

# PATENTLMM: Large Multimodal Model for Generating Descriptions for Patent Figures

Shreya Shukla<sup>1\*</sup> Nakul Sharma<sup>1\*</sup> Manish Gupta<sup>2</sup> Anand Mishra<sup>1</sup>

<sup>1</sup>IIT Jodhpur, India <sup>2</sup>Microsoft, India

{shukla.12,sharma.86}@iitj.ac.in, gmanish@microsoft.com, mishra@iitj.ac.in

## Abstract

Writing comprehensive and accurate descriptions of technical drawings in patent documents is crucial to effective knowledge sharing and enabling the replication and protection of intellectual property. However, automation of this task has been largely overlooked by the research community. To this end, we introduce PATENTDESC-355K, a novel large-scale dataset containing  $\sim 355$ K patent figures along with their brief and detailed textual descriptions extracted from 60K+ US patent documents. In addition, we propose PATENTLMM – a novel multimodal large language model specifically tailored to generate high-quality descriptions of patent figures. Our proposed PATENTLMM comprises two key components: (i) PATENTMME, a specialized multimodal vision encoder that captures the unique structural elements of patent figures, and (ii) PATENTLLAMA, a domain-adapted version of LLaMA fine-tuned on a large collection of patents. Extensive experiments demonstrate that training a vision encoder specifically designed for patent figures significantly boosts the performance, generating coherent descriptions compared to fine-tuning similar-sized off-the-shelf multimodal models. PATENTDESC-355K and PATENTLMM pave the way for automating the understanding of patent figures, enabling efficient knowledge sharing and faster drafting of patent documents. We make the code and data publicly available at <https://v12g.github.io/projects/PatentLMM/>.

## 1 Introduction

Patents are a cornerstone of intellectual property protection, granting inventors exclusive rights to their creations. Effective communication of these inventions is crucial for patent examiners, courts, and the technical community to appreciate the inventiveness of these inventions and assess their novelty. Patent documents rely heavily on figures and their corresponding textual descriptions to present technical details. Writing accurate descriptions of these figures is essential for an unambiguous understanding of the invention and its components and facilitates knowledge sharing within the technical community. Comprehensive descriptions also ensure that the invention is adequately protected against potential infringements by others. However, manually crafting

such descriptions is time-consuming and laborious, hindering the efficiency of patent processing and analysis.

One of the major challenges for generating patent figure descriptions in an automated way is the lack of large-scale labeled datasets. Existing datasets, while invaluable for advancing research in natural and scientific figure captioning, do not adequately capture the nuances and complexities inherent to patent illustrations. To address this gap, we curate PATENTDESC-355K, a novel large-scale dataset containing  $\sim 355$ K patent figures and their brief and detailed textual descriptions extracted from 60K+ patent documents. This dataset offers a rich and diverse collection of patent figures that span various technical domains, along with their corresponding descriptions, enabling the development and evaluation of models specifically tailored for this task.

Typically, patent figures are associated with brief and detailed descriptions. In our proposed PATENTDESC-355K dataset, we found that they span an average of  $\sim 34$  and  $\sim 1680$  tokens, respectively. Thus, unlike existing image captioning benchmarks, for example COCO (Lin et al. 2014), TextCaps (Sidorov et al. 2020) and NoCaps (Agrawal et al. 2019) where captions span an average of  $\sim 12$  tokens, the descriptive captioning of patent figures in our dataset is much more challenging. Moreover, unlike the natural scene images of the existing captioning datasets, patent figures are structured technical illustrations that adhere to a more standardized visual style for technical and legal documentation.

The emergence of Large Language Models (LLMs) and Large Multimodal Models (LMMs) has revolutionized almost every vision and language task. These models exhibit a remarkable ability to understand and generate coherent language across diverse domains. However, applying these models to the generation of patent descriptions presents unique challenges. The length of descriptions and the complexity inherent to patent diagrams underscore the need to focus on various elements of the figure, such as arrows, nodes, and text annotations. Further, contrary to dense document images, patent figures are sparse and comprise several elements like text, nodes, node labels (a number associated with nodes in the patent figure), figure numbers, and arrows in different styles, i.e., uni-direction and bidirectional, solid, and dotted, among others. Please refer to Fig. 3 in the Appendix for an overview of these elements.

As can be seen in Fig. 1, the detailed description of patent

\*These authors contributed equally.

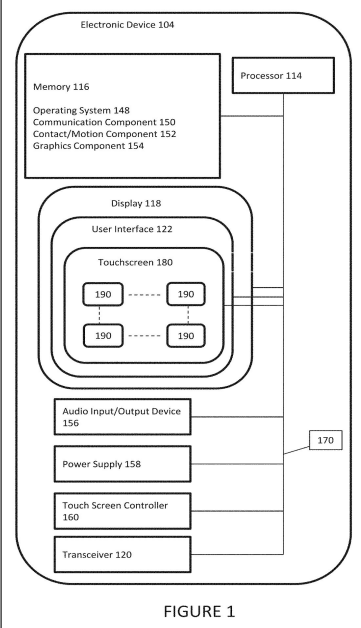
Electronic Device 104	Generated Brief and Detailed Descriptions	Ground Truth Brief and Detailed Descriptions
 <p style="text-align: center;">FIGURE 1</p>	<p>FIG. 1 illustrates a mobile smartphone in accordance with an aspect of the invention.</p> <p>Referring now to FIG. 1, an exemplary electronic device 104 is illustrated. It should be understood that the electronic device 104 may be any one of a number of electronic devices including, but not limited to, a cell phone, a smart phone, a personal digital assistant, a laptop computer, a desktop computer, a netbook computer, a server computer, etc. The electronic device 104 includes a processor 114, a memory 116, a display 118, a user interface 122, a transceiver 120, and an optional touch screen 160. The processor 114 is configured to execute instructions received from the memory 116, and is, for example, a general purpose processor, a field programmable gate array, or any other suitable processor. The processor 114 is configured to execute instructions received from the memory 116, including instructions for displaying a user interface on the display 118. The processor 114 may also be configured to execute instructions received from the transceiver 120, including instructions for receiving data from the transceiver 120 and instructions for transmitting data from the transceiver 120. The memory 116 is configured to store data received from the processor 114 and instructions to be executed by the processor 114. The memory 116 may be, for example, a random access memory device, a solid state memory device, or any other suitable storage device. The display 118 is configured to visually display data received from the processor 114. The display 118 may be, for example, a liquid crystal display, an organic light emitting diode display, a cathode ray tube display, or any other suitable display. The user interface 122 is configured to allow a user to interact with the electronic device 104. The user interface 122 may include, for example, a keypad, a touchscreen 160, a microphone, a speaker, etc. The transceiver 120 is configured to send and receive...</p>	<p>FIG. 1 shows the details of an exemplary electronic device in accordance with aspects of the invention. The electronic device 104 includes a processor 114, memory 116, display 118, user interface 122, and the like. The processor 114 may be a central processing unit configured to execute instructions including instructions related to software programs. The display 118 may be a liquid crystal display having a backlight to illuminate the various color liquid crystals to provide a colorful display. The user interface 122 may be any type of physical input having buttons and further may be implemented as a touchscreen 180. The electronic device 104 may further include in the memory 116, an operating system 148, a communication component 150, a contact/motion component 152, a graphics component 154, and the like. The operating system 148 together with the various components providing software functionality for each of the components of the electronic device 104. The memory 116 may include a high-speed random-access memory. Also, the memory 116 may be a non-volatile memory, such as magnetic fixed disk storage, flash memory or the like. These various components may be connected through various communication lines including a data bus 170. Additionally, the electronic device 104 may include an audio input/output device 156. The audio input/output device 156 may include speakers, speaker outputs, and in the like, providing sound output; and may include microphones, microphone inputs, and the like, for receiving sound inputs. The audio input/output device 156 may include and analog to digital converter and a digital to audio converter for audio input and output functions respectively. When implemented as a wireless device, the electronic device 104 may include a transceiver 120 and the like. The electronic device 104 may provide radio and...</p>

Figure 1: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

figures heavily makes use of these elements to convey the semantics of the figure. Given this dramatic difference between captions of natural scenes versus patent figures, it was anticipated that recent image captioning methods (Li et al. 2022; Wang et al. 2022a,b) and multimodal LLMs (Ye et al. 2023b; Liu et al. 2024a; Zhu et al. 2024) would perform poorly for our task in a zero-shot setting. Surprisingly, these approaches demonstrated suboptimal performance even after fine-tuning on our dataset. These unique properties of patent figures require specialized system design to ensure the accurate and concise generation of descriptions without introducing hallucinations or irrelevant details.

In this paper, we propose PATENTLMM – a novel model to generate descriptions of patent figures. The model contains two important components: PATENTMME and PATENTLLAMA. PATENTMME is a specialized multimodal vision encoder for patent figures, trained using masked language modeling loss, along with two other novel loss functions focused on learning structure from sparse patent figures. PATENTLLAMA is a domain-adapted version of LLaMA fine-tuned on a large collection of patent text from the Harvard USPTO Dataset (HUPD) (Suzgun et al. 2024). PATENTLMM combines the PATENTMME encoder and the PATENTLLAMA using a projection layer.

The major contributions of our work are as follows. (i) We present a large-scale dataset of  $\sim 355K$  patent figures and their brief and detailed descriptions. (ii) We propose a novel multimodal model PATENTLMM, comprising a patent domain-specialized vision encoder trained using objectives specifically tailored to capture the structure of patent documents and an LLM fine-tuned on patent data. (iii) We extensively benchmark existing captioning models and multimodal LLMs and show that our proposed approach surpasses their best performance on the average BLEU metric by 10.22% and 4.43% on an absolute scale for generating

brief and detailed descriptions, respectively.

## 2 Related Work

**Image Captioning in Pre-LMMs era:** The patent figure description task is broadly similar to the image captioning task, which has been an active research area in the last decade. Some representative early work on image captioning includes the combination of a CNN encoder with an LSTM decoder (Vinyals et al. 2015), a multimodal RNN architecture that uses local and global image features (Andreas et al. 2016), an adaptive attention model (Lu et al. 2017), and a bottom-up and top-down attention model (Anderson et al. 2018). Recent works have also focused on improving caption diversity (Shetty, Roumeliotis, and Laaksonen 2017), novel object captioning (Lu et al. 2018), and incorporating external knowledge (Gu et al. 2019). As discussed in the previous section, our task differs significantly from these previous efforts on image captioning in terms of the length of descriptions and the structure of patent figures.

**Describing Scientific Figures:** Patent figures are a specific form of scientific illustrations. Although previous work on generating descriptions of patent figures has been sparse, ample research has been done to caption scientific figures. Chen et al. (2019, 2020) create and leverage FigCAP and adapt an LSTM-based model (Hochreiter and Schmidhuber 1997) for captioning. Recently, Hsu, Giles, and Huang (2021) collected the SciCap dataset from articles published on arXivIn (Yang et al. 2023), the authors augment the SciCap dataset with additional information such as OCR text from figures and referring sentences from the text to curate SciCap+, and demonstrate the performance boost achieved by incorporating extra information. Kantharaj et al. (2022) and Tang, Boggust, and Satyanarayan (2023) address the problem of captioning various visualization charts

of data. Certain works go beyond natural language descriptions to generate code, particularly for flowcharts. For example, Shukla et al. (2023) and Liu et al. (2022) specifically address the generation of code from flow chart images. A parallel work PatFig (Aubakirova, Gerdes, and Liu 2023) scrapes a similar dataset as ours with 17K training samples and 2K test samples, and demonstrates the performance of MiniGPT-4 (Zhu et al. 2024) in the proposed dataset. In this work, we contribute a  $\sim 20\times$  larger dataset and propose a novel model, PATENTLMM, which is almost twice as effective as MiniGPT-4 in BLEU-4 for PATENTDESC-355K.

**Large Multimodal Models:** Recent work in the multimodal (vision and language) community has focused on leveraging the world knowledge implicitly encoded in large language models for multimodal tasks such as visual question answering and image captioning (Zhu et al. 2024; Li et al. 2023; Liu et al. 2024a; Achiam et al. 2023; Ye et al. 2023b, 2024; Wang et al. 2022b; Team et al. 2023; Alayrac et al. 2022), visual grounding (Ye et al. 2024; Zhu et al. 2024; Team et al. 2023; Achiam et al. 2023) and image-text matching (Li et al. 2022, 2023). This is achieved by feeding an image representation as input along with the prompt to the language model and modeling the output using the language modeling objective. Recent advances include Flamingo (Alayrac et al. 2022), which inserts trainable gated cross-attention layers into a pretrained LLM (Hoffmann et al. 2022). BLIP-2 (Li et al. 2023) leverages pre-trained ViT (Dosovitskiy et al. 2021) and LLaMA (Touvron et al. 2023), combined with QFormer, to translate image embeddings into LLM prompt embeddings. MiniGPT-4 (Zhu et al. 2024) builds upon pretrained BLIP-2 and finetunes an additional linear layer to project queries into the LLM on a curated dataset. In contrast, LLaVA-1.5 (Liu et al. 2024a) proposes a relatively simple and effective two-stage approach. In addition, document-specific LLMs such as LayoutLLM (Luo et al. 2024), UReader (Ye et al. 2023a) and TextMonkey (Liu et al. 2024b) have shown impressive performance on Document VQA task. We compare with several of these models and show that these models do not perform competitively for the task of generating descriptions from patent figures.

### 3 PATENTDESC-355K: A Novel Dataset of Patent Figures with Descriptions

We introduce PATENTDESC-355K – a novel large-scale dataset tailored for generating descriptions for patent figures. Our proposed dataset comprises 355K patent figures sourced from Google Patents<sup>1</sup>, with each image accompanied by its brief and detailed descriptions extracted from the corresponding patent documents. The dataset is available for download on our project website: <https://vl2g.github.io/projects/PatentLMM/>. Fig. 1 visualizes a (patent figure, brief description, detailed description) triplet from our dataset. With our primary focus on US patents published after 2004, our dataset spans over 60K patents from assignees like Amazon, Microsoft, LinkedIn, Google, Yahoo, etc. To assess the quality of the dataset, we manually evaluated a random set of 100 patent figures with their brief and detailed

<sup>1</sup><https://patents.google.com>

Table 1: PATENTDESC-355K: Dataset Statistics.

	Train	Validation	Test
Number of Images	320,717	17,286	17,336
Avg. number of tokens in brief descriptions	34.37	34.28	34.30
Avg. number of tokens in detailed descriptions	1,677.85	1,676.71	1,697.16
Number of Unique Patents	50,448	8,027	7,964
Avg. number of images per patent	6.36	2.15	2.18

descriptions and computed the sentence-level precision and recall of the extracted descriptions against the ground-truth descriptions. For brief descriptions, both precision and recall scores were 100%. For detailed descriptions, precision and recall were 90.81% and 91.96%, respectively. More details on data set curation, preprocessing, description extraction, and quality assessment are provided in Appendix A.

**Dataset Analysis:** Table 1 presents detailed statistics of the 355K image-description triplets in our dataset. During the creation of training, validation and test set splits, we ensure absolute exclusivity between patents in the train set and those in the combined validation and test sets, to enable robust out-of-sample evaluation. To achieve this, we randomly sampled  $\sim 12.6K$  patents from  $\sim 60K$ , representing  $\sim 82.5K$  images. From this isolated subset of images, we sample  $\sim 17K$  images each for the val and test set, and discard the remaining images. This sampling technique also helps maintain the diversity within the validation and test sets, thereby providing a fair and representative evaluation. Our detailed descriptions span  $\sim 1.7K$  tokens on average, which is much larger compared to an average token length for popular image captioning benchmarks (Lin et al. 2014; Chen et al. 2015; Sidorov et al. 2020).

## 4 Methodology

Our approach is inspired by the recent success of large multimodal models like MiniGPT-4 (Zhu et al. 2024) and LLaVA (Liu et al. 2023, 2024a), which have demonstrated state-of-the-art performance on several benchmarks by effectively aligning visual and textual modalities. We introduce PATENTLMM, which combines our domain-adapted version of the LLaMA language model, namely PATENTLLAMA, with our novel visual encoder specialized for patent figures, namely PATENTMME. In this section, we describe the architectures of PATENTMME and PATENTLLAMA, and the overall framework of PATENTLMM.

### 4.1 PATENTMME: Encoder for Patent Figures

The Vision Transformer (ViT) (Dosovitskiy et al. 2021), commonly used as a vision encoder in existing image captioning frameworks, is typically pre-trained on natural scene images, which are fundamentally different from patent figures. A better suited encoder is perhaps LayoutLM (Xu et al. 2020, 2021; Huang et al. 2022) which has shown impressive performance in document image understanding tasks. However, patent figures have a sparse layout compared to dense document images and are characterized by specific structured visual syntax. Unlike document images, patent figures comprise labeled nodes interconnected with arrows and accompanied by textual elements. The semantic relationship

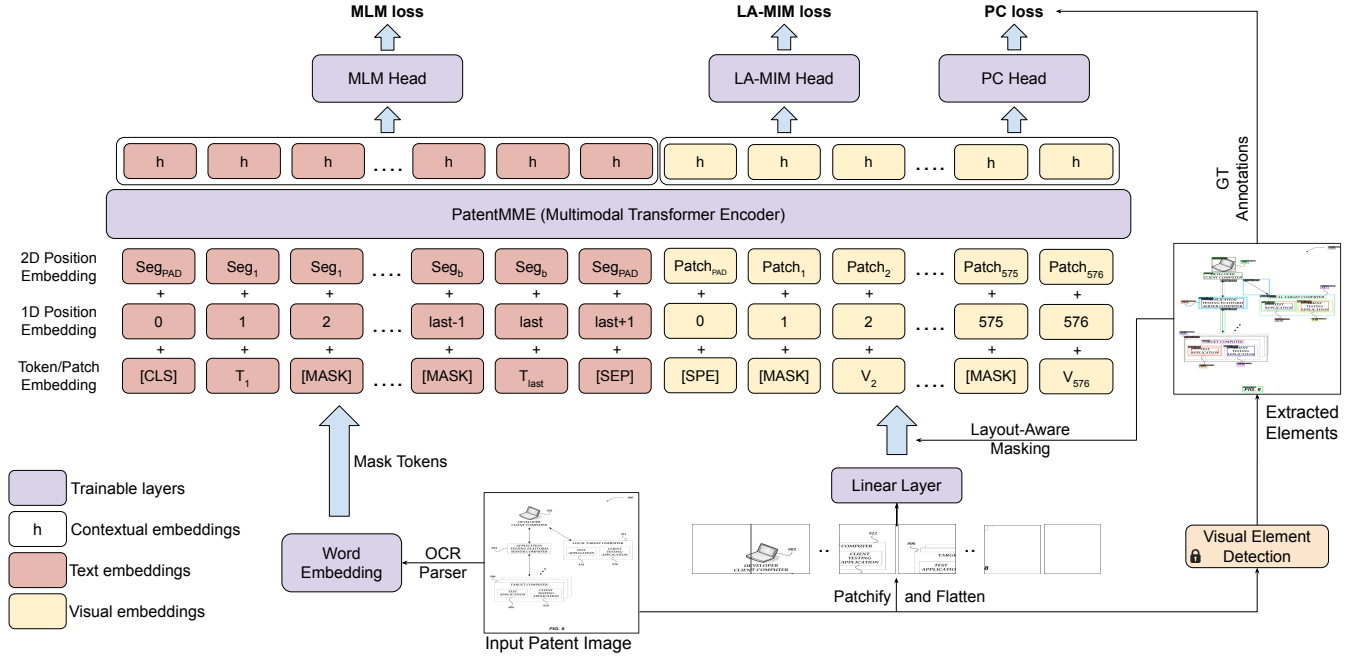


Figure 2: PATENTMME Architecture. We jointly process OCR tokens and visual embeddings to produce multimodal context-aware embeddings. These contextual embeddings are optimized using our proposed MLM, LA-MIM and PC objectives.

between these diagrammatic constituents is paramount for decoding the inventive concepts and technical specifications elucidated within the patent figures. We, therefore, build on the existing document image understanding capabilities of LayoutLMv3 (Huang et al. 2022) and pre-train it with novel objectives, specifically tailored to capture the structural information of patent figures.

**Architecture:** The proposed PATENTMME shares its architecture with LayoutLMv3 (Huang et al. 2022) and is a multi-modal transformer model that processes image, text, and document layout information jointly. The overall architecture of the model is illustrated in Fig. 2. Given an input patent figure  $I$ , the OCR text is extracted using off-the-shelf Tesseract OCR engine (Kay 2007). The image is then down-scaled to  $H \times W$  and split into non-overlapping patches of  $p$  dimensions each, resulting in  $M = HW/p^2$  image patches. The OCR extracted text is tokenized using the BPE tokenizer (Shibata et al. 1999) and represented using a learnable embedding matrix. Following (Huang et al. 2022), learnable 1D-position embeddings and 2D segment-level layout-position embeddings are added to the word embeddings, resulting in the final text embeddings. The image embeddings are created by linear projection of flattened image patches and combining them with learnable 1D position embeddings and 2D spatial embeddings. We use images of size  $I \in \mathbb{R}^{3 \times 384 \times 384}$ , i.e.,  $H = W = 384$ . With  $p = 16$  this results in  $M = 576$  patches. The higher resolution helps preserve intricate structural details of patent figures, such as node labels and arrows.

**Pre-training data and annotations:** To enable large-scale in-domain pre-training of PATENTMME, we crawled a

set of 900K+ patent figures corresponding to the patent IDs from the Harvard USPTO Patent Dataset (HUPD) (Suzgun et al. 2024). For a fair evaluation, appropriate care has been taken to avoid any overlap of the sample with the validation and testing split of our dataset.

For robust patent-figures’-specific pretraining, we define loss functions that leverage patent diagram specific elements like nodes, node labels, figure labels, text and arrows. To extract such elements, we train a Faster-RCNN (Ren et al. 2015) based visual element detection network on 350 manually annotated patent figures, sampled randomly from our training data. The trained model is then used to infer elements from all training images, which is used to provide weak ground-truth labels during PATENTMME training. We show inference samples of this model in Appendix B.

**Pre-training Loss Formulations:** To enhance the vision encoder’s capability in capturing fine-grained structural details of patent figures, we pre-train PATENTMME using novel layout-aware masked image modeling (LAMIM) and image patch classification (PC) objectives, along with the established masked language modeling (MLM) loss. We describe these losses in the following text.

**Notation:** We use  $R$  and  $T$  to denote the set of image patches (regions) and OCR tokens, respectively. Further,  $X_m$  and  $X_{um}$  denote the masked and unmasked parts of the modality  $X$ . The probability distribution generated by our PATENTMME model and the set of categories of visual elements that can be detected by our detection network by  $p_\theta$  and  $C$ , respectively.

(i) **Masked Language Modeling (MLM).** Similar to LayoutLMv3 (Huang et al. 2022), we randomly mask 30% of

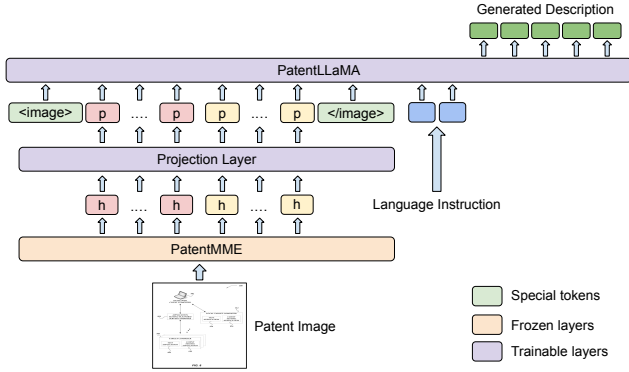


Figure 3: PATENTLMM Architecture. Language Instruction is a fixed prompt guiding the model to generate either brief or detailed descriptions.

the OCR text tokens and optimize the model to predict the masked tokens, encouraging it to learn patent-specific textual semantics. This loss is computed as follows:

$$\mathcal{L}_{MLM}(\theta) = - \sum_{i \in T_m} \log p_{\theta}(t_i | R_{um}, T_{um}), \quad (1)$$

where  $t_i$  denotes the correct masked text tokens.

(ii) **Layout-Aware Masked Image Modeling (LAMIM)**. We utilize masked image modeling to learn visual representations by randomly masking 40% of the image patches. Since the patent figures are more sparse compared to dense document images, we mask only the image patches that contain at least one of the following five elements: nodes, node labels, figure labels, text and arrows. This strategy helps to avoid masking blank regions in the patent figures, and hence learn robust visual representations. Our formulation of the LAMIM objective is similar to BEiT (Bao et al. 2022) and therefore requires a discrete image tokenizer. We choose OCR-VQGAN (Rodriguez et al. 2023) since its tokenized image representation is capable of handling textual information better than competing works dVAE (Ramesh et al. 2021) and VQGAN (Esser, Rombach, and Ommer 2021). We compute this loss as follows:

$$\mathcal{L}_{MIM}(\theta) = - \sum_{i \in R_m} \log p_{\theta}(r_i | R_{um}, T_{um}), \quad (2)$$

where  $p_i$  denotes the correct masked image patches.

(iii) **Patch Classification (PC)**. In this multi-label binary classification objective, we classify each of the  $M$  image patches into one or more of the following five categories: *node*, *node label*, *figure label*, *text*, and *arrows*. This objective which is mathematically computed as follows, helps the model learn discriminative representations for different visual elements in patent figures.

$$\mathcal{L}_{PC} = - \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^{|C|} [y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})], \quad (3)$$

where  $\hat{y}_{ij}$  denotes the probability of patch  $i$  belonging to class  $j$ , and  $y_{ij}$  is the binary ground truth label obtained using the visual element detector.

## 4.2 PATENTLLAMA: Description Generator

PATENTLLAMA is a domain-adapted version of the LLaMA-2 7B model for the patent domain. We continue to pre-train the LLaMA-2 7B model using LoRA (Hu et al. 2022) adapters, on the descriptions from HUPD patent dataset (Suzgun et al. 2024), to bias the model to generate the language inherent to patent documents. To avoid any train-test leakage, we ensure that we use the HUPD dataset after removing patent documents corresponding to the validation and test splits of our PATENTDESC-355K dataset.

## 4.3 PATENTLMM

Inspired by recent multimodal LLM studies like MiniGPT-4 (Zhu et al. 2024) and LLaVA (Liu et al. 2023, 2024a), we integrate PATENTMME and PATENTLLAMA through a single MLP network to exploit their pre-trained representations. The detailed architecture for PATENTLMM is illustrated in Fig. 3. Given a patent figure, we first obtain its layout-aware text and visual representations from frozen PATENTMME. These representations are projected into the input embedding space of PATENTLLAMA using a projection MLP, and the PATENTLLAMA is finetuned to maximize the likelihood of the corresponding description conditioned on these projected representations.

## 5 Experiments

### 5.1 Experimental Setup

**PATENTMME:** PATENTMME is initialized with LayoutLMv3-Large to inherit its document understanding capabilities. For each of the three losses discussed in Section 4, the text and image embeddings obtained from PATENTMME are projected through separate MLPs (loss heads) before the loss is calculated. Since the network weights already have a good initialization, to prevent major changes in weights of the multimodal transformer, we adopt two-step training. During Step-1, the weights of the multimodal transformer remain frozen and only the loss heads are trained for 1 epoch with a higher learning rate of  $1e-3$  and 1K warm-up steps to learn good initialization. During Step 2, the entire model is trained end-to-end for 8 epochs with a lower learning rate of  $5e-5$  and with 10K warm-up steps. The PATENTMME model is trained on  $8 \times V100$  GPUs, with an effective batch size of 64 and Adam (Kingma and Ba 2014) optimizer.

**PATENTLMM:** Following the standard practice (Liu et al. 2024a), we train our PATENTLMM model in two stages. To align the patent figure representations obtained from PATENTMME with the input latent space of PATENTLLAMA, we train only the projection layer in the first stage, keeping all other parameters frozen. During stage 2, we add LoRA adapters to all the linear layers of the PatentLLaMA module, except for the language modeling head, whose weights remain frozen. The weights of PATENTMME are kept frozen throughout. We train our PATENTLMM with an effective batch size of 192 on  $3 \times A100$  GPUs (40 GB). Stage 1 training progresses at a higher learning rate of  $1e-3$ , and stage 2 training takes place

Table 2: Quantitative results on PATENTDESC-355K (test set) for brief and detailed description generation (B=BLEU, R=ROUGE, M=METEOR). Number in parenthesis under #Parameters column denote number of trainable parameters.

Setup	Method	# Parameters	B-2	B-4	Avg. B	R-1	R-2	R-L	M	B-2	B-4	Avg. B	R-1	R-2	R-L	M
			Brief						Detailed							
Zero-shot	BLIP-2	2.7B	1.01	0.03	1.62	15.43	1.70	12.47	6.72	0.00	0.00	0.01	3.24	0.49	2.84	1.00
	TextMonkey	9.8B	0.91	0.11	1.12	13.00	4.60	12.16	7.14	0.10	0.03	0.10	6.18	2.38	4.83	2.64
	PEGASUS	568M	3.18	0.13	4.20	14.68	2.33	11.46	13.24	0.86	0.04	1.12	12.26	1.96	9.47	5.18
	mPLUG-owl2	7.5B	3.64	0.36	4.47	21.47	5.10	19.41	13.07	3.65	0.49	3.40	23.75	5.39	14.85	11.83
	UReader	7.2B	3.54	0.35	4.50	20.90	4.56	17.85	13.45	0.05	0.01	0.06	5.15	1.54	4.49	2.04
	LLaVA-1.5	7.4B	4.52	0.24	4.71	17.59	3.27	14.63	15.74	3.65	0.37	3.36	23.75	4.63	14.71	11.69
	GPT-4V	Unknown	20.74	8.56	18.68	36.07	15.65	31.89	32.88	19.61	6.05	18.26	39.95	12.14	20.16	27.31
Finetuned	Pegasus	568M	2.44	0.14	4.03	13.86	1.55	11.52	11.62	5.80	0.41	6.33	19.28	2.24	15.27	12.11
	GIT	681M	26.95	15.33	24.78	45.28	27.17	42.29	44.27	6.33	1.18	6.23	13.66	3.17	10.87	10.68
	BLIP	252M	24.62	12.52	22.40	42.59	23.78	39.16	42.84	5.45	1.05	5.31	12.42	2.89	9.46	9.55
	MiniGPT-4	7.8B (3.2M)	30.57	17.96	28.13	43.53	25.33	40.35	43.03	11.01	2.81	10.26	28.91	6.23	15.67	16.65
	OFA	472M	33.01	21.76	31.24	54.26	37.94	51.47	44.89	15.76	7.23	14.93	33.20	13.70	22.89	21.17
	LLaVA-1.5	7.4B (341M)	36.64	25.00	34.37	48.92	32.01	45.87	48.23	20.90	11.12	19.81	36.86	15.68	24.48	24.71
	PATENTLMM	7.4B (341M)	<b>46.40</b>	<b>36.66</b>	<b>44.59</b>	<b>56.68</b>	<b>42.63</b>	<b>54.18</b>	<b>56.44</b>	<b>25.42</b>	<b>15.02</b>	<b>24.24</b>	<b>40.70</b>	<b>19.27</b>	<b>27.54</b>	<b>28.39</b>

at a learning rate of  $2e-4$  with a cosine schedule, for 12K steps using Adam optimizer. We train separate LMMs for brief and detailed descriptions.

Overall, training PATENTLMM is a three-phase process. Firstly, we train the PATENTMME encoder in a semi-supervised fashion by leveraging a vast amount of patent figures corresponding to patents in the HUPD dataset. Secondly, we domain-adapt the LLaMA-2 7B model on the HUPD patent text data to create PATENTLLAMA. Lastly, we integrate PATENTMME and PATENTLLAMA to create PATENTLMM, and train it following the two-stage process.

## 5.2 Baselines

We benchmark the performance of various baselines on our proposed PATENTDESC-355K dataset in the zero-shot and fine-tuned setup. We benchmark the text-only baseline Pegasus (Zhang et al. 2020) by generating patent figure descriptions from OCR tokens extracted from patent figures. For image captioning baselines, we study the state-of-the-art models GIT (Wang et al. 2022a), BLIP (Li et al. 2022) and OFA (Wang et al. 2022b). We further compare our method with recent multimodal LLMs such as UReader (Ye et al. 2023a), TextMonkey (Liu et al. 2024b), mPLUG-owl2 (Ye et al. 2024), BLIP-2 (Li et al. 2023), MiniGPT-4 (Zhu et al. 2024), LLaVA-1.5 (Liu et al. 2024a) and the closed GPT-4V model (Achiam et al. 2023). GPT-4V prompt is listed in Appendix C.3.

To measure the description generation performance of these models, we use standard image captioning metrics such as BLEU (Papineni et al. 2002), ROUGE (Lin 2004) and METEOR (Banerjee and Lavie 2005). Higher values for all the scores are desired. A detailed description of these metrics is provided in Appendix C.1.

## 5.3 Results and Discussion

The quantitative performance comparison for the brief and detailed description generation task is reported in Table 2. In the zero-shot setting, GPT-4V demonstrated superior performance among baselines across all metrics, significantly

outperforming other baselines owing to its large scale and the diverse data it has seen during its pre-training. The poor zero-shot performance of other baselines highlights the gap in their pre-training data and the nature of patent figures and descriptions. The fine-tuned models outperform their zero-shot counterparts, highlighting the importance of task-specific training for these models. MiniGPT-4 and LLaVA-1.5 utilize a frozen pre-trained ViT trained on web-scale natural images, which results in suboptimal representation of patent figures. Similarly, OFA also enforces these priors by utilizing a pre-trained discrete image tokenizer. On the other hand, PATENTLMM gives a boost of  $\sim 8\%$  across all metrics, signifying the importance of better domain knowledge embedded in it through the proposed PATENTMME pretraining and PatentLLaMA.

Similar to brief description generation, GPT-4V outperformed all other baselines for the detailed description generation task in the zero-shot setting. We observe that majority of the baselines struggle with performance in the zero-shot setup. In the fine-tuned setting, our PATENTLMM maintained its superior performance, achieving the highest scores across all metrics. This consistent top performance for both brief and detailed descriptions suggests the efficacy of our proposed approach for the task of generating descriptions from patent figures. The overall lower scores for detailed descriptions can be attributed to their comprehensiveness, complexity, and length, requiring models to capture and generate more nuanced and detailed information.

**Ablations:** We perform the following three ablation studies to quantify the impact of different components of our proposed PATENTLMM model:

(i) **PATENTMME Pre-training objectives:** Table 3 shows the ablation results with combinations of pre-training objectives for the brief description generation. We observe that using a combination of MLM and LAMIM leads to better results compared to the pre-trained LayoutLMv3. Further, the PC loss also improves the performance of the model, when pre-trained with HUPD images data. A similar ablation for

Table 3: Ablation study to quantify the impact of pre-training objectives of PATENTMME on the overall performance of PATENTLMM on brief descriptions generation task. All models are trained with PATENTLLAMA.

Pre-training	B-2	B-4	Avg. B	R-1	R-2	R-L	M
Pretrained LayoutLMv3	42.81	32.50	40.86	53.68	38.88	51.07	53.34
w/ MLM + LAMIM	45.24	35.33	43.39	55.69	41.38	53.20	55.34
w/ MLM+LAMIM+PC	<b>46.39</b>	<b>36.65</b>	<b>44.59</b>	<b>56.68</b>	<b>42.62</b>	<b>54.18</b>	<b>56.44</b>

Table 4: Ablation study to quantify the importance of OCR tokens on the overall performance of PATENTLMM on brief descriptions generation task.

OCR in training?	OCR in Inference?	B-2	B-4	Avg. B	R-1	R-2	R-L	M
No	No	30.32	19.17	28.30	41.61	25.21	38.95	41.46
Yes	No	11.51	2.77	9.83	24.38	7.92	21.68	22.52
Yes	Yes	<b>46.40</b>	<b>36.66</b>	<b>44.59</b>	<b>56.68</b>	<b>42.63</b>	<b>54.18</b>	<b>56.44</b>

detailed descriptions is reported in Appendix C.2.

**(ii) Importance of OCR tokens:** In this ablation, we study whether avoiding passing OCR tokens to PATENTLMM causes any drop in the performance of brief description generation. We experiment with two ablations: (1) OCR tokens are used for PATENTMME pretraining but not for PATENTLMM training, and (2) OCR tokens are used for PATENTMME pretraining and for PATENTLMM training but not at inference time. Table 4 shows that it is important to use OCR tokens in the entire pipeline for the best results.

**(iii) PatentLMM Training:** We report an additional ablation study in Appendix C.2 to quantify the advantage of using PatentLLaMA against the pre-trained LLaMA model.

**Qualitative Analysis:** Fig. 1 shows an example brief and detailed description generated by PATENTLMM for a test sample. The generated brief description, more specifically, terms the electronic device shown in the image as a mobile smartphone. The generated detailed description provides a comprehensive overview of the electronic device 104, its components, and their functions. It covers most of the key elements mentioned in the ground truth, including the processor 114, the memory 116, the display 118, and the user interface 122. However, there are some omissions, like Graphics Component 154 and Communication Component 150. More case studies are provided in Appendix D.

**Error analysis:** We perform a thorough manual error analysis on a set of 50 samples drawn from our test set to identify some prominent errors in the descriptions generated by our PATENTLMM model. We identify five main error categories as follows. (i) Hallucination in figure labeling occurs in 3 brief and 3 detailed descriptions. (ii) Hallucination in 4 brief descriptions and 7 detailed descriptions was due to little or no OCR detectable text in the figures. (iii) Incorrect association of node labels occurs when the wiggly arrows connecting node labels to respective nodes are misinterpreted or ignored due to downsampling of the image before being passed to PATENTMME. This was observed in 10 detailed descriptions. (iv) A similar misinterpretation due to down-

Table 5: GPT-4V evaluation on a set of 1K samples.

Description	Method	Rel.	Acc.	Compl.	Coh.	Fluency	Cover.
Brief	LLaVA-1.5	1.38	1.06	1.01	1.85	1.98	0.98
	Ours	1.44	1.18	1.17	1.91	2.00	1.15
Detailed	LLaVA-1.5	0.75	0.75	0.73	1.07	1.69	0.71
	Ours	0.90	0.78	0.76	1.15	1.85	0.75

sampling is often the cause of hallucinated node labels in 12 detailed descriptions. (v) Cross-figure references in the descriptions establish the interconnection between various aspects and provide a complete picture of the presented technical invention. The figures may be related hierarchically (systems vs components), sequentially (steps of a process), different views (top-bottom-left-right), or in other ways. Since we train PATENTLMM to generate descriptions for individual patent figures, our model hallucinates the cross-figure references for 2 brief and 5 detailed descriptions. Qualitative examples are presented in Appendix D.2.

**GPT-4V Evaluation Results:** Apart from small-scale manual error analysis, we utilize the GPT-4V model to qualitatively evaluate the performance of LLaVA-1.5 and our proposed PATENTLMM model on the brief and detailed description generation task for a set of 1000 samples. We input the GPT-4V model with the patent figure, the ground truth description and the description generated using these models, along with the special instruction prompt. The instruction prompt instructs the GPT-4V model to rate the generated description on the following criterion: Relevance, Accuracy, Completeness (with respect to input image), Fluency and Coverage (with respect to input image and ground truth description) on an integer scale of 0 to 2. To mitigate randomness in scores, we set the temperature parameter to 0 for the GPT-4V model and created five versions of the instruction prompt. The scores obtained from each of the prompts for each criterion are then averaged. Table 5 shows that our system generates high-quality results.

## 6 Conclusion and Future Work

Our work addresses the existing gap in the automated generation of patent figure descriptions by introducing PATENTDESC-355K, a comprehensive dataset of patent figures and their corresponding brief and detailed descriptions. We further proposed PATENTLMM, a large multi-modal model comprising a domain-specialized image encoder PATENTMME and a domain-adapted patentLLaMA model for generating brief and detailed descriptions from patent figures. Extensive experiments demonstrated that our proposed PATENTLMM outperforms competent baselines by significant margins. Future research in this direction can explore experiments with patents in multiple languages, patent document-level reasoning to allow for cross-figure references while generating descriptions, incorporating external knowledge bases from technical domains to improve the performance of detailed description generation, and generation of grounded descriptions.

**Acknowledgements:** This work was supported by the Microsoft Academic Partnership Grant (MAPG) 2023.



## References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Agrawal, H.; Desai, K.; Wang, Y.; Chen, X.; Jain, R.; Johnson, M.; Batra, D.; Parikh, D.; Lee, S.; and Anderson, P. 2019. Nocaps: Novel object captioning at scale. In *ICCV*.
- Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hassan, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. In *NeurIPS*.
- Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.
- Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In *CVPR*.
- Aubakirova, D.; Gerdes, K.; and Liu, L. 2023. PatFig: Generating Short and Long Captions for Patent Figures. In *ICCVW*.
- Banerjee, S.; and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL Workshop*.
- Bao, H.; Dong, L.; Piao, S.; and Wei, F. 2022. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*.
- Chen, C.; Zhang, R.; Kim, S.; Cohen, S.; Yu, T.; Rossi, R.; and Bunescu, R. 2019. Neural Caption Generation over Figures. In *UbiComp/ISWC*.
- Chen, C.; Zhang, R.; Koh, E.; Kim, S.; Cohen, S.; and Rossi, R. 2020. Figure Captioning with Relation Maps for Reasoning. In *WACV*.
- Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.
- Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming Transformers for High-Resolution Image Synthesis. In *CVPR*.
- Gu, J.; Wang, J.; Cai, J.; and Jiang, H. 2019. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *CVPR*.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780.
- Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de las Casas, D.; Hendricks, L. A.; Welbl, J.; Clark, A.; Hennigan, T.; Noland, E.; Millican, K.; van den Driessche, G.; Damoc, B.; Guy, A.; Osindero, S.; Simonyan, K.; Elsen, E.; Vinyals, O.; Rae, J. W.; and Sifre, L. 2022. An empirical analysis of compute-optimal large language model training. In *NeurIPS*.
- Hsu, T.-Y.; Giles, C. L.; and Huang, T.-H. 2021. SciCap: Generating Captions for Scientific Figures. In *EMNLP 2021 (Findings)*.
- Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.
- Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *ACM-MM*.
- Kantharaj, S.; Leong, R. T.; Lin, X.; Masry, A.; Thakkar, M.; Hoque, E.; and Joty, S. 2022. Chart-to-Text: A Large-Scale Benchmark for Chart Summarization. In *ACL*.
- Kay, A. 2007. Tesseract: an open-source optical character recognition engine. *Linux J.*, 2007(159): 2.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.
- Li, J.; Li, D.; Xiong, C.; and Hoi, S. C. H. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*.
- Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.
- Lin, T.-Y.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.
- Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved baselines with visual instruction tuning. In *CVPR*.
- Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. *ArXiv*, abs/2304.08485.
- Liu, Y.; Yang, B.; Liu, Q.; Li, Z.; Ma, Z.; Zhang, S.; and Bai, X. 2024b. TextMonkey: An OCR-Free Large Multimodal Model for Understanding Document. *arXiv preprint arXiv:2403.04473*.
- Liu, Z.; Hu, X.; Zhou, D.; Li, L.; Zhang, X.; and Xiang, Y. 2022. Code Generation From Flowcharts with Texts: A Benchmark Dataset and An Approach. In *EMNLP (Findings)*, 6069–6077.
- Lu, J.; Xiong, C.; Parikh, D.; and Socher, R. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*.
- Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2018. Neural baby talk. In *CVPR*.
- Luo, C.; Shen, Y.; Zhu, Z.; Zheng, Q.; Yu, Z.; and Yao, C. 2024. LayoutLLM: Layout Instruction Tuning with Large Language Models for Document Understanding. In *CVPR*.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.
- Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*.



- Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*.
- Rodriguez, J. A.; Vazquez, D.; Laradji, I.; Pedersoli, M.; and Rodriguez, P. 2023. OCR-vqgan: Taming text-within-image generation. In *WACV*.
- Shetty, R.; Roumeliotis, G.; and Laaksonen, J. 2017. Speaking the same language: Matching machine to human captions for image captioning. In *ICCV*.
- Shibata, Y.; Kida, T.; Fukamachi, S.; Takeda, M.; Shinohara, A.; and Shinohara, T. 1999. Byte Pair Encoding: A Text Compression Scheme That Accelerates Pattern Matching. *ResearchGate*.
- Shukla, S.; Gatti, P.; Kumar, Y.; Yadav, V.; and Mishra, A. 2023. Towards Making Flowchart Images Machine Interpretable. In *ICDAR*.
- Sidorov, O.; Hu, R.; Rohrbach, M.; and Singh, A. 2020. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*.
- Suzgun, M.; Melas-Kyriazi, L.; Sarkar, S.; Kominers, S. D.; and Shieber, S. 2024. The harvard uspto patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications. In *NeurIPS*.
- Tang, B.; Boggust, A.; and Satyanarayan, A. 2023. VisText: A Benchmark for Semantically Rich Chart Captioning. In *ACL*.
- Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *ArXiv*, abs/2302.13971.
- Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.
- Wang, J.; Yang, Z.; Hu, X.; Li, L.; Lin, K.; Gan, Z.; Liu, Z.; Liu, C.; and Wang, L. 2022a. GIT: A Generative Image-to-text Transformer for Vision and Language. *Transactions on Machine Learning Research*.
- Wang, P.; Yang, A.; Men, R.; Lin, J.; Bai, S.; Li, Z.; Ma, J.; Zhou, C.; Zhou, J.; and Yang, H. 2022b. OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. In *ICML*.
- Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *KDD*, 1192–1200.
- Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; et al. 2021. LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In *ACL-IJCNLP*.
- Yang, Z.; Dabre, R.; Tanaka, H.; and Okazaki, N. 2023. SciCap+: A Knowledge Augmented Dataset to Study the Challenges of Scientific Figure Captioning. *ArXiv*, abs/2306.03491.
- Ye, J.; Hu, A.; Xu, H.; Ye, Q.; Yan, M.; Xu, G.; Li, C.; Tian, J.; Qian, Q.; Zhang, J.; et al. 2023a. UReader: Universal OCR-free Visually-situated Language Understanding with Multimodal Large Language Model. In *EMNLP 2023 (Findings)*.
- Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; et al. 2023b. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.
- Ye, Q.; Xu, H.; Ye, J.; Yan, M.; Hu, A.; Liu, H.; Qian, Q.; Zhang, J.; and Huang, F. 2024. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *CVPR*.
- Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *ICML*.
- Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2024. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. In *ICLR*.

# Supplementary material for PATENTLMM: Large Multimodal Model for Generating Descriptions for Patent Figures

## A PATENTDESC-355K

### A.1 Dataset Curation

To create a comprehensive dataset, we crawled a diverse set of over 90K US patent documents published between 1900 and January 2023. This involved searching for various companies on Google Patents and downloading respective CSV files, with detailed relevant patent metadata like ID, assignee, publication date, patent URLs, etc. We use the patent-ids and URLs to download the HTML documents. We parsed these HTML documents to extract the image URLs, and downloaded  $\sim 900\text{K}$  images, ensuring a rich and representative corpus of technical illustrations. In the following sections, we describe the preprocessing and description extraction stages:

**Pre-processing of Patent Images** We manually analyzed a random set of 500 patent images from our collection, and encountered considerable noise attributed to the vast diversity. To alleviate this noise, we implemented a series of filtering steps as follows.

1. **Correcting Image Orientation:** We identified that  $\sim 40\%$  of the analysed images were vertically oriented. To rectify this automatically, we compared the average length of OCR tokens extracted using PaddleOCR (Du et al. 2021) for the original image and the  $90^\circ$ -rotated image, and saved the image with greater average OCR length.
2. **Redundancy Removal:** We eliminated the first occurrence of representative figure images, which were repeated twice for each patent.
3. **Discarding Multi-Figure Images:** Around  $7\%$  of the analysed images had multiple figures per image. To maintain a focus on singular representative visuals per image, we extract figure labels using PaddleOCR (Du et al. 2021). Then, we remove a small proportion of images containing multiple occurrences of figure labels.
4. **Graph/Plot/Chart Removal:** Around  $5\%$  images in our analysed data depicted graphical plots. Prior works (Tang, Boggust, and Satyanarayan 2023; Mahinpei, Kostic, and Tanner 2022; Masry et al. 2022; Liu et al.

2023) have studied their captioning in detail through specialized handling, so we discard these images by training a ResNet-50-based binary classifier on a subset of 300 manually annotated images, achieving a 98% validation accuracy.

5. **Publication Date Filtering:** We observed a specific convention in HTML tags for patents published after 2004. So, to ensure consistency in HTML tags for easier description extraction, we discarded patents published before 2005. This resulted in our final set of patents published from Jan 2005 to Jan 2023.

After these image-based pre-processing steps, we end up with a final set of  $\sim 429\text{K}$  images corresponding to  $\sim 64\text{K}$  unique patents.

**Extracting Descriptions of Patent Figures** We obtain the brief and detailed descriptions for each image from the corresponding patent HTML document as follows. We first extract figure labels from the images utilizing PaddleOCR (Du et al. 2021). Next, we extract the content within the *brief-description-of-drawings* tag in the HTML. Within this tag, there is a child *description-line* or *description-paragraph* tag corresponding to every image. Hence, for each image, using its figure label, we extract the text enclosed within the corresponding *description-line* or *description-paragraph* tag as its brief description.

For detailed descriptions, we consider the paragraphs falling after the *brief-description-of-drawings* tag. For each new *description-line*/*description-paragraph* tag, we check if its first sentence contains the *figref* tag. If yes, the tag text is attributed to the referred figure, else we append the tag text to the previously referred figures' description. In rare cases, if the OCR extracts incorrect figure labels, we cannot obtain descriptions for such images and hence we exclude such images from our dataset. Overall, this leads to our final dataset with  $\sim 355\text{K}$  images spanning  $\sim 60\text{K}$  patent documents.

### A.2 Quality Assessment for the Proposed PATENTDESC-355K

To assess the quality of our automatic description extraction heuristics, we manually annotated a random set of 100 patent images with their brief and detailed descriptions and computed the sentence-level precision and recall of the extracted descriptions against the ground-truth descriptions.

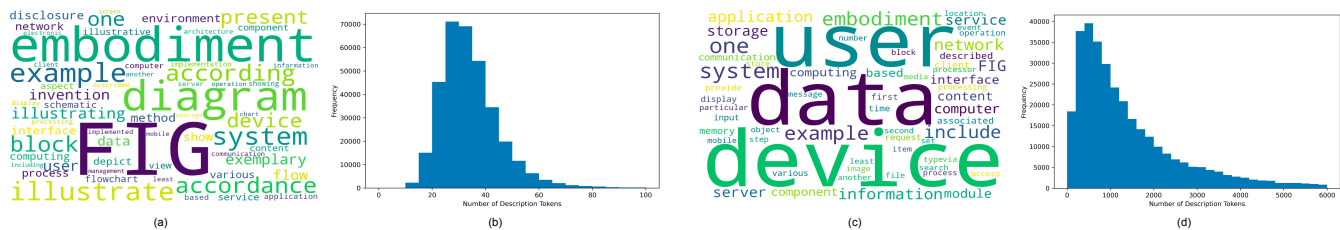


Figure 1: PATENTDESC-355K Analysis. (a) and (c): Word clouds of most common occurrences in brief and detailed descriptions respectively. (b) and (d): frequency distribution of description lengths for the brief and detailed descriptions respectively.

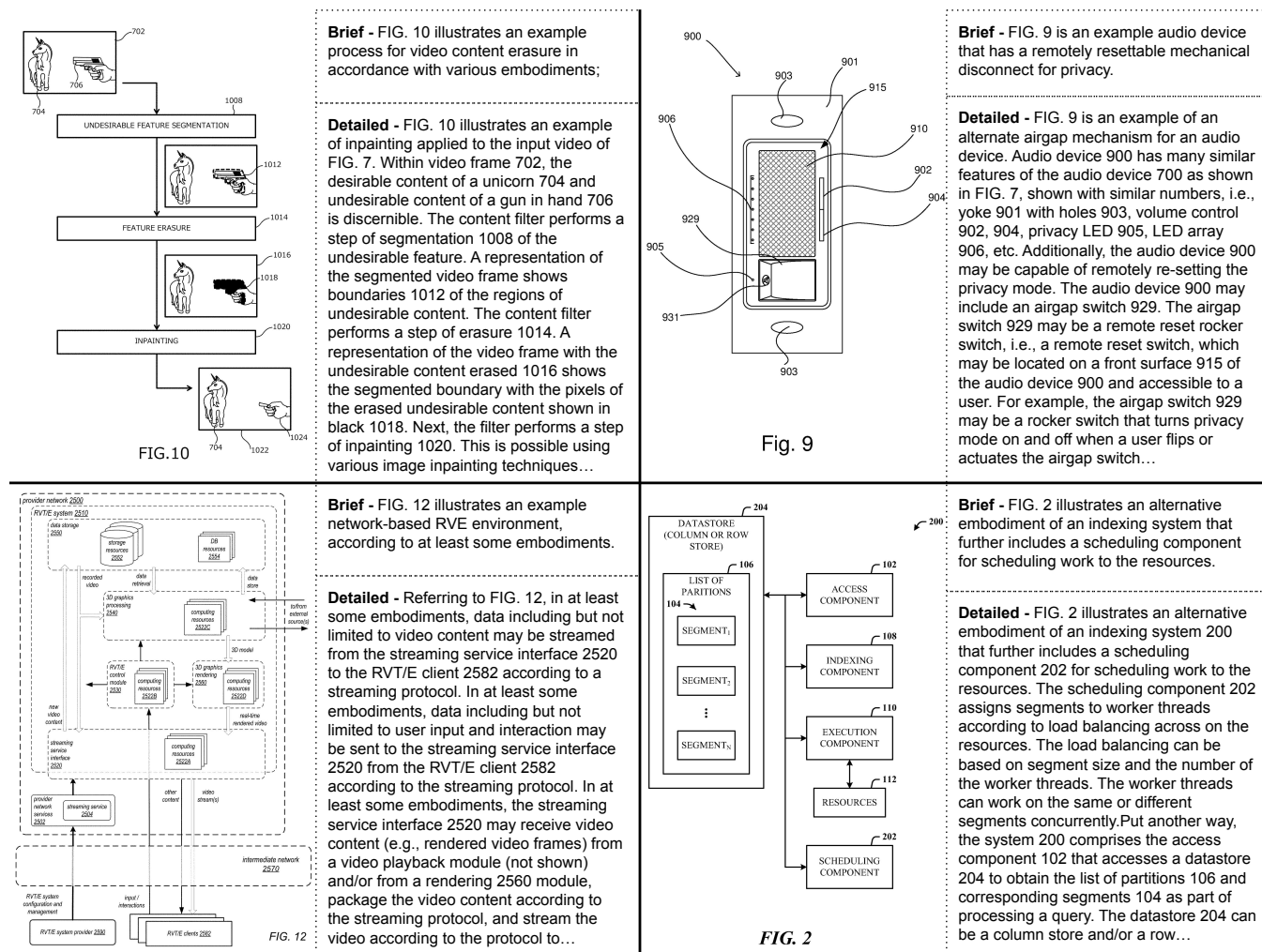


Figure 2: Samples from our PATENTDESC-355K dataset showing (patent figure, brief description, detailed description) triplets.

For brief descriptions, both precision and recall scores were 100%. This is expected since brief descriptions span single tags, making their rule-based extraction and OCR-based matching almost error-free. For detailed descriptions, the extracted descriptions match with the ground truth with a precision of 90.81% and a recall of 91.96%. Through manual analysis, we identified two primary reasons for the lower scores on detailed descriptions: (a) for the last referred figure in the HTML, sometimes the description includes con-

cluding paragraphs not specifically relevant to the figure, reducing the precision of the descriptions for such figures; (b) some sentences in the description contain references to multiple figures. In this paper, we study the problem of generating descriptions for individual figures. Hence, we discard such sentences, leading to slightly lower recall for those figures. Finally, to establish a consistent evaluation framework across baselines with varying context windows, we clip the detailed descriptions to 500 tokens.

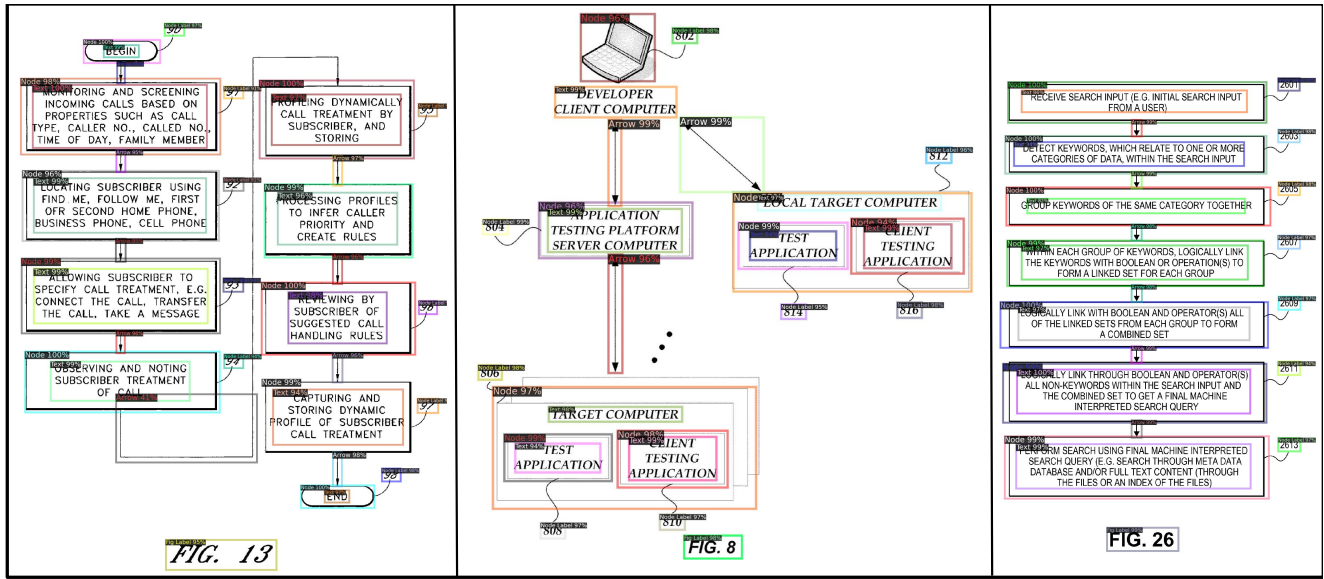


Figure 3: Annotations of patent image elements obtained using our visual element detection model on a selection of test samples.

Overall, the quality of the proposed dataset is robust, with high precision and recall scores for brief descriptions and slightly lower but still strong scores for detailed descriptions.

### A.3 Dataset Analysis and Examples

Fig. 1(a) and (c) illustrate the word clouds of the most frequent words in brief and detailed descriptions, respectively. Further, Fig. 1(b) and (d) show the frequency distribution of description lengths for the brief and detailed descriptions, respectively.

Fig. 2 shows a selection of example samples – (patent figure, brief description, detailed description) from our PATENTDESC-355K dataset.

## B Details on patent Visual Element Detector

We manually annotated 400 patent images sampled randomly from our training data. The annotations were in the form of bounding boxes for the following five categories: nodes, node labels, text, arrows, and figure labels. We split this dataset into a training split consisting of 350 samples and a test split consisting of 50 samples. We then use the training split to fine-tune a F-RCNN (Ren et al. 2015) model with ResNet101 (He et al. 2016) backbone, pre-trained on the MS-COCO dataset. We finetune this model with a learning rate of  $1e-4$  until convergence. This helps us obtain an AP@50 score of 92.52, and an AP@75 score of 64.34 on the test set. We show a few examples of the annotations obtained using the trained network in Fig. 3.

## C Additional Experiments and Details

### C.1 Evaluation Metrics

To measure the description generation performance, we use the standard image captioning metrics such as BLEU (Pa-

pineni et al. 2002), ROUGE (Lin 2004) and METEOR (Banerjee and Lavie 2005). Higher values for all the scores are desired.

**BLEU-n** (Papineni et al. 2002) calculates the n-gram overlap between the generated and reference texts, taking into account precision with which captions are generated, and a brevity penalty for shorter texts. We report BLEU-2, BLEU-4 and Avg. BLEU for all our experiments. The Avg. Blue score averages BLEU-1, BLEU-2, BLEU-3 and BLEU-4 metric.

**ROUGE** (Lin 2004) also measures the overlap between the generated and reference texts. ROUGE-N measures N-gram overlap while ROUGE-L measures the overlap based on the longest common subsequence between the generated and reference texts. We report ROUGE-1, ROUGE-2 and ROUGE-L for our experiments.

**METEOR** (Banerjee and Lavie 2005) is computed based on the explicit word-to-word matches between the generated and reference texts. It considers not only exact word matches but also stem, synonym, and paraphrase matches, as well as applies weighted penalties for incorrect word order.

### C.2 Additional Ablations

**PATENTLMM Training:** Table 1 shows the ablation results with different design choices for training of the decoder LLM of PATENTLMM for the brief description generation task. We show results without stage 2 training of the PATENTLMM (rows 1 and 3). We also show results when the decoder LLM is initialized using LLaMA-2 versus PATENTLLAMA. We observe that stage-1 training is clearly not enough and the decoder LLM needs to be finetuned for the patent description task to generate reasonable descriptions. We also observe that using PATENTLLAMA leads to significantly better results compared to just using the standard LLaMA-2 model. This shows that domain adaptive pre-

Table 1: Ablation study to quantify the impact of the decoder LLM on the overall performance of PATENTLMM on brief descriptions generation task. We report the results of PATENTLMM with PatentMME vision encoder, at both stages of training.

Stage	LLM Init.	B-2	B-4	Avg. B	R-1	R-2	R-L	M
1	LLaMA-2	1.06	0.02	1.58	6.70	0.86	5.84	10.04
2	LLaMA-2	43.54	33.50	41.66	54.35	39.76	51.81	53.82
1	PATENTLLAMA	1.08	0.02	1.61	7.08	0.87	5.99	10.32
2	PATENTLLAMA	<b>46.39</b>	<b>36.65</b>	<b>44.59</b>	<b>56.68</b>	<b>42.62</b>	<b>54.18</b>	<b>56.44</b>

Table 2: Ablation study to quantify the impact of pre-training objectives of PATENTMME on the overall performance of PATENTLMM on detailed descriptions generation task. All models are trained with PATENTLLAMA.

Pre-training	B-2	B-4	Avg. B	R-1	R-2	R-L	M
Pretrained LayoutLMv3	23.84	13.84	22.73	39.28	18.06	26.44	27.04
w/ MLM+LAMIM+PC	<b>25.42</b>	<b>15.02</b>	<b>24.24</b>	<b>40.70</b>	<b>19.27</b>	<b>27.54</b>	<b>28.39</b>

training with HUPD patent text is important for better performance.

**Pre-training ablation of PatentMME for detailed description:** In Table 2, we observe that using a combination of MLM, LAMIM and PC losses leads to better results compared to the pretrained LayoutLMv3. Thus, domain specific pre-training with HUPD image data is helpful even for detailed description generation.

### C.3 Evaluation of GPT-4V as a baseline

We utilize the following prompt to evaluate the patent description generation capabilities of the GPT-4V model as one of the baselines in the zero-shot setting.

**System Prompt:** You have been hired to draft patents for the world’s largest companies. Your primary task is to help your company in drafting patent documents.

**User Prompt:** You will be provided with a figure which will be part of a new patent that your company is planning to file. Your task is to generate a brief description for the patent figure and it will be used as a part of the patent document, and will be included under the ‘BRIEF DESCRIPTION OF THE DRAWINGS’ section of the patent document.

IMAGE:#image\_url#

Output the brief description in <results></results> tag.

Please note that the provided prompt generates brief descriptions. For detailed descriptions, we simply replace all occurrences of “brief” with “detailed” in the prompt.

### C.4 Using GPT-4V for qualitative evaluation

For evaluating the quality of the brief and detailed descriptions generated by our PATENTLMM model, we utilize GPT-4V as an evaluator. More specifically, we provide the GPT-4V model with the patent image, the ground truth description, and the description generated using our model. These inputs are accompanied by a prompt that instructs the GPT-4V model to rate the generated description against the ground truth by giving it an integer score from 0, 1, 2. We use the following prompt to accomplish this:

**System Prompt:** You are a helpful legal assistant that has been hired to quantitatively evaluate the quality of brief descriptions of patent figures generated from a black-box system, against the original brief description and the image of the patent figure.

**User Prompt:** You are acting as a critical assistant to a patent examiner. We have developed a model that creates both detailed and brief descriptions from images, and we seek your expertise to evaluate the quality of these generated outputs. Your evaluation should focus on the following criteria:

**Relevance:** The degree to which the description corresponds with the content of the image.

**Accuracy:** The correctness of the specific details provided in the description.

**Completeness:** The extent to which the description addresses all significant elements of the image.

**Coherence:** The logical consistency, clarity, and readability of the description.

**Fluency:** The grammatical and stylistic quality of the text, ensuring it reads smoothly.

**Coverage:** Whether the generated description adequately covers the essential concepts from the ground truth text or image.

You will be given the image, the reference ground truth description, and the description generated by the model. Please provide an integer score of 0, 1, or 2 for each criterion, with 0 indicating worse performance, 1 indicating reasonable, and 2 indicating the perfect score.

Keep in mind that while the generated description may not exactly match the ground truth, it should still faithfully represent the content of

the image. Pay close attention to the patent figure when making your assessments.

Ensure that your output is formatted as follows:

```
Relevance: <your score>
Accuracy: <your score>
Completeness: <your score>
Coherence: <your score>
Fluency: <your score>
Coverage: <your score>
```

```
IMAGE:#img_url#
GROUND TRUTH:#gt_desc#
GENERATED DESCRIPTION:#gen_desc#
```

Output the scores for each metric in the prescribed format in <results></results> tag.

We replace all the instances of “brief” with “detailed” in the System Prompt to evaluate the detailed descriptions generated by our PATENTLMM.

## D Additional Qualitative Analysis

### D.1 Case Studies

We perform rigorous case studies by carefully going through randomly chosen 8 patent images and their corresponding brief and detailed descriptions generated by our approach, critically comparing them against respective ground truths. We summarize our observations for all these examples in the following text:

In Fig. 4, the generated brief description accurately describes the flowchart as a method for providing the decision of a priority arbiter in a network device for forwarding an incoming packet. The generated detailed description provides a comprehensive breakdown of the flowchart steps, such as receiving an incoming packet at a network device, classifying the packet for processing by VRF subsystems, processing the packet and generating action codes, generating decisions using priority arbiters, selecting a particular priority arbiter and providing a decision for forwarding the packet. Although the steps and their purposes are described accurately in the generated description, however, when compared to the ground truth detailed description, the generated description is missing reference to Figs. 3 and 4 (cross-figure references) of the same patent. This is because our PATENTLMM is trained to describe each patent image independently, limiting the model’s ability to draw connections between related images within the same patent document.

In Fig. 5, the generated brief description is more specific compared to the ground truth by mentioning ‘property page’, but loses its overall meaning by not talking about ‘user profile data’. The generated detailed description provides a comprehensive explanation of the flowchart, covering all the steps shown. It accurately describes the process of detecting user interactions, analyzing content, examining user profile data, generating a user interface with suggestions, and up-

dating the user profile. When compared, the generated and ground truth descriptions mostly differ in their language and level of details only.

In Fig. 6, the generated brief description focuses on using a network service for sharing spreadsheet objects, which aligns well with the ground truth. The generated detailed description provides a fairly comprehensive overview of the system’s components and their interactions like the sharing manager 26, the web browser 222, the application 224, the user interface 216, etc. However, it does not provide as much context on the types of computing devices and network configurations that can be used in the system, unlike the ground truth. Moreover, the term ‘codeless sharing’, a concept central to the ground truth, is omitted in the generated descriptions.

In Fig. 7, the generated brief description correctly describes the flowchart as a method for checking open orders in a stock, which aligns well with the ground truth’s emphasis on viewing open order status. The generated detailed description demonstrates a comprehensive understanding of the process flow, accurately describing key steps such as selecting open orders function, displaying the list of open orders, selecting a particular stock, highlighting orders for possible actions, selecting operations like cancel, change, or replace, and populating a trade ticket with information. When compared to the ground truth, some cross-figure references are absent in the generated description. It is however commendable how our PATENTLMM successfully captured the unusual flow indicated by arrows and node labels in this figure despite the downsampling of the image before being passed to the model.

In Fig. 8, the generated brief description accurately identifies the block diagram as a KPI management system. The generated detailed description further captures many key elements correctly. These include the server 120, the application component 110, the interface component 120 and the KPI definition component 134. The generated description however does not mention the data source 130 and the data 132 components shown in the patent figure. Moreover, the description of the capabilities of the server 120 and the nature of the databases 130 is less detailed in the generated description compared to the ground truth.

In Fig. 9, the generated brief description correctly identifies the image as a block diagram illustrating a VMM (Virtual Machine Manager) pool with dedicated hardware resources. The generated detailed description captures many system components accurately including the hardware resources 202, the load balancer 206, the cloning manager 152, the VM pool 204 and the VMM 102. While our model accurately describes these components in detail, it incorrectly identifies the system as 200 instead of 300. This misidentification likely stems from the model observing other node labels in the diagram that start with 2 (such as 202, 204, 206, 208, 214) and erroneously extending this pattern to the overall system number.

In Fig. 10, the generated brief description accurately captures the general idea of a content delivery system. The generated detailed description correctly identified many components like the client 205, the content sources 290A-290N,

direction 810, source 815, schedule 820, reports 825, and phase 830. The description further elaborates on some of these components. However, the generated description focused more on the configuration of various components for monitoring and diagnostics, while the ground truth emphasized on download behaviors and content delivery methods.

In Fig. 11, the generated brief description, though less comprehensive compared to ground truth, correctly identifies the image as a schematic overview of a system for providing rack configuration to a device. The generated detailed description accurately lists the main components of the system such as the device discovery module 140, the rack management module 160, the rack management communication interface (150), the user interface module (130) and the examples of managed devices 170 (servers, switches, routers). When compared, the generated and ground truth descriptions mostly differ in their level of details for certain components.

## D.2 Failure Cases

Fig. 12 demonstrates a few failure cases that occur in the brief descriptions generated by our PATENTLMM model. Specifically, Fig. 12(a) corresponds to the case when the model confuses the OCR (reads Fig 20 as 2C) and generates the figure label incorrectly, Fig. 12(b) demonstrates a case when the model hallucinates reference to another figure of the same patent (also discussed for detailed description for case study 1 (Fig. 4) in section D.1), and Figs. 12(c) and (d) demonstrate the case when the model hallucinates the description due to negligible OCR-detectable text in the figure.

These three error categories are also encountered in detailed descriptions generated by our model. Additionally, we identify two more failure cases in the detailed descriptions generated by our PATENTLMM model, as shown in Fig. 13, which occurs when the node labels are associated incorrectly with concepts presented in nodes, and in Fig. 14, which happens when the model hallucinates node labels. These cases usually occur due to the distortion of node labels or wiggly arrows connecting the nodes and node labels, during down-sampling of images before being passed to PatentMME.

## E Future directions to mitigate failures

In this section, we describe potential approaches to address the observed failure cases in our model’s generated descriptions. Specifically, we focus on reducing hallucinations and inaccuracies arising from missing contextual information, including figure references and technical details.

**Document-level Reasoning for better Cross-Figure References** Our analysis reveals that the model occasionally hallucinates cross-figure references (as illustrated in our case studies (Appendix D.1)), particularly when an invention’s component is illustrated across multiple diagrams. To mitigate this issue, we propose enhancing document-level reasoning by linking interdependent figures throughout the patent document. By enabling the model to track and reconcile components and their relations across multiple figures, we can ensure that figure references are more accurate and context-aware.

**Incorporation of External Knowledge Bases** When key textual cues are absent or insufficient within a given figure, the model may hallucinate technical details. To address this limitation, we suggest integrating external technical knowledge sources—such as domain-specific knowledge bases or authoritative patent databases—into the generation process. By drawing on these external resources, the model can retrieve and incorporate accurate relevant information rather than hallucinating it. To this end, techniques inspired by Retrieval-Augmented Generation (RAG) can be employed to query large, domain-specific repositories and return the most relevant knowledge snippets. This retrieval step provides a verifiable grounding for the generated descriptions, significantly reducing the likelihood of hallucinated technical content.

**Grounded Description Generation** Combining the above strategies can lead to a unified approach for grounded description generation, where figure content is consistently anchored to both internal textual references and external knowledge bases. By ensuring that each descriptive element corresponds to verifiable information, we can produce more reliable, less hallucinated, and overall higher-quality figure descriptions. This combination of document-level reasoning and external resource integration is a crucial step towards generating robust and trustworthy descriptions of patent figures.



	Generated Brief and Detailed Descriptions	Ground Truth Brief and Detailed Descriptions
<p style="text-align: center;"><b>FIG. 9</b> <span style="float: right;"><u>900</u></span></p>	<p>FIG. 9 illustrates a flowchart of a method for providing a decision of a priority arbiter in a network device for forwarding an incoming packet according to some embodiments of the present disclosure.</p> <p>FIG. 9 illustrates a flowchart of a method 900 for providing a QoS arbiter in a VRF subsystem. In one embodiment, method 900 may be performed by an INE 204. At step 902, an incoming packet is received at a network device. In one embodiment, the network device is a network processor. At step 904, the incoming packet is classified for processing by one of a plurality of VRF subsystems. In one embodiment, the classification is performed by a traffic manager. At step 906, the incoming packet is processed and the action codes generated by multiple packet processors of the VRF subsystem are provided to a plurality of priority arbiters. In one embodiment, the priority arbiters are implemented in hardware. At step 908, a plurality of decisions are generated by the plurality of priority arbiters using a mapping of the action codes to a set of priorities associated with each priority arbiter. In one embodiment, the mapping is static. At step 910, a particular priority arbiter from the plurality of priority arbiters is selected based on an association of the VRF subsystem with the particular priority arbiter. In one embodiment, the priority arbiter is selected based on the priority associated with the VRF subsystem. At step 912, the decision is provided as an output decision for forwarding the incoming packet. In one embodiment, the output decision is provided to the traffic manager. In one embodiment, the output decision is provided to the VRF subsystem.</p>	<p>FIG. 9 illustrates a computer implemented method for packet forwarding using multiple priority arbiters in one embodiment of the disclosed technologies;</p> <p>FIG. 9 illustrates a computer implemented method 900 for packet forwarding using multiple priority arbiters in one embodiment of the disclosed technologies. The method 900 may be performed by the network device 300 as discussed with reference to FIG. 3. In step 902, an incoming packet may be received by a network device. As discussed with reference to FIG. 3, the network device 300 may receive the incoming packet 112 via the input interface 302. In some implementations, the incoming packet 302 may include only parts of the header. For example, the header may include source and destination physical addresses (e.g., IP addresses of respective devices), a protocol version (e.g., IPv4 or IPv6), a length of the packet, error correction data, and any other useful information. In step 904, the incoming packet may be classified for processing by one of a plurality of VRF subsystems. As discussed with reference to FIG. 3, the incoming packet 112 may be classified by the VRF classification logic 102 for processing by one of the VRF subsystems from the VRF subsystems 304 a-304 n. Each VRF subsystem may include a plurality of packet processors as discussed with reference to FIG. 4. In some embodiments, the VRF classification logic 102 may determine a VRF ID associated with the packet 112 and send the packet 112 to the VRF subsystem associated with that VRF ID. In one instance, based on the classification, the packet 112 may be sent to the first VRF subsystem 304 a for processing by multiple packet processors of the first VRF subsystem 304 a. For example...</p>

Figure 4: Case Study 1: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

	Generated Brief and Detailed Descriptions	Ground Truth Brief and Detailed Descriptions
<p style="text-align: center;"><b>Figure 16</b></p>	<p>FIG. 16 illustrates a flow chart of a method for providing a completion suggestion within a property page.</p> <p>FIG. 16 shows a flowchart of an example of a computer implemented method 1600 for providing a user interface related to user profile data, performed in accordance with some implementations. Method 1600 may be described with reference to FIGS. 19-23. At block 1610, a server performing method 1600 detects user interactions in a property page, the user interactions include interactions associated with the user and interactions associated with the social contacts of the user. For example, the user may have interacted with a post by one of his social contacts, or a record, or a feed. The interactions can be detected in the page provided to the user. At block 1620, the server performing method 1600 analyzes content of the user interactions to determine context. For example, the context can be determined by analyzing the words in a post, or the subject matter of a record. The context can be in the form of keywords, or concepts as described above. At block 1630, the server performing method 1600 examines user profile data to determine a field of information related to the context that needs to be filled or updated. There are many fields that can be related to the context. As described above, a field can be a value in a user profile. In some implementations, the field can be a user interaction. In some implementations, the field can be a user profile attribute. Other types of fields may exist. At block 1640, the server performing method 1600 generates a user interface with a suggestion/query related to the field, the suggestion/query including information from context of the user interactions obtained from the analysis. For example, if the user is looking at a post by one of his social contacts, and the context of the post indicates that it is related to a product, then the suggestion/query can...</p>	<p>FIG. 16 illustrates method operations for completing user profile data, in accordance with an embodiment of the invention.</p> <p>With the above general understanding of the function of the profile completion module, a method for completing user profile will now be described with reference to FIG. 16. The method begins in operation 1610, when user interactions at a property page are detected. The user interactions may include social interactions, such as comments, posts, pictures, etc., generated by a user or by the social contacts of the user or entities or groups of entities that are socially connected to the user or general actions/interactions associated with the user or actions/interactions associated with social contacts of the user. The user interactions are analyzed to determine the context of the interaction and/or data content associated with the user or social contacts of the user, as illustrated in operation 1620. User profile information is examined to determine a field related to the context that needs to be filled or updated, as illustrated in operation 1630. Based on the examination of the user profile information and the analysis of the user interactions, a user interface with a query/suggestion related to the field is generated, as illustrated in operation 1640. The query/suggestion includes information from context of the user interactions obtained from the analysis. The user interface may also include a confirmation statement to confirm the suggestion or a suggested user action to be taken at the user interface, an input area for responding to the query and/or the suggestion, if needed. The query/suggestion may be identified from a pre-defined list of queries/suggestions and is related to a specific field of the user profile that is to be filled. Response to the query/suggestion at the user interface is monitored and used to update the field in the user profile based on user action at the user interface, as illustrated in...</p>

Figure 5: Case Study 2: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

<p style="text-align: center;"><b>FIG. 2</b></p>	<p style="text-align: center;"><b>Generated Brief and Detailed Descriptions</b></p> <p>FIG. 2 shows a system for using a network service for sharing spreadsheet objects;</p> <p>FIG. 2 illustrates an embodiment of a system for sharing spreadsheet content. The system 200 may include a computing device 1 210, a computing device 2 220, and a spreadsheet server 240. The computing device 1 210 may include a spreadsheet application 212, a sharing manager 26, and a user interface 216. The computing device 2 220 may include a web browser 222 and an application 224. The sharing manager 26 may be a software application that may manage sharing of spreadsheet content between the computing device 1 210 and the computing device 2 220. The web browser 222 may be a web browser such as Internet Explorer, Mozilla Firefox, Safari, or other similar web browser. The application 224 may be a software application that may display and/or interact with spreadsheet content. The user interface 216 may be a user interface such as a graphical user interface. The spreadsheet server 240 may include a web application 242, a spreadsheet web renderer 244. A network service 250 may include a network service application 252. The web application 242 may be a software application that may provide spreadsheet content to the computing device 1 210 and the computing device 2 220. The spreadsheet...</p>	<p style="text-align: center;"><b>Ground Truth Brief and Detailed Descriptions</b></p> <p>FIG. 2 shows a system for codeless sharing of spreadsheet objects with a network service;</p> <p>FIG. 2 shows a system for codeless sharing of spreadsheet objects with a network service. As illustrated, system 200 includes computing device 1 (210), computing device 2 (220), network share 230, spreadsheet server 240 and network service 250. The computing devices may be any type of computing device that is configured to perform the operations relating to the sharing of one or more spreadsheet objects with a network service. For example, some of the computing devices may be: mobile computing devices (e.g. cellular phones, tablets, smart phones, laptops, and the like); desktop computing devices and servers. Some computing devices may be arranged to provide an online cloud based service (e.g. interacting with spreadsheets online), some may be arranged as data shares, some may be arranged in local networks, some may be arranged in networks accessible through the Internet, and the like. The computing devices are coupled through network 18. Network 18 may be an IP network, a carrier network for cellular communications, and the like. Generally, network 18 is used to transmit data between computing devices, such as computing device 1, computing device 2, network share 230, spreadsheet server 240 and network service 250. Computing device 1 includes spreadsheet application 212, spreadsheet 214 and user interface 216. As illustrated, computing device 1 is used by a user to select one or more objects to share with a network service, such as a social networking site, a blog, a web page hosted by a service, or some other network location. Spreadsheet 214 comprises different cells/spreadsheet objects that may be selected for sharing with a network service. The...</p>
--	---	--

Figure 6: Case Study 3: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

<p style="text-align: center;"><b>FIG. 34</b></p>	<p style="text-align: center;"><b>Generated Brief and Detailed Descriptions</b></p> <p>FIG. 34 illustrates a flow chart of a method for checking open orders in a stock;</p> <p>FIG. 34 is a block diagram that illustrates the function of the open orders window in the main trade screen, in accordance with an embodiment of the present invention. Step 3410. The user seeks to review the list of open orders in all stocks that are active in the account. The user selects the open orders function from function bar 2725. Step 3415. The application displays an entire list of the open orders window in the master trade screen. Step 3420. The user can then scroll through the open order list and can focus on a particular open order. The user can also select the open orders function from function bar 2725 to display the open orders window in the master trade screen. Step 3425. The user selects a particular stock from the stock summary display and then selects the open orders function from function bar 2725. Step 3430. The application displays only the selected open order in the open orders window. Step 3435. The user selects a desired open order from the list displayed in the open orders window. The application highlights the order for possible actions by the user, including: cancel, change, or replace. Step 3440. The user selects the desired open order from the list displayed in the open orders window. Step 3445. The user selects an operation from the open orders window. The application automatically cancels, changes, or deletes the selected open order. Step 3450. The application populates a trade ticket with information from the open order window. Step 3455. The user executes the desired function according to trade ticket functions. Step 3460. To hide the open orders window, the user selects the open orders function from function bar 2725. Step 3465. The application removes the open orders window and removes the indication of open orders from the open orders function.</p>	<p style="text-align: center;"><b>Ground Truth Brief and Detailed Descriptions</b></p> <p>FIG. 34 is a flow diagram illustrating software which enables a user to view the status of his open orders in a stock.</p> <p>The application software used in the practice of the invention is described in greater detail in conjunction with FIGS. 17-40 below. FIG. 34 is a flow diagram illustrating software which enables a user to view the status of his open orders in a stock. Initially, at step 3410, the user seeks to review his open orders in all stocks active in his account. At step 3415, the user selects the open orders function from the function buttons display (see FIGS. 5 &amp; 15). At step 3420, the application displays the entire list of open orders in the open orders (see display in FIGS. 5 &amp; 7) display of the master trade screen. At step 3425, the user views the display showing the open orders. The user selects a stock, at step 3430, from the stock summary display (see FIGS. 5 &amp; 12) and selects the Open orders function. The application then, at step 3435, displays only open orders for the selected stock in the open orders display. The user selects, at step 3440, the desired open order from the list. The application highlights the order for actions by the user, such actions including cancel, change, or replace. At step 3445, the user selects an operation (e.g., cancel, change, or delete) from the open orders display. The application automatically populates the trade ticket, at step 3450 (see FIGS. 5 &amp; 11), with information from the open orders display on the selected stock. At step 3455, the user executes the desired function as requested in the trade ticket. To hide open orders, the user selects, at step 3460, the Open orders function from the function button display (function bar). The application then removes the open orders display and removes the indication from the open orders function, at step 3465.</p>
---	---	---

Figure 7: Case Study 4: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

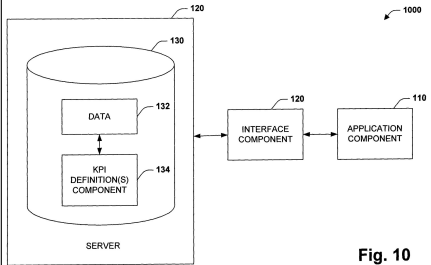
	Generated Brief and Detailed Descriptions	Ground Truth Brief and Detailed Descriptions
 <p style="text-align: right;"><b>Fig. 10</b></p>	<p>FIG. 10 is a block diagram of a definition-based KPI management system that facilitates the linking of KPIs to an application.</p> <p>FIG. 10 illustrates a system 1000 that employs KPI definition(s) in connection with automatically scaling a target component. The system 1000 includes the server 120, the application component 110, and the interface component 130, which are substantially similar to the server 120, the application component 110, and the interface component 120, respectively, described in connection with FIG. 1. The system 1000 further includes a KPI definition component 134 that can provide KPI definition(s) to the server 120. It is to be appreciated that the KPI definition component 134 can be implemented as a component within the server 120, as a separate server, etc. For instance, the KPI definition component 134 can be associated with an enterprise and can provide KPI definition(s) that are specific to the enterprise. In another example, the KPI definition component 134 can provide KPI definition(s) that are more generic and can be utilized by multiple enterprises. The KPI definition(s) provided by the KPI definition component 134 can be related to various aspects that are to be monitored by the server 120. For example, the KPI definition(s) can be related to an amount of traffic associated with the application component 110, an amount of resources utilized by the application component 110, a latency associated with the application component 110, etc. According to an example, the KPI definition(s) can be related to a performance of the application component 110 in connection with a particular service. For example, the KPI definition(s) can be related to a performance of the application component 110 in connection with a chat service, a voice service, a video service, etc. According to another example, the KPI definition(s) can be related to a performance of the application component 110 in connection...</p>	<p>FIG. 10 is a block diagram of a KPI system in accordance with an aspect of the subject invention.</p> <p>FIG. 10 depicts a KPI system 1000 in accordance with an aspect of the subject invention. KPI system 1000 includes an application component 110, an interface component 114, and a server 120 including data source 130, data 132 and KPI component(s) 134. Application component 110 can be a computer program or computer program product including a plurality of executable instructions for performing one or more functions. One of those functions can include presentation, interaction, or processing of key performance indicators. Server 140 is a computer or computing device that provides data to other computers or applications over a network. Server 140 can manage the resources such as one or more data stores or databases 130. According to one aspect of the invention, the server 140 can be an analysis server and the database 130 can be a multidimensional database. However, it should be appreciated that the invention is not so limited. Various aspects of the present invention can be performed on a server designated for a different purpose and possibly storing data in a relational format. Database 130 stores data 130. Similarly, database 130 can store a specific type of data namely metadata. Metadata is simply data about or concerning other data. Such metadata can be a schema describing how other data is organized, for example, cubes, dimensions, hierarchies, measures, and so forth for a multidimensional database. According to another aspect of the invention, additional metadata can be stored corresponding to a KPI component(s) 134. KPIs are collections of attributes or elements. Accordingly, a KPI component can provide for and define a KPI as previously described. Interface component 120 can facilitate retrieval and transmission of data concerning...</p>

Figure 8: Case Study 5: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

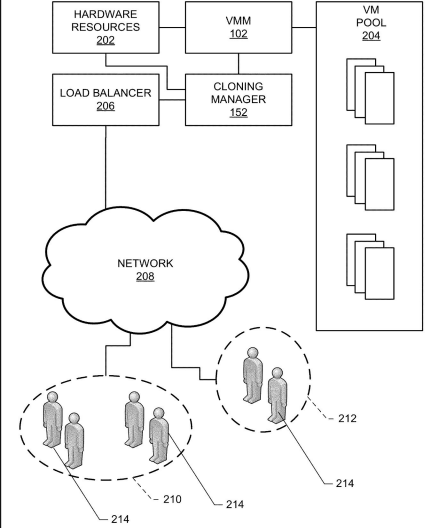
	Generated Brief and Detailed Descriptions	Ground Truth Brief and Detailed Descriptions
 <p style="text-align: center;"><b>FIG. 2</b></p>	<p>FIG. 2 is a block diagram illustrating a dedicated VMM pool of virtual machines with dedicated VMM hardware resources, according to various embodiments.</p> <p>FIG. 2 is a block diagram of an example system 200 for maintaining a virtualized environment. The system 200 includes hardware resources 202, a load balancer 206, a pool 204 of virtual machine instances, a cloning manager 152, and a virtual machine migration service (VMMS) 102. The hardware resources 202 can include one or more servers. The one or more servers can include one or more CPUs, memory, storage, and/or networking resources. The hardware resources 202 can be virtualized, meaning the hardware resources 202 can be presented to virtual machines as multiple virtual machines having multiple virtualized CPUs, memory, storage, and/or networking resources. The hardware resources 202 can be virtualized by a hypervisor or virtual machine monitor (VMM) that manages the hardware resources 202. The hardware resources 202 can be virtualized by a single VMM and present to multiple virtual machines. The load balancer 206 can be a computer system or a server that manages a computer cluster or a cloud computer cluster. The load balancer 206 can be responsible for distributing resources, for example, workloads, input or output among a network of servers, for example, in a computer cluster. The load balancer 206 can be responsible for balancing the resources, for example, the number of client requests by a client application running on different servers, to optimize resource use, maximize throughput, minimize response time, and avoid overload. The load balancer 206 can be responsible for increasing or decreasing resources responsive to demand. The load balancer 206 can use a least connections algorithm to route client requests to servers. The pool 204 of virtual machine...</p>	<p>FIG. 2 is a schematic illustration of an example implementation of a cloning manager of the example system of FIG. 1B.</p> <p>FIG. 2 is a schematic illustration of an example system 300 to provision virtual machine resources. In addition to the VMM 102 and the cloning manager 152 described above in connection with FIG. 1B, the illustrated example of FIG. 2 includes hardware resources 202, a VM pool 204, and a load balancer 206 communicatively coupled to a network 208. The example hardware resources 202 of FIG. 2 may include any number of CPUs, storage resources (e.g., hard disk drive(s), optical disk drive(s), flash disk drive(s), etc.) and/or any amount of memory, such as the example RAM 112 and/or nonvolatile memory. The illustrated example of FIG. 2 also includes a first customer 210 and a second customer 212, each of which may include any number of users 214. Although the illustrated example of FIG. 2 includes two customers, any number of customers may be serviced by example methods, apparatus, systems and/or articles of manufacture disclosed herein. In operation, the example cloning manager 152 of FIG. 2 invokes the example VMM 102 to instantiate a parent VM, such as the example VM1 104 of FIG. 1B, which results in an established (running) VM. As described above, the parent VM performs a number of disk accesses to acquire and store pages to physical memory 112. The parent VM may allocate other hardware resources 202, such as one or more CPUs and/or other processors dedicated to the parent VM. The example parent VM may be generally associated with a type of task and/or operation including, but not limited to OLAP, banking, ecommerce, and/or web services. In the event one or more task(s) are deemed dissimilar, additional parent VMs may be instantiated, as desired. When the parent VM...</p>

Figure 9: Case Study 6: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

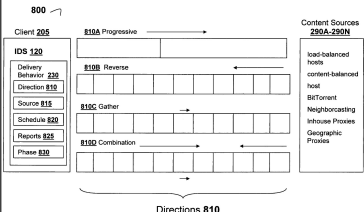
	Generated Brief and Detailed Descriptions	Ground Truth Brief and Detailed Descriptions
 <p style="text-align: center;"><b>Fig. 8A</b></p>	<p>FIG. 8A is a block diagram of an embodiment of a system for progressively loading content;</p> <p>Referring to FIG. 8A, a diagrammatic view of a system for remote diagnostic analysis and management of networked devices is shown. As shown, the system 800 may include a client 205 and one or more content sources 290A-290N. The client 205 may include a number of modules, such as a direction 810, a source 815, a schedule 820, a reports 825 and a phase 830. The content source 290A-290N may include a number of modules, such as a load-balanced host, a content-balanced host, a load-balanced bitstream, a neighborcasting module, a geographic proxy, and a geographic proxy. The progressive diagnostic engine 810 may be configured to perform progressive diagnostics on the content sources 290A-290N. The progressive diagnostic engine 810 may be configured to perform progressive diagnostics on the content sources 290A-290N. The source 815 may be configured to identify one or more content sources 290A-290N to be monitored by the system 800. The direction 810 may be configured to identify a direction in which to send the monitored content source 290A-290N. The schedule 820 may be configured to identify a schedule for sending the monitored content source 290A-290N. The phase 830 may be configured to identify a phase of the schedule 820 for sending the monitored content source 290A-290N. The reports 825 may be configured to generate one or more reports associated with the monitored content source 290A-290N. The directions 810 may be configured to identify one or more directions in which to send the monitored content source 290A-290N.</p>	<p>FIG. 8A is a diagrammatic view of another embodiment of downloading according to a delivery behavior;</p> <p>Referring now to FIG. 8A, the environment 800 provides a diagrammatical view of delivery strategies 810 used between a client 205 and one or more content source 290A-290N. In overview of environment 800, a client 205 comprises the IDS 120 in communication over a network with one or more content sources 290A-290N. The client 205 may use a delivery behavior 230 to download one or more files from a content source 290A-290N. The delivery behavior 230 may identify or specify one or more of the following: 1) direction 810, 2) source 815, 3) schedule 820, 4) report 825, and 5) phase 830. The content sources 290A-290N may comprise a variety of types of content sources providing a range of performance or download characteristics. The content source 290A-290N may comprise any of the following: 1) a load-balance host or server, 2) a content-balance host or server, 3) a peer-to-peer client, host, server or peer, such as a BitTorrent tracker or seed host, 4) an inhouse proxy, 5) a geographic proxy, or 6) may otherwise use neighborcasting to be directed to a content source. In further detail, the direction 810 of the delivery behavior 230 may identify or specify any of the following types of directions: 1) progressive 810A, 2) reverse 810B, 3) gather 810C, and combination 810D. Any one of these directions 810A-810D may be a default download direction used by the client 205, IDS 120 or download manager 220, and may be a default download direction for a user, application or content source 290A-290N. For a progressive download direction 810A, the download manager 220 of the IDS 120 downloads the content, such as file, from the beginning to end of the content in order, i.e. from front-to-back or start-to-end order.</p>

Figure 10: Case Study 7: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

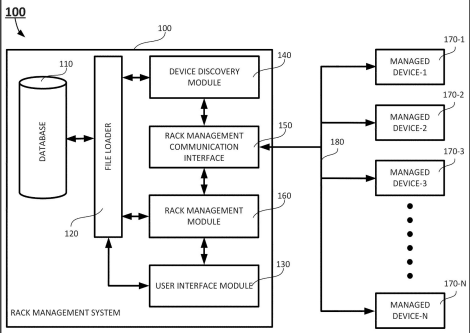
	Generated Brief and Detailed Descriptions	Ground Truth Brief and Detailed Descriptions
 <p style="text-align: center;"><b>FIG. 1</b></p>	<p>FIG. 1 is a schematic overview of a system for providing a rack configuration to a device according to one embodiment.</p> <p>FIG. 1 is a diagram of an example rack management system 100. The rack management system 100 includes a device discovery module 140, a rack management module 160, a rack management communication interface 150, and a user interface module 130. The rack management system 100 communicates with managed devices 170. The managed devices 170 may include any number of managed devices 170-1, 170-2, 170-3, . . . , 170-N. The managed devices 170 may include, for example, servers, storage devices, switches, routers, or any other type of data center component. The rack management system 100 may communicate with the managed devices 170 using any type of communication protocol including, for example, a proprietary protocol, an Ethernet protocol, a fiber channel protocol, a serial attached SCSI protocol, a SONET protocol, or a Fibre Channel over Ethernet protocol. The device discovery module 140 is configured to discover managed devices 170. For example, the device discovery module 140 may communicate with the managed devices 170 using a rack management protocol. The rack management protocol may be a proprietary protocol or a standard protocol. The rack management protocol may be based on, for example, the Distributed Management Task Force (DMTF) Open System Manufacturer's Alliance (OSMA) standard. The device discovery module 140 may send a rack management protocol communication to each of the managed devices 170. The rack management protocol communication may request the managed devices 170 to provide data such as, for example, identification data identifying the managed...</p>	<p>FIG. 1 schematically shows a block diagram of a rack management system for constructing, configuring, monitoring and managing the managed devices on a rack according to one embodiment of the present disclosure;</p> <p>The present disclosure will now be described more fully hereinafter with reference to the accompanying drawings, FIGS. 1-7, in which embodiments of the disclosure are shown. This disclosure may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the disclosure to those skilled in the art. Like numbers refer to like elements throughout. In computer server centers or data centers, a large amount of computers, servers, routers, disk arrays, and switches are mounted on one or more racks and powered by power distribution units (PDUs). Monitoring and managing the operations of these managed devices on the rack are critically important. The managed devices on the rack can be any type of device, including, but not limited to, routers, access servers, switches, bridges, hubs, IP telephones, IP video cameras, computer hosts, and printers. A power distribution unit (PDU) is a device fitted with multiple outputs designed to distribute electric power, especially to racks of computers and networking equipment located within the computer server centers or data centers. Most computer servers, storage and network devices provide remote access. Common methods are accessible through SNMP and include a RS-232 serial connection (for local management) using a command-line interface (CLI) or a LAN network-controller (for remote management) using a web page. This allows an administrator to monitor and manage the managed devices on the rack from a remote terminal and...</p>

Figure 11: Case Study 8: An example of generated and ground truth brief and detailed descriptions using our proposed PATENTLMM.

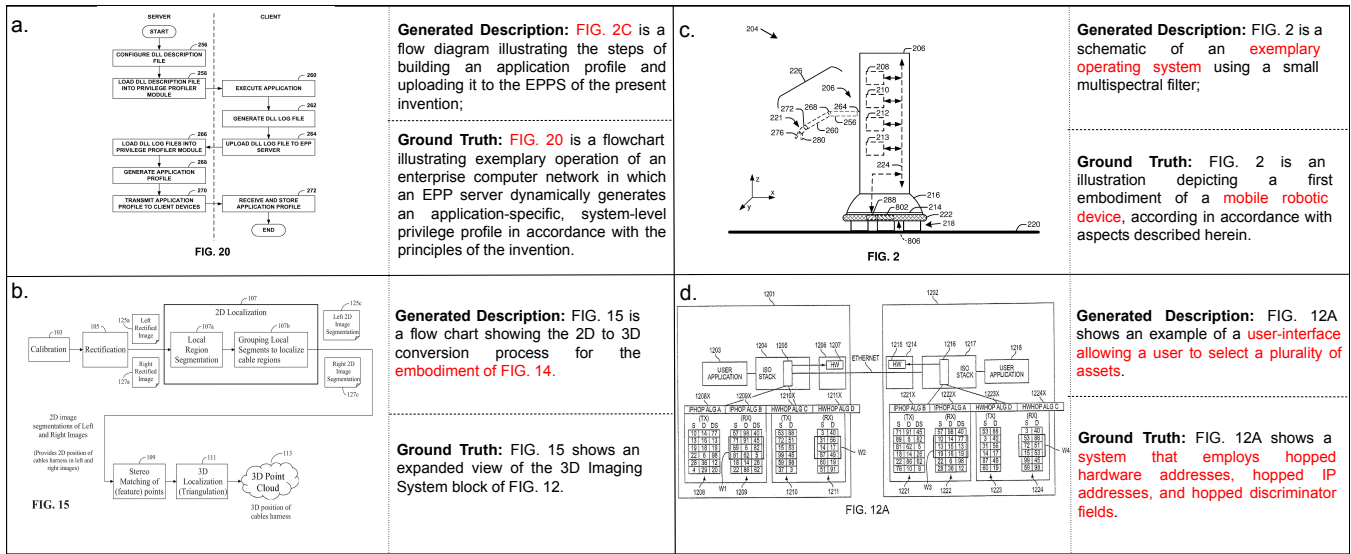


Figure 12: Failure Cases for the brief descriptions generated by PATENTLMM.

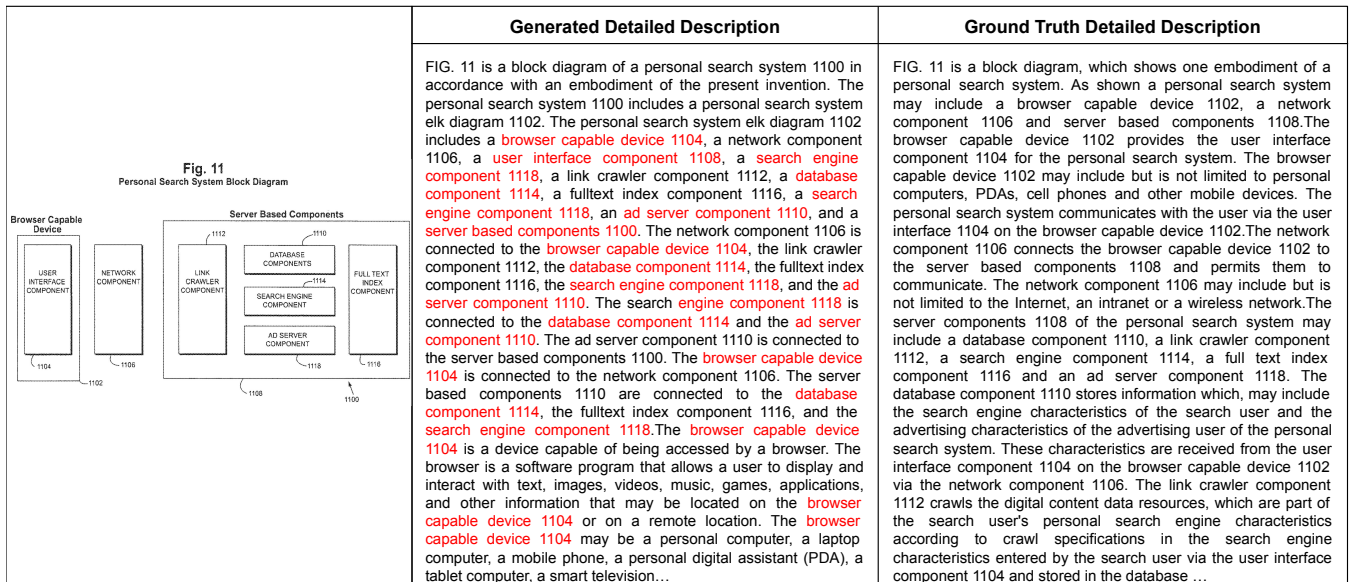


Figure 13: Failure Case for the detailed description generated by PATENTLMM. The text in red highlights the incorrect association of nodes and node labels in the generated description.

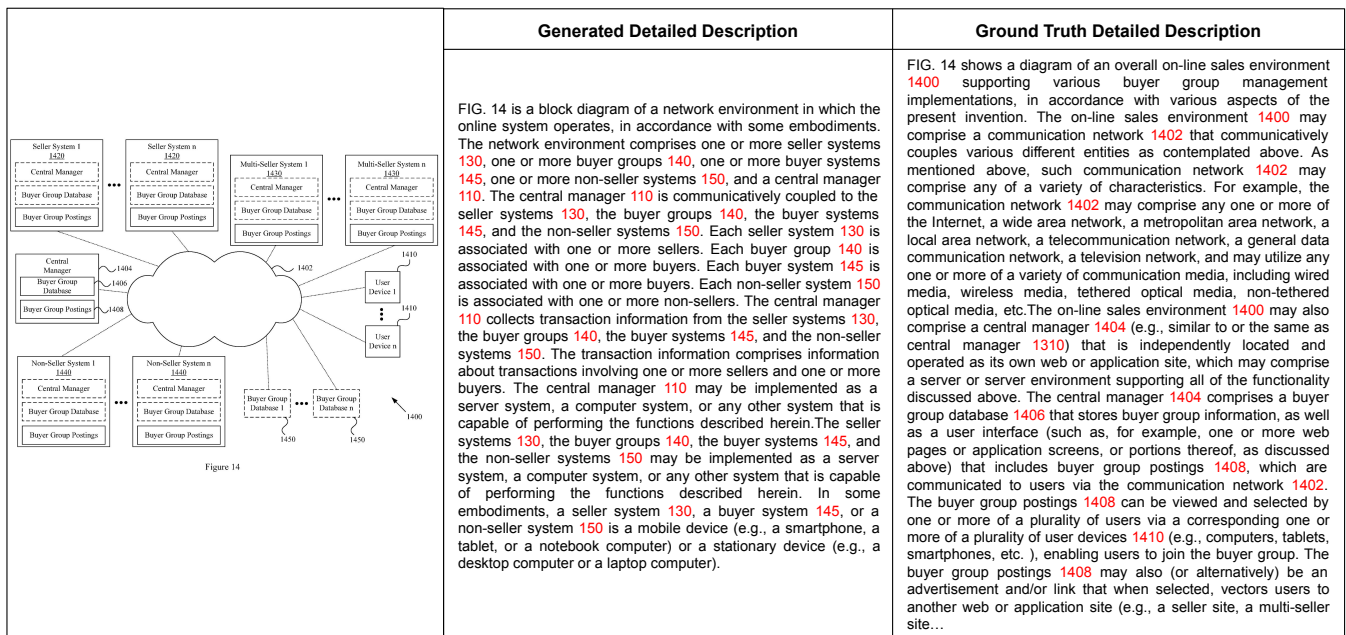


Figure 14: Failure Case for the detailed description generated by PATENTLMM. The text in red highlights the hallucinated node labels.

## References

- Banerjee, S.; and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL Workshop*.
- Du, Y.; Li, C.; Guo, R.; Cui, C.; Liu, W.; Zhou, J.; Lu, B.; Yang, Y.; Liu, Q.; Hu, X.; et al. 2021. PP-OCRv2: Bag of tricks for ultra lightweight ocr system. *arXiv preprint arXiv:2109.03144*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.
- Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.
- Liu, F.; Piccinno, F.; Krichene, S.; Pang, C.; Lee, K.; Joshi, M.; Altun, Y.; Collier, N.; and Eisenschlos, J. 2023. MatCha: Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12756–12770.
- Mahinpei, A.; Kostic, Z.; and Tanner, C. 2022. Linecap: Line charts for data visualization captioning models. In *Visualization and Visual Analytics (VIS)*.
- Masry, A.; Do, X. L.; Tan, J. Q.; Joty, S.; and Hoque, E. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *ACL (Findings)*, 2263–2279.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.
- Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*.
- Tang, B.; Boggust, A.; and Satyanarayan, A. 2023. VisText: A Benchmark for Semantically Rich Chart Captioning. In *ACL*.