Boosting Text-to-Chart Retrieval through Training with Synthesized Semantic Insights

Anonymous ACL submission

Abstract

Text-to-chart retrieval, enabling users to find relevant charts via natural language queries, has gained significant attention. However, evaluating models in real-world business intelligence (BI) scenarios is challenging, as current benchmarks fail to simulate realistic user queries or test for deep semantic understanding with static chart images. To address this evaluation gap, we introduce CRBench, the first benchmark for this task sourced from real-world BI applications, with 21,862 charts and 326 validated text queries. Testing on CRBench reveals that existing methods, which rely primarily on visual features, perform poorly and fail to capture the rich analytical semantics of charts. To address this performance bottleneck, we propose a semantic insights synthesis pipeline that automatically generates three hierarchical levels of insights for charts: visual patterns, statistical properties, and practical applications. Using this pipeline, we produced 207,498 semantic insights for 69,166 charts as training data. We use this data to fine-tune the LongCLIP architecture, creating a specialized model we name ChartFinder. Experimental results show ChartFinder significantly outperforms state-of-the-art methods on CRBench, achieving up to 66.9% NDCG@10 for precise queries (an 11.58% improvement) and an average increase of 5% across nearly all metrics for fuzzy queries. This work provides the community with a much-needed benchmark for realistic evaluation and demonstrates a powerful data synthesis paradigm for enhancing a model's semantic understanding of charts.

1 Introduction

Visualization charts are crucial tools in data analysis and business decision-making, providing intuitive ways for users to understand complex data (Wu et al., 2024; Luo et al., 2018a; Shen et al., 2022; Qin et al., 2020). Consequently, text-to-chart retrieval systems that allow users to find relevant charts based on natural language queries (*i.e.,* text queries) have become increasingly valuable, attracting significant attention from both the database community (Ji et al., 2024b,a; Luo et al., 2023) and industry practitioners (Setlur et al., 2023). In real-world BI scenarios, queries range from *precise queries* targeting specific analytical information (*e.g.,* "Find charts showing hotel booking distribution in Europe for 2022") to *fuzzy queries* for exploratory analysis (*e.g.,* "Use a pie chart to compare different sale methods") (Ji et al., 2024a,b). Both require a deep understanding of the rich analytical semantics embedded in charts, such as data relationships and statistical trends (Qin et al., 2020). This challenge is exacerbated when charts exist only as static images without accessible metadata, a common scenario where visual-based methods inherently struggle (Luo et al., 2023; Ji et al., 2024a; Li et al., 2022a).

A core obstacle impeding progress in this field is the lack of a dedicated benchmark for realistic evaluation. Existing chart-related datasets are primarily designed for tasks like captioning and fail to simulate real-world retrieval challenges. They lack concise, intent-driven queries and a large repository of visually and semantically similar charts to test a model's true discriminative power. To address this critical evaluation gap, we first curate and introduce CRBench, the first benchmark for text-to-chart retrieval sourced from real-world BI scenarios. CRBench contains 326 text queries and 21,862 chart images, with all query-chart labels verified by crowd workers to capture the diversity of real user needs.

Benchmarking existing models, including powerful Multimodal Large Language Models (MLLMs) (Liu et al., 2022; Xiao et al., 2024b; Zhou et al., 2024b; Zhang et al., 2024), on CRBench reveals their significant limitations. These models struggle to capture the deep analytical semantics required to differentiate between challenging charts. To address this performance bottle-

neck, we argue that the key lies in providing higher-quality training data that explicitly teaches models to bridge the gap between visual features and their analytical meaning. To this end, we propose our second major contribution: an automatic semantic insights synthesis pipeline. Inspired by automatic data visualization (Luo et al., 2018a) and insight generation methods (Ma et al., 2021), this pipeline automatically synthesizes three hierarchical levels of insights: *(1) Visual-oriented*, *(2) Statistics-oriented*, and *(3) Task-oriented*.

To validate the challenge of our benchmark and the effectiveness of our data pipeline, we use our synthesized data to fine-tune several CLIP-based models. We name the model resulting from fine-tuning Long-CLIP (Zhang et al., 2025) as ChartFinder. The core idea is to leverage the synthesized insights exclusively during the training stage to learn robust visual-semantic representations. During retrieval, the model relies solely on chart images.

Contributions. We make the following contributions.

(1) **CRBench Benchmark.** We curate CRBench, *the first* benchmark for evaluating text-to-chart retrieval sourced from *real-world BI scenarios*. It includes 326 text queries and 21,862 charts, with query-chart labels verified by crowd workers to capture the diversity of real user needs (Section 3).

(2) **A Semantic Insights Synthesis Pipeline for Charts.** We develop an automatic pipeline for generating semantic insights from chart metadata, synthesizing three levels of semantic insights, *i.e.,* visual-oriented, statistics-oriented, and task-oriented insights, *as training data*. This approach addresses the challenge of manual annotation by automating the process, enabling the creation of diverse insights that enrich the model's ability to comprehensively understand charts (Section 4).

(3) **A High-Performance Model Validating the Pipeline.** By fine-tuning the CLIP architecture with our synthesized data, we produce ChartFinder. It not only validates the effectiveness of our data synthesis pipeline by establishing a new state-of-the-art but also serves as a strong baseline for future research on CRBench.

(4) **Extensive Experiments.** Extensive experiments validate both the challenge of CRBench as a benchmark and the effectiveness of our synthesis pipeline. The results demonstrate that our synthesized data significantly improves the performance of multiple CLIP-based architectures across several benchmarks, including in challenging zero-shot scenarios (Section 5).

2 Related Work

Chart Retrieval. Chart retrieval aims to find similar charts from a repository using a chart as a query. Existing methods follow two main paradigms: perception-based and data-based. Perception-based approaches analyze the chart image, using either visual style and structure (Saleh et al., 2015; Hoque and Agrawala, 2019), deep visual embeddings (Luo et al., 2023), or hand-drawn sketches of trendlines to find matches (Mannino and Abouzied, 2018). In contrast, data-based approaches assume access to the underlying data of the query chart and use it to guide the retrieval process (Wang and Palpanas, 2021; Lekschas et al., 2020; Siddiqui et al., 2016). Chart search has similarities with text-to-chart retrieval, particularly in terms of underlying semantic complexity. However, our work fundamentally differs by addressing the additional challenge of aligning natural language queries with chart semantics, especially when metadata is unavailable.

Text-to-Chart Retrieval. Previous work on text-to-chart retrieval has largely followed two paths: visualization recommendation systems that guide user exploration (Vartak et al., 2017; Ye et al., 2024; Qin et al., 2018b,a; Bendeck et al., 2024; Srinivasan and Setlur, 2021) and natural language interfaces that accept analytical queries (Shen et al., 2022; Tang et al., 2022; Luo et al., 2018b, 2022b, 2021; Setlur et al., 2023; Srinivasan and Setlur, 2023). While significant progress has been made, these methods often struggle to fully understand user intent. We believe embedding queries and charts into a shared latent space offers a more robust solution. The development of Multimodal Large Language Models (MLLMs) (Li et al., 2022b, 2023; Yuan et al., 2021; Xiao et al., 2024a) and pre-trained models like CLIP (Radford et al., 2021) has enabled text-to-chart retrieval to be treated as a cross-modal task. Numerous variants have been developed to enhance CLIP's performance, including those with improved visual encoders like EVA-CLIP (Sun et al., 2023; Fang et al., 2024), better text encoders like Jina-CLIP-v2 (Xiao et al., 2024b; Koukounas et al., 2024; Sturua et al., 2024), or mod-

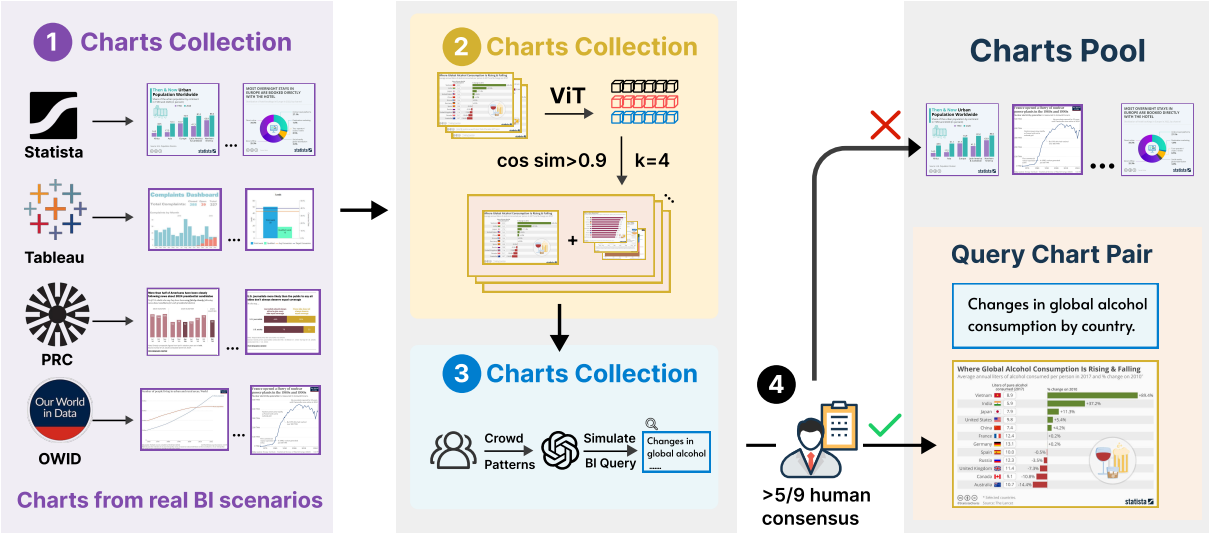


Figure 1: The construction workflow of the CRBench.

els improved through large-scale fine-tuning such as UniVL-DR (Liu et al., 2022), MagicLens (Zhang et al., 2024), VISTA (Zhou et al., 2024a), and MARVEL (Zhou et al., 2024b). However, research focusing MLLM-based methods specifically on chart retrieval remains limited. This paper addresses that gap by using contrastive learning to inject deep semantic insights into our model, ChartFinder.

3 CRBench Benchmark

In this section, we first discuss the motivation for curating a new text-to-chart retrieval benchmark (Section 3.1), followed by an introduction to the benchmark curation process (Section 3.2). After that, we introduce the key characteristics of the first text-to-chart benchmark CRBench (Section 3.3). Finally, we discuss the data quality and limitations of CRBench.

3.1 Motivation and Design Consideration

To rigorously evaluate text-to-chart retrieval models, a specialized benchmark is required. However, existing chart-related datasets are ill-suited for this task. While chart-to-text datasets like *VisText* (Tang et al., 2023) and *Chart-to-Text* (Kantharaj et al., 2022) are valuable resources, they were fundamentally designed for generation tasks, such as captioning, not retrieval. This creates several critical limitations when they are repurposed for retrieval.

First, their "queries" are typically long, descriptive captions, rather than the concise, intent-driven natural language phrases in real-world BI systems. Second, and more importantly, they often feature a one-to-one or one-to-two chart-to-text ratio. This kind of ratio fails to simulate a realistic retrieval environment, where a single user query can be matched against a massive repository containing thousands of visually and semantically similar charts.

Consequently, a significant gap exists for a benchmark that can evaluate a model's performance on realistic, intent-driven queries against a large and challenging chart repository.

Design Consideration. To fill this gap, we aim to propose a real-world dataset designed for text-to-chart retrieval, enabling the fair comparison of retrieval models across various use cases. Several key factors were considered, as outlined below:

(1) *Precise and Fuzzy Queries*: As discussed in Section 1, user queries in real-world applications can be both precise (focused on specific details) and fuzzy (more general). Our benchmark includes both types of queries to reflect real-world user behavior and more accurately evaluate model performance across varying query complexities.

(2) *Distributional Separation*: To ensure a fair and rigorous evaluation of model generalization, we maintain a strict separation between the data sources of our training set and the benchmark itself. Therefore, while our training data is generated from Kaggle datasets, the charts for CRBench are sourced exclusively from different, real-world applications not used in training.

(3) *Query Realism and Unambiguity*: The bench-

mark must feature queries that reflect the full spectrum of real-world user behavior, including both precise and fuzzy types. Crucially, to enable rigorous and reliable evaluation, each query must be constructed to have a single, unambiguous ground-truth chart, even in the presence of highly similar distractors.

3.2 Benchmark Construction

The key challenge in constructing CRBench is ensuring each query has a clear, unambiguous ground-truth chart, particularly when evaluating retrieval performance on visually similar charts. To systematically address this, we adopt a clearly defined four-stage methodology, as illustrated in the Figure 1.

Step 1: Real-world Chart Collection. To ensure CRBench reflects real-world chart retrieval scenarios, we collected charts from reliable sources widely used in business intelligence and data analysis: (i) *Tableau*: A leading data visualization platform for charting and business analytics (Tableau, 2003). (ii) *Statista*: A platform providing statistical data across various sectors (statista, 2007). (iii) *Pew Research Center*: A non-profit organization focusing on social science research (Research, 2004). (iv) *Our World in Data*: An open-access platform researching global challenges through charts on topics like economics, health, and society (OWID, 2011). From these sources, *we gathered* ***21,862 charts*** covering various domains, including finance, business, e-commerce, technology, and more.

Step 2: Similarity-based Grouping. To construct a challenging retrieval environment based on the "target-and-distractor" paradigm, we first embedded all charts into 768-dimensional vectors using a pre-trained *ViT* and computed their pairwise cosine similarities.

The grouping process was then performed as follows: we iterated through each chart and identified its four most visually similar counterparts. A challenging group was formed only if the similarity score between this initial chart and all four of its counterparts was ≥ 0.90. If this strict condition was met, the initial chart that served as the anchor for the group was designated as the target chart, with the other four serving as distractors. Through this systematic process, we ultimately generated 247 distinct groups, each containing one clearly defined target and four highly similar distractors.

Step 3: Text Query Generation.

To generate high-quality and unambiguous queries, we initiated the process with an exploratory crowdsourcing study on the Appen platform. In this study, we presented crowdworkers with the previously created groups of five visually similar charts (one target and four distractors). Their task was to author a query for the target chart that could uniquely distinguish it from the other four.

However, we found that the quality of these manually generated queries was often poor. Many queries were either too general to be effective for differentiation or were simply literal paraphrases of the chart's title. This outcome highlighted the need for a more controlled and reliable generation process.

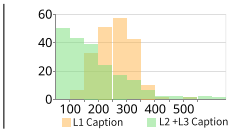
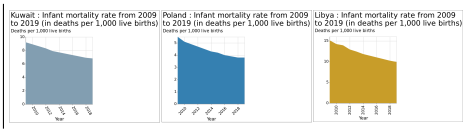
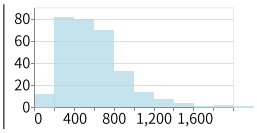
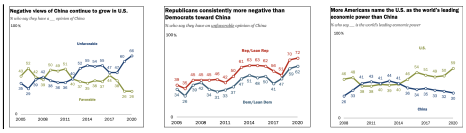
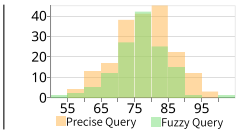
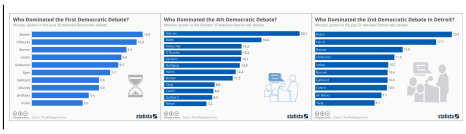
Therefore, we transitioned to a controlled query-generation methodology using GPT-4o. Crucially, the prompt designed for GPT-4o was informed by the patterns and shortcomings observed in the initial human-generated queries from the first step. The model was explicitly instructed to generate precise and fuzzy queries aimed at distinguishing a given target chart from its four distractors, thereby overcoming the generality issues we had identified and ensuring the queries were highly targeted.

Step 4: Human Consensus Validation. Finally, we performed a stringent multi-worker validation to ensure each query was unambiguous. Each GPT-4o-generated query, along with its corresponding five similar charts (one target and four distractors), was reviewed by nine independent crowdworkers. A query was accepted into CRBench only if a strong consensus of at least five out of the nine workers was reached. Through this rigorous quality control, we ultimately retained 195 precise queries and 131 fuzzy queries, each validated to have a single, unambiguous ground-truth chart.

3.3 CRBench Overview

In this section, we compare CRBench with two existing benchmarks, Chart-to-Text (Kantharaj et al., 2022) and VisText (Tang et al., 2023), both of which focus on chart-related tasks like caption generation and description. While these benchmarks are useful for evaluating chart captioning and description generation, they do not address the text-to-chart retrieval task. In our study, we repurpose the captions in these benchmarks as queries for retrieval tasks, enabling us to compare their effectiveness with CRBench in retrieving relevant charts

Table 1: Comparison of Existing Benchmarks. The statistics for Chart-to-Text and VisText are from the respective test sets. The column "*C/Q*" indicates the ratio for charts (C) and text queries (Q). The column "*Dist. of Query Len.*" Shows the distribution of lengths of text queries in characters. The column "*CT*" means the types of charts.

Datasets	CT	#-Charts	#-Queries	C/Q	Dist. of Query Len.	Samples of Similar Charts
VisText	3	882	1764	0.5		
Chart-to-Text	6	1393	1393	1		
CRBench	11	21,862	326	67		

based on text queries. Specifically, in VisText, each chart is paired with two types of captions: L1 captions (*e.g.,* basic properties) and L2+L3 captions (*e.g.,* trends and statistics). This results in a chart-to-query ratio of 1:2. In contrast, Chart-to-Text pairs each chart with a single caption, resulting in a 1:1 ratio. In CRBench, however, the ratio is much higher (67:1), meaning the number of charts significantly exceeds the number of queries, which is more representative of real-world scenarios where charts vastly outnumber queries.

Table 1 shows the distribution of query lengths in each benchmark. In VisText, the query lengths for L1 and L2+L3 captions typically range from 100 to 400 characters. Chart-to-Text features a wider range of query lengths, with some queries reaching up to 2400 characters. CRBench, however, features more concise queries, with both 195 precise and 131 fuzzy queries generally ranging from 0 to 100 characters, better reflecting how queries are typically structured in real-world applications.

Moreover, we compared examples of similar charts across the benchmarks. In VisText, similar charts show strong visual resemblance but often differ significantly in semantic content. In Chart-to-Text, charts deemed similar tend to align well semantically but differ visually. On the other hand, CRBench ensures that similar charts are closely aligned both visually and semantically, posing a significant challenge for models to retrieve the ground truth. Overall, CRBench presents a more challenging and realistic benchmark for evaluating text-to-chart retrieval models.

3.4 Data Quality and Limitations

While crowdsourcing was used to generate queries, we addressed quality issues through a second round of crowdsourcing and GPT-4o. This process helped ensure the relevance and clarity of the queries, resulting in a high-quality set of precise and fuzzy queries.

Despite careful curation, crowdsourcing variability and the evolving landscape of data visualizations may require periodic updates to CRBench to maintain its relevance and accuracy.

4 Semantic Insights Synthesis Pipeline

As we mentioned in the Introduction, we need high-quality caption data for model training to retrieve correct charts. However, manually annotating charts to generate semantic insights is time-consuming and inefficient. To address this, we propose a training data development pipeline that automatically synthesizes semantic insights from raw data, ensuring the data's effectiveness in supporting the training of text-to-chart retrieval models. Figure 2 shows our proposed pipeline, which consists of three key steps: data collection, automatic data visualization, and semantic insights generation based on the metadata (Figure 2: ❷–❺).

4.1 Data Collection and Visualization

Step-1: Data Collection. The first step in generating high-quality semantic insights is selecting a reliable source of datasets. To meet the above criteria, we select datasets from Kaggle, which offers several advantages: first, almost all the datasets are derived from real-world applications or competi-

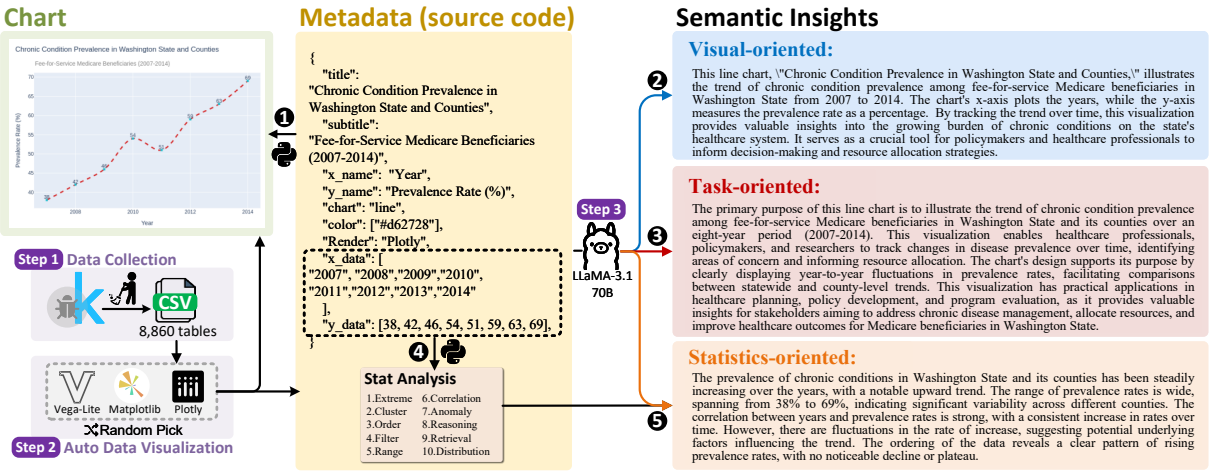


Figure 2: An overview of the semantic insights synthesis pipeline.

tions. Second, Kaggle provides a scoring system for datasets, with those receiving a score of 10 typically offering comprehensive content and high data quality.

We crawled 9,003 datasets with a score of 10 from Kaggle and performed local preprocessing, including removing missing values. After cleaning, we retained ***8,860 high-quality tables*** (in CSV format), completing the data preparation for subsequent steps.

Step-2: Automatic Data Visualization. Next, we used DeepEye (Luo et al., 2018a), an automatic data visualization system, to generate a variety of charts, including bar charts, pie charts, line charts, scatter plots, grouped line charts, stacked bar charts, and grouped bar charts, covering widely-used chart types (Luo et al., 2022a). In our implementation, DeepEye generated 69,166 charts based on the 8,860 tables. For each visualization, we also derived its chart metadata in JSON format, which includes details like axis labels, chart titles, and data points. To enhance the visual diversity of the charts, we re-rendered them by randomly using a visualization libraries from Matplotlib, Plotly, and Vega-Lite (see Figure 2: ❶).

We further enhance the visual diversity of generated charts by systematically randomizing various visual parameters. For example, adjusting line type, marker shape, or switching between sub-types such as pie and donut charts. We also store the metadata produced by DeepEye (Luo et al., 2018a) in the source code of the visualization, as shown in Figure 2. This metadata is essential for deriving semantic insights in Step 3. In total, we generated ***69,166 charts*** using Matplotlib, Plotly, or Vega-Lite, along with their corresponding *metadata*.

4.2 Semantic Insights Synthesis

The third step is to generate detailed chart semantic insights based on the metadata of the charts. To comprehensively interpret the charts, we design three levels of semantic insights: *(1)* The apparent visual patterns observed in the chart. *(2)* Data-driven statistical insights that uncover trends, comparisons, and relationships. *(3)* Task-oriented insights that provide context on how the chart can be applied in real-world scenarios for decision-making and resource allocation. This layered approach allows for a holistic interpretation of each chart, supporting diverse user queries and enhancing the retrieval process. As illustrated in Figure 2 (❷–❺), the semantic insights I are generated through the following steps:

Visual-oriented Insight Generation (Figure 2-❷). In this step, we generate visual-oriented insights by analyzing the chart's visual patterns and trends. This insight summarizes the chart's overall visual presentation, such as identifying significant trends, distributions, and patterns in the data.

Task-oriented Insight Generation (Figure 2-❸). Next, we use both the metadata and the visual insights to derive task-oriented insights. This step contextualizes the chart for practical applications, helping users understand how the chart can be used in real-world scenarios.

Statistics-oriented Insight Generation (Figure 2-❺). First, we apply a set of 10 statistical analysis tasks (Wu et al., 2024) on the chart's metadata (*e.g.,* the X/Y-axes), as shown in Figure 2-❹. These

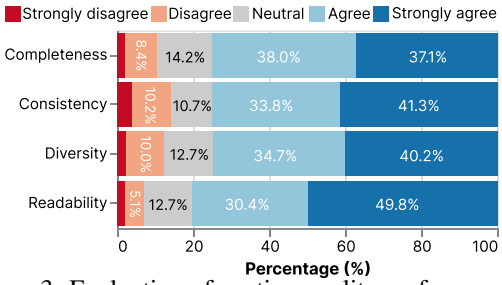


Figure 3: Evaluation of caption quality on four aspects

tasks help identify statistical properties such as trends, correlations, and anomalies within the data. In the second step of our pipeline, we generate statistics-oriented insights by summarizing this statistical information.

This provides a deeper understanding of the data patterns and their significance.

After all the processes, we got a total of ***69,166 charts*** and corresponding ***207,498 semantic insights***.

4.3 The Quality of Semantic Insights

To verify the quality of the semantic insights generated, we conducted a crowdsourcing experiment. We randomly selected 50 semantic insights and invited 100 crowdworkers to rate them on four dimensions: Completeness, Consistency, Diversity, and Readability, using a 5-point scale (from "strongly disagree" to "strongly agree"). As shown in Figure 3, the evaluation results are very positive. The total number of positive ratings of "agree" and "strongly agree" exceeded 74% in all four dimensions, especially readability, which received more than 80% approval. This result fully confirms the high quality of our synthesized insights. These high-quality data are the cornerstone of our subsequent training of text-graph retrieval models. The good generalization ability of the model in other benchmarks also further verifies the effectiveness of our data generation pipeline.

5 Experiments

In this section, we conduct a set of experiments to evaluate whether the synthesized semantic insights can enhance text-to-chart retrieval performance across multiple benchmarks and settings. We also assess whether the ChartFinder model achieves state-of-the-art performance against all baselines. The following experiments are performed:

Exp-1: How does the overall performance of ChartFinder compare with other baselines on CRBench?

Exp-2: How does ChartFinder perform when the text query becomes a long caption?

Exp-3: Ablation study on the combination of synthesized semantic insights.

Exp-4: The generalizability of synthesized semantic insights.

Exp-5: The impact of center crop strategy on text-to-chart retrieval. In the main body of the paper, we will focus on the detailed analysis of Exp-1. Due to page limitations, the comprehensive results and discussions for Exp-2 through Exp-5 are provided in the Appendix A.

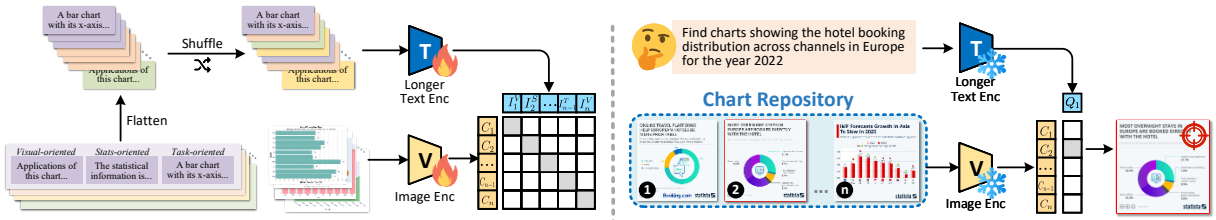
Metrics. To evaluate the effectiveness of the model in text-to-chart retrieval, we use the following metrics: Recall@1, Recall@5, Recall@10, MRR@10, and NDCG@10. These metrics are widely adopted in information retrieval tasks (Qin et al., 2020)

Methods. We evaluate 10 methods in our experiments: (1) VISTA (Zhou et al., 2024a), (2) Univl-dr (Liu et al., 2022), (3) MARVEL (Zhou et al., 2024b), (4) CLIP (Radford et al., 2021), (5) CLIP-DPR (Liu et al., 2022), (6) EVA-CLIP (Sun et al., 2023), (7) CoCa (Cherti et al., 2023), (8) Jina-CLIP-v2 (Koukounas et al., 2024), (9) Long-CLIP (Zhang et al., 2025) (with Long-CLIP-B for the Base version and Long-CLIP-L for the Large version, which differ by the size of the vision encoder used). (10) Our Method: **ChartFinder** and **ChartFinder-B**: Both versions use the same Long-CLIP text encoder but differ in their visual encoder. *ChartFinder uses the larger ViT-L/14, while ChartFinder-B uses the smaller ViT-B/16.* In all experiments, we changed the image preprocessing operation from center crop to direct resize, because we found that cropping would crop important information such as the title of the image.

Training Details. We trained ChartFinder using 4 A800 GPUs with a learning rate of $1e-6$ for 20 epochs. For the base version of the model, we set the batch size to 512, while for the larger version, we reduced the batch size to 256 due to the increased memory usage associated with the larger model.

5.1 Exp-1: Overall Results on CRBench

Experiment Settings. The purpose of this experiment is to evaluate and compare the performance of



(a) Training Process with Semantic Insights of Charts

(b) Retrieval Process based on User Query

Figure 4: (a) ChartFinder is trained using both chart images and their corresponding semantic insights. (b) ChartFinder takes a query and a set of chart images as input, retrieving the most relevant charts based on embedding similarity.

Table 2: Performance Comparison of Text-to-Chart Retrieval Models on CRBench Benchmark.

Models	**#Param**	Precise Query					Fuzzy Query				
		R@1	R@5	R@10	MRR@10	NDCG@10	R@1	R@5	R@10	MRR@10	NDCG@10
VISTA (Zhou et al., 2024a)	196M	8.21	20.00	27.18	13.15	16.42	9.16	19.08	24.43	13.16	15.8
Univl-DR (Liu et al., 2022)	151M	4.10	12.82	15.38	7.84	9.65	3.82	10.69	14.50	6.90	8.70
MARVEL (Zhou et al., 2024b)	310M	1.54	5.13	7.69	3.08	4.63	0.00	0.76	1.53	0.46	2.23
CLIP (Radford et al., 2021)	151M	12.31	26.15	33.33	19.01	22.43	10.69	21.37	25.95	15.69	18.15
CLIP-DPR (Liu et al., 2022)	151M	9.74	18.46	23.59	13.60	15.95	5.34	13.74	16.79	8.51	10.48
EVA-CLIP (Sun et al., 2023)	149M	5.13	19.49	24.10	11.08	14.21	3.05	9.92	13.74	6.40	8.14
CoCa (Cherti et al., 2023)	151M	8.72	21.54	25.1	14.11	16.76	3.05	8.40	9.92	5.19	6.33
Jina-CLIP-v2 (Koukounas et al., 2024)	865M	4.10	16.92	25.13	10.01	13.58	3.05	10.69	14.50	5.84	7.87
Long-CLIP-B (Zhang et al., 2025)	149M	26.67	53.85	62.56	37.99	43.90	22.90	40.46	51.91	30.29	35.33
ChartFinder-B	149M	38.97	62.05	70.26	49.30	61.30	29.77	50.38	61.83	38.96	44.41
Long-CLIP-L (Zhang et al., 2025)	427M	41.03	75.90	79.49	55.32	55.32	38.93	67.18	74.81	50.99	56.77
ChartFinder	427M	**47.18**	**80.00**	**84.10**	**61.23**	**66.90**	**41.98**	**75.57**	**80.15**	**55.31**	**61.40**

different models in precise and fuzzy query tasks.

Overall Results. Experimental results are shown in Table 2. We can observe that ChartFinder and ChartFinder-B perform best in both query tasks, and multiple metrics are ahead of other methods by more than 5% or even 10%. For example, in the precise query task, ChartFinder's R@1, R@5, and R@10 are 47.18%, 80.00%, and 84.10%, respectively, significantly higher than the second-place Long-CLIP-L's 41.03%, 75.90%, and 79.49%. In the fuzzy query task, ChartFinder also surpasses Long-CLIP-L, with R@1, R@5, and R@10 being 3.05%, 8.39%, and 5.34% higher, respectively. In comparison, although Jina-CLIP-v2 (865M) has the largest number of parameters, it performs poorly in both types of tasks. The R@1 in the precise query task is only 4.10%, and the R@1 in the fuzzy query task is also only 3.05%.

6 Limitations

Dependence and Scope of Semantic Insight Generation. The insight generation process utilizes a LLM(LLaMA-3.1), meaning the quality of the output is subject to the model's inherent knowledge and potential biases. Although the study validated a small sample of insights via crowdsourcing, it cannot guarantee that all 207,498 synthesized insights are entirely free of errors or misinterpretations.

Limitations of the CRBench Benchmark Construction. The queries in CRBench were generated by GPT-4o, not from organic user query logs. While this approach was designed to simulate realistic precise and fuzzy queries and was validated through a rigorous crowdsourcing process, these synthetic queries may not fully capture the diversity, ambiguity, and unforeseen intents of genuine user behavior.

Model Architecture and Application Scenario Constraints. The core architecture of ChartFinder is based on CLIP. Although the training strategy is innovative, its fundamental performance is still bound by the inherent capabilities and limitations of the underlying vision-language models.

Generalizability of Data Sources . The datasets used for generating the training data were sourced exclusively from Kaggle. While these are high-quality and derived from real-world applications, they may not fully represent the entire spectrum of data types and structures found within private, enterprise Business Intelligence (BI) systems, which can be more domain-specific or messy.

References

Alexander Bendeck, Dennis Bromley, and Vidya Setlur. 2024. Slopeseeker: A search tool for exploring a dataset of quantifiable trends. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 817–836.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. 2023. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2818–2829. IEEE.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. 2024. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171.

Enamul Hoque and Maneesh Agrawala. 2019. Searching the visual style and structure of d3 visualizations. *IEEE transactions on visualization and computer graphics*, 26(1):1236–1245.

Daomin Ji, Hui Luo, Zhifeng Bao, and J. Shane Culpepper. 2024a. Navigating data repositories: Utilizing line charts to discover relevant datasets. *Proc. VLDB Endow.*, 17(12):4289–4292.

Daomin Ji, Hui Luo, Zhifeng Bao, and J. Shane Culpepper. 2024b. The story behind the lines: Line charts as a gateway to dataset discovery. *CoRR*, abs/2408.09506.

Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*.

Andreas Koukounas, Georgios Mastrapas, Bo Wang, Mohammad Kalim Akram, Sedigheh Eslami, Michael Günther, Isabelle Mohr, Saba Sturua, Scott Martens, Nan Wang, and 1 others. 2024. jina-clip-v2: Multilingual multimodal embeddings for text and images. *arXiv preprint arXiv:2412.08802*.

Fritz Lekschas, Brant Peterson, Daniel Haehn, Eric Ma, Nils Gehlenborg, and Hanspeter Pfister. 2020. Peax: Interactive visual pattern search in sequential data using unsupervised deep representation learning. In *Computer Graphics Forum*, volume 39, pages 167–179. Wiley Online Library.

Haotian Li, Yong Wang, Aoyu Wu, Huan Wei, and Huamin Qu. 2022a. Structure-aware visualization retrieval. In *CHI*, pages 409:1–409:14. ACM.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022b. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Zhenghao Liu, Chenyan Xiong, Yuanhuiyi Lv, Zhiyuan Liu, and Ge Yu. 2022. Universal vision-language dense retrieval: Learning a unified representation space for multi-modal retrieval. *arXiv preprint arXiv:2209.00179*.

Yuyu Luo, Xuedi Qin, Chengliang Chai, Nan Tang, Guoliang Li, and Wenbo Li. 2022a. Steerable self-driving data visualization. *IEEE Trans. Knowl. Data Eng.*, 34(1):475–490.

Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018a. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*, pages 101–112. IEEE.

Yuyu Luo, Xuedi Qin, Nan Tang, Guoliang Li, and Xinran Wang. 2018b. Deepeye: Creating good data visualizations by keyword search. In *SIGMOD Conference*, pages 1733–1736. ACM.

Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021. Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks. In *SIGMOD Conference*, pages 1235–1247. ACM.

Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2022b. Natural language to visualization by neural machine translation. *IEEE Trans. Vis. Comput. Graph.*, 28(1):217–226.

Yuyu Luo, Yihui Zhou, Nan Tang, Guoliang Li, Chengliang Chai, and Leixian Shen. 2023. Learned data-aware image representations of line charts for similarity search. *Proc. ACM Manag. Data*, 1(1):88:1–88:29.

Pingchuan Ma, Rui Ding, Shi Han, and Dongmei Zhang. 2021. Metainsight: Automatic discovery of structured knowledge for exploratory data analysis. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 1262–1274, New York, NY, USA. Association for Computing Machinery.

Miro Mannino and Azza Abouzied. 2018. Qetch: Time series querying with expressive sketches. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1741–1744.

OWID. 2011. Our world in data.

Xuedi Qin, Yuyu Luo, Nan Tang, and Guoliang Li. 2018a. Deepeye: An automatic big data visualization framework. *Big Data Min. Anal.*, 1(1):75–82.

Xuedi Qin, Yuyu Luo, Nan Tang, and Guoliang Li. 2018b. Deepeye: Visualizing your data by keyword search. In *EDBT*, pages 441–444. OpenProceedings.org.

Xuedi Qin, Yuyu Luo, Nan Tang, and Guoliang Li. 2020. Making data visualization more efficient and effective: a survey. *VLDB J.*, 29(1):93–117.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Pew Research. 2004. Pew research public.

Babak Saleh, Mira Dontcheva, Aaron Hertzmann, and Zhicheng Liu. 2015. Learning style similarity for searching infographics. *arXiv preprint arXiv:1505.01214*.

Vidya Setlur, Andriy Kanyuka, and Arjun Srinivasan. 2023. Olio: A semantic search interface for data repositories. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16.

Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2022. Towards natural language interfaces for data visualization: A survey. *IEEE transactions on visualization and computer graphics*, 29(6):3121–3144.

Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2016. Effortless data exploration with zenvisage: an expressive and interactive visual analytics system. *arXiv preprint arXiv:1604.03583*.

Arjun Srinivasan and Vidya Setlur. 2021. Snowy: Recommending utterances for conversational visual analysis. In *The 34th annual ACM symposium on user interface software and technology*, pages 864–880.

Arjun Srinivasan and Vidya Setlur. 2023. Bolt: A natural language interface for dashboard authoring. In *EuroVis (Short Papers)*, pages 7–11.

statista. 2007. Chart of the day.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and 1 others. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *arXiv preprint arXiv:2409.10173*.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint: 2303.15389*.

Tableau. 2003. Tebleau public. Website. https://public.tableau.com/app/discover.

Benny J Tang, Angie Boggust, and Arvind Satyanarayan. 2023. Vistext: A benchmark for semantically rich chart captioning. *arXiv preprint: 2307.05356*.

Jiawei Tang, Yuyu Luo, Mourad Ouzzani, Guoliang Li, and Hongyang Chen. 2022. Sevi: Speech-to-visualization through neural machine translation. In *SIGMOD Conference*, pages 2353–2356. ACM.

Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, and Aditya Parameswaran. 2017. Towards visualization recommendation systems. *Acm Sigmod Record*, 45(4):34–39.

Qitong Wang and Themis Palpanas. 2021. Deep learning embeddings for data series similarity search. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1708–1716.

Yifan Wu, Lutao Yan, Leixian Shen, Yunhai Wang, Nan Tang, and Yuyu Luo. 2024. Chartinsights: Evaluating multimodal large language models for low-level chart question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12174–12200.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2024a. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829.

Han Xiao, Georgios Mastrapas, and Bo Wang. 2024b. Jina clip: Your clip model is also your text retriever. In *Multi-modal Foundation Model meets Embodied AI Workshop@ ICML2024*.

Yilin Ye, Jianing Hao, Yihan Hou, Zhan Wang, Shishi Xiao, Yuyu Luo, and Wei Zeng. 2024. Generative AI for visualization: State of the art and future directions. *Vis. Informatics*, 8(1):43–66.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, and 1 others. 2021. Florence: A new foundation model for computer vision. *arXiv preprint: 2111.11432*.

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. 2025. Long-clip: Unlocking the long-text capability of clip. In *European Conference on Computer Vision*, pages 310–325. Springer.

Kai Zhang, Yi Luan, Hexiang Hu, Kenton Lee, Siyuan Qiao, Wenhu Chen, Yu Su, and Ming-Wei Chang. 2024. Magiclens: Self-supervised image retrieval with open-ended instructions. *arXiv preprint arXiv:2403.19651*.

Junjie Zhou, Zheng Liu, Shitao Xiao, Bo Zhao, and Yongping Xiong. 2024a. Vista: Visualized text embedding for universal multi-modal retrieval. *arXiv preprint arXiv:2406.04292*.

Tianshuo Zhou, Sen Mei, Xinze Li, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, Yu Gu, and Ge Yu. 2024b. Marvel: Unlocking the multi-modal capability of dense retrieval via visual module plugin. In

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14608–14624.

A Complete Experiments

A.1 Exp-2: ChartFinder in the Zero-Shot Setting

To evaluate the generalization ability of ChartFinder, we conducted zero-shot experiments on two external public benchmarks, VisText (Tang et al., 2023) and Chart-to-Text (Kantharaj et al., 2022). In this setting, we directly applied the model trained on our synthetic data to these new tasks without any fine-tuning to test the transferability of its learned semantic understanding ability.

A.1.1 Experiments on VisText Benchmark.

We use the test set of VisText (Tang et al., 2023), comprising 882 charts with each chart corresponding to two captions mentioned in Section 3.3, to conduct this group of experiments. We take the two-level caption as the query, and the annotated chart as the retrieval target to perform text-to-chart retrieval tasks.

Overall Results. As shown in Table 3, ChartFinder performs best in all metricss in L1 caption, significantly ahead of other models. For example, Recall@1 reaches 96.15%, far exceeding other models. Recall@10 reaches 99.66%, close to a full match. MRR@10 and NDCG@10 are 96.90% and 97.44%, respectively, further verifying their advantage in ranking quality. ChartFinder-B still achieves excellent performance with fewer parameters. Recall@1 reaches 93.20%, which is comparable to Long-CLIP-L and about 3 percentage points higher than Long-CLIP-B. On Recall@10, ChartFinder-B reaches 98.85%, surpassing all models with similar parameter scales. L2+L3 captions are more inclined to detailed analytical descriptions. Thus, the retrieval difficulty is significantly higher than the L1 task, and the overall metrics have declined. ChartFinder still leads in this task with R@1 reaching 67.80%, R@10 reaching 89.12%, MRR@10 and NDCG@10 reaching 75.20% and 78.78%, respectively.

A.1.2 Experiments on Chart-to-Text Benchmark

We use the Chart-to-Text dataset (Kantharaj et al., 2022) to conduct this experiment. Specifically, this dataset has 1,393 text and chart pairs in the test set. We use the same metrics as other experiments.

Overall Results. As shown in Table 5, ChartFinder outperforms other models in all metrics. For example, Recall@1 reaches 97.06%, 2.09% higher than the second-best model (Long-CLIP-L, 94.97%). Recall@10 reaches 99.86%, almost close to the full score, showing the excellent performance of the model in the top-k retrieval task. MRR@10 and NDCG@10 are 98.41% and 98.80%, respectively, further verifying the effectiveness of Liter in ranking quality. ChartFinder-B also demonstrates obvious performance, surpassing Long-CLIP-B in all metrics, especially in Recall@1 (94.90% vs. 90.81%) and NDCG@10 (97.53% vs. 95.35%).

Finding 1: When we use long captions as the query, the accuracy of all models is improved compared to CRBench. This also shows that in real BI scenarios, retrieval is more difficult when the query is shorter and more brief.

Although the Chart-to-Text dataset can effectively test the performance of the model in clear semantic scenarios, its task design also has certain limitations. For example, the same pictures exist in the dataset, which narrows the search scope. Most of the queries in Chart-to-Text are descriptive sentences generated based on charts. There is a lack of user intent-driven queries, which is not enough to fully evaluate the generalization and upper-limit capabilities of the model. In contrast, the queries in CRBench are more complex and closer to actual applications, which can better test the performance of the model in real retrieval scenarios.

A.2 Exp-3: Ablation Study on the Combination of Synthesized Semantic Insights

Experiment Settings. To verify the necessity and synergy of the three semantic insights we proposed, we designed a series of ablation experiments. We used ChartFinder-B as the base model and arranged and combined the three semantic insights to obtain a total of 7 training configurations (including the initial model). Table 4 shows the performance of each configuration in the two scenarios of precise query and fuzzy query, and the evaluation indicators include R@10, MRR@10, and NDCG@10.

Overall Results. The ablation study in Table 4 reveals several key insights. First, adding any single type of semantic insight significantly improves per-

Table 3: Performance Comparison of Text-to-Chart Retrieval Models on VisText Benchmark.

Models	**L1 Caption**				**L2+L3 Caption**			
	R@5	R@10	MRR@10	NDCG@10	R@5	R@10	MRR@10	NDCG@10
VISTA	87.76	89.80	79.85	82.27	63.95	70.86	37.61	41.80
Univl-DR	81.75	86.62	71.57	75.22	52.83	60.77	42.61	46.93
MARVEL	66.55	72.79	55.61	59.73	39.80	48.64	31.25	35.37
CLIP	89.46	92.74	83.05	85.41	68.25	74.83	56.81	61.14
CLIP-DPR	88.10	90.93	80.59	83.14	64.74	72.11	53.51	57.97
EVA-CLIP	87.07	89.68	78.50	81.26	49.43	57.26	39.39	43.64
CoCa	84.35	87.87	79.69	81.65	58.96	66.44	50.54	54.30
Jina-CLIP-v2	90.93	94.67	84.85	87.23	74.49	80.50	62.72	67.01
Long-CLIP-B	94.78	95.35	81.88	83.50	78.12	83.45	67.99	71.73
ChartFinder-B	97.85	98.75	95.17	96.05	79.48	84.01	69.64	73.11
Long-CLIP-L	98.07	99.09	95.21	96.15	82.99	88.66	73.43	77.09
ChartFinder	**99.43**	**99.66**	**96.90**	**97.44**	**84.58**	**89.12**	**75.20**	**78.78**

Table 4: The Effectiveness of ChartFinder on CRBench by Incorporating Different Semantic Insights.

Visual-o	Statistics-o	Task-o	Precise Query			Fuzzy Query			Overall
			R@10	MRR@10	NDCG@10	R@10	MRR@10	NDCG@10	
			62.56	37.99	43.90	51.91	30.29	35.33	43.66
✓			69.64	44.39	48.69	56.28	27.27	34.29	46.67
	✓		66.15	45.18	50.18	57.25	32.75	38.60	48.35
		✓	65.64	44.44	49.52	58.78	32.24	38.59	48.20
✓		✓	**71.79**	44.96	51.40	60.31	35.24	41.17	50.81
✓	✓		66.15	43.26	48.74	54.96	32.20	37.66	47.16
	✓	✓	65.64	43.53	48.87	54.96	30.44	36.23	46.61
✓	✓	✓	70.26	**49.30**	**61.30**	**61.83**	**38.96**	**44.41**	**54.34**

Table 5: Experimental Results on Chart-to-Text Benchmark.

Models	**R@5**	**R@10**	**MRR@10**	**NDCG@10**
VISTA	68.46	73.20	58.94	62.39
Univl-DR	67.91	73.08	57.60	61.35
MARVEL	38.19	46.30	28.97	33.08
CLIP	88.16	92.03	80.25	83.11
CLIP-DPR	82.56	85.86	73.09	76.21
EVA-CLIP	83.42	87.80	74.73	77.90
CoCa	83.92	87.65	74.49	77.92
Jina-CLIP-v2	83.78	88.23	73.51	77.08
Long-CLIP-B	98.28	98.85	94.18	95.35
ChartFinder-B	99.07	99.43	96.89	97.53
Long-CLIP-L	99.35	99.78	97.06	97.75
ChartFinder	**99.86**	**99.86**	**98.41**	**98.80**

formance over the baseline, with statistics-oriented insights providing the largest individual boost (from 43.66 to 48.35). Second, combining two types of insights generally yields further improvement, with the visual-task combination being the most effective pair (50.81). However, the greatest performance leap occurs when all three insights are used together, achieving the top overall score of 54.34.

The experiments demonstrate that visual-oriented insights primarily assist in capturing general visual patterns, while statistics-oriented insights enhance the model's analytical sensitivity to underlying data trends. Task-oriented insights further contextualize these representations, significantly improving retrieval precision in complex scenarios. Therefore, the optimal performance achieved by combining all three insights highlights their complementary nature and synergistic effects.

This suggests that while each insight is valuable, they form a complete information processing chain from visual perception (visual-oriented) to data-driven analysis (statistics-oriented) and finally to practical application (task-oriented). The full combination provides a holistic semantic context that allows the model to learn the most effective representations, demonstrating a powerful synergistic effect that surpasses any partial combination.

In summary, the experimental results strongly support our hypothesis: efficient chart-to-text retrieval models need to comprehensively utilize multi-dimensional semantic insights rather than simply superimposing features.

Finding 2: The three semantic insights are not merely additive but synergistic, with the complete set being crucial for optimal performance.

Table 6: Experiment results on Text-to-chart retrieval task.

Models	Precise Query					Fuzzy Query				
	R@1	R@5	R@10	MRR@10	NDCG@10	R@1	R@5	R@10	MRR@10	NDCG@10
CLIP-DPR	4.10	15.38	20.00	8.90	11.55	5.34	7.63	10.69	6.60	7.54
FT-CLIP-DPR	8.21	19.49	24.62	13.08	15.80	7.63	14.50	19.08	10.97	12.88
	↑4.11	↑4.11	↑4.62	↑4.18	↑4.25	↑2.29	↑6.87	↑8.39	↑4.37	↑5.34
UniVL-DR	3.08	11.79	17.44	6.59	9.13	5.34	6.87	9.92	6.26	7.11
FT-UniVL-DR	8.21	17.95	26.15	13.25	16.29	12.21	18.32	20.61	15.08	16.42
	↑5.13	↑6.16	↑8.71	↑6.66	↑7.16	↑6.87	↑11.45	↑10.69	↑8.82	↑9.31
Long-CLIP-B	26.67	53.85	62.56	37.99	43.90	22.90	40.46	51.91	30.29	35.33
ChartFinder-B	38.97	62.05	70.26	49.30	61.30	29.77	50.38	61.83	38.96	44.41
	↑12.30	↑8.20	↑7.70	↑11.31	↑17.40	↑6.87	↑9.92	↑9.92	↑8.67	↑9.08
Long-CLIP-L	41.03	75.90	79.49	55.32	55.32	38.93	67.18	74.81	50.99	56.77
ChartFinder	47.18	80.00	84.10	61.23	66.90	41.98	75.57	80.15	55.31	61.40
	↑6.15	↑4.10	↑4.61	↑5.91	↑11.58	↑3.05	↑8.39	↑5.34	↑4.32	↑4.63

A.3 Exp-4: The Generalizability of Synthesized Semantic Insights

Experiment Settings. To verify the versatility of the three designed semantic insights, we apply them to train different types of CLIP-based models to observe their impact on model performance. In this experiment, we select CLIP-DPR and UniVL-DR as base models and compare them with fine-tuned versions (FT-CLIP-DPR and FT-UniVL-DR) that incorporate semantic insights. All models are evaluated on the CRBench dataset, using two test scenarios: exact query and fuzzy query. Evaluation metrics include R@10 and MRR@10.

Overall Results. As shown in Table 6, the introduction of semantic insights has brought significant performance improvements to different models. For the CLIP-DPR, in the precise query scenario, R@10 and MRR@10 increased by 4.62 and 4.18 percentage points, respectively; in the fuzzy query scenario, the improvement was more significant, with R@10 increasing by 8.39 percentage points. The UniVL-DR achieved greater performance gains after the introduction of semantic insights, especially in the fuzzy query scenario, with R@10 increasing by 10.69 percentage points and MRR@10 increasing by 8.82 percentage points. This result fully proves that the semantic insights we designed have good versatility and can effectively enhance the performance of retrieval models with different architectures, especially showing stronger robustness when processing fuzzy queries. This further verifies the universal value of multi-dimensional semantic insights for chart retrieval tasks.

Finding 4: We found that fine-tuning other models using semantic insights can also significantly improve their retrieval capabilities, proving the generalizability of the three semantic insights.

A.4 Exp-5: The Impact of Center Crop Strategy

Experiment Settings. We mentioned in Section 5 Training Details that for the text-to-chart retrieval task, we replaced the traditional center crop preprocessing method in the CLIP-based models with a direct resize strategy. To verify the necessity of this modification, we selected Long-CLIP-B and Long-CLIP-L as the base models and conducted comparative tests on the urban1k and Chart-to-Text datasets, respectively. Urban1k (Zhang et al., 2025) is a retrieval dataset containing natural images of urban scenes, which contains a variety of urban buildings, streets, and landscapes, representing a typical natural image retrieval task.

Overall Results. The results in Table 7 show that the preprocessing strategies have very different effects on different retrieval tasks. On the Urban1k dataset, direct resize caused a slight decrease in model performance, with Long-CLIP-B and Long-CLIP-L decreasing by 0.9 and 1.5 percentage points on average, respectively. However, on the Chart-to-Text dataset, the same strategy brought significant improvements, with the two models gaining approximately 12.3 and 13.1 percentage points in performance, respectively. This shows that in chart retrieval tasks, retaining the complete structure of the chart (including edge axis labels, legends, etc.) is more important than maintaining precise proportions, proving that our adjust-

Table 7: Ablation Study on Center Crop. *W/* direct resize means we directly resize the image to the required size instead of cropping the chart to the target size.

Models	R@10	MRR@5	MRR@10	NDCG@10
		Urban1k		
Long-CLIP-B	94.1	97.4	85.8	88.6
w/ direct resize	93.2	96.1	85.1	87.8
	↓0.9	↓1.3	↓0.7	↓0.8
Long-CLIP-L	96.7	98.2	90.7	92.6
w/ direct resize	95.1	97.5	88.9	90.9
	↓1.6	↓0.7	↓1.8	↓1.5
		Chart-to-Text		
Long-CLIP-B	87.29	90.38	79.87	82.44
w/ direct resize	98.28	98.85	94.18	95.35
	↑10.99	↑8.47	↑14.31	↑12.91
Long-CLIP-L	88.51	91.03	81.20	83.61
w/ direct resize	99.35	99.78	97.06	97.75
	↑10.84	↑8.75	↑15.86	↑14.14

ments to chart-specific preprocessing strategies are necessary and effective.

Finding 5: In natural images, a lot of important semantic information is often in the center of the image, so the center crop processing allows the model to pay more attention to the query target. However, in the chart, important information such as the title, x-axis, and y-axis names are all around the chart, which makes it easy for the center crop to cut out important information.

B Case Study

Figure 5 compares the top-1 retrieval performance of our model and CLIP on CRBench. In Query #1, ChartFinder successfully retrieves a chart that includes both *urban* and *rural population*, while CLIP retrieves a chart focusing only on *urban population*, failing to meet the query requirements. In Query #2, ChartFinder correctly aligns with the semantic meaning of *booking distribution*, retrieving a pie chart about hotel booking channels. In contrast, CLIP retrieves a chart about optimizing distribution channels, which does not match the query's focus. In Query #3, ChartFinder captures the keyword *change* and retrieves a chart showing trends in alcohol consumption. In contrast, CLIP retrieves a chart ranking alcohol consumption, ignoring the notion of *change*. In Query #4, ChartFinder identifies *distribution* and retrieves a stacked area chart illustrating income distribution. In contrast, CLIP retrieves a chart focusing on income growth trends, failing to address the *distribution* aspect.

(i) For precise query #1, ChartFinder successfully retrieves the charts that accurately match the query by capturing the keywords and filter conditions *urban and rural population* correctly. However, the chart CLIP retrieved only reflects the urban population, failing to meet the retrieval requirements of the rural population.

(ii) For precise query #2, our model aligns the semantic information about the hotel booking channels described by the pie chart with the *booking distribution* in the query. In contrast, although the retrieved chart of the CLIP is also about hotel information, the description is not about the different ways to book hotels, but the optimization of hotel distribution channels.

(iii) For fuzzy query #3, our model successfully captures the keyword *change* and retrieves a chart that clearly shows the trend in alcohol consumption. By contrast, although the chart retrieved by CLIP is also related to alcohol consumption, its content only focuses on the absolute ranking of alcohol consumption ("Who's the Booziest"), completely ignoring the core element of *change*.

(iv) For fuzzy query #4, *distribution* in red refers directly to the concept of data distribution. Our model successfully retrieves a stack area chart showing the distribution of income levels. However, the chart retrieved by CLIP is biased toward the trend of household income growth, although this graph is related to income and does not reflect the analytical information on income level distribution.

Through the four cases above, we can see the effectiveness of our model when matching both precise queries and fuzzy queries with their target charts.

C Prompt

Yaml config for running Qwen2.5-Math-7B on LiteCoT

```
You are a user searching for visualizations in a
↪  database. You will see 5 visualizations, but you
↪  should ONLY focus on the FIRST image when
↪  generating queries.
The other four images are very similar to the first
↪  one, make sure your queries CAN NOT match any of
↪  the other four images.
Generate two meaningful queries (10-15 words each):

\textbf{\textit{1. Precise Query}}:
Generate a query about specific content in the FIRST
↪  visualization that includes:

- Meaningful combination of axis labels/categories

- Important data values or time ranges

- Key categories or measurements
```

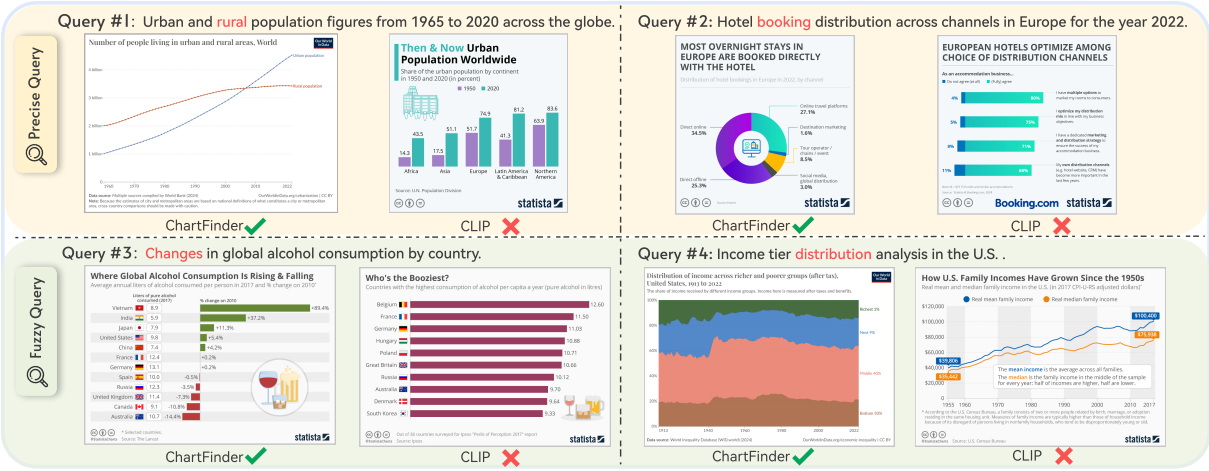


Figure 5: Case study on Top-1 Text-to-Chart Retrieval Results of the ChartFinder and CLIP.

```
- If the image contains time, you should include the
↪  time range in the query because it is important
↪  for the user to search for the specific time
↪  range and distinguish it from other similar
↪  visualizations.

- When covering the time range, do not directly use
↪  the initial time range in the chart. For example,
↪  if the initial time range in the x-axis is
↪  2012-2019, you can use 2014 - 2018 or 2013 - 2016.
↪  Make sure the time range in your query is
↪  included in the initial time range.

- You don't need to cover a very specific time range,
↪  like specific months or days, unless you need this
↪  to distinguish from other similar visualizations.

- Do not use a comma in your query.

- Do not directly use the initial text in the image
↪  as the query; always find similar words or
↪  phrases to describe the content of the image. For
↪  example, if the title contains "weekly", you
↪  should use every 7 days in the query.

\textbf{Example: ``global temperature change
↪  1990-2020 in the United States'' (good, time and
↪  location are included)}

\textbf{Bad example: ``global temperature change''
↪  (bad, no time and location to distinguish from
↪  other similar visualizations).}

\textbf{\textit{2. Fuzzy Query}}:
Generate a query about the visualization purpose that
↪  includes:

- Line charts: trend analysis or comparison over time

- Bar charts: value comparison or ranking purposes

- Pie/stacked charts: distribution analysis or
↪  proportion comparison

- Scatter plots: correlation or pattern analysis

\textbf{Example: ``annual economic growth comparison
↪  between countries'' (enough information to
↪  distinguish from other similar visualizations)}

\textbf{Bad example: ``growth comparison''(too little
↪  information)}

Format your response as:\\
\{\
    ``Precise query'': ``10-15 word query about
    ↪  content'',
```

```
    ``Fuzzy query'': ``10-15 word query about
    ↪  purpose''
\}\

Remember:

- ONLY consider the FIRST image

- Include relevant context

- Be specific and meaningful

- Articles (a, an, the) and prepositions count as
↪  words
```