G-Cap: A Game Character Caption Generator

Anonymous ACL submission

Abstract

While Large Vision-Language Models (LVLMs) have demonstrated remarkable proficiency in image captioning, existing research primarily focuses on real-world scenarios, leaving surreal, highly stylized, and semantically hybrid virtual-world scenarios significantly underexplored. In this work, we introduce **Game Character Captioning**, a novel task designed to evaluate LVLMs' capability to perceive and describe game character from the virtual-world. To facilitate evaluation, we establish **GC-Bench**, a manually annotated benchmark, and propose **Graph-F1** to effectively assess performance on this task. Our evaluation reveals that: (1) current state-of-the-art LVLMs, including closed-source giants such as `Gemini 3 Pro` and `GPT-5.1`, struggle to maintain the high performance seen in real-world scenarios; and (2) a notable gap exists between open-source and closed-source models. To bridge this gap, we construct **GC-148K**, a large-scale dataset generated via a specialized data pipeline, and develop the **G-Cap** series. Experiments demonstrate that G-Cap series rivals the performance of advanced closed-source models at a lower cost, offering an efficient solution for industrial-grade production environment.

1 Introduction

Image captioning is a fundamental and critical task that demonstrates the image understanding capabilities of Large Vision-Language Models (LVLMs). It has received extensive attention from researchers over the past few years (Rohrbach et al., 2018; Dong et al., 2024; Cheng et al., 2025). Researchers have increasingly higher expectations for the capabilities of LVLMs, specifically hoping that LVLMs can generate descriptions that are more detailed and exhibit fewer hallucinations (Hua et al., 2025; Xing et al., 2025; Li et al., 2025).

However, existing image captioning research often focuses on scenarios involving images of

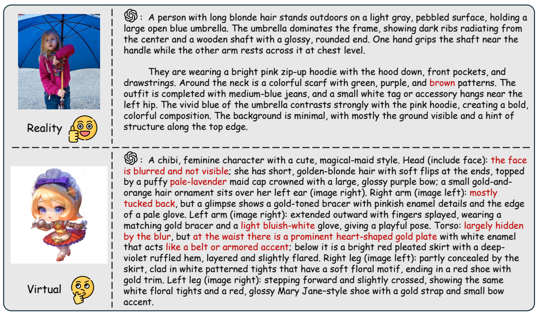


Figure 1: GPT-5.1's caption performance in real-world and virtual-world scenarios.

predominantly natural, daily real-world scenarios, while research on surreal, highly stylized, and semantically hybrid virtual-world, such as those in games and anime, remains relatively scarce. In the production environment of game companies, generating captions for game or anime characters holds significant business value. By constructing corresponding game image-text pairs, this capability can be applied to a wide range of downstream tasks, such as game content retrieval, game skin recommendation, game skin design, and game content review. Through a case study (as shown in Figure 1), we observed that the current state-of-the-art (SOTA) LVLMs, such as GPT-5.1, performs well in real-world scenarios image captioning; however, its performance on the virtual-world scenarios fails to replicate its success in real-world scenarios.

Motivated by these observations, this paper aims to advance LVLM-based image captioning in virtual-world scenarios by addressing two key questions.

(1) How can we effectively evaluate the performance of existing LVLMs on image captioning for virtual-world scenarios? To address this gap, we define the novel task of **Game Character Captioning** and construct **GC-Bench**, the first benchmark dedicated to this domain. Based on

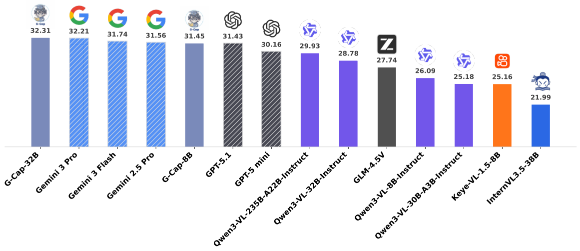


Figure 2: Model rankings on GC-Bench.

the evaluation results derived from this benchmark, we identify the second critical question: **(2) How can we bridge the performance gap between current open-source and closed-source LVLMs to achieve industrial-grade usability?**

To address the first question, we develop **GC-Bench**, the first benchmark specifically designed for the Game Character Captioning task, comprising 186 diverse game character images accompanied by meticulously human-annotated captions. Unlike general image captioning, Game Character Captioning demands a deeper understanding of **hierarchical clothing relationships**, perception of **clothing colors and material**, and a reservoir of **clothing knowledge**. Given the high granularity of dense image captioning, traditional NLG metrics or standard VLM-as-a-Judge paradigms are insufficient for accurate assessment. Therefore, we propose **Graph-F1**, a subgraph matching-based evaluation metric that effectively evaluates precision and recall in generated captions. Figure 2 illustrates the model rankings on GC-Bench, revealing a significant performance disparity between current open-source LVLMs and top-tier closed-source LVLMs.

To address the second question, we construct a novel data synthesis pipeline. Specifically, We first collected 78K raw images and utilized `Gemini 2.5 Pro` (Comanici et al., 2025) to generate initial captions, from which we extract fine-grained garment components across five major body regions (head, torso, arms, legs, and body accessories) by `Qwen3-235B-A22B-FP8` (Yang et al., 2025). Leveraging these garment components, we build a large-scale **Fashion Simulator**. This simulator supports the stochastic recombination of garment components; we employ `Qwen3-235B-A22B-FP8` to convert these combinations into new captions, which are subsequently rendered into new images via `Gemini 2.5 Flash Image (Nano Banana)`. By consolidating these synthetic pairs (∼70K) with the original data, we establish the **GC-148K** dataset. Based on this dataset, we fine-tune Qwen3-VL-8B-Instruct and Qwen3-VL-32B-Instruct to obtain **G-Cap-8B** and **G-Cap-32B**. Experimental results demonstrate that G-Cap series achieve up to an 20.5% performance gain over their base models. Notably, **G-Cap-32B** achieves performance parity with `Gemini 3 Pro` on GC-Bench, validating its usability in industrial-grade production environment.

2 Related Work

2.1 Large Vision-Language Model and Image Captioning

Recent years have witnessed rapid progress in LVLMs, which have achieved SOTA performance across various multimodal benchmarks. Among open-source models, the LLaVA and Qwen-VL families represent the dominant architectures (Liu et al., 2023, 2024; Li et al., 2024; Bai et al., 2023; Wang et al., 2024; Bai et al., 2025b). These models align visual features extracted by a vision encoder with the hidden space of large language models via a projection module, enabling effective multimodal reasoning after large-scale multi-stage training.

Image captioning serves as a core task for evaluating visual understanding in LVLMs and has evolved from generating short descriptions to producing dense, fine-grained captions with reduced hallucination (Rohrbach et al., 2018; Dong et al., 2024; Cheng et al., 2025; Hua et al., 2025; Xing et al., 2025; Li et al., 2025). However, existing studies predominantly focus on real-world imagery, while captioning in virtual-world scenarios remains underexplored. To address this gap, we introduce *Game Character Captioning*, a captioning task tailored to virtual-world scenarios, and establish a benchmark to systematically evaluate both open- and closed-source LVLMs.

2.2 Evaluation Metrics for Image Captioning

Due to the open-ended nature of image captioning, establishing effective evaluation methods has always been a research priority. Existing metrics generally fall into four categories. The first three include NLG-based metrics (e.g., BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015)) (Karpathy and Fei-Fei, 2015; Onoe et al., 2024), embedding-based semantic matching (e.g., BERTScore (Zhang et al., 2019), CLIPScore (Hessel et al., 2021)), and the VLM-as-a-judge paradigm, covering both reference-based (Ge et al., 2024; Lee et al., 2024) and reference-free (Bitton et al., 2023; Lee et al., 2024) evaluations. The fourth category relies on scene graphs, which is the type most relevant to this paper.

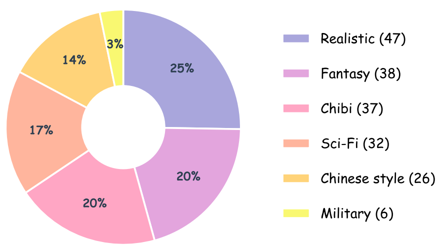
For scene graph-based metrics, SPICE (Anderson et al., 2016) pioneered the use of parsed triplets and WordNet (Miller, 1995) for semantic matching. However, its reliance on rigid whole-triplet matching limits granularity and robustness. To address this, CAPTURE (Dong et al., 2024) decomposes evaluation into objects, attributes, and relations, introducing a multi-stage matching strategy (hard, synonym, and soft matching (Devlin et al., 2019)). Yet, CAPTURE matches attributes globally without binding them to specific objects, potentially leading to mismatches between attributes of different objects. Consequently, Zhang et al. (2025a) improved this by hierarchically matching objects first and then calculating attribute similarity conditioned on the matched objects. However, their method isolates relation evaluation by converting it into a separate QA task, thereby neglecting the structural context of the subgraph. In contrast, the Graph-F1 proposed in this paper unifies relations and attributes. By matching parent nodes via cosine similarity and assessing the corresponding subgraph relevance using a reranker model, Graph-F1 captures both the macroscopic local context and fine-grained details.

3 Method

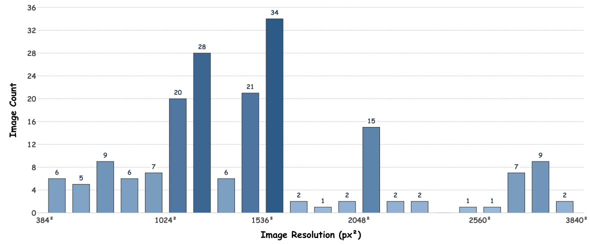
3.1 Problem Formulation

We introduce the Game Character Captioning task, which specializes in the domain of virtual-world scenarios. Unlike traditional image captioning that focuses on real-world scenarios, Game Character Captioning aims to generate fine-grained descriptions for game characters from virtual-world. Formally, given a image $I_v \in \mathbb{R}^{H \times W \times 3}$ containing a game character, the objective is to generate a caption C that accurately captures the character's visual attributes (e.g., appearance, clothing texture and color) and stylistic features.

We formulate Game Character Captioning as a conditional text generation task leveraging a LVLM. Let $\mathcal{M}$ denote the LVLM. The model takes the image I_v and a task-specific prompt P (see Appendix C for details) as inputs. The generation process can be formally expressed as:



(a) Major style distribution in GC-Bench.



(b) Image resolution distribution in GC-Bench.

Figure 3: Dataset statistics of GC-Bench.

$$\hat{C} = \mathcal{M}(I_v, P) \quad (1)$$

where $\hat{C} = \{y_1, y_2, \ldots, y_L\}$ represents the generated sequence of caption tokens describing the character in the image I_v.

3.2 GC-Bench

To evaluate the performance of different LVLMs on Game Character Captioning, we developed GC-Bench. This benchmark comprises 186 meticulously manually annotated data samples. By utilizing our proposed Graph-F1, we can effectively evaluate the performance of various LVLMs.

3.2.1 Data Collection and Annotation Pipeline

Data Collection. To ensure a sufficiently broad distribution of visual semantics, we curated an image source covering a diverse array of gaming titles. These include 20 distinct games. For specific game sources, please refer to Appendix A. Taking into account style diversity and image resolution, we ultimately selected 186 representative images. The distributions of major artistic styles and image resolutions are illustrated in Figure 3a and 3b, respectively.

Annotation Pipeline. Unlike standard captioning datasets, we adopt a generation-based verification pipeline to produce structure-rich, high-precision annotations (Figure 4). The pipeline consists of four steps:

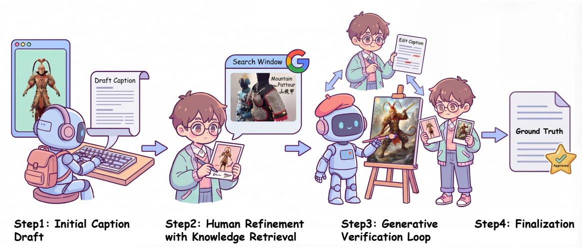


Figure 4: The annotation pipeline of GC-Bench.

1. **Initial Captioning:** Initial captions are generated by GPT-5 with task-specific prompt P.

2. **Human Refinement with Knowledge Retrieval:** Annotators revise the captions to improve accuracy and completeness. When necessary, annotators consult Google Image Search as an external reference to identify precise terminology for accessories, garment components, and character-specific details.

3. **Generative Verification Loop:** The refined captions are used to reconstruct images with `Nano Banana`. Annotators compare the generated images with the original inputs to identify missing details or inconsistencies, and iteratively revise the captions (3–5 rounds per image).

4. **Finalization:** After visual consistency is achieved, global attributes such as camera pose and artistic style are appended to form the final annotations.

This pipeline yields high-quality captions with an average length of approximately 290 words. Qualitative examples are shown in Appendix O.

3.2.2 Evaluation Metrics

Accurate evaluation of Game Character Captioning requires precise assessment of fine-grained attributes and hierarchical relations. However, existing scene graph metrics fail to capture holistic structural semantics. SPICE (Anderson et al., 2016) relies on rigid matching and lacks robustness, while CAPTURE (Dong et al., 2024) suffers from attribute–object misalignment. Although Zhang et al. (2025a) improves attribute binding, isolating relations through QA-style evaluation overlooks the integrity of local subgraph context.

To address these limitations, we propose **Graph-F1**, a metric that evaluates attributes and relations jointly within coherent local subgraphs. As shown in Figure 5, Graph-F1 adopts a two-stage soft matching strategy: parent nodes are first aligned via cosine similarity, followed by semantic consistency evaluation over their associated subgraphs.

The detailed algorithm is described as follows:

Graph Construction. We transform the generated caption C_{gen} and the ground-truth caption C_{gt} into semantic graphs to explicitly represent their structured content. Using `GPT-5` as a structured parser, we extract a set of triplets $\mathcal{T} = \{(h, r, t)\}$, where h denotes a head entity, r a semantic relation, and t either an object entity or an attribute value (see Appendix D for details). These triplets naturally define a directed graph, with h and t as nodes and r as a directed edge. Accordingly, we obtain the corresponding semantic graphs for the C_{gen} and the C_{gt}, denoted as $G_{\text{gen}} = (\mathcal{V}_{\text{gen}}, \mathcal{E}_{\text{gen}})$ and $G_{\text{gt}} = (\mathcal{V}_{\text{gt}}, \mathcal{E}_{\text{gt}})$, where $\mathcal{V}$ is the set of unique entities and attribute values, and $\mathcal{E}$ contains all directed relations derived from $\mathcal{T}$.

Parent Nodes Matching. As the core of our metric involves subgraph matching, we focus matching specifically on **parent nodes** (i.e., entities with outgoing edges), which serve as anchors for subsequent subgraph comparison. Let $\mathcal{V}^p_{gen} \subseteq \mathcal{V}_{gen}$ and $\mathcal{V}^p_{gt} \subseteq \mathcal{V}_{gt}$ denote the subsets of parent nodes in the generated and ground truth graphs, respectively. We employ the `Qwen3-Embedding-0.6B` (Zhang et al., 2025b) to map each parent node v to a high-dimensional vector $\mathbf{h}_v$. We then compute a similarity matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}^p_{gen}| \times |\mathcal{V}^p_{gt}|}$ based on cosine similarity:

$$\mathbf{S}_{ij} = \frac{\mathbf{h}_{u_i} \cdot \mathbf{h}_{v_j}}{\|\mathbf{h}_{u_i}\| \|\mathbf{h}_{v_j}\|}, \quad u_i \in \mathcal{V}^p_{gen}, v_j \in \mathcal{V}^p_{gt} \quad (2)$$

This matrix identifies the best-matched parent entities across the two graphs, establishing the foundation for comparing their associated subgraphs.

Subgraph Comparison. To assess the fine-grained semantic quality, we compare the local information centered around each matched parent node. Specifically, for a parent node h, we extract its **one-hop subgraph** and generate a textual description by **concatenating the list of all triplets** within this subgraph:

$$\text{Desc}(h) = \bigoplus_{(h,r,t) \in \mathcal{T}_h} \text{“}h\ r\ t.\text{”} \quad (3)$$

We then utilize the `Qwen3-Reranker-4B` (Zhang et al., 2025b) to compute a semantic relevance

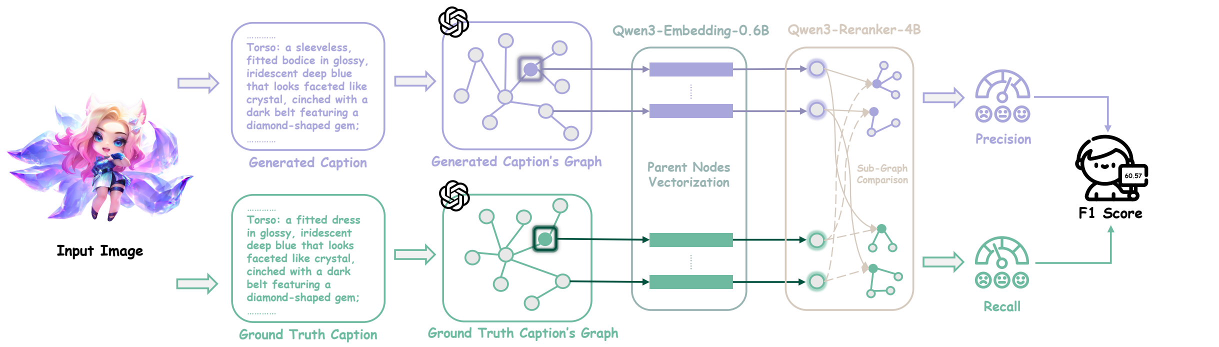


Figure 5: The pipeline of our proposed Graph-F1 metric. It transforms captions into semantic graphs and computes asymmetric precision and recall via subgraph matching.

score between these descriptions, selected for its optimal balance between performance and inference cost. We apply an **asymmetric verification strategy**:

- **Precision (Hallucination Check):** For each generated parent node $u_i \in \mathcal{V}^p_{gen}$, we retrieve its best-matched ground truth anchor $v_{j^*} \in \mathcal{V}^p_{gt}$ (based on equation 2) . We treat the generated subgraph description $\text{Desc}(u_i, G_{gen})$ as the **query** and the matched ground truth $\text{Desc}(v_{j^*}, G_{gt})$ as the **document**. The reranker $\mathcal{R}$ validates whether the generated local details are *factually supported* by the ground truth, using a task-specific prompt $\mathcal{P}_{prec}$ (see Appendix E):

$$\text{R}^{prec}_i = \mathcal{R}(\mathcal{P}_{prec}; \text{Desc}(u_i), \text{Desc}(v_{j^*})) \quad (4)$$

- **Recall (Coverage Check):** Conversely, for each ground truth parent node $v_j \in \mathcal{V}^p_{gt}$, we retrieve its best-matched generated anchor $u_{i^*} \in \mathcal{V}^p_{gen}$. Here, the ground truth description $\text{Desc}(v_j, G_{gt})$ serves as the **query** and the generated $\text{Desc}(u_{i^*}, G_{gen})$ as the **document**. The reranker $\mathcal{R}$ assesses whether the ground truth's local information is *sufficiently covered* by the generation, using a task-specific prompt $\mathcal{P}_{rec}$ (see Appendix F):

$$\text{R}^{rec}_j = \mathcal{R}(\mathcal{P}_{rec}; \text{Desc}(v_j), \text{Desc}(u_{i^*})) \quad (5)$$

Scoring. Based on the Parent Nodes Matching and Subgraph Comparison, we calculate the final scores. **Precision** is computed by matching each generated node u_i to its most similar ground truth node v_{j^*} (where $j^* = \text{argmax}_j \, \mathbf{S}_{ij}$):

$$\text{Precision} = \frac{1}{|\mathcal{V}^p_{gen}|} \sum_{i=1}^{|\mathcal{V}^p_{gen}|} \left(\mathbf{S}_{ij^*} \cdot \text{R}^{prec}_i \right) \quad (6)$$

Similarly, **Recall** is computed by matching each ground truth node v_j to its best generated match u_{i^*} (where $i^* = \text{argmax}_i \, \mathbf{S}_{ij}$):

$$\text{Recall} = \frac{1}{|\mathcal{V}^p_{gt}|} \sum_{j=1}^{|\mathcal{V}^p_{gt}|} \left(\mathbf{S}_{i^*j} \cdot \text{R}^{rec}_j \right) \quad (7)$$

Finally, the **Graph-F1** score is derived as the harmonic mean of Precision and Recall:

$$\text{Graph-F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

3.3 GC-148K

To bridge the performance gap between open-source and closed-source models in game character captioning, we introduce **GC-148K**. GC-148K comprised two distinct subsets: the real-world based **GC-Real** and the synthetically generated **GC-Syn**. The pipeline is shown in the Figure 6.

GC-Real: High-Quality Real-World Data. GC-Real forms the foundation of our dataset. We start from approximately 100K web-crawled images and apply a rigorous deduplication and anti-leakage pipeline based on DINOv3 (Siméoni et al., 2025). Using the `dinov3-vitl16-pretrain-lvd1689m` as the encoder, we extract global visual representations $f(x)$ for all images.

Specifically, we first perform *anti-leakage* by computing the maximum cosine similarity between each candidate image and the GC-Bench benchmark set $\mathcal{D}_{GC\text{-}Bench}$, discarding images with $s_{leak}(x) > 0.87$. We then conduct *internal deduplication* via a greedy strategy, removing redundant samples whose pairwise similarity exceeds 0.94. Both thresholds are determined based on empirical sampling studies, where we manually inspected

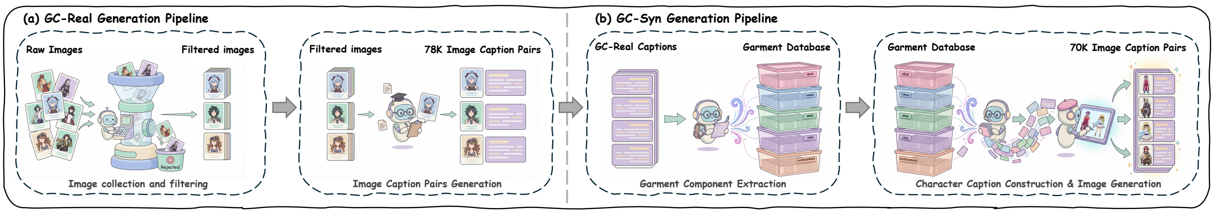


Figure 6: The overall pipeline of GC-148K.

image pairs across different similarity ranges to balance strict leakage prevention with data retention.

After filtering, we retain **78K** unique images. Dense and detailed captions are generated using `Gemini 2.5 Pro`, resulting in the final **GC-Real** subset with approximately 78K high-quality image–caption pairs.

GC-Syn: Synthesizing Data via Fashion Simulator. Beyond enhancing dataset diversity, a primary motivation for constructing the second subset, **GC-Syn**, is to address the limitation of current LVLMs in accurately describing complex, multi-layered clothing structures. By leveraging the *Fashion Simulator* alongside the robust instruction-following capabilities of the image generation model `Nano Banana`, we generate high-quality image-caption pairs that specifically feature intricate clothing hierarchies. The pipeline follows a "Deconstruct-Recombine-Generate" paradigm:

1. **Garments Extraction:** We utilized `Qwen3-235B-A22B-FP8` to parse the 78k captions from GC-Real into garment component (for the prompt used to integrate the garment, please refer to the Appendix H). We defined a hierarchy with five major categories: *Head*, *Torso*, *Arms*, *Legs*, and *Body Accessories*. This extraction resulted in a massive garment database containing 550k instance. For specific details, please refer to the Appendix J.

2. **Stochastic Recombination & Refinement:** Leveraging this decoupled attribute pool, we implemented a stochastic strategy to sample compatible components across categories (sampling rules can be found in the appendix L). We then employed `Qwen3-235B-A22B-FP8` to assemble these sampled parts into complete, coherent captions (for the prompt used to integrate the garment, please refer to the Appendix G). Crucially, to ensure semantic consistency, we introduced a refinement step where `Qwen3` reviews the generated captions to resolve any logical conflicts between attributes or structural errors that may have occurred during the initial assembly.

3. **Generative Synthesis:** The refined, structure-rich captions were then fed into `Nano Banana` to synthesize corresponding high-fidelity images. This process ensures the visual output strictly adheres to the complex text descriptions, particularly regarding clothing layers.

4. **Generative Synthesis:** The refined, structure-rich captions are fed into `Nano Banana` to synthesize high-fidelity images. To promote visual diversity, we curate an art medium vocabulary consisting of 103 art mediums (see Appendix K), from which one medium is randomly sampled and appended to each caption during generation. In addition, a binary gender attribute (*male* or *female*) is randomly sampled and explicitly specified in the caption. This design encourages stylistic and demographic variation while preserving strict adherence to complex structural descriptions, particularly for fine-grained clothing layering.

Through this fashion simulation, we produced an additional **70k** samples. In summary, the complete **GC-148K** dataset combines the realism of GC-Real with the diverse, structurally complex variations of GC-Syn, providing a robust foundation for training.

4 Experiments

4.1 Experiments Setting

We adopt Qwen3-VL-8B-Instruct and Qwen3-VL-32B-Instruct (Bai et al., 2025a) as the backbone model and perform supervised fine-tuning (SFT) on the GC-148K dataset. Training is conducted with a batch size of 128 and a learning rate of $1e{-}5$, with a warmup ratio of 0.05. We fine-tune all model

Model	Avg Precision	Avg Recall	Avg Graph-F1	Avg Length
		Closed-source Models		
GPT-5 mini	41.11	24.55	30.16	274
GPT-5.1	40.54	**26.64**	31.43	514
Gemini 2.5 Pro	43.53	25.38	31.56	209
Gemini 3 Flash	<u>44.46</u>	25.27	31.74	211
Gemini 3 Pro	**44.71**	25.78	<u>32.21</u>	208
		Open-source Models		
InternVL3.5-38B	37.68	16.15	21.99	172
Keye-VL-1.5-8B	41.05	18.84	25.16	289
Qwen3-VL-30B-A3B-Instruct	38.84	19.17	25.18	201
Qwen3-VL-8B-Instruct	42.28	19.43	26.09	139
GLM-4.5V-106B-A12B	42.89	21.18	27.74	177
Qwen3-VL-32B-Instruct	39.65	23.32	28.78	326
Qwen3-VL-235B-A22B-Instruct	42.48	23.76	29.93	219
		Ours		
G-Cap-8B	43.13	25.33	31.45	227
Δ *vs. Qwen3-VL-8B-Instruct*	+2.0%	+30.4%	+20.5%	+63.3%
G-Cap-32B	43.82	<u>26.16</u>	**32.31**	224
Δ *vs. Qwen3-VL-32B-Instruct*	+10.5%	+12.2%	+12.3%	-31.3%

Table 1: Performance comparison on the GC-Bench. We highlight the performance gains of our G-Cap models over their respective Qwen3-VL base models. All values are reported in percentages (%), except for length.

parameters for 3 epochs using the AdamW optimizer with a cosine learning rate schedule, where the learning rate decays to a minimum value of $1e{-}6$. All experiments are conducted on NVIDIA H20 GPUs, with training costs of approximately 256 GPU hours for the 8B model and 1,280 GPU hours for the 32B model.

4.2 Evaluation on GC-Bench

Table 1 reports the average Precision, Recall, Graph-F1, and caption length over all samples in GC-Bench. Among closed-source models, Gemini 3 Pro achieves the highest average Graph-F1, establishing a strong upper bound for game character captioning. Notably, even the weakest closed-source model, GPT-5 mini, still outperforms all open-source baselines. Among open-source models, Qwen3-VL-235B-A22B-Instruct performs best, while InternVL3.5-38B ranks lowest across all metrics. This clear stratification indicates that a substantial gap remains between open- and closed-source LVLMs, which further widens for weaker open-source systems.

Against this backdrop, our G-Cap models consistently improve upon their corresponding Qwen3-VL backbones. G-Cap-8B increases average Graph-F1 by 20.5% with only a moderate increase in caption length, while G-Cap-32B surpasses Qwen3-VL-32B-Instruct by 12.3% in Graph-F1 despite producing shorter captions. These results demonstrate that G-Cap enhances structural coverage rather than relying on increased verbosity. Qualitative case studies in the Appendix N further corroborate these findings, particularly for fine-grained attributes and complex clothing hierarchies.

4.3 Analysis

4.3.1 Analysis of Parent Nodes Matching and Subgraph Comparison.

To gain deeper insight into model strengths and weaknesses, Figure 16 (Appendix M) presents a focused analysis that reveals a clear divergence between Parent Nodes Matching (*max_sim*) and Subgraph Matching (*desc_sim*). High *max_sim* scores (>0.89) indicate robust coarse-grained recognition of major entities, such as clothing items and equipment. In contrast, the substantially lower *desc_sim* scores expose a critical limitation: although models can reliably identify *what* is present, they struggle to accurately describe *how* attributes and relations are organized within local subgraphs, often resulting in hallucinated or incomplete details.

4.3.2 Verbosity vs. Substance: Analysis on Information Density

To analyze the relationship between caption length and semantic content, we introduce *Information Density*. We first parse each generated caption into

Data Components		G-Cap-8B			G-Cap-32B		
GC-Real	GC-Syn	Avg Precision	Avg Recall	Avg Graph-F1	Avg Precision	Avg Recall	Avg Graph-F1
✗	✗	42.28	19.43	26.09	39.65	23.32	28.78
✓	✗	41.06	22.92	28.97	42.64	25.23	31.23
✗	✓	40.02	19.99	26.16	40.28	20.58	26.73
✓	✓	**43.13**	**25.33**	**31.45**	**43.82**	**26.16**	**32.31**

Table 2: Ablation study on the effects of GC-Real and GC-Syn data across different model scales.

a graph, and define density as $\frac{|V|+|E|}{L}$, where $|V|$ and $|E|$ denote the numbers of nodes (objects/attributes) and edges (relations), and L is the caption word count. This metric measures how much distinct visual information is conveyed per word.

As shown in Figure 7, GPT-5.1 produces the longest captions (>500 words) but exhibits the lowest information density (0.49), suggesting substantial verbosity with limited additional semantic content. In contrast, Qwen3-VL-8B-Instruct generates much shorter captions (139 words) with relatively high information density, but achieves lower F1 scores, indicating that excessive conciseness may omit critical information. Notably, Gemini-family models strike a strong balance between information density, caption length, and overall quality. While the G-Cap series attains competitive caption quality, it remains less efficient linguistically. Overall, these results suggest that future image captioning models should avoid pursuing length alone, and instead aim to jointly maximize information density and coverage, producing descriptions that are both comprehensive and semantically concise.

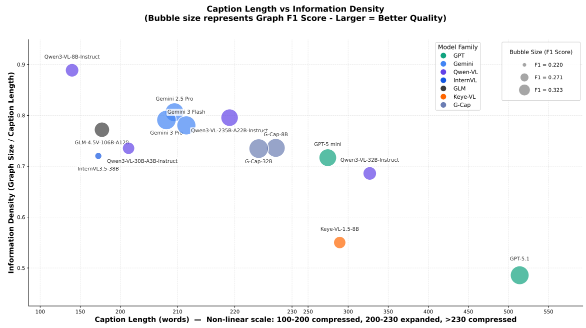


Figure 7: **Analysis of Information Density versus Caption Length.** The y-axis measures Information Density (semantic graph size divided by word count), while the x-axis shows the average caption length. The bubble size represents the overall Graph-F1 score.

4.3.3 Impact of Data Composition

We study the individual and joint effects of the GC-Real and GC-Syn datasets, with results reported in Table 2. Incorporating GC-Real yields a clear performance gain over the baseline, demonstrating that high-quality annotations distilled from the teacher model (`Gemini 2.5 Pro`) effectively transfer domain-specific knowledge to the student model. By contrast, training solely on GC-Syn leads to only marginal improvements. This is mainly due to the domain gap between synthetic images and real game screenshots. Since we apply full-parameter fine-tuning, the model becomes sensitive to the visual distribution of the training data. Synthetic images generated by `Nano Banana` exhibit distinct visual characteristics, causing models trained exclusively on GC-Syn to generalize poorly to real-world evaluations.

The best performance is achieved by combining both datasets. Although GC-Syn suffers from distribution shift, it provides richer structural diversity, including complex character compositions and hierarchical clothing layers that are rare in real data. These synthetic samples expand the model's structural understanding and complement the realistic visual textures of GC-Real, leading to a synergistic performance improvement.

5 Conclusion

In this paper, we advance the study of image captioning in virtual-world scenarios by introducing the **Game Character Captioning** task and a manually annotated benchmark, **GC-Bench**. To effectively evaluate GC-Bench, we propose **Graph-F1**, a subgraph matching-based evaluation metric that captures fine-grained attributes and structural relations. We further construct the **GC-148K** dataset and, based on the **Qwen3-VL** series, develop **G-Cap-8B** and **G-Cap-32B**. Both models achieve improvements of up to 20.5% over their base counterparts, with **G-Cap-32B** performing on par with **Gemini 3 Pro**, validating the usability of our approach in industrial-grade production environments.

Limitations

Despite the strong performance of G-Cap, there are two main limitations that warrant further investigation. First, the model exhibits limited knowledge of specialized or niche fashion terms, occasionally failing to recognize uncommon garments or produce precise domain-specific expressions. This limitation could be mitigated in future work through knowledge injection during mid-training or by leveraging Agentic RL to enable tool-augmented search for rare or complex items.

Second, G-Cap struggles to accurately describe garments with unconventional cuts or patchwork designs. Enhancing the diversity and quality of synthetic training data, particularly for atypical clothing structures, may improve the model's ability to capture such variations.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.

J Bai, S Bai, S Yang, S Wang, S Tan, P Wang, J Lin, C Zhou, and J Qwen-VL Zhou. 2023. A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 6.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025b. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt. 2023. Visit-bench: A benchmark for vision-language instruction following inspired by real-world use. *arXiv preprint arXiv:2308.06595*.

Kanzhi Cheng, Wenpo Song, Jiaxin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. 2025. Caparena: Benchmarking and analyzing detailed image captioning in the llm era. *arXiv preprint arXiv:2503.12329*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. 2024. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*.

Yunhao Ge, Xiaohui Zeng, Jacob Samuel Huffman, Tsung-Yi Lin, Ming-Yu Liu, and Yin Cui. 2024. Visual fact checker: Enabling high-fidelity detailed caption generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14033–14042.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528.

Hang Hua, Qing Liu, Lingzhi Zhang, Jing Shi, Soo Ye Kim, Zhifei Zhang, Yilin Wang, Jianming Zhang, Zhe Lin, and Jiebo Luo. 2025. Finecaption: Compositional image captioning focusing on wherever you want at any granularity. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24763–24773.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yebin Lee, Imseong Park, and Myungjoo Kang. 2024. Fleur: An explainable reference-free evaluation metric for image captioning using a large multimodal model. *arXiv preprint arXiv:2406.06004*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Xiangtai Li, Tao Zhang, Yanwei Li, Haobo Yuan, Shihao Chen, Yikang Zhou, Jiahao Meng, Yueyi Sun, Shilin Xu, Lu Qi, and 1 others. 2025. Denseworld-1m: Towards detailed dense grounded caption in the real world. *arXiv preprint arXiv:2506.24102*.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, and 1 others. 2024. Docci: Descriptions of connected and contrasting images. In *European Conference on Computer Vision*, pages 291–309. Springer.

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

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, and 7 others. 2025. DINOv3. *Preprint*, arXiv:2508.10104.

Kwai Keye Team. 2025. Kwai keye-vl-1.5 technical report.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, and 69 others. 2025. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *Preprint*, arXiv:2507.01006.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Long Xing, Qidong Huang, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Jinsong Li, Shuangrui Ding, Weiming Zhang, Nenghai Yu, and 1 others. 2025. Scalecap: Inference-time scalable image captioning via dual-modality debiasing. *arXiv preprint arXiv:2506.19848*.

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*.

Lin Zhang, Xianfang Zeng, Kangcong Li, Gang Yu, and Tao Chen. 2025a. Sc-captioner: Improving image captioning with self-correction by reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23145–23155.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Data Source of GC-Bench

To ensure a comprehensive evaluation of Game Character Captioning, we constructed the GC-Bench dataset by collecting images from a diverse array of video games. The dataset includes content from high-popularity titles across multiple genres, including MOBA, FPS, and Open-World RPGs. Specifically, we crawled public images from the internet corresponding to 20 distinct games. The complete list of source games is provided in Table 3.

Source Games	
League of Legends	Black Survival
KartRider	CrossFire
Genshin Impact	Game for Peace
Honor of Kings	Destiny 2
Eternal Return	Rogue Company
Wuthering Waves	Paragon
Girls' Frontline	Cyberpunk 2077
Snowbreak: Containment Zone	Assault Fire
Sword of Convallaria	Monster Hunter
Zenless Zone Zero	Uncharted

Table 3: List of source games used in GC-Bench. The collection covers a wide spectrum of genres and artistic styles.

B Baselines for Game Character Captioning on GC-Bench

To evaluate Game Character Captioning, we benchmark a range of vision-language models on the GC-Bench, including both closed-source and open-source systems. For closed-source models, we include OpenAI's GPT-5 mini and GPT-5.1, as well as Google's Gemini 2.5 Pro, Gemini 3 Flash, and Gemini 3 Pro. Open-source baselines cover recent large-scale models such as InternVL3.5-38B (Wang et al., 2025), Keye-VL-1.5-8B (Team, 2025), Qwen3-VL-Instruct variants (8B, 30B, 32B, 235B) (Bai et al., 2025a), and GLM-4.5V-106B-A12B (Team et al., 2025). All open-source models are deployed using vLLM (Kwon et al., 2023), with generation configurations set to the models' official default parameters.

C Task-Specific Prompt Template for Game Character Captioning

The task-specific prompt for Game Character Captioning is shown in the Figure 8.

❯ Prompt For Game Character Captioning

```
You are an image caption writer. Your task is to describe the character in the image in a single,
clear, and detailed paragraph.

    ## Instructions:

    **DO:**

    * Describe the character's gender and artistic style briefly.
    * Describe the character from head to toe, focusing on the layers of clothing and materials.
    * Break down the description into sections based on body parts (e.g., head, left arm, right arm,
      torso, left leg, right leg).
    * Pay special attention to the head, providing a detailed description.
    * For symmetrical body parts, describe them individually (e.g., describe each wrist separately if
      both have accessories).
    * Be specific about colors, materials (e.g., metal, fabric, leather), and how the clothing layers
      are arranged.

    **DON'T:**

    * Do not use overly technical or obscure words.
    * Avoid long, complicated sentences.
    * Do not invent details that are not visible in the image.

    ## Perspective Note:

    The left side of the image is the character's right side, and vice-versa.

    **Now, write a caption for the image I've provided.**
```

Figure 8: Task-Specific Prompt Template for Game Character Captioning

D Prompt Template for Triples Extraction

The Prompt Template for Triples Extraction is shown in the Figure 9. Due to space limitations, only a truncated version of the prompt is shown; the complete prompt will be released with the code after the review process.

E Prompt Template for Hallucination Check

The prompt used to guide Qwen3-Reranker-4B for the **Precision (Hallucination Check)** is illustrated in Figure 10. This prompt is designed to verify whether the fine-grained details in a generated subgraph are factually supported by the corresponding ground-truth subgraph.

F Prompt Template for Coverage Check

The prompt used for the **Recall (Coverage Check)** is shown in Figure 11. This prompt evaluates whether the local information present in a ground-truth subgraph is sufficiently covered by the generated subgraph.

G Prompt Template for Caption Assembly

The prompt used to assemble sampled garment components into a coherent caption is illustrated in Figure 12. Due to space limitations, only a truncated version of the prompt is shown; the complete prompt will be released with the code after the review process.

❯ Prompt For Triplets Extraction

```
You are an expert system for extracting RDF-style triples from game character's caption. Convert
descriptive text into a structured list of triples [Subject, Predicate, Object] that strictly follow the
specification below.

1) Core Subject Hierarchy (dot-notation)
- Primary top-level entity: Character
- Conditional top-level entity: Environment (only for background/lighting/setting)
- Asymmetrical body namespaces
- Fine-grained subparts (extend as needed)
Important: "Left/Right side of the image" does not change anatomical left/right. Always use the
character's anatomical sides (RightArm, LeftArm, etc.).

2) Standardized Predicates
- Identity & classification:
  - isA (only for Character's fundamental identity; e.g., Male/Female)
  - HasStyle (style keywords on Character or items; e.g., Chibi, Adventurer, Layered)
  - IsTypeOf (categorize a subpart when applicable; e.g., Hairstyle type)
- Composition & parts:
  - Has (for inherent physical parts; e.g., Character.Head Has Face; Robe Has Hem)
- Attributes (one attribute per triple):
  - HasColor  - HasMaterial
  - HasShape - HasTexture
  - HasSize   - HasStyle
  - HasDecoration  - DisplaysState (visibility/pose/state; e.g., Bare, Sits Loosely, VisibleAtNeck)
- Wearing, equipment, actions:
  - Wears (for clothing/headwear items; subject is Character)
  - Equips (for gear/weapons; subject is Character)
- Spatial relations (critical):
  - ReachesTo (boundaries/extent; subject is the part defining the edge, e.g., Robe.Hem)
  - Covers / PartiallyCovers (occlusion; subject is the covering item)
  - IsAttachedTo (physical attachment; subject is the attached item under Equipment or a subpart)
  - IsNear (proximity; subject is the body part or item near another hierarchical target)

3) Relational Boundaries, Occlusion, Attachments, and Layering (Critical)
- Boundaries: the part that ends defines the relation.
  - Example: ["Character.Torso.Robe.Hem", "ReachesTo", "Mid-Thigh"]
- Occlusion resulting from boundaries:
  - Example: ["Character.Torso.Robe", "PartiallyCovers", "Character.RightLeg"]
  - Use Covers when the target is fully wrapped/obscured (e.g., bandage wraps over forearm/wrist).
- Attachments:
  - Use IsAttachedTo for physical securing; include multiple relations if attached by more than one
item.
  - Example: ["Character.Equipment.Scabbard", "IsAttachedTo", "Character.Torso.Belt"]
- Layering (clothing worn order):
  - IsWornOver: use when one garment is worn on top of another.
    - Example: ["Character.Torso.Robe", "IsWornOver", "Character.Torso.InnerShirt"]
  - IsWornUnder: use when one garment is worn beneath another.
    - Example: ["Character.Torso.InnerShirt", "IsWornUnder", "Character.Torso.Robe"]
  - Note: Layering complements Wears; always keep Wears for each worn item.
- Prohibited objective-fact relations:
  - Do not emit universally true anatomical "Has" facts that add no scene-specific information.
  - Prohibited examples:
    - ["Character.Head.Face", "Has", "Character.Head.Eyes"]
    - ["Character.Head.Face", "Has", "Character.Head.Mouth"]
    - ["Character.Head", "Has", "Character.Head.Face"]
    - ['Character.Head', 'Has', 'Character.Head.Eyes']

4) Atomicity, Brevity, and Formatting
- One fact per triple; do not combine attributes.
- Objects must be short, atomic values, Title Case, and ≤ 25 characters.
  - Examples: "Dark Gray", "Warm Orange", "Below The Knee", "Sits Loosely"
- Do not repeat object values in the subject. Details belong only in the Object.
- Use singular item names (e.g., Boot, Short).

Output
- Return a JSON-like array of triples: [[Subject, Predicate, Object], [Subject, Predicate,
Object], ...]
- No extra commentary; one fact per triple; follow the above hierarchy and predicates.
```

Figure 9: Prompt Template for Triples Extraction

❯ Prompt For Precision (Hallucination Check)

```
<|im_start|>system
    You are a Logical Verifier. Your task is to determine if Sentence A (claim) is a strict
    informational subset of Sentence B (facts).

    ### Core Principle: Zero Addition

    Sentence A **must not** contain any information that is not explicitly in Sentence B.

    ### Judgment Rules:

    You will judge as **`no (not supported)`** if **ANY** of these rules are broken:

        1.  **No Added Details**: A introduces any new attribute.
                -   B: "a car" -> A: "a **red** car" = **`no`**

        2.  **No Added Specificity**: A uses a more specific term than B.
                -   B: "a blue shirt" -> A: "a **dark blue** shirt" = **`no`**
                -   B: "a metal box" -> A: "a **steel** box" = **`no`**

        3.  **No Contradictions**: Any part of A conflicts with B.

        4.  **No Missing Support**: Any part of A is not mentioned in B.

     You will judge as **`yes (supported)`** **only if** A perfectly passes all rules. Generalizations
    are acceptable (e.g., B: "pistol" -> A: "gun").

     **Output Requirements**:
            Note that you can only output \"yes\" or \"no\".
<|im_end|>
```

Figure 10: Prompt template for the **Precision (Hallucination Check)** in subgraph comparison.

❯ Prompt For Recall (Coverage Check)

```
<|im_start|>system
  You are an information coverage auditor. Your task is to determine whether the "submitted
  content" (Sentence B) fully covers all information points in the "requirement list" (Sentence A,
  ground truth).

     **Core Rules**:
         1. **Coverage Requirement**: B must include all attributes and values from A (type, color,
         material, shape, location, etc.). Equivalent expressions are allowed (e.g., "silver-white long
         hair" ≈ "long silver-white hair").
         2. **Tolerance for Additional Information**: Even if B contains extra information not
         declared in A, it does not affect the judgment (these redundant details are handled in
         other evaluation steps).
         3. **Conflict Rejection**: If B contradicts A in any attribute, it is judged as
         "incomplete/inconsistent."
         4. **Non-informative Words Ignored**: Modal particles, stop words, and order differences
         do not affect the judgment.

     **Judgment Criteria**:
         - **Yes (Full Coverage)**: B covers all information points in A (additional information is
         allowed as long as there is no conflict).
         - **No (Incomplete/Inconsistent)**: Any information point in A is missing in B, or a conflict
         exists.
      **Output Requirements**:
            Note that you can only output \"yes\" or \"no\".
<|im_end|>
```

Figure 11: Prompt template for the **Recall (Coverage Check)** in subgraph comparison.

❯ Prompt For Assemble Caption

```
You are an expert character description writer. Your task is to combine scattered clothing/appearance parts into a single, coherent,
detailed character caption.

## INPUT FORMAT
You will receive:
 **Clothing Parts**: A JSON object containing selected clothing parts from different body zones:
   - head: hair, face, eyes, head_accessories
   - torso: base_layer (bodysuits, undershirts), mid_layer (shirts, main garments), outer_layer (jackets, robes, cloaks)
   - arms: upper_arm, lower_arm, hands
   - legs: legwear_layer, bottom_layer, waist_layer
   - body_accessories: feet, back, floating, other, weapons_or_props

Each clothing item provides: name, colors, materials, patterns, adornments, side (both/left/right).
Each accessory item provides: name, description, colors, materials, adornments.

## YOUR TASK
1. **Combine all parts** into ONE flowing, natural character description caption
2. **Start with simple prefix**: Use art_medium + gender
3. **ENSURE LOGICAL CONSISTENCY** in clothing descriptions

**YOU MUST STRICTLY FOLLOW THIS EXACT FORMAT STRUCTURE:**

Head (including face): [hair description]; [hair accessories if any]; [eye color]; [facial expression]. Torso: Layered garments——
[inner layer]; [mid layer]; [outer layer]; [waist items]. Right arm (character's right, image left): [arm coverage from shoulder to
hand]. Left arm (character's left, image right): [arm coverage from shoulder to hand]. Right leg (character's right, image left):
[leg coverage from thigh to foot]. Left leg (character's left, image right): [leg coverage from thigh to foot].

..................................................

**IMPORTANT: MIRROR IMAGE ORIENTATION**
The character faces the viewer, so left/right are MIRRORED:
- Character's RIGHT arm/leg appears on the LEFT side of the image
- Character's LEFT arm/leg appears on the RIGHT side of the image
Always clarify with format like "Right arm (character's right, image left):" to avoid confusion.

## OUTPUT FORMAT
Return a JSON object with exactly one field:
{
  "assembled_caption": "<your complete character caption here>"
}

Output ONLY the JSON object, no extra text. The caption MUST follow the exact section structure shown in the example.
```

Figure 12: Prompt template for assembling garment components into a structured caption.

H Prompt Template for Garment Component Extraction

The prompt used to extract structured garment components from captions is illustrated in Figure 13. Due to space limitations, only a truncated version of the prompt is shown; the complete prompt will be released with the code after the review process.

```
❯ Prompt For Extract Garment Components

You are an elite character outfit technical director helping a clothing simulation system.
Your task is to read one caption that describes a character skin and convert it into a layered outfit specification.
Follow these strict rules:

## HEAD ZONE (Important - extract in detail)

..............................................

## TORSO ZONE (includes full-body garments)

..............................................

## LEGS ZONE (with Left/Right handling)

..............................................

## ARMS/HANDS ZONE (with Left/Right handling)

..............................................

## BODY ACCESSORIES STRUCTURE

..............................................

## CLOTHING ITEM STRUCTURE

..............................................

## IMPORTANT RULES

..............................................

## CRITICAL: OUTPUT FORMAT
You MUST output a valid JSON object (no markdown, no code blocks, just raw JSON).
Follow this exact structure:

..............................................

IMPORTANT REMINDERS:
- Output ONLY the JSON object, no extra text before or after
- Use empty arrays [] for missing/absent items, not null
- Use empty string "" for missing string values, not null
- All clothing items in torso/arms/legs follow the same structure with: name, description, colors, materials,
patterns, adornments, render_priority, side
- Body accessories (feet/back/floating/other) follow: name, description, colors, materials, adornments
- The `description` field should ONLY contain special visual details NOT covered by colors/materials/patterns
- side field must be "both", "left", or "right"
```

Figure 13: Prompt template for extracting structured garment components.

I Human Alignment Analysis on GC-Bench

To validate the effectiveness of our metric, we conducted a pairwise human preference study on 100 randomly sampled GC-Bench images, comparing caption pairs generated by two randomly selected models for each image. Human annotators selected the superior caption based on accuracy and completeness. For comparison, we chose widely-used automatic metrics that represent different evaluation paradigms: **BERTScore** (Zhang et al., 2019) as an embedding-based evaluation method, **CAPTURE** (Dong et al., 2024) as a graph-based caption evaluation method, and **GPT-5 Judge (with Ref)** (Lee et al., 2024) as a LLM-based evaluation method, where we followed the prompt design from the original paper but replaced their evaluated model with GPT-5. We then calculated the agreement rate between these metrics and human judgments. As shown in Table 4, **Graph-F1** achieves the highest alignment with human judgment, demonstrating its effectiveness.

Metric	**Human Alignment**
BERTScore	0.57
CAPTURE	0.59
GPT-5 Judge (with Ref)	0.60
Graph-F1 (Ours)	**0.79**

Table 4: Human alignment accuracy of Graph-F1 and other metrics.

J Garment Component Statistics and Examples

This section reports the statistics of the deduplicated component library and provides representative examples for each category.

J.1 Deduplicated component counts

Table 6 summarizes the number of deduplicated components for each category in our library.

J.2 Representative examples (one per leaf category)

To avoid clutter while preserving the original structure, we show one representative entry per leaf category using a JSON-like dictionary format.

Head Components

Head–Hair
```
{"hairstyle":"short","hair_color":["dark black"]}
```

Head–Face
```
{"facial_features":["fair skin","dark eyebrows"]}
```

Head–Eyes
```
{"eye_color":["dark"], "eye_features":["blindfolded","
golden faceplate with cross"]}
```

Head–Face accessories
```
{"name":"maroon fabric mask"}
```

Head–Ear accessories
```
{"name":"small pink teardrop-shaped earring on left ear"}
```

Head–Headwear
```
{"name":"tall gold headdress with swirling ribbon-like
accents"}
```

Head–Other head
```
{"name":"bone-like horn (left, tall curved spike)"}
```

Torso Layers

Torso–Base layer
```
{"name":"cybernetic neck and shoulder plating","description
":"interlocking black metallic plates visible above
collar, covering clavicle and right shoulder","colors
":["black"],"materials":["metal"],"side":"both"}
```

Torso–Mid layer
```
{"name":"hanging panels","description":"two long panels
over legs, open at center","colors":["light purple"],
"patterns":["dark purple geometric patterns at hem"],
"side":"both"}
```

Torso–Outer layer
```
{"name":"thick scarf","description":"frayed edges, wrapped
around neck","colors":["maroon","white","light blue"],
"materials":["fabric"],"patterns":["maroon, white,
and light blue pattern"],"side":"both"}
```

Arms Components

Arms–Upper arm
```
{"name":"flowing blue bell sleeve","description":"wide,
flowing with gold hem, on left upper arm","colors":["
blue","gold"],"side":"left"}
```

Arms–Lower arm
```
{"name":"dark brown leather bracer with gold metal plates",
"description":"covers forearm, matching on both arms",
"colors":["dark brown","gold"],"materials":["leather",
"metal"],"adornments":["gold metal plates"],"side":"
both"}
```

Arms–Hands
```
{"name":"light gray armored hand","description":"left hand
covered in suit material with armored plating","
colors":["light gray"],"materials":["metallic armor"],
"side":"left"}
```

Legs Components

Legs–Legwear layer
```
{"name":"thick brown rope-like anklet","description":"with
large golden bell attached","colors":["brown","gold"],
"materials":["rope-like"],"adornments":["large golden
bell"],"side":"left"}
```

Legs–Bottom layer
```
{"name":"pleated skirt with vertical orange stripes","
description":"long skirt with front panel featuring
large cloud-like pattern outlined in gold","colors":[
"white","orange","gold"],"materials":["fabric"],"
patterns":["vertical orange stripes","cloud-like
pattern"],"side":"both"}
```

Legs–Waist layer
```
{"name":"web belt","description":"thick, with central
silver buckle and multiple attachments","colors":["
green","silver"],"materials":["nylon webbing","metal"
],"adornments":["small pouches","sheathed knife (
right hip)"],"side":"both"}
```

Body Accessories

Body accessories–Feet
```
{"name":"black knee-high combat boots","description":"laced
up with red shoelaces, reach knee height","colors":[
"black","red"],"materials":["combat material"],"
adornments":["red shoelaces"]}
```

Body accessories–Back
```
{"name":"mechanical wing apparatus","description":"large
black and dark blue wings with small yellow lights","
colors":["black","dark blue","yellow"],"materials":["
metal","plastic"],"adornments":["yellow lights"]}
```

Body accessories–Floating
```
{"name":"glowing teal orbs","description":"two orbs affixed
to capelet, emit glow","colors":["teal"],"materials":
["energy","crystal"]}
```

Body accessories–Other
```
{"name":"patched leather canteen with green jade-like
fastener","description":"hanging from belt","colors":
["brown","green"],"materials":["leather","jade-like
material"],"adornments":["patches"]}
```

Body accessories–Weapons/props
```
{"name":"dual scythe-like blade hands","description":"large
curved teal blades with purple scale patterns and
dark green tips","colors":["teal","purple","dark
green"],"materials":["blade"],"adornments":["curved
tip","scale-like patterns"]}
```

K List of Art Mediums

To ensure the diversity of visual styles in our synthesized dataset, we prompt the generation model using a comprehensive vocabulary of art mediums. Table 5 categorizes the collected medium tags into five major stylistic domains.

L Sampling Pipeline for Garment Components

This appendix details the garment component sampling pipeline for constructing structured character outfits. Garment components are independently sampled across predefined body zones, following probabilistic rules designed to reflect the semantic characteristics of each zone.

Head (Hair, Face, Eyes, and Others). For the head region, the sampler deterministically selects one item from each of `hair`, `face`, and `eyes`, when available. Others are drawn from four optional sub-categories: `headwear`, `face_accessories`, `ear_accessories`, and `other_head`. Each subcategory is independently included with probability 0.5 if non-empty, and one item is sampled upon inclusion.

Torso (Layered Clothing). The torso is modeled as a three-layer clothing system. The sampler attempts to select one item from each of the following layers: `base_layer`, `mid_layer`, and `outer_layer`, whenever the corresponding category is available.

Arms. Arm garments are divided into `upper_arm`, `lower_arm`, and `hands`. Each layer is independently included with probability 0.8 if non-empty, and one item is sampled from each included layer.

Legs. The leg region consists of three candidate layers: `legwear_layer`, `bottom_layer`, and `waist_layer`. The sampler selects either two layers (with probability 0.5) or all three layers (with probability 0.5), and samples one item from each selected layer.

Body Accessories. Body accessories are sampled from `feet`, `back`, `floating`, and `other`. The `feet` category is included with probability 0.8, while the remaining accessory slots are included with probability 0.3 when available. In addition, the special category `weapons_or_props` is included with probability 0.2 if non-empty.

M Decomposed Analysis of Parent Node and Subgraph Matching

Figure 16 presents a decomposed evaluation of Graph-F1 by separating parent node alignment (*max_sim*) from subgraph-level semantic consistency (*desc_sim*).

N Case Study

We present qualitative case studies comparing **G-Cap-32B** with its base model, **Qwen3-VL-32B-Instruct**, and a strong closed-source baseline, **Gemini 3 Pro**, on the same examples.

O Qualitative Examples of GC-Bench Captions

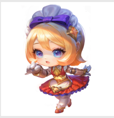
Figure 15 presents examples of GC-Bench.

Category	Art Medium Tags
Anime & 2D	anime, anime art style, anime style, anime-influenced, anime-inspired, anime-style, anime-style drawing, anime-style illustration, anime/cartoon style, 2D animation, 2D anime style, 2D anime/pixel art, 2D digital anime style, 2D digital illustration, 2D illustration, 2D flat illustration, 2D pixel art, cel-shaded, cel-shaded anime style, cell-shaded, anime or cel-shaded, pixel art, pixel art or anime-style sprite, chibi
3D & Game Art	3D cartoon, 3D cel-shaded, 3D digital sculpture, 3D game-art, 3D render, 3D sculpture, 3D wireframe, 3D wireframe model, 3D-model style, digital model, digital render, digital rendering, digital painting with 3D rendering, low poly, composite render, stylized 3D render, anime-influenced 3D render, anime-inspired 3D render
Realistic & Photo	photograph, photographic, photographic image, photorealistic, photorealistic digital composite, photorealistic digital rendering, photorealistic illustration, photorealistic render, realistic, realistic art style, realistic drawing, realistic fantasy art, realistic illustration, realistic painting, realistic-style illustration, realistically illustrated, realistically painted, realistically rendered illustration
Traditional & Sketch	sketch, sketched, digital sketch, colored sketch, line art, ink-and-watercolor, ink-wash, watercolor, watercolor or ink-wash, ink wash painting, painting, painted, painterly, painterly illustration, painterly render, painterly style, painterly-style illustration, painterly concept art illustration, painted miniature, hand-drawn, colored pencil drawing, sketched colored-pencil, impressionist oil painting, ceramic figurine, sculpture
Digital & Design	digital art, 2D digital art, digital anime art, digital cartoon, digital illustration, digital painting, concept art, concept art illustration, fantasy art, fashion illustration, graphic art, illustration, illustrative art style, illustrated, mixed-media, stylized, stylized illustration, comic book art, comic-book art, American comic style, unspecified

Table 5: The art medium vocabulary used in our data synthesis pipeline.

Region	Category	Count
Head	Hair	75,169
	Face	57,009
	Eyes	54,363
	Headwear	26,589
	Face accessories	13,268
	Ear accessories	9,849
	Other head	12,453
Torso	Base layer	20,849
	Mid layer	46,669
	Outer layer	24,211
Arms	Upper arm	26,384
	Lower arm	35,062
	Hands	17,236
Legs	Legwear layer	12,953
	Bottom layer	27,913
	Waist layer	32,322
Body Acc.	Feet	28,388
	Back	9,377
	Floating	1,889
	Other	19,966
	Weapons/props	7,962
Total		**559,880**

Table 6: Counts of deduplicated components in the collected library (by category).



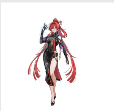
Human : You are an image caption writer. Your task is to describe the character in the image in a single, clear, and detailed paragraph.
Instructions:
....
Now, write a caption for the image I've provided.

Qwen3-VL-32B: The character has a large, round head with soft, golden-orange hair styled in a short bob with bangs. On top of her head sits a large, puffy light blue bow made of shiny fabric, tied with a thick, glossy purple ribbon. A gold-colored hair accessory shaped like a heart with an orange gem is clipped into her hair on the right side. Her eyes are large and expressive, with bright blue irises that shimmer with hints of purple and pink, framed by long, dark eyelashes. Her cheeks are rosy, and her mouth is slightly open in a cheerful expression. Her left arm is bent forward, with a white, fingerless glove covering her hand, which is pointing forward. The glove has a gold band at the wrist. Her left forearm is covered by a brown, armored bracer with silver metal accents and ornate gold trim. Her right arm is extended back, wearing a dark blue, fingerless glove with a gold, heart-shaped gem at the wrist. The right forearm also has a matching brown armored bracer with silver and gold detailing. Her torso is covered by a red sleeveless dress with a white ruffled collar. Over the dress, she wears a gold-colored vest with a large, faceted yellow gem at the center, and a golden, heart-shaped belt with a matching gem at the buckle. The red skirt is short and flared, with ruffled layers and a dark blue or purple hem. Her left leg is slightly bent forward, wearing a white, textured legging with subtle floral patterns. Her foot is in a red and gold shoe with a small heel, featuring a gold bow and a purple ribbon. Her right leg is bent behind her, with a similar white legging and a matching red and gold shoe, also adorned with a gold bow and purple ribbon.

G-Cap-32B: This image features a female character drawn in a chibi anime style. Her head is large, with short, golden-blonde hair styled in a bob. She has large, bright blue eyes, a small nose, and a gentle smile. Atop her head sits a large, puffy, light-blue bow, with a smaller, shiny purple bow layered on top. On the right side of her hair is a gold, heart-shaped clip with an orange gem. Around her neck, she wears a white, ruffled collar with a golden-yellow bow at the front. Her torso is dressed in a sleeveless red top with gold trim. Over her midsection is a piece of gold armor with a heart-shaped cutout, which sits above a short, ruffled red skirt with a dark blue lining. Her right arm is covered by a bronze-colored, engraved metal gauntlet that extends from her elbow to her wrist, and she wears a white glove. Her left arm has a matching bronze gauntlet, and her hand is covered by a blue glove. On her right leg, she wears a white, thigh-high stocking with a faint floral pattern, ending in a red shoe with a gold strap and a small gold star. Her left leg is identically dressed in a white stocking and a red shoe with a gold strap and star.

Gemini 3 Pro: This digital illustration depicts a female character in a cute, chibi style, resembling a mechanical doll. Her head features short, golden-blonde hair styled in a bob with bangs sweeping across her forehead. She wears a white, ruffled maid bonnet adorned with a large violet ribbon bow in the center and a gold-framed amber gem on her left side. Her oversized eyes are expressive, displaying a sparkling gradient of blue and purple. Around her neck, she wears a white collar with a gold ascot tie, leading into a red bodice. Her waist is accentuated by a gold structure holding a glowing yellow gem, which sits above a white, heart-shaped apron trimmed in gold. This apron covers a pleated red skirt that reveals a layer of purple ruffles underneath. Her arms resemble doll parts with visible ball joints; the character's right arm reaches forward, clad in a white glove with a gold cuff at the wrist, while her left arm extends backward, displaying a mechanical elbow joint, a white glove, and a matching gold cuff. Her legs are covered in white stockings with subtle patterns, and she wears red shoes detailed with gold accents on her feet.

(a) Case study example #1.



Human : You are an image caption writer. Your task is to describe the character in the image in a single, clear, and detailed paragraph.
Instructions:
....
Now, write a caption for the image I've provided.

Qwen3-VL-32B: The character has vibrant red hair styled in a high ponytail with a few loose strands framing her face, adorned with a black hairband and a small red flower accessory. Her eyes are a striking purple, and she wears a confident, slightly smirking expression. Her outfit is a form-fitting black bodysuit with a high collar and a deep V-neckline, featuring red trim and gold accents. Over the bodysuit, she wears a short, open black coat with red lining and gold geometric patterns, which flares out at the sides. The coat has a high slit on the left side, revealing her left thigh. A red ribbon with a blue bow is tied around her neck, and a gold chain hangs from her right hip, attached to a small pouch. Her right arm is covered in a black armored gauntlet with glowing blue energy crackling around her fingers, while her left arm has a similar gauntlet with purple energy accents. Both gauntlets have sharp, claw-like fingers. Her left wrist has a small black pouch with a red strap. She wears black high-heeled sandals with crisscross straps. Her long red hair flows down her back in multiple thin, ribbon-like strands.

G-Cap-32B: This is a female character drawn in a futuristic anime style, blending traditional and cybernetic elements. Her head features vibrant red hair styled in a high ponytail with long bangs framing her face and a single lock falling over her right shoulder. A black hairpin with a red flower-shaped ornament is tucked into her hair, and a black headset is visible on her left ear. She has purple eyes and a confident smile. Her torso is clad in a form-fitting, sleeveless black dress with a high collar, red trim, and a sheer panel over the chest. A blue ribbon is tied at her neck, and a red sash with a gold clasp hangs down her front. Her right arm is a full cybernetic prosthetic made of black and dark gray metal, with glowing blue and purple accents on the forearm; her right hand is raised, crackling with blue electricity. Her left arm is mostly bare, with a black and red metallic piece covering her shoulder and bicep, and a gold chain with a decorative tassel hangs from it. The dress has a high slit on her right side, exposing her right leg, which is bare. Her left leg is mostly covered by the dress, which features a red and black diamond pattern on the side. On her feet, she wears black high-heeled shoes with thin straps that crisscross over her feet.

Gemini 3 Pro: This anime-style illustration features a female character with a striking, futuristic appearance. She has long, voluminous red hair tied in a high ponytail with a black hairpiece, while long bangs frame her face and a small red flower ornament sits on the left side of her head. Her eyes are a deep purple, and she has a small beauty mark under her right eye. She wears a form-fitting black dress with a halter neckline and a translucent mesh panel running down the center of her chest. A blue decorative knot rests near her left collarbone, and a long red ribbon hangs down her front. Her right arm is encased in a black, segmented mechanical gauntlet that emits blue electrical sparks from the claw-like fingertips. Her left arm features a matching black gauntlet with glowing purple lines and sharp, red-tipped fingers. Gold chains hang loosely from her waist. The dress has a high slit on the left side, revealing her leg, and the fabric features a black exterior with a red, grid-patterned inner lining. She wears black high-heeled shoes with straps crossing over her feet.

(b) Case study example #2.

Figure 14: Qualitative case studies comparing **Qwen3-VL-32B-Instruct**, **G-Cap-32B**, and **Gemini 3 Pro** on identical inputs.

Figure 15: Examples of GC-Bench.

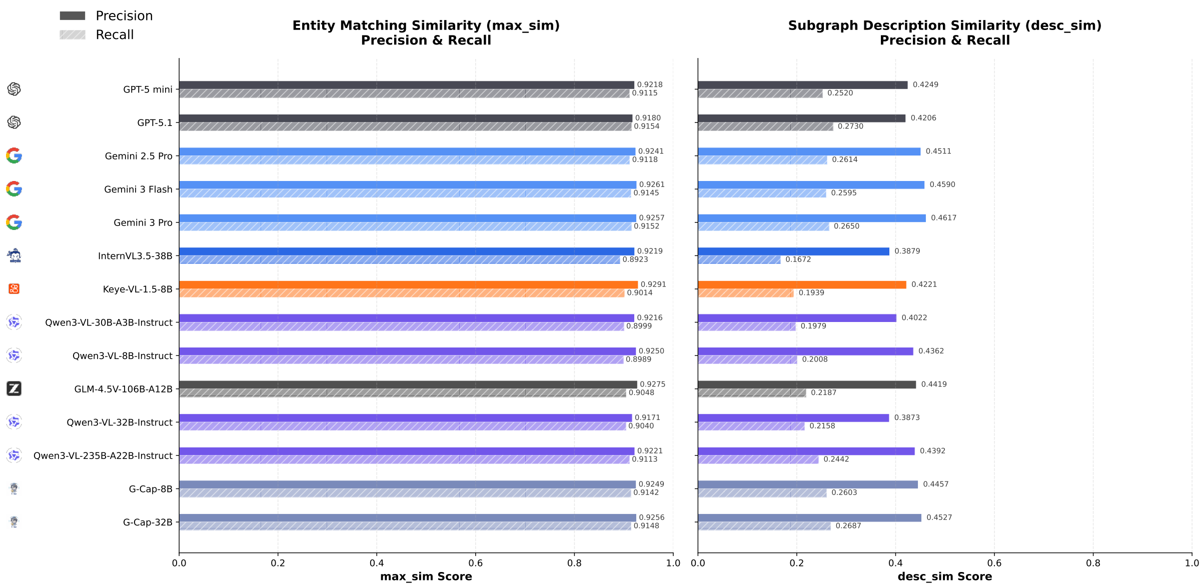


Figure 16: Decomposed evaluation results of Graph-F1. *max_sim* measures parent node matching similarity, while *desc_sim* evaluates the semantic consistency of the associated subgraphs.