interpretability by inspecting which parts of the input each $\langle Q_{\tau_k} \rangle$ attends to when forming its estimate.

[1]Including the query text and any retrieved neighbor context.

3.3 Retrieval-based Quantile Regression

Semantically similar inputs tend to exhibit similar outcome distributions. For example, similar product descriptions yield similar price distributions, and similar questions receive similar response-time distributions. When conditioning only on the query text, the model must implicitly learn these distributional patterns from the training data, which can be challenging. We therefore augment quantile regression with *neighbors*, semantically similar, label-bearing instances retrieved from a candidate pool, to explicitly provide the model with distributional evidence from similar instances.

Given an input text sequence X, we retrieve the top-K semantically similar neighbors from a candidate pool using dense embedding similarity. Each retrieved neighbor is provided to the model along with its empirical distribution represented as a small set of quantiles. Implementation details, including the choice of embedding model, retrieval features, input formatting, and which quantiles are selected for neighbors, are described in Section 4.1.

3.4 Loss Functions and Theoretical Analysis

A critical design choice in distribution learning is the objective used to align the predicted quantile vector $\hat{q}_{\boldsymbol{\tau}}(X) = \big(\hat{q}_{\tau_1}(X), \ldots, \hat{q}_{\tau_Q}(X)\big)$ with the

supervision derived from the outcome set $\mathcal{Y}_i = \{y_{i1}, \ldots, y_{iM_i}\}$. Standard quantile regression applies the pinball loss to *raw* outcomes y. In contrast, our training targets $\widehat{Q}_i(\tau)$ are *empirical quantile estimators* computed from a finite sample $\mathcal{Y}_i$ and then interpolated to a fixed grid. We therefore analyze losses that are appropriate for *quantile supervision*, and clarify what each objective targets in population.

ℓ_1 and ℓ_2 losses on empirical quantiles (Wasserstein matching). We treat the interpolated empirical quantiles $\widehat{Q}_i(\tau_k)$ as noisy measurements of the latent population quantiles $Q^*(X_i, \tau_k)$ and minimize an element-wise ℓ_p loss:

$$\mathcal{L}_{\ell_1}(\theta) = \frac{1}{NQ} \sum_{i=1}^{N} \sum_{k=1}^{Q} \left| \widehat{Q}_i(\tau_k) - \hat{q}_{\tau_k}(X_i) \right|, \quad (3)$$

$$\mathcal{L}_{\ell_2}(\theta) = \frac{1}{NQ} \sum_{i=1}^{N} \sum_{k=1}^{Q} \left(\widehat{Q}_i(\tau_k) - \hat{q}_{\tau_k}(X_i) \right)^2. \quad (4)$$

In one dimension, the p-Wasserstein distance admits the quantile representation $W_p^p(F, G) = \int_0^1 |Q_F(u) - Q_G(u)|^p \, du$ (Villani, 2009). When $\boldsymbol{\tau}$ is a dense, approximately uniform grid, $\mathcal{L}_{\ell_1}$ and $\mathcal{L}_{\ell_2}$ can be interpreted as discrete approximations to $W_1(\widehat{F}_i, \widehat{F}_\theta(\cdot \mid X_i))$ and $W_2^2(\widehat{F}_i, \widehat{F}_\theta(\cdot \mid X_i))$, respectively, where $\widehat{F}_\theta$ is the distribution implied by the predicted quantiles. Here we assume the predicted quantiles are monotonic in τ (or are monotonized by sorting) so they define $\widehat{F_\theta}$.

Mismatched pinball on empirical quantiles (Pinball-Q). A seemingly natural alternative is to apply the standard pinball loss to the empirical quantile targets:

$$\mathcal{L}_{\text{Pinball-Q}}(\theta) = \frac{1}{NQ} \sum_{i=1}^{N} \sum_{k=1}^{Q} \rho_{\tau_k}\left(\widehat{Q}_i(\tau_k) - \hat{q}_{\tau_k}(X_i) \right), \quad (5)$$

where $\rho_\tau(u) = u\big(\tau - \mathbb{I}[u < 0]\big)$. While pinball is proper for learning $Q^*(X, \tau)$ from raw outcomes, here the "outcome" fed to pinball is itself a random quantile estimator. As a result, (5) generally targets the τ-th quantile of the *estimator distribution* $\widehat{Q}_i(\tau) \mid X_i$, not the underlying parameter $Q^*(X_i, \tau)$ (Appendix A). This yields a systematic "inflation/deflation" effect away from $\tau = 0.5$.

Scalarized pinball on a single statistic (Pinball-Med). Prior LLM regression work often associates each input with a single scalar label and trains all quantile heads against that scalar using pinball. To mirror this setting under distribution supervision, we define a scalar pseudo-target $y_i := \widehat{Q}_i(0.5)$ (the sample median)[2] and optimize

$$\mathcal{L}_{\text{Pinball-Med}}(\theta) = \frac{1}{NQ} \sum_{i=1}^{N} \sum_{k=1}^{Q} \rho_{\tau_k}(y_i - \hat{q}_{\tau_k}(X_i)). \quad (6)$$

Appendix A shows that (6) is generally *inconsistent* for learning the full conditional distribution when $M_i > 1$: it learns the conditional distribution of the statistic y_i (which concentrates around $Q^*(X_i, 0.5)$ as M_i grows), causing the predicted quantile function to collapse.

Theoretical implications and empirical ordering. Under the latent-sample model in Appendix A, for each fixed (X, τ) the empirical quantile obeys an asymptotic expansion $\widehat{Q}_i(\tau) = Q^*(X_i, \tau) + \varepsilon_{i,\tau}$ where $\varepsilon_{i,\tau}$ is approximately centered and symmetric with variance scaling as $\mathrm{Var}(\varepsilon_{i,\tau} \mid X_i) \propto \tau(1-\tau)/M_i$ (up to density factors). Consequently, $\mathcal{L}_{\ell_1}$ (median) and $\mathcal{L}_{\ell_2}$ (mean) are Fisher-consistent for Q^* in the large-M_i regime (with ℓ_1 offering additional robustness under heavy tails or outliers). In contrast, $\mathcal{L}_{\text{Pinball-Q}}$ introduces a bias of order $M_i^{-1/2}$ whose magnitude increases toward extreme quantiles, while $\mathcal{L}_{\text{Pinball-Med}}$ discards distributional shape and collapses toward the median. This analysis predicts the empirical ordering observed in our experiments: $\mathcal{L}_{\ell_1}$ performs best overall, $\mathcal{L}_{\text{Pinball-Q}}$ is competitive but biased, and $\mathcal{L}_{\text{Pinball-Med}}$ underperforms due to systematic loss of tail information.

The same theory suggests variance-aware weighting by (M_i, τ_k) (and density factors) to downweight noisy tail quantiles when M_i is small; we leave this extension to future work (Appendix A).

4 Experiments

4.1 Datasets

We experiment upon two publicly available diverse text-to-distribution datasets: Inside Airbnb (Inside Airbnb, 2025) and StackSample (Stack Overflow, 2019, 2025). For both datasets, we construct ground-truth distributions from multiple observed outcomes per instance, keeping only instances with at least 4 observations (an empirical threshold chosen to balance label quality with dataset size). We use stratified sampling for train/val/test splits. For retrieval, we compute dense embeddings using `Qwen/Qwen3-Embedding-8B`[3] over the full text of

[2] If an application only provides a subset of quantiles, one can analogously set y_i to the median of the available reported quantiles.

[3] https://huggingface.co/Qwen/Qwen3-Embedding-8B

Dataset	Split	# Samples	Avg. # Tokens
Inside Airbnb			
	train	768,001	385
	val	50,000	354
	test	50,000	355
	test_la (OOD)	61,551	385
StackSample			
	train	46,544	320
	val	5,818	322
	test	5,819	324

Table 1: Dataset split statistics. We report number of samples and average number of tokens per sample.

each instance, and attach nine representative quantiles [4] of each neighbor's empirical distribution spanning extremes to quartiles to the model input. Full statistics are in Table 1. See Appendix C for full construction details.

Airbnb. The task is to predict the price distribution for an Airbnb listing given its textual description and metadata. We construct the dataset from Inside Airbnb (Inside Airbnb, 2025), collecting all available cities from 2024-09 to 2025-08 (119 cities) and converting prices to U.S. dollars. Each listing is represented by its title, description, amenities, location, and property type. We construct the ground-truth price distribution from observed monthly prices across time snapshots, yielding ∼840k samples from 55 cities after filtering. We hold out Los Angeles to form an out-of-domain (OOD) test set.

StackSample. The task is to predict the distribution of response times (time from question posting to receiving an answer) for a Stack Overflow question given its text. We use StackSample (Stack Overflow, 2019, 2025), a Kaggle-hosted subset of Stack Overflow Q&A. Each question is represented by its title, body, and tags. We construct the ground-truth response-time distribution from observed answer response times. Following prior work (Vedula et al., 2025), we apply log transformation to handle the wide range of response times (from minutes to hours).

4.2 Experimental Setup

We fine-tune Qwen3 models (1.7B–14B parameters) (Yang et al., 2025) with LoRA (Hu et al., 2022), predicting $Q{=}99$ uniformly spaced quantiles. We compare $\text{QR}_{K=0}$ (baseline quantile regression (Vedula et al., 2025)), $\text{QR}_{K=8}$ (QR with $K{=}8$ retrieved neighbors), and $\text{QT}_{K=8}$ (quantile tokens with $K{=}8$ neighbors). We evaluate using average MAPE (Mean Absolute Percentage Error), wMAPE, sMAPE for point accuracy, and CRPSS (Continuous Ranked Probability Skill Score) and RCIW (Relative Coverage Interval Width) for distributional quality, mostly following Vedula et al. (2025). Full details on dataset construction, experimental setup, hyperparameters, and metric definitions are in Appendix D, G, and E.

4.3 Experimental Results

Our Quantile Tokens Regression approach (QT) consistently outperforms the quantile regression (QR) baseline across both datasets and all model sizes (Tables 2 and 3). The method ranking is stable: $\text{QT}_{K=8}$ outperforms $\text{QR}_{K=8}$, which in turn outperforms $\text{QR}_{K=0}$ on all reported metrics. Compared to retrieval-augmented QR, QT improves both accuracy and sharpness, consistently reducing average MAPE and producing markedly narrower prediction intervals, with RCIW reduced by multiple factors in both datasets. The advantage of QT is especially pronounced on StackSample, the smaller and more challenging dataset with response times spanning from 1.01 mins to 12 hrs. The $\text{QR}_{K=0}$ baseline produces extremely poor distributions with very wide confidence intervals (RCIW@99 of 4.55×10^4), while $\text{QT}_{K=8}$ converges more reliably. Comparing $\text{QT}_{K=8}$ to the baseline $\text{QR}_{K=0}$, $\text{QT}_{K=8}$ reduces average MAPE from 266.65 to 84.30[5] (68% reduction) and shrinks RCIW@99 from 4.55×10^4 to 346.90 (131× reduction). Even compared to retrieval-augmented $\text{QR}_{K=8}$, $\text{QT}_{K=8}$ achieves 14% lower average MAPE (84.30 vs 98.56) and 6× narrower intervals.

Neighbor retrieval provides substantial gains across all configurations, with particularly large impact when training data are limited. On StackSample, retrieval yields dramatic improvements: for the QR baseline, average MAPE drops from 266.65 to 98.56 (63% reduction) when moving from $K{=}0$ to $K{=}8$. On Airbnb, retrieval also consistently improves average MAPE for QR across all model sizes: for Qwen3-4B, average MAPE drops from 30.31 to 27.78 (8% reduction), and similar gains hold at other scales. These substantial improvements empirically validate the hypothesis (Section 3.3) that semantically similar inputs exhibit

[4] 1, 5, 10, 25, 50, 75, 90, 95, and 99 percentiles.

[5] The absolute MAPE on StackSample remains high due to the limited dataset size and the inherently high uncertainty in response-time prediction compared to price prediction.

Model	Method	K	avg MAPE↓	wMAPE↓	sMAPE↓	CRPSS↑	RCIW@90↓	RCIW@95↓	RCIW@99↓
Qwen3-1.7B	Quantile Regression	0	32.60	53.60	32.09	0.4408	12.72	16.29	19.06
	Quantile Regression	8	29.27	52.12	29.43	0.4588	16.73	22.80	29.84
	Quantile Token	8	27.18	50.18	27.39	0.4677	3.91	5.45	7.58
Qwen3-4B	Quantile Regression	0	30.31	52.06	30.28	0.4536	9.58	12.30	14.66
	Quantile Regression	8	27.78	50.75	27.95	0.4700	11.77	15.08	17.94
	Quantile Token	8	26.89	49.99	27.14	0.4700	5.12	7.17	9.64
Qwen3-8B	Quantile Regression	0	29.02	51.18	29.05	0.4616	7.61	9.69	11.34
	Quantile Regression	8	26.64	49.88	27.07	**0.4800**	7.38	9.48	11.27
	Quantile Token	8	26.56	**49.63**	26.75	0.4700	**3.89**	**5.44**	**6.96**
Qwen3-14B	Quantile Regression	0	30.23	51.85	29.91	0.4575	10.89	14.25	16.87
	Quantile Regression	8	27.96	50.63	27.83	0.4754	17.46	22.98	27.51
	Quantile Token	8	**26.40**	49.67	**26.67**	0.4741	4.61	7.39	11.10

Table 2: Results on the Airbnb test set with various model sizes. QR denotes quantile regression, QT denotes quantile token regression, and K is the number of anchors. Best value per metric column is in bold.

Model	Method	K	avg MAPE↓	wMAPE↓	sMAPE↓	CRPSS↑	RCIW@90↓	RCIW@95↓	RCIW@99↓
Qwen3-4B	Quantile Regression	0	266.65	75.00	67.41	0.0668	3900.57	6949.75	45480.31
Qwen3-4B	Quantile Regression	8	98.56	74.97	67.19	0.3001	545.50	1025.38	2110.01
Qwen3-4B	Quantile Token	8	**84.30**	**73.02**	**64.86**	**0.3375**	**274.20**	**315.40**	**346.90**

Table 3: Results on the StackSample dataset.

similar outcome distributions, as retrieved neighbors' distributions provide informative context for predictions. This contrast aligns with dataset scale: Airbnb is much larger, with ∼840k listings across 55 cities, while StackSample contains ∼58k questions. Therefore, retrieval has greater impact when training data are more limited, offering explicit distributional evidence from similar instances that helps ground predictions.

Model scaling shows diminishing returns at larger sizes on Airbnb. Moving from 1.7B to 4B parameters reduces average MAPE by 7% for $\text{QT}_{K=8}$ (from 27.18 to 26.89), while moving from 8B to 14B yields only 1% improvement (from 26.56 to 26.40). This aligns with empirical scaling laws (Kaplan et al., 2020) but shows that gains saturate in our settings. Notably, distributional metrics do not monotonically improve with size: for example, $\text{QR}_{K=8}$ shows wider intervals at 14B than at 8B (RCIW@99 increases from 11.27 to 27.51). Since we tune hyperparameters per model, this suggests that larger backbones can be more sensitive to optimization and regularization choices, and better point accuracy does not necessarily translate into sharper confidence intervals.

To summarize, these results suggest that retrieval augmentation and quantile tokens are especially critical for harder, higher-uncertainty text-to-distribution tasks rather than providing only incremental gains. We further evaluate generalization by holding out Los Angeles in the Airbnb dataset during training and OOD testing on its listings, with full results in Appendix F.

4.4 Ablation Studies

4.4.1 Loss Functions

Table 4 validates the theoretical comparison in Section 3.4 and Appendix A by contrasting Wasserstein losses with pinball variants under empirical-quantile supervision.

Loss	avg MAPE↓	CRPSS↑	RCIW@95↓
Pinball-Med	32.80	**0.5331**	151.78
Pinball-Q	32.66	**0.5332**	151.27
ℓ_1 Wasserstein	**26.55**	0.4682	**3.55**
ℓ_2 Wasserstein	26.64	0.4737	4.15

Table 4: Ablation on loss functions on Airbnb dev set using Qwen3-4B with K=8 neighbors.

The two Wasserstein objectives, which are Fisher-consistent for the target quantiles as the number of labels per instance increases, achieve the best practical accuracy and sharpness. In particular, ℓ_1 Wasserstein yields the lowest average MAPE and the tightest confidence intervals, while ℓ_2 Wasserstein is competitive but slightly worse on both average MAPE and RCIW@95. In contrast, the pinball-based objectives perform poorly for distribution learning in our setting: PINBALL-Q applies pinball loss to the empirical quantile

targets, and PINBALL-MED uses only the empirical median as supervision. Both incur much larger average MAPE and extremely wide intervals, consistent with the predicted bias of PINBALL-Q and the loss of distributional information under PINBALL-MED. Although the pinball losses attain higher CRPSS than Wasserstein, this primarily reflects a sharpness–calibration tradeoff: they produce overly conservative forecasts with inflated dispersion, which can improve skill relative to the reference while yielding uncertainty estimates that are too unsharp to be useful in practice, as indicated by RCIW@95 above 150 versus 3–4 for Wasserstein. Overall, these results support using Wasserstein objectives for empirical-quantile supervision, with ℓ_1 Wasserstein providing the best accuracy–sharpness tradeoff in our experiments.

4.4.2 Number of Neighbor

Table 5 studies number of retrieved items K on the Airbnb dev set using Qwen3-4B with QT and ℓ_1 Wasserstein loss, applying postprocess monotonicity (described in Section 4.4.3).

# Neighbors	avg MAPE↓	CRPSS↑	RCIW@95↓
0	29.54	0.4509	4.90
2	27.02	0.4625	4.45
4	26.64	0.4676	4.34
8	26.55	0.4682	3.55
16	**25.85**	**0.4735**	**3.47**

Table 5: Ablation on the number of retrieved neighbors on Airbnb dev set using Qwen3-4B.

Increasing K consistently improves average MAPE and CRPSS while tightening intervals. The improvement is most pronounced when neighbors are first introduced, especially from $K{=}0$ to $K{=}2$, and exhibits diminishing returns as K increases. Using $K{=}16$ performs best overall, reducing average MAPE from 29.54 to 25.85 and improving CRPSS from 0.4509 to 0.4735, while lowering RCIW@95 from 4.90 to 3.47. However, larger K increases the number of input tokens, which raises memory usage and training cost (e.g., $K{=}16$ requires approximately $2\times$ memory per sample and $1.4\times$ total training time compared to $K{=}8$)..

4.4.3 Monotonicity

Our proposed Quantile Token regression approach provides no guarantee that the predicted quantiles will satisfy the monotonicity constraint, which can cause issues like the 90th percentile prediction being lower than the 80th percentile. Table 6 therefore compares three approaches to ensure monotonicity on the Airbnb dev set using Qwen3-4B with QT, ℓ_1 Wasserstein loss, and $K{=}8$ neighbors.

Method	avg MAPE↓	CRPSS↑	RCIW@95↓
BASELINE	26.55	0.4682	**3.55**
CUMSUM	26.51	**0.4749**	7.16
POSTPROCESS	**26.35**	0.4701	**3.55**

Table 6: Ablation on monotonicity method on Airbnb dev set using Qwen3-4B with $K{=}8$ neighbors.

BASELINE applies no monotonicity constraint and uses the raw predicted quantiles as-is. CUMSUM enforces monotonicity during both training and inference by predicting non-negative gaps between adjacent quantiles and cumulatively summing them to form ordered quantiles. POSTPROCESS keeps training unchanged and enforces monotonicity only at inference time by sorting the predicted quantiles. Empirically, CUMSUM slightly improves average MAPE and yields the best CRPSS, but substantially widens intervals. In contrast, POSTPROCESS achieves the lowest average MAPE, while maintaining comparable CRPSS and intervals.

5 Conclusion

We studied text-to-distribution prediction under empirical-quantile supervision, where each input has multiple observed outcomes and the target distribution is represented by a dense quantile grid. We introduced a *retrieval-augmented* approach that grounds distribution estimates with retrieved neighbor instances and their empirical distributions, and *Quantile Token Regression*, which predicts each quantile from a dedicated token representation formed through self-attention. Across Inside Airbnb and StackSample, both methods improve accuracy and yield sharper predictive intervals, with especially large gains on the smaller, more challenging StackSample dataset. We also analyzed training objectives, showing that Wasserstein matching better fits quantile-target supervision than pinball variants and offers a stronger accuracy–sharpness tradeoff in practice. Overall, combining retrieval-based grounding with quantile-specific representations is a simple, effective approach for scalable text-to-distribution prediction, motivating future work on variance-aware weighting, calibration under sharpness constraints, and broader applications.

Limitations

Our evaluation relies on empirical quantiles constructed from multiple observed outcomes per input and interpolated to a fixed quantile grid; since ground-truth values at every quantile level are typically unavailable, this interpolation can introduce approximation error, especially when each instance has only a few labels, and the resulting distributions may not fully reflect real-world conditional outcome distributions. We also evaluate only the Qwen3 model family, so the observed gains from retrieval augmentation and quantile tokens may depend on the backbone and should be validated on additional LLM families. In addition, we report point estimates for all metrics but do not provide uncertainty quantification such as bootstrap confidence intervals, which would strengthen statistical reliability when comparing close configurations. Finally, we do not study text-generation-based quantile prediction; prior work found that generation underperforms regression-style decoding for point estimation (Vedula et al., 2025), so we focus on regression-based quantile outputs and leave generative methods for distribution prediction to future work.

References

Siddharth Arora, James W. Taylor, and Ho-Yin Mak. 2023. Probabilistic forecasting of patient waiting times in an emergency department.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

R. R. Bahadur. 1966. A note on quantiles in large samples. *The Annals of Mathematical Statistics*, 37(3):577–580.

Zsolt Bitvai and Trevor Cohn. 2015. Non-linear text regression with a deep convolutional neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 180–185, Beijing, China. Association for Computational Linguistics.

Tingting Chen and Shijing Si. 2024. Predicting rental price of lane houses in Shanghai with machine learning methods and large language models. *arXiv preprint arXiv:2405.17505*.

Cheng-Han Chiang, Hung-yi Lee, and Michal Lukasik. 2025. TRACT: Regression-aware fine-tuning meets chain-of-thought reasoning for LLM-as-a-judge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2934–2952, Vienna, Austria. Association for Computational Linguistics.

Nicolai Dorka. 2024. Quantile regression for distributional reward models in rlhf. *arXiv preprint arXiv:2409.10164*.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Nate Gillman, Daksh Aggarwal, Michael Freeman, Saurabh Singh, and Chen Sun. 2025. Fourier head: Helping large language models learn complex probability distributions. In *Proceedings of ICLR*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36.

Wenjun Gu, Yihao Zhong, Shizun Li, Changsong Wei, Liting Dong, Zhuoyue Wang, and Chao Yan. 2024. Predicting stock prices with FinBERT-LSTM: Integrating news sentiment analysis. In *2024 8th International Conference on Cloud and Big Data Computing*.

Ragıp Gürlek, Francis de Véricourt, and Donald K.K. Lee. 2024. Boosted generalized normal distributions: Integrating machine learning with operations knowledge. Working paper, SSRN. Available at SSRN, Posted: August 1, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Joseph Marvin Imperial. 2021. BERT embeddings for automatic readability assessment. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 611–618, Held Online. INCOMA Ltd.

Inside Airbnb. 2025. Get the data. `https://insideairbnb.com/get-the-data/`. Dataset downloads (city snapshots). Licensed under CC BY 4.0. Accessed 2025-12-17.

Ryan Jacobs, Maciej P. Polak, Lane E. Schultz, Hamed Mahdavi, Vasant Honavar, and Dane Morgan. 2024. Regression with large language models for materials and molecular property prediction. *arXiv preprint arXiv:2409.06080*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Thomas Kneib, Alexander Silbersdorff, and Benjamin Säfken. 2023. Rage against the mean–a review of distributional regression approaches. *Econometrics and Statistics*, 26:99–123.

Roger W Koenker and Gilbert Bassett. 1978. Regression quantiles. *Econometrica*, 46(1):33–50.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Michal Lukasik, Zhao Meng, Harikrishna Narasimhan, Yin-Wen Chang, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. 2025. Better autoregressive regression with llms via regression-aware fine-tuning. In *Proceedings of ICLR*.

Xingyou Song, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, and Yutian Chen. 2024. Omnipred: Language models as universal regressors. *Trans. Mach. Learn. Res.*, 2024.

Stack Overflow. 2019. Stacksample: 10% of stack overflow q&a. Kaggle dataset. Kaggle-hosted dataset. The page shows "Updated 6 years ago" when accessed on 2025-12-17, implying a last-updated year of approximately 2019.

Stack Overflow. 2025. What is the license for the content i post? Stack Overflow Help Center. States Stack Overflow user contributions are licensed under CC BY-SA, with version depending on contribution date (2.5/3.0/4.0).

Eric Tang, Bangding Yang, and Xingyou Song. 2025. Understanding LLM embeddings for regression. *Transactions on Machine Learning Research*.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. 2024. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. *Preprint*, arXiv:2404.07544.

Nikhita Vedula, Dushyanta Dhyani, Laleh Jalali, Boris N. Oreshkin, Mohsen Bayati, and Shervin Malmasi. 2025. Quantile regression with large language models for price prediction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12396–12415, Vienna, Austria. Association for Computational Linguistics.

Cédric Villani. 2009. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer.

Hairu Wang, Sheng You, Qiheng Zhang, Xike Xie, Shuguang Han, Yuchen Wu, Fei Huang, and Jufeng Chen. 2025. Llp: Llm-based product pricing in e-commerce. *Preprint*, arXiv:2510.09347.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

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. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Appendix

A Theoretical Analysis of Quantile Supervision

This appendix formalizes the behavior of different objectives when training labels are *empirical quantiles* computed from finite samples. We justify the empirical ordering observed in our experiments: $\mathcal{L}_{\ell_1}$ performs best, $\mathcal{L}_{\text{Pinball-Q}}$ is slightly worse but better than $\mathcal{L}_{\text{Pinball-Med}}$, and pinball-style objectives are appropriate only in the extremely sparse supervision regime.

A.1 Latent sample model and quantile-label noise

Fix an input instance $X = x$. Let $F^*(\cdot \mid x)$ denote the true conditional distribution of $Y \mid X = x$ with (left-continuous) quantile function

$$Q^*(x, \tau) := \inf\{t \in \mathbb{R} : F^*(t \mid x) \geq \tau\}, \qquad \tau \in (0, 1).$$

For simplicity, here we assume M_i are equal to M. In our data, each input X_i is paired with a multiset of outcomes $\mathcal{Y}_i = \{y_{i1}, \ldots, y_{iM}\}$, which we model as

$$y_{i1}, \ldots, y_{iM} \overset{\text{i.i.d.}}{\sim} F^*(\cdot \mid X_i),$$

with a variable sample size M across instances. From $\mathcal{Y}_i$ we construct the empirical CDF $\widehat{F}_i(t) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}[y_{im} \leq t]$, and define the empirical quantile estimator (with the same interpolation rule as in the main text)

$$\widehat{Q}_i(\tau) := \widehat{F}_i^{-1}(\tau).$$

For theoretical analysis, we treat $\widehat{Q}_i(\tau)$ as a standard sample quantile estimator; the additional linear interpolation changes the estimator by at most $O(1/M)$ when $f(Q) > 0$ that does not change the asymptotics in large M regime.

Regularity assumption. We assume that for the quantile levels of interest $\tau \in [\tau_{\min}, 1 - \tau_{\min}]$: (i) $F^*(\cdot \mid x)$ is continuously differentiable in a neighborhood of $Q^*(x, \tau)$; (ii) the conditional density $f^*(t \mid x) = \partial_t F^*(t \mid x)$ exists and satisfies $f^*(Q^*(x, \tau) \mid x) > 0$.

Under these conditions, sample quantiles admit a Bahadur-type expansion and a central limit theorem (e.g., "Bahadur representation", (Bahadur, 1966)):

$$\widehat{Q}_i(\tau) = Q^*(X_i, \tau) + \frac{\tau - \widehat{F}_i(Q^*(X_i, \tau))}{f^*(Q^*(X_i, \tau) \mid X_i)} + r_{i,\tau}, \qquad r_{i,\tau} = o_p(M^{-1/2}). \tag{7}$$

As a consequence,

$$\sqrt{M}\Big(\widehat{Q}_i(\tau) - Q^*(X_i, \tau)\Big) \Rightarrow \mathcal{N}\left(0, \frac{\tau(1-\tau)}{f^*(Q^*(X_i, \tau) \mid X_i)^2}\right). \tag{8}$$

Thus $\widehat{Q}_i(\tau)$ can be viewed as a noisy measurement of $Q^*(X_i, \tau)$ with heteroskedastic noise that is (asymptotically) centered and symmetric. While the *vector* $(\widehat{Q}_i(\tau_1), \ldots, \widehat{Q}_i(\tau_Q))$ is correlated across τ (since all quantiles are computed from the same sample $\mathcal{Y}_i$), our population minimizer statements below are pointwise in τ and rely only on the marginal distribution in (8).

A.2 Population minimizers for ℓ_1 and ℓ_2 losses

Fix x and τ and define the random label $Z := \widehat{Q}_i(\tau) \mid (X_i = x)$. Consider a scalar prediction $a \in \mathbb{R}$.

Proposition 1. *The conditional risk minimizers satisfy:*

$$\arg\min_{a \in \mathbb{R}} \mathbb{E}\big[(Z-a)^2 \mid X = x\big] = \mathbb{E}[Z \mid X = x], \qquad \arg\min_{a \in \mathbb{R}} \mathbb{E}\big[|Z-a| \mid X = x\big] \in \text{Median}(Z \mid X = x).$$

Proof. Both losses are convex in a. For squared loss, differentiate the conditional risk and set to zero. For absolute loss, the subdifferential of $a \to \mathbb{E}|Z - a|$ is

$$\partial = \left[\mathbb{P}(Z < a|x) - \frac{1}{2}, \mathbb{P}(Z \le a|x) - \frac{1}{2}\right]$$

so $0 \in \partial$ if and only iff $\mathbb{P}(Z < a|x) \le \frac{1}{2} \le \mathbb{P}(Z \le a|x)$. This yields the median characterization. □

Proposition 2 (Asymptotic Fisher consistency of $\mathcal{L}_{\ell_1}$ and $\mathcal{L}_{\ell_2}$)**.** *Under the latent sample model and regularity assumptions above, for each fixed $(x, \tau) \in \mathcal{X} \times [\tau_{\min}, 1 - \tau_{\min}]$, the population targets of $\mathcal{L}_{\ell_1}$ and $\mathcal{L}_{\ell_2}$ converge to the true quantile:*

$$\mathbb{E}\Big[\widehat{Q}_i(\tau) \mid X_i = x\Big] = Q^*(x, \tau) + O(M^{-1}), \qquad \mathrm{Median}\Big(\widehat{Q}_i(\tau) \mid X_i = x\Big) = Q^*(x, \tau) + o(M^{-1/2}).$$

In particular, as $M \to \infty$, both losses are Fisher-consistent for $Q^*(x, \tau)$.

Proof sketch. Equation (8) implies that $\widehat{Q}_i(\tau)$ is asymptotically normal and hence asymptotically symmetric about $Q^*(x, \tau)$, yielding median consistency. Finite-sample bias of sample quantiles is typically $O(1/M)$ under smoothness, yielding the stated mean expansion. Applying Proposition 1 completes the argument. □

A.3 Bias of mismatched pinball on empirical quantiles (Pinball-Q)

Proposition 3. *Let $\rho_\tau(u) = u(\tau - \mathbb{I}[u < 0])$. For any scalar random variable Z,*

$$\arg\min_{a \in \mathbb{R}} \mathbb{E}\big[\rho_\tau(Z - a)\big] \in \mathrm{Quantile}_\tau(Z),$$

i.e., pinball loss targets the τ-th quantile of the label distribution.

Proof. Similar as the above proof for ℓ_1, the subdifferential of $a \mapsto \mathbb{E}[\rho_\tau(Z - a)]$ is

$$\partial = [\mathbb{P}(Z < a|x) - \tau, \mathbb{P}(Z \le a|x) - \tau]$$

so the optimality condition is $\mathbb{P}(Z < a|x) \le \tau \le \mathbb{P}(Z \le a|x)$. □

Proposition 4 (Bias of $\mathcal{L}_{\text{Pinball-Q}}$)**.** *Fix (x, τ) and write the empirical quantile label as*

$$\widehat{Q}_i(\tau) = Q^*(x, \tau) + \varepsilon_{i,\tau}, \qquad \varepsilon_{i,\tau} := \widehat{Q}_i(\tau) - Q^*(x, \tau).$$

The population minimizer of $\mathcal{L}_{\textit{Pinball-Q}}$ at level τ satisfies

$$q^*_{\textit{Pinball-Q}}(x, \tau) = Q^*(x, \tau) + \mathrm{Quantile}_\tau\big(\varepsilon_{i,\tau} \mid X_i = x\big).$$

Under the normal approximation in (8), the resulting inflation/deflation bias is approximately

$$q^*_{\textit{Pinball-Q}}(x, \tau) \approx Q^*(x, \tau) + \Phi^{-1}(\tau)\sqrt{\frac{\tau(1-\tau)}{M}}\,\frac{1}{f^*(Q^*(x, \tau) \mid x)}. \tag{9}$$

Proof. By Proposition 3, $\mathcal{L}_{\text{Pinball-Q}}$ targets the τ-quantile of the random label $\widehat{Q}_i(\tau) \mid X_i = x$. Quantiles are translation-equivariant, giving the first display. Under (8), $\varepsilon_{i,\tau}$ is approximately normal with standard deviation $\sqrt{\tau(1-\tau)}/(\sqrt{M}\, f^*(Q^* \mid x))$, and the τ-quantile of a centered normal is $\Phi^{-1}(\tau)$ times its standard deviation. □

Equation (9) shows that the leading $O(M^{-1/2})$ inflation term vanishes at $\tau = 0.5$, but Pinball-Q systematically inflates upper quantiles ($\tau > 0.5$) and deflates lower quantiles ($\tau < 0.5$). The magnitude of this mismatch is $O(M^{-1/2})$ and grows away from $\tau = 0.5$ over typical interior grids and can be further amplified when f^* is small (often in tails).

A.4 Scalarized pinball is inconsistent for distribution learning (Pinball-Med)

Define the scalar pseudo-target

$$y_i := \widehat{Q}_i(0.5),$$

and recall the scalarized objective $\mathcal{L}_{\text{Pinball-Med}}(\theta) = \frac{1}{NQ}\sum_{i,k}\rho_{\tau_k}(y_i - \hat{q}_{\tau_k}(X_i))$.

Proposition 5 (What $\mathcal{L}_{\text{Pinball-Med}}$ learns)**.** *For each $\tau \in (0,1)$ and fixed x, the population minimizer of $\mathcal{L}_{\textit{Pinball-Med}}$ at level τ satisfies*

$$q^*_{\textit{Pinball-Med}}(x,\tau) \in \text{Quantile}_\tau\big(y_i \mid X_i = x\big).$$

Consequently, $\mathcal{L}_{\textit{Pinball-Med}}$ is proper for the conditional distribution of the statistic $y_i = \widehat{Q}_i(0.5)$, not for $Y \mid X$.

Proof. Fix (x,τ) and view y_i as the random label. Applying Proposition 3 to $Z = y_i \mid X_i = x$ yields the claim. □

Proposition 6 (Collapse in the large-M regime)**.** *Assume $M \to \infty$ and the regularity conditions above. Then $y_i = \widehat{Q}_i(0.5) \to Q^*(x, 0.5)$ in probability, and hence*

$$q^*_{\textit{Pinball-Med}}(x,\tau) \to Q^*(x,0.5) \qquad \textit{for all } \tau \in (0,1).$$

If $F^*(\cdot \mid x)$ is non-degenerate, this implies a non-vanishing distributional error; for example,

$$W_1\big(F^*(\cdot \mid x),\ \delta_{Q^*(x,0.5)}\big) = \int_0^1 \big|Q^*(x,u) - Q^*(x,0.5)\big|\,du \ > \ 0.$$

Proof sketch. Consistency of the sample median follows from standard quantile consistency. As y_i concentrates, the conditional distribution of $y_i \mid X_i = x$ converges to a point mass at $Q^*(x, 0.5)$, whose τ-quantile equals the same point for every τ. The W_1 identity is standard in one dimension. □

A.5 Why $\mathcal{L}_{\text{Pinball-Q}}$ can outperform $\mathcal{L}_{\text{Pinball-Med}}$

Propositions 4-6 formalize a bias-information trade-off. Compared to Pinball-Med, Pinball-Q uses the full set of empirical quantiles $\{\widehat{Q}_i(\tau_k)\}_{k=1}^{Q}$, preserving supervision about distributional shape. Although Pinball-Q incurs an $O(M^{-1/2})$ bias away from $\tau = 0.5$, Pinball-Med introduces a *structural* error by collapsing the predicted quantile function toward the median when $M > 1$. In intermediate regimes where M is small but greater than one, the error from losing tail/shape information can dominate the mismatch inflation term, making $\mathcal{L}_{\text{Pinball-Q}}$ empirically stronger than $\mathcal{L}_{\text{Pinball-Med}}$. Finally, ℓ_1 on empirical quantiles combines the shape information of Pinball-Q with asymptotic Fisher consistency, explaining the empirical ordering $\mathcal{L}_{\ell_1} \succ \mathcal{L}_{\text{Pinball-Q}} \succ \mathcal{L}_{\text{Pinball-Med}}$ observed in our experiments.

A.6 Towards a variance-aware weighting

Equation (8) implies heteroskedastic noise across (M,τ). A variance-aware extension replaces the unweighted losses with weights proportional to the inverse asymptotic variance,

$$w_{i,k} \ \propto \ \frac{M\, f^*(Q^*(X_i,\tau_k) \mid X_i)^2}{\tau_k(1-\tau_k)},$$

yielding a quasi-likelihood weighted least squares objective. In practice, the unknown density factor can be estimated from the predicted quantile slope via $f^*(Q^*(x,\tau) \mid x) = 1/\partial_\tau Q^*(x,\tau)$ (when the quantile function is differentiable and strictly increasing), suggesting a fully data-adaptive weighting scheme. We leave a systematic study of these weights to future work.

B Quantile Interpolation

Each instance is associated with a variable-size set of observed outcomes $\mathcal{Y}_i = \{y_{i1}, \ldots, y_{iM_i}\}$, where M_i can differ across instances. We sort them to obtain order statistics $y_{i(1)} \leq \cdots \leq y_{i(M_i)}$ and treat these samples as defining an empirical quantile function. To obtain a fixed-dimensional training target, we interpolate the empirical quantile function to a dense grid of $Q = 99$ quantile levels $\boldsymbol{\tau} = \{0.01, 0.02, \ldots, 0.99\}$, producing

$$\widehat{\mathbf{q}}_i = \left(\widehat{Q}_i(0.01), \ldots, \widehat{Q}_i(0.99)\right) \in \mathbb{R}^{99}. \tag{10}$$

For each τ_k, we compute a fractional rank $r_{ik} = 1 + (M_i - 1)\tau_k$ and set $\widehat{Q}_i(\tau_k)$ by linear interpolation between $y_{i(\lfloor r_{ik} \rfloor)}$ and $y_{i(\lceil r_{ik} \rceil)}$.

C Dataset Construction Details

Airbnb. We collect all available cities from Inside Airbnb data snapshots between 2024-09 and 2025-08, covering 119 cities. We drop listings with fewer than 4 price observations and remove cities with fewer than 10k samples after filtering, yielding ∼840k samples from 55 cities. For retrieval, we build one training index per city and restrict retrieved neighbors to the same city. For each retrieved neighbor, we append only its title to the model input.

StackSample. We filter out answers with response time exceeding 12 hours, convert each remaining response time to minutes, then apply log transformation to ensure a manageable range. We perform quantile interpolation over the log-transformed response times to create the ground-truth quantile distribution. For retrieval, we build an index from training questions only. For each retrieved neighbor question, we append only its title to the model input.

D Experimental Setup Details

We run all experiments on a single AWS GPU cluster with NVIDIA H100 and H200 GPUs, set $Q = 99$ with uniformly spaced quantile levels, and tune hyperparameters (e.g., number of epochs, batch sizes, learning rates, and maximum sequence length) across settings, model sizes, and datasets to report the best configuration. Unless otherwise specified, we use $K = 8$ retrieved neighbors. We fine-tune the Qwen3 model family (Yang et al., 2025) with LoRA (Hu et al., 2022) using the HuggingFace Transformers library (Wolf et al., 2020).

E Evaluation Metrics

We evaluate quantile predictions on a test set $\{(x_i, y_i)\}_{i=1}^n$, where the model outputs $\hat{q}_\tau(x_i)$ for $\tau \in \{\tau_k\}_{k=1}^Q$. Since each sample has a ground-truth distribution, we first compute the Mean Absolute Percentage Error (MAPE) at each quantile in a coarse set of target quantiles $\mathcal{T} = \{0.1, 0.2, \ldots, 0.9\}$. Then we compute the average over them, which we refer to as averageMAPE:

$$\text{MAPE@}\tau = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{\hat{q}_\tau(x_i) - q_\tau(x_i)}{q_\tau(x_i)} \right|, \tag{11}$$

$$\text{avgMAPE} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \text{MAPE@}\tau, \tag{12}$$

We additionally report the Weighted Mean Absolute Percentage Error (wMAPE) and the Symmetric Mean Absolute Percentage Error (sMAPE) using the median prediction $\hat{y}_i = \hat{q}_{0.5}(x_i)$. We use $y_i = q_{0.5}(x_i)$ as the benchmark value, which is the median of the ground-truth distribution.

$$\text{wMAPE} = 100 \cdot \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{\sum_{i=1}^n |y_i|}, \tag{13}$$

$$\text{sMAPE} = \frac{200}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{|\hat{y}_i| + |y_i|}. \tag{14}$$

Model	Method	K	avg MAPE↓	wMAPE↓	sMAPE↓	CRPSS↑	RCIW@90↓	RCIW@95↓	RCIW@99↓
Qwen3-1.7B	QR	0	32.01	45.93	37.18	0.4706	14.06	17.98	21.05
	QR	8	24.95	37.68	25.64	0.5750	15.01	21.57	29.66
	QT	8	22.82	34.78	24.16	0.5901	3.72	5.18	7.35
Qwen3-4B	QR	0	32.01	45.64	37.36	0.4680	10.00	12.76	14.92
	QR	8	23.54	35.79	24.30	0.5904	9.60	12.37	14.89
	QT	8	22.70	34.59	24.13	0.5952	5.10	7.12	9.64
Qwen3-8B	QR	0	30.92	43.31	36.12	0.4950	8.26	10.52	12.29
	QR	8	22.56	34.32	23.30	0.6036	6.65	8.48	9.91
	QT	8	**22.35**	33.55	23.51	0.6058	**3.76**	**5.16**	**6.58**
Qwen3-14B	QR	0	29.11	40.53	32.21	0.5332	11.54	14.72	17.20
	QR	8	23.33	35.23	24.50	0.6016	13.23	16.88	19.78
	QT	8	22.36	**33.31**	**22.95**	**0.6128**	5.38	8.41	11.70

Table 7: Results on the Airbnb OOD test set (Los Angeles).

For the evaluation of distributional quality, we report the Continuous Ranked Probability Skill Score (CRPSS) and Relative Coverage Interval Width (RCIW). CRPSS is

$$\mathrm{CRPSS} = 1 - \frac{\mathrm{CRPS}_{\text{model}}}{\mathrm{CRPS}_{\text{ref}}}, \tag{15}$$

where $\mathrm{CRPS}_{\text{ref}}$ is computed from a reference forecast. We report $\mathrm{RCIW}@c$ for $c \in \{90, 95, 99\}$:

$$\mathrm{RCIW}@c = \frac{100}{n} \sum_{i=1}^{n} \frac{\hat{q}_{\tau_u(c)}(x_i) - \hat{q}_{\tau_\ell(c)}(x_i)}{|\hat{q}_{0.5}(x_i)|}, \tag{16}$$

where (τ_ℓ, τ_u) are chosen to approximate the desired central coverage (for Q=99, we use $(0.05, 0.95)$, $(0.02, 0.98)$, and $(0.01, 0.99)$ for c=$90, 95, 99$, respectively).

F Experiments Results on OOD Dataset

As in 7, the LA test set can appear easier than the multi-city test split for two complementary reasons. First, Los Angeles is a large, high-density city in the dataset, so retrieval can find closer and more informative neighbors, leading to stronger grounding for distribution prediction. Second, U.S. cities and listings constitute a substantial portion of our training data, so LA is not far from the dominant training distribution in both language and pricing patterns. As a result, the holdout primarily reflects a city-level split rather than a severe domain shift, which helps explain why OOD performance can be comparable to or even better than the stratified test set.

G Hyperparameter Configurations

This section provides the hyperparameter configurations used in our experiments. All models were trained using LoRA fine-tuning on the Qwen3 family. Table 8 shows hyperparameters for Inside Airbnb experiments, and Table 9 shows hyperparameters for StackSample experiments.

Model	Method	K	LR	Epochs	Train BS	Eval BS	LoRA Rank	Max Len
Qwen3-1.7B	QR	0	3e-6	5	64	64	384	1024
Qwen3-1.7B	QR	8	3e-6	5	64	64	384	2048
Qwen3-1.7B	QT	8	3e-6	5	16	8	384	2048
Qwen3-4B	QR	0	2e-6	5	32	32	384	1024
Qwen3-4B	QR	8	2e-6	5	32	32	384	2048
Qwen3-4B	QT	8	2e-6	5	16	16	384	2048
Qwen3-8B	QR	0	2e-6	5	16	16	384	1024
Qwen3-8B	QR	8	2e-6	5	16	16	384	2048
Qwen3-8B	QT	8	2e-6	5	16	16	384	2048
Qwen3-14B	QR	0	1e-6	5	32	32	384	1024
Qwen3-14B	QR	8	1e-6	5	16	16	384	2048
Qwen3-14B	QT	8	1e-6	5	8	8	384	2048

Table 8: Hyperparameters for Inside Airbnb experiments. K = number of neighbors, LR = learning rate, BS = batch size per device, Max Len = maximum sequence length. All models use weight decay = 0.01, LoRA dropout = 0.1, and Wasserstein W1 loss.

Model	Method	K	LR	Epochs	Train BS	Eval BS	LoRA Rank	WD	Warmup	Max Len
Qwen3-4B	QR	0	3e-7	8	32	64	64	0.01	0.0	1280
Qwen3-4B	QR	8	3e-7	8	32	32	64	0.01	0.0	2304
Qwen3-4B	QT	8	3e-7	5	16	16	192	0.01	0.1	2304

Table 9: Hyperparameters for StackSample experiments. K = number of neighbors, LR = learning rate, BS = batch size per device, WD = weight decay, Warmup = warmup ratio, Max Len = maximum sequence length. QR models use LoRA dropout = 0.1, QT model uses LoRA dropout = 0.15. All use Wasserstein W1 loss.