Improving Visual Commonsense in Language Models via Late Fusion of Multiple Image Generation

Anonymous authors
Paper under double-blind review

Abstract

Commonsense reasoning is fundamentally based on multimodal knowledge. However, Large Language Models (LLMs), trained using textual data only, are limited in their ability to incorporate essential visual information. In contrast, Visual Language Models (VLMs), which excel at visually-oriented tasks, often fail at non-visual tasks such as textual commonsense reasoning. This divergence highlights a critical challenge - the integration of robust visual understanding with foundational text-based reasoning. To this end, we introduce a method aimed at enhancing LLMs' visual commonsense while maintaining textual modeling and commonsense reasoning performance. Specifically, our method is based on test-time compute scaling. We generate multiple images based on the input text prompt and integrate these into the model's decision-making process by mixing their prediction probabilities. To facilitate multimodal grounded language modeling, we employ a late-fusion layer that combines the projected visual features with the output of a pre-trained LLM conditioned on text only. This late-fusion layer enables predictions based on comprehensive image-text knowledge as well as text-only when required. We evaluate our approach using several visual commonsense reasoning tasks together with traditional NLP tasks, including commonsense reasoning and reading comprehension. Our experimental results demonstrate significant superiority over existing baselines. When our method is applied to recent LLMs (e.g., Llama 3), we observe improvements not only in visual commonsense but also in NLP benchmarks.

1 Introduction

Large language models (LLMs) have shown significant success in advancing a variety of natural language understanding and generation tasks (Devlin et al., 2019; Radford et al., 2019; Zhang et al., 2022b; Team et al., 2024; Touvron et al., 2023). As human knowledge is grounded in multimodal information, Vision Language Models (VLMs) have emerged, incorporating both images and text (Alayrac et al., 2022; Liu et al., 2023b;a; Li et al., 2023a; Dai et al., 2023; Cha et al., 2024), thus enabling significant advances in multimodal tasks such as visual commonsense and visual question answering (Zhang et al., 2022a; Xia et al., 2023; Li et al., 2023c; Jin et al., 2024). However, while VLMs excel at visually oriented tasks, this success may come at the expense of their performance on non-visual tasks. Specifically, we categorize VLMs into three groups: (i) VLMs that train the LLM weights exclusively on multimodal data and therefore suffer from forgetting of language capabilities (Lu et al., 2024; McKinzie et al., 2024), (ii) VLMs that freeze the LLM during training and thus preserve their language capabilities (Li et al., 2023a), and (iii) VLMs that balance the proportion of language and multimodal data during training to mitigate forgetting of language capabilities (Lu et al., 2024; McKinzie et al., 2024). With these directions, the key question now is whether we can further improve the integration of robust visual understanding with foundational text-based language reasoning, enhancing rather than merely preserving language capabilities (Yun et al., 2021). We note that one cause for this divergence is the VLM's over-reliance on a single visual input, even when such input contains little relevant information (Chen et al., 2024).

To mitigate such discrepancies Visually-augmented Language Models (VaLMs) were proposed (Wang et al., 2023; Guo et al., 2023; Zhang et al., 2022a; Cui et al., 2024; Tan & Bansal, 2020). VaLMs suggest augmenting text-based models with additional visual information. Recent studies suggest VaLM-like models improve visual commonsense performance in NLP benchmarks (Zhang et al., 2024; Lu et al., 2022; Tang et al., 2023; Zhang et al., 2022a; Yang et al., 2022). Notice, unlike VLMs, VaLMs focus on utilizing relevant visual information to improve visual commonsense in language-oriented tasks, whereas VLMs are aimed at reason over visual inputs such as visual question answering and image captioning (Wang et al., 2023).

In this study, we propose a novel VaLM like approach for improving visual commonsense reasoning in LLMs. The proposed approach comprised of two main components: (i) a novel architecture, that allows for the late fusion of text and images, and (ii) an inference-based procedure that integrates multiple images generated by a pre-trained text-to-image model conditioned on the input text.

More specifically, in training, given an image and a corresponding caption, our method first encodes the image using a pre-trained multimodal encoder, mapping the input into a common representation space of text and images. Next, this encoded representation is passed through a projector, which maps this encoding to a sequence of embeddings $z_1^v, \dots, z_n^v$. Simultaneously, the input text is passed through a pre-trained LLM, producing text token embeddings $z_1^x, \dots, z_k^x$. Finally, we combine $z_1^v, \dots, z_n^v$ and $z_1^x, \dots, z_k^x$ through a late-stage attention-like mechanism, which allows for text tokens to attend to the pseudo-text tokens generated from the visual input. Unlike previous work, this integration is done once, just before the model's prediction, and not as input to the LLM. This late fusion enables the model to better focus on the input text to predict the next token while also enabling it to use visual information if this is required to predict the next token. This formulation strikes the right balance, allowing success in both visual understanding and text-based language reasoning.

The second component of our approach involves the integration of multiple visual inputs at inference. Unlike training, we do not have access to images corresponding to the input text at inference. So, instead, we generate multiple images conditioned on the input text using a pre-trained text-to-image model. More specifically, we consider different variations of the input text and pass it to a pre-trained text-to-image generator to generate k image variations. Each generated image is fed into our visually augmented LLM to generate k different predictions (probability vectors) and a prediction when no input image is given, thus generating $k+1$ predictions. Lastly, all probability vectors are weighted-averaged to produce the final output. By integrating different probability vectors, our prediction is based on several visualizations conditioned on the input text. Further, the aggregated probability vector will be highly influenced by confident predictions, being of low entropy. By providing an option not to use an input image at all, we also enable the prediction to be made based on the input text alone when this is required.

We evaluate our approach on a set of object and visual common-sense tasks together with text-based commonsense reasoning. For object commonsense, we employ the zero-shot benchmark proposed by Wang et al. (2023), which focuses on questions related to colors, shapes, and sizes of different objects. For visual commonsense, we consider a more challenging benchmark, the ImageNetVC (Xia et al., 2023) dataset. ImageNetVC is composed of high-quality question-answer pairs over diverse domains. For commonsense reasoning, we assess our method using standard benchmarks, similarly to Dubey et al. (2024); Touvron et al. (2023); Team et al. (2023); Almazrouei et al. (2023a). We also consider the task of reading comprehension, where we adhere to the benchmark framework suggested by Touvron et al. (2023). When considering object and visual commonsense tasks, the proposed approach significantly outperforms the evaluated baselines across a variety of architectures and model sizes. Interestingly, following the proposed approach also slightly improves performance in text-based common-sense reasoning tasks. We conclude the experimental section with an ablation study, analyzing the importance of each of the components composing our method.

2 Related Work

Large Language and Vision Models. LLMs have demonstrated remarkable capabilities in various natural language processing tasks (Devlin et al., 2019; Radford et al., 2019; Zhang et al., 2022b; Team et al., 2024; Touvron et al., 2023). Their potential expands significantly when integrated with visual modalities, giving rise to vision language models (VLMs) (Alayrac et al., 2022; Liu et al., 2023b;a; Li et al., 2023a; Dai et al., 2023; Cha et al., 2024). By incorporating text and image data during training, VLMs have enabled a new set of multimodal understanding capabilities, allowing breakthroughs in tasks such as visual question answering (VQA), image captioning, and visual commonsense reasoning (Zhang et al., 2022a; Xia et al., 2023; Li et al., 2023c; Jin et al., 2024). Despite their exceptional performance in visually oriented tasks, VLMs frequently exhibit a drop in performance in non-visual tasks that necessitate fundamental common-sense reasoning. In this work, we aim to improve performance in visual reasoning tasks while maintaining (or even slightly improving) commonsense reasoning compared to language models.

Visually-Augmented Language Models. Numerous studies explored approaches to augment text-only Language Models with visual information. One set of approaches retrieves images related to the input text and uses them as contextual input to the language model (Tan & Bansal, 2020; Lu et al., 2022; Wang et al., 2023). Similarly, Tang et al. (2021) employs a knowledge distillation approach to fuse visual knowledge. Other works (Zhang et al., 2024; Guo et al., 2023; Li et al., 2023c) distill visual knowledge from multimodal embedding methods such as CLIP (Radford et al., 2021) into text-only language models. Similarly, MORE (Cui et al., 2024) distills visual knowledge from BLIP-2's Q-Former (Li et al., 2023a) into text-only models. Another set of works utilizes pre-trained text-to-image generative models. In the context of diffusion-based text-to-image models, Z-LaVi (Yang et al., 2022) leverages generated visuals that match possible label predictions of a given text-only language model. Our method, instead, considers visuals that match the input text. LiVE (Tang et al., 2023) introduces a vision-text plug-and-play vision-text fusion layer inserted within transformed blocks of pre-trained LMs. iNLG (Zhu et al., 2023) uses generated images as additional visual supervision to guide the language model in text generation, where the visual input is provided as an additional input to the LM in the form of a visual prefix. Unlike LiVE and iNGL, which integrate visual knowledge as input to the LM or as an integrated layer, we, instead, use the output of an unmodified pre-trained LLM together with an encoding of a generated image, using a late-fusion layer. This enables focusing on the input text and use visual information. Second, instead of a specialized attention-like mechanism or a mapping network, our work aggregates scores simply by averaging predictions made using different generated images obtained from variations of the input text. This enables our method to use a diverse set of predictions obtained using diverse visual "experts" and gauge its final prediction towards the more confident predictions.

Multiple Generations Agreement. Several works encourage an agreement, or consistency, between the predictions of a language model given perturbations of the input (Bachman et al., 2014; Sajjadi et al., 2016; Xie et al., 2020; Zhai et al., 2019). In contrast, we model this agreement by aggregating predictions given different visual inputs generated through a pre-trained text-to-image model conditioned on the input text. Our work is also related to the ability to obtain the confidence of LLMs, as derived by Portillo Wightman et al. (2023). They showed that one can estimate the confidence of LLMs by aggregating their predictions under different prompts. Our motivation is similar but uses the agreement of different visually generated inputs. In addition, while their work focuses on estimating confidence, we aim to improve visual commonsense reasoning.

3 Method

Our approach, denoted as vLMIG , (stands for improving visual Language Models via Multiple Image Generation), aims to leverage visual cues to improve object and visual commonsense capabilities in LLMs while maintaining their performance in standard text benchmarks (i.e., commonsense reasoning and reading comprehension tasks). For that, vLMIG adopts a multi-modal learning approach, where we incorporate visual cues within textual representation to perform next-token prediction. During training, we utilize two types of input data: (i) a pair of images and their corresponding text description, and (ii) a text and a synthetically generated image obtained from a text-to-image model. During inference, given an input text prompt, we generate multiple images corresponding to different parts of the input text, feed them into the model, and aggregate their probability vectors based on their alignment with the input prompt. In the following subsections we: (i) outline the process of model optimization (Section 3.1); and (ii) introduce our visually driven inference method (Section 3.2).

3.1 Visually Enhanced Language Model

Our training process aims to equip the LM with the ability to utilize visual knowledge and align it with textual information. To this end, vLMIG is comprised of four main components: (i) a pre-trained LLM;

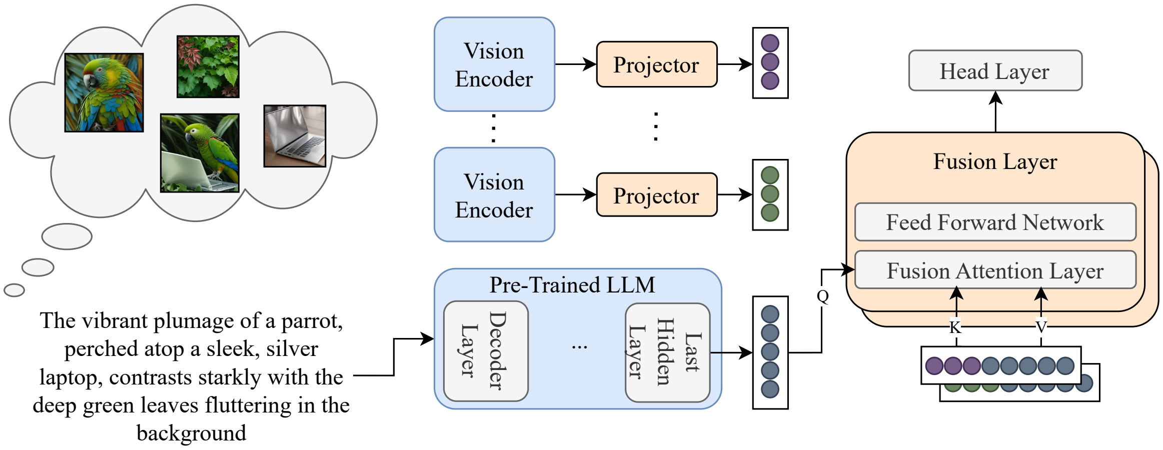


Figure 1: **Illustration of the proposed method.** During *training*, we utilize two types of data: (i). a pair of images and the corresponding text description, or (ii) a text and synthetically generated image conditioned on the input text. Each image is passed through a prerained vision encoder and then through a visual token projector, which projects the visual encoding onto pseudo-textual tokens. Simultaneously, the input text is passed through a pre-trained LLM, producing textual tokens. Next, our fusion layer fuses the visual pseudo-textual tokens and textual tokens, and produces a prediction of the next textual token. In this fusion layer, an attention-like mechanism is performed where queries are taken to the textual tokens, and the keys and values are taken as both the textual tokens and visual pseudo-textual tokens. In blue are fixed pretrained components while in orange are trainable components. At *inference*, the same process is applied, but to k different images conditionally generated using the input text. The predictions resulting from different images are then integrated as a form of ensemble using Eq. 6 and Eq. 7.

(ii) a pre-trained Vision Encoder; (iii) a Visual Token Projector (VTP); and (iv) a Late Fusion Attention Layer (LFAL). To preserve the integrity of their learned representations, the Vision Encoder and the LLM are kept frozen during the training process (refer to Figure 1). The following sections will elaborate on the VTP and LFAL components.

Given an image $v \in \mathbb{R}^{3\times 224\times 224}$ and its corresponding caption $x = (x_{(1)}, \dots, x_{(n_x)})$, where n_x is the number of tokens in the caption, the objective during training is to maximize the log-likelihood:

$$\max_{\theta} \log P_\theta(x_{(t)}|x_{(<t)}, v). \tag{1}$$

Our method begins with the vision encoder V that extracts visual features $z^v = \mathrm{V}(v)$, where $z^v \in \mathbb{R}^{n_v\times d_v}$. Here, n_v is the number of image patches produced by the image, which are subsequently used to extract visual features of dimension d_v using a visual extractor, which in our case is a vision encoder. These features are then transformed by the Visual Token Projector.

Visual Token Projector (VTP). The VTP intuitively projects the visual representation of the input image, z^v, into a pseudo-text latent embedding. Such representation does not represent actual words but aligns with the dimensions of the embedded text tokens, hence allowing us to fuse this visual representation with the input prompt later via attention blocks. The VTP comprises two linear layers,

$$u^v = W_1\sigma(W_2 z^v), \tag{2}$$

where $W_2 \in \mathbb{R}^{d_v \times d_{\mathrm{VTP}}}$, $W_1 \in \mathbb{R}^{d_{\mathrm{VTP}} \times d_x}$, d_{VTP} is the hidden embedding dimension, σ is a non-linear function, and d_x is the text embedding dimension of the LLM. Overall we obtain $u^v \in \mathbb{R}^{n_v\times d_x}$.

Late Fusion Attention Layer (LFAL). The LFAL aims to incorporate visual cues with textual context. The LFAL is a late fusion module, i.e., it is added before the logits output. The design of this layer is similar to that of a standard Transformer block. The trainable parameters of this layer are the modules that transform the input into Q, K, V representations accordingly. We fuse the visual representations with the text representations by concatenating them along the time dimension,

$$K = V = [z^v; z^x_{(<t)}], \quad Q = z^x_{(<t)}, \tag{3}$$

where z^x is the latent representation of the input text obtained by the pre-trained LLM, and $K, V \in \mathbb{R}^{(t+n_v)\times d_x}$, $Q \in \mathbb{R}^{t\times d_x}$. Thus, the attention mechanism facilitates the integration of visual context into the language model's predictions by computing

$$\Phi \propto \left(QK^T\right); \quad X_v = \Phi \cdot V, \tag{4}$$

where $\Phi \in \mathbb{R}^{t\times(t+n_v)}$, $X_v \in \mathbb{R}^{t\times d_x}$.

Finally, we introduce a linear layer to convert the embedding dimension to the dimensions of the vocabulary size. This can be represented as:

$$\hat{X}_v = WX_v, \tag{5}$$

where $W \in \mathbb{R}^{d_x\times N}$ are the trainable weights and N represents the size of the vocabulary.

3.2 Visually Driven Inference

vLMIG grounds the inference process with visual information. For instance, when asked, *Does the Samoyed have a spotted pattern on its back?* an image of a Samoyed could provide the necessary visual information. However, since the text lacks an associated image for visual reasoning in our inference setup, we employ a text-to-image module to generate the required images.

To enhance visual robustness, we generate k images corresponding to the entire prompt. For a prompt containing multiple sentences, an image is generated per sentence. If the number of sentences does not match k, we randomly sample k from the prompt's pool of sentences. We also ensure that each generated image corresponding to the same sentence is generated with a unique seed.

The images are integrated as a form of ensemble, i.e., we run our model with different images, allowing for a robust visual representation. This also allows for the integration of diverse visual information, i.e.,

$$\sum_{i=1}^{k} P_\theta(x_t \mid x_1, \ldots, x_{t-1}, v_i). \tag{6}$$

To add another layer of resilience to corrupted images, we additionally measure the alignment score between the text and the generated image by applying

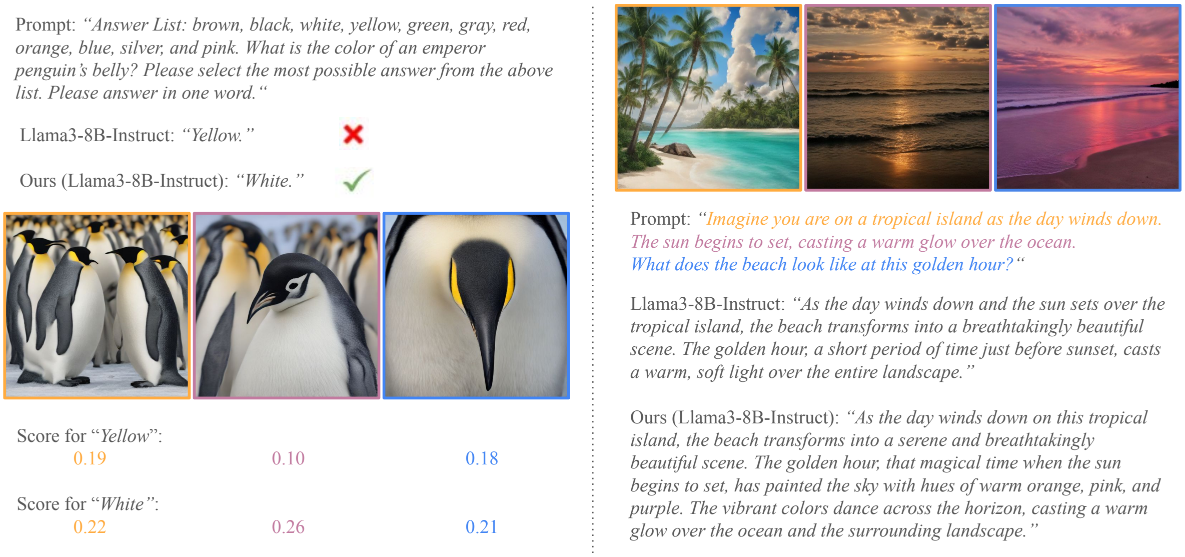
$$\sum_{i}^{k} f(\bar{x}_i, v_i)P(x_t \mid x_1, \ldots, x_{t-1}, v_i) + (1 - f(\bar{x}_i, v_i))P(x_t \mid x_1, \ldots, x_{t-1}) \tag{7}$$

where $f(\bar{x}_i, v_i)$ is a normalized CLIP score of the generated image v_i and its corresponding text $\bar{x}_i$. This allows us to determine the confidence of the generated image w.r.t the input text. Intuitively, when the score is high, we rely more on the language model with the image guidance, and otherwise on the original language model. Finally, we sample from Equation (7), to produce the text output. An example of our model's inference is shown in Fig. 2.

4 Experimental Setup

4.1 Datasets

We optimize vLMIG using a combination of natural and artificial text-image pairs, obtained by applying a pre-trained text-to-image model on texts from text-only datasets. We use the Visual Genome Regions dataset



Prompt: *"Answer List: brown, black, white, yellow, green, gray, red, orange, blue, silver, and pink. What is the color of an emperor penguin's belly? Please select the most possible answer from the above list. Please answer in one word."*

Llama3-8B-Instruct: *"Yellow."* ✗

Ours (Llama3-8B-Instruct): *"White."* ✓

Score for "*Yellow*": 0.19 0.10 0.18

Score for "*White*": 0.22 0.26 0.21

Prompt: *"Imagine you are on a tropical island as the day winds down. The sun begins to set, casting a warm glow over the ocean. What does the beach look like at this golden hour?"*

Llama3-8B-Instruct: *"As the day winds down and the sun sets over the tropical island, the beach transforms into a breathtakingly beautiful scene. The golden hour, a short period of time just before sunset, casts a warm, soft light over the entire landscape."*

Ours (Llama3-8B-Instruct): *"As the day winds down on this tropical island, the beach transforms into a serene and breathtakingly beautiful scene. The golden hour, that magical time when the sun begins to set, has painted the sky with hues of warm orange, pink, and purple. The vibrant colors dance across the horizon, casting a warm glow over the ocean and the surrounding landscape."*

Figure 2: An illustrative example of our method at inference. On the LHS, we address visual commonsense reasoning with the prompt *What is the color of an emperor penguin's belly?* vLMIG correctly answers 'White', while Llama3-8B-Instruct incorrectly answers 'Yellow". Our method generates 3 images and assigns higher weights (CLIP-scores) to the correct prediction. Similar illustrations can be found in 4. On the RHS, for text generation, our method generates 3 images aligned with different parts of the sentence, resulting in a visually cohesive response. In contrast, Llama's answer is less visually aligned.

(Krishna et al., 2016), which consists of 5.4M images with region descriptions. We also leverage Laion-220K (Schuhmann & Bevan, 2023), which comprises 220K captioned images from the LVIS dataset (Gupta et al., 2019), and Wikitext-103-raw-v1 (Merity et al., 2016), a collection of over 100 million tokens extracted from verified Wikipedia articles. To simulate inference with generated images, we randomly sample 2% of data from the Wikipedia textual dataset and use it to generate the corresponding image.

4.2 Implementation Details

In all experiments, we use CLIP-ViTB/32 (Radford et al., 2021) to compute the CLIP score for text-image pairs and as the vision encoder. For text-to-image generation, we utilize SDXL-turbo (Sauer et al., 2023). Model optimization was performed using four A100 GPUs following a dual training pipeline. The model was first trained for 40K iterations with a batch size of 256, employing AdamW with a learning rate of 5×10^{-4} and a constant scheduler. It was then fine-tuned for 10K iterations with a batch size of 128 and a learning rate of 5×10^{-5}. For all training runs, only the LFAL and VTP were optimized, while all other components remained frozen. Training vLMIG required approximately 192 A100 GPU hours on Llama-3, 90 hours on Gemma-2B and OPT-2.7B, and 50 hours on GPT-2. In inference, unless stated otherwise, we generate 10 images per sample.

4.3 Evaluation Benchmarks

Object Commonsense (Object Color, Shape, and Relative Size). For object commonsense evaluation, we employ the zero-shot evaluation benchmark proposed by Wang et al. (2023) which focuses on question-answering tasks related to colors, shapes, and sizes of objects. For color evaluation, we adapt the Memory Color (Norlund et al., 2021) and Color Terms (Bruni et al., 2012) datasets, and for shape assessment, we use the ViComTe shape dataset (Zhang et al., 2022a). Size evaluation employs the dataset inspired by Bagherinezhad et al. (2016). All these tests adhere to the guidelines provided by Wang et al. (2023).

Visual Commonsense. We evaluate the proposed method on ImageNetVC (Xia et al., 2023), a human-annotated dataset designed specifically for zero and few-shot visual commonsense evaluation across 1,000

ImageNet categories (Deng et al., 2009). It comprises more than 4,076 high-quality QA pairs over diverse domains such as color, shape, material, component, and general questions. For *base models*, we employ the prompts in Table 7 of Xia et al. (2023) and measure accuracy by selecting, from a predefined candidate set, the answer with the highest likelihood. For instruct models (i.e. Vicuna, Llama-3B-Instruct), we use the instruction-style prompt in Table 11 of the same work and compute the top-1 accuracy over the model's one-word response.

Commonsense Reasoning. For commonsense reasoning, we consider the same benchmark tests from Touvron et al. (2023): PIQA (Bisk et al., 2019), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC in both its easy and challenge forms (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), and CommonsenseQA (Talmor et al., 2018). To gauge accuracy across these various tests, we adopt the metric proposed by Shwartz et al. (2020), which calculates accuracy by selecting the candidate with the highest likelihood from a predefined candidate set.

Reading Comprehension. We adhere to the benchmark of Touvron et al. (2023) and assess performance on BoolQ (Clark et al., 2019), SQuAD 2.0 (Rajpurkar et al., 2018), and QuAC (Choi et al., 2018). We evaluate SQuAD and QuAC using the settings recommended by Ouyang et al. (2022) and report the exact match (EM) score. For BoolQ, we consider a zero-shot binary setup by selecting the highest probability between the yes/no tokens.

4.4 Baselines

We consider two sets of baseline methods. First, to evaluate object color, shape, and relative size, collectively representing the object visual commonsense benchmark, we compared our method with VaLMs, which are primarily focused on improving visual commonsense in language models. Specifically, we considered Vokenization (Tan & Bansal, 2020), based on BERT. X-adapter (Zhang et al., 2024) is based on both BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Z-LaVI (Yang et al., 2022) is built on GPT-neo-1.3B (Gao et al., 2020). iNLG (Zhu et al., 2023) uses the MS-COCO pretrained base model of BART (Lewis et al., 2019). Additionally, LIVE (Tang et al., 2023) leverages both BART and T5 (Raffel et al., 2023), as does the multimodal version of MORE (Cui et al., 2024), built on T5, and VaLM (Wang et al., 2023). We also directly compared these models with pure LMs, namely BERT and GPT-2 (Radford et al., 2019), as well as fine-tuned versions of these LMs trained on the same data as our method without incorporating images.

Second, to assess visual commonsense, commonsense reasoning, and reading comprehension, we conducted evaluations with LMs and VLMs across a range of model sizes and architectures. We aimed to compare our method with LLMs to ensure that we not only improve visual commonsense reasoning but also maintain performance on other language abilities. Additionally, we demonstrate that VLMs, which excel at visually-oriented tasks, are suboptimal compared to our method on non-visual tasks, such as basic commonsense reasoning. The LMs: GPT-2, OPT-2.7B (Zhang et al., 2022b), Gemma-2B (Team et al., 2024), Vicuna-7B (Zheng et al., 2023), and both the base and instruct versions of Llama3-8B (AI@Meta, 2024). The VLMs: InstructBLIP (Dai et al., 2023), built on Vicuna-7B, and Llava-Next (Liu et al., 2024), built on both Vicuna-7B and Llama3-8B.

5 Results

5.1 Main Results

We first examine ways of improving weaker language models (in terms of data and size) with visual capabilities. Following previous work, we focus on two types of language models: masked language models (BERT) and causal language models (GPT-2). Both models lack visual commonsense and fail to answer simple questions like, *What is the color of a banana?*. For GPT-2-based models, we measured accuracy using direct zero-shot predictions. For BERT-based models, we followed the approach from Zhang et al. (2024), masking the sequence immediately after the last word and predicting the masked token. For a fair comparison in this

Table 1: Performance results on object commonsense tasks (Memory Color, Color Terms, Object Shape, and Relative Size). The table compares (i) Masked Language Models, where vLMIG is based on BERT, and (ii) Causal Language Models, where vLMIG is based on GPT-2. Models marked * retrieve images during inference, † are zero-shot, and ‡ generate images during inference.

Model	Base Model	Memory Color	Color Terms	Object Shape	Relative Size
BERT	-	31.6	30.7	28.1	38.1
BERT (FT)	BERT	33.9	31.5	21.5	35.7
Vokenization*	BERT	14.2	20.0	43.2	72.4
X-adapter*	RoBERTa	59.6	53.8	-	-
X-adapter*	BERT	64.1	60.0	-	-
vLMIG ‡	BERT	**74.5**	**72.5**	**67.3**	**78.4**
GPT-2	-	32.4	34.6	44.5	43.1
GPT-2 (FT)	GPT-2	33.3	34.9	39.3	38.2
Z-LaVI*†	GPT-neo	50.4	49.2	64.4	76.8
LIVE‡	T5	42.4	41.5	36.4	70.1
LIVE‡	BART	49.6	46.7	41.5	66.7
iNLG‡	BART	48.6	44.8	39.5	51.1
MORE*	T5	47.5	45.6	33.5	65.8
VaLM* ($k = 4$)	GPT-2	54.0	52.7	62.8	85.0
VaLM* ($k = 8$)	GPT-2	58.6	50.2	59.4	62.4
vLMIG ‡	GPT-2	**72.5**	**69.2**	**66.8**	**85.5**

experiment, we limited the training of our method to the Visual Genome dataset. Additional details about the baselines' setup can be found in Appendix A.

Table 1 summarizes the results [1]. vLMIG significantly improves all tasks and model variations when considering BERT-based models. As for GPT-2, vLMIG significantly outperforms the base model and VaLM across all setups, with minor improvement when considering Relative Size (85.0 vs. 85.5). Furthermore, we trained vLMIG on COCO with the same settings as iNLG. The results, 66.9 vs. 48.6 on Memory Color, 65.8 vs. 44.8 on Color Terms, 63.1 vs. 39.5 on Object Shape, and 73.5 vs. 51.1 on Relative Size show that our method consistently outperforms iNLG across all tasks. We hypothesize that the improvement is a result of our unique integration of multiple images, whereas in the baselines, a single image could sometimes be incorrect. Further, our novel late fusion mechanism that uses multiple image generation provides a significant advantage over the rest of the baselines that incorporate images in earlier layers or Z-LaVI that sum probabilities and do not fuse images.

In Appendix B, we provide additional comparisons with baselines on the object commonsense task: (i) using image retrieval instead of image generation, (ii) generating a different number of multiple images for our method and the baseline, and (iii) evaluating vLMIG on additional object visual commonsense tasks. Specifically, we present a comparison on the VEC benchmark (Li et al., 2023c), showing that vLMIG outperforms baselines in tasks involving height, mass, temperature, and hardness.

Next, we evaluate vLMIG on more complex benchmarks of visual commonsense, commonsense reasoning, and reading comprehension. Results are reported in Table 2. Results per subtask can be found in Appendix C.2. We show that vLMIG consistently outperforms LMs across all model sizes: small (GPT-2), mid (Gemma-2B), and large-scale (Vicuna-7B, Llama3-8B, and Llama3-8B-Instruct). Interestingly, our method also slightly improves performance in commonsense reasoning and reading comprehension, tasks that are primarily text-oriented and typically do not require visual reasoning. We observe an improvement of $\sim$1 absolute points over the LMs in commonsense reasoning while maintaining comparable performance in reading comprehension.

When comparing with VLMs, such as the VLM variants of Vicuna-7B, InstructBLIP and Llava-Next, while these models boost visual commonsense (e.g., Llava-Next achieves 49.3 vs. 43.5 for Vicuna-7B), they suffer

---

[1] The reported results of GPT-2, BERT, Z-LaVI, iNLG, MORE, and LIVE were obtained by running the official codebase, while the results for the other models were taken from their respective papers. As no codebase exists for the X-adapter, we could not obtain a result for Object Shape and Relative Size.

Table 2: Results for visual commonsense, commonsense reasoning, and reading comprehension. We report results for LLMs: GPT-2, Gemma-2B, OPT-2.7B, Vicuna-7B, Llama3-8B, and Llama3-8B-Instruct. * indicates VLM models that train on large-scale image-text data. We apply vLMIG to the base LLM in each table block for a fair comparison.

Model	Base Model	Tasks			
		Visual Commonsense	Commonsense Reasoning	Reading Comprehension	Avg.
		Small-Scale Models			
GPT-2	-	30.3	46.1	30.5	35.6
vLMIG	GPT-2	**38.6**	**46.7**	**32.2**	**39.2**
		Mid-Scale Models			
Gemma-2B	-	45.6	63.8	48.8	52.7
vLMIG	Gemma-2B	**50.1**	**65.1**	**48.9**	**54.7**
OPT-2.7B	-	41.0	50.9	44.6	45.5
vLMIG	OPT-2.7B	45.4	**51.6**	**44.7**	**47.2**
		Large-Scale Models			
Vicuna-7B	-	43.5	56.6	57.5	52.5
InstructBLIP*	Vicuna-7B	48.4	52.5	53.6	51.5
Llava-Next*	Vicuna-7B	**49.3**	53.7	54.7	52.5
vLMIG	Vicuna-7B	47.6	**56.8**	**57.9**	**54.1**
Llama3-8B	-	52.0	72.0	57.9	60.6
Llava-Next*	Llama3-8B	**55.6**	68.5	54.8	59.6
vLMIG	Llama3-8B	55.0	**72.9**	**58.0**	**62.0**
Llama3-8B-Instruct	-	53.0	71.6	59.2	61.2
vLMIG	Llama3-8B-Instruct	**55.6**	**71.7**	**60.9**	**62.7**

trade-offs in commonsense reasoning (53.7 vs. 56.6) and reading comprehension (54.7 vs. 57.5). InstructBLIP shows a similar trend, with 48.4 in visual commonsense but reduced performance in commonsense reasoning (52.5 vs. 56.6) and reading comprehension (53.6 vs. 57.5). In contrast, our method not only improves visual commonsense (47.6 for Vicuna-7B) but also enhances performance in commonsense reasoning (56.8) and reading comprehension (57.9). This consistency demonstrates that, despite extensive VLM training, late fusion adaptation enhances visual commonsense capabilities without compromising other language tasks.

In addition, we evaluate whether vLMIG , which leverages an image generation model and an image feature extraction model-adding approximately 2.5B parameters to the backbone model-bridges the gap between smaller and larger models in the visual commonsense task, i.e., achieving performance comparable to significantly larger models of the same architecture. Table 3 presents the accuracy of vLMIG applied to models of different sizes. When applied to OPT-2.7B, vLMIG improves accuracy from 41.0 to 45.4, achieving performance comparable to the significantly larger OPT-66B model (45.7). Similarly, applying vLMIG to Llama3-8B improves accuracy from 52.0 to 55.0, closely matching Falcon-40B (Almazrouei et al., 2023b) (55.0) while using significantly fewer parameters. Moreover, using three LFAL layers, as presented in Appendix C under "Effect of scaling in fusion strategy on performance," further boosts Llama3-8B-based vLMIG to 56.3, approaching the performance of Llama3-70B (56.9).

A potential concern is that generating multiple images during inference could increase runtime, reducing the practicality of the proposed method. However, we believe the proposed approach could be viewed as test-time compute scaling, which leads to better task performance on the expense of longer inference time. Having said that, to fairly assess this, we estimate the inference time of vLMIG and baseline methods using the vLLM inference package (Kwon et al., 2023), while ensuring model performance remains roughtly the same. We evaluate the inference time of vLMIG using Llama3-8B and compare it to Llama3-70B, with both models achieving similar task performance. vLMIG (Llama3-8B) runs at 2425 ms per sample, while Llama3-70B requires 2802 ms. In both settings, we use a context length of 1000 tokens and measure runtime over a loop of 38 predictions. The Llama3-8B variant is evaluated on a single A100 GPU, whereas Llama3-70B runs on 8×A100 GPUs. These results demonstrate that vLMIG achieves competitive inference efficiency while avoiding the computational overhead of larger models.

Table 3: Scaling analysis: vLMIG improves smaller models to match the performance of larger models.

Model	# Parameters	Accuracy
OPT-based Models		
OPT-2.7B	2.7B	41.0
vLMIG (OPT-2.7B)	5.3B	45.4
OPT-66B	66B	45.7
Llama3 and Falcon-based Models		
Falcon-7B	7B	49.4
Llama3-8B	8B	52.0
vLMIG (Llama3-8B)	10.7B	55.0
vLMIG (Llama3-8B) - 3 LFALs	11.1B	56.3
Falcon-40B	40B	55.0
Llama3-70B	70B	56.9

Table 4: Performance comparison of image generation, CLIP text embedding, and the baseline (Gemma-2B) on Visual Commonsense, Commonsense Reasoning, and Reading Comprehension. Additionally, we report the average inference time (in milliseconds) for a single token prediction on the Color dataset.

Method	Visual Commonsense	Commonsense Reasoning	Reading Comprehension	Inference time (ms)
Gemma-2B	45.6	63.8	48.8	**20**
CLIP Text Embedding	47.9	65.0	48.9	58
Generated Images	**50.1**	**65.1**	**48.9**	779

5.2 Ablation Studies

We present four ablation studies: (i) analyzing the effect of using a multi-modal representation, (ii) comparing the effect of late versus early fusion layers, (iii) examining the effect of the number of generated images, and (iv) comparing our approach with a fine-tuned baseline. Additional results and ablations can be found in Appendix C. Specifically, we ablate the fusion depth, explore different visual encoder choices and image generation models, report inference efficiency, and evaluate the image selection strategy—all of which contribute to vLMIG 's effectiveness.

CLIP text embedding vs. image generation. One might argue that multi-modal representations, which might serve as a bridge between image and text modalities, could be used instead to inject visual information. For instance, one could extract a CLIP representation for each input prompt and obtain a visually driven text representation. Such representation could be later used under the same modeling setup instead of synthetic image generation. And so the natural question is, *do synthetically generated images hold more information than multi-modal text representation?*

To address this, we compare vLMIG both of which are based on the Gemma-2B architecture, against text representations obtained from a pre-trained CLIP (Radford et al., 2021) model, specifically the CLIP-ViTB/32 version. For a fair comparison, we adapt the proposed model architecture, datasets, and implementation details and only replace the visual representations with multi-modal textual representations. Initially, we tried to embed the full-text prompt using CLIP. However, this resulted in poor performance. Instead, as suggested by Guo et al. (2023), we extract noun entities from the text prompt using a part-of-speech tagger. Then, we embed this pre-processed text using CLIP text encoder. We report the visual commonsense, commonsense reasoning, and reading comprehension results under the same settings described in Section 4.3. We also report the average inference running time, of a token prediction, tested on the color test. Results are reported in Table 4.

For visual commonsense, image generation significantly boosts performance compared to the CLIP text embedding and the baseline (Gemma-2B), while CLIP text embedding improves over the baseline. In commonsense reasoning and reading comprehension, both image generation and CLIP text embedding methods

Table 5: Ablation of late vs. early fusion of visual information on Visual Commonsense* (includes the Color test set), Commonsense Reasoning* (includes the PIQA test set), and Reading Comprehension* (includes the BoolQ test set).

Architecture	Visual Commonsense*	Commonsense Reasoning*	Reading Comprehension*
Early fusion	41.9	75.2	65.9
Prepend fusion	42.4	70.1	65.4
Late fusion	**45.4**	**77.7**	**67.0**

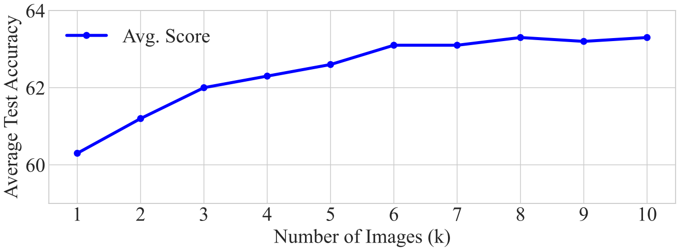


Figure 3: Average impact of the number of generated images per inference on performance, aggregating results from three tests: Color (Xia et al., 2023), PIQA (Bisk et al., 2019), and BoolQ (Clark et al., 2019). This graph displays the average performance scores for values of k from 1 to 10, illustrating the general trend across varied test scenarios under identical settings.

either outperform or match the baseline. For inference time, using CLIP text embeddings helps close the gap between the generated image method and the baseline. Overall, while the generated images method achieves the highest performance, CLIP text embeddings offer a balanced trade-off between accuracy and inference run-time, providing alternative for scenarios where computational resources or latency are limited.

Late vs. early fusion of visual information. Visual information can be fused into LLMs in different ways. We consider how fusing or injecting visual information in various stages affects downstream task performance. We report the results of vLMIG when considering either early or late fusion. In the case of early fusion, we apply our fusion layer to our visual pseudo tokens and textual tokens, which are the output of a single encoding layer of the pretrained LLM. We then pass the resulting output to the rest of the pretrained LLM. We additionally provide a comparison for an alternative design choice, in which we omit the fusion layer and optimize the vision projector layer while prepending such representation to the text input (*Prepend Fusion*), which is then fed into the LLM. The loss function and other training settings remain unchanged. As shown in Table 5, late fusion achieves the best performance across all benchmarks.

The effect of k (number of generated images). We analyze the impact of the number of generated images (k) during inference. We examined our method across various values of k, from $k = 1$ to $k = 10$. Due to resource constraints, we consider a single test from each of the three benchmarks (i.e., visual commonsense, commonsense reasoning, and reading comprehension). Specifically, we utilized the color test from the ImageNetVC benchmark (Xia et al., 2023) for visual commonsense. For commonsense reasoning, we use the PIQA benchmark (Bisk et al., 2019), and for reading comprehension, we consider BoolQ (Clark et al., 2019). Each test was performed under the same settings as in Section 4. The average results are shown in Figure 3, and the results per benchmark can be found in Table 23 on Appendix C.2. As expected, we observe an improved performance as we increase k, with a performance saturation when ~ 6 images are used, where scaling the number of images is especially helpful for improving performance on visual commonsense tasks.

vLMIG effectiveness vs. fine-tuned LLM. Lastly, to directly assess the impact of image integration, we compare our approach with a baseline using Gemma-2B (Team et al., 2024), which was fine-tuned on the same datasets but without visual elements. The Gemma-2B model was fine-tuned with a learning rate of 5×10^{-5}, identical to our training settings but excluding the additional visual layers. Results on the Color test from ImageNetVC (Xia et al., 2023), PIQA (Bisk et al., 2019), and BoolQ (Clark et al., 2019), detailed in Table 6, demonstrate that including images significantly enhances performance across all benchmarks.

Table 6: Performance comparison of our method vs. Gemma-2B fine-tuned (FT) LLM on *Visual Commonsense (includes the Color testset (Xia et al., 2023)), *Commonsense Reasoning (includes the PIQA testset (Bisk et al., 2019)), and *Reading Comprehension (includes the BoolQ testset (Clark et al., 2019)).

Method	*Visual Commonsense	*Commonsense Reasoning	*Reading Comprehension
Gemma-2B (FT)	35.2	76.1	64.9
vLMIG	**45.4**	**77.7**	**67.0**

6 Discussion

Limitations. While Tab. 3 shows that vLMIG achieves comparable inference time with fixed task performance, its main drawback remains inference speed, as it requires generating k images. This limitation is particularly evident when compared to models of the same size. Although recent diffusion models can produce high-quality images in a single step, inference still incurs noticeable latency. That said, this trade-off aligns with the prevailing trend of scaling test-time compute to enhance output quality (Wei et al., 2023; Yao et al., 2023; Snell et al., 2024), reaffirming the "no-free-lunch" principle that higher performance typically demands longer runtime.

For scenarios where inference speed is paramount, one can replace generated images with multimodal representations within the same framework—dramatically improving runtime at the cost of some performance (see Tab. 4). Additionally, simple optimizations such as parallelizing text processing and image generation or using lower-resolution images can further mitigate latency: preliminary results with Llama3-8B demonstrate that reducing image size from 512×512 to 256×256 lowers inference time from 1938 ms to 1588 ms with minimal impact on task performance. Future work will explore more advanced strategies to optimize inference time.

Finally, we have not yet evaluated vLMIG on the newest flagship LLMs (e.g., DeepSeek (DeepSeek-AI et al., 2025)) nor trained it beyond the 8B-parameter scale due to computational constraints. Larger-scale studies are left for future work.

Conclusion. We introduce vLMIG , a method designed to enhance the visual commonsense capabilities of LLMs while maintaining their foundational text-based language reasoning capabilities. To enable this, vLMIG introduces two main novel components: (i) a novel training pipeline consisting of a late fusion layer applied over the output of a text-only LLM and a visually adapted pseudo-tokens, and (ii) the integration of multiple visual "experts" through the generation of multiple images from a text-to-image model and the aggregation of their "vote" (or vector probabilities), enabling the model to leverage diverse visual perspectives. We conduct a comprehensive evaluation demonstrating our approach's effectiveness across various visual commonsense tasks, significantly outperforming existing baselines. Notably, vLMIG excels in visual tasks while also maintaining or slightly enhancing performance in text-based commonsense reasoning and reading comprehension. In future work, we plan to explore its potential in more complex visual reasoning tasks, including those requiring deeper object relationships and contextual understanding.

Broader Impact Statement

The broader impact of our method has both potential risks and benefits associated with using LLMs, visual encoders, and text-to-image generators. As our method uses these components, it inherits their associated issues. The following are points that should be considered: (i) *Hallucinations*: Similar to LLMs, our model might generate outputs that are not grounded in facts. In our case, this can also happen in the text-to-image model, where the model generates factually incorrect visuals; (ii) *Biases*: Biases can be performed using the pre-trained LLM, the CLIP encoder, and the text-to-image generator and transferred into our model. This may lead to biased output or to unfair representations of diverse content; and (iii) *Energy consumptions*: While our model primarily uses pre-trained foundation models as part of our model design and only adapts a lightweight vision projector fusion layer, training such pre-trained models requires significant energy consumption. Further, inference time queries, which are performed many times, may be costly.

References

AI@Meta. Llama 3 model card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023a.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023b. URL `https://arxiv.org/abs/2311.16867`.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014.

Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi. Are elephants bigger than butterflies? reasoning about sizes of objects, 2016.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam Khanh Tran. Distributional semantics in technicolor. In *Annual Meeting of the Association for Computational Linguistics*, 2012. URL `https://api.semanticscholar.org/CorpusID:8712237`.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm, 2024.

Meiqi Chen, Yixin Cao, Yan Zhang, and Chaochao Lu. Quantifying and mitigating unimodal biases in multimodal large language models: A causal perspective, 2024. URL `https://arxiv.org/abs/2403.18346`.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Wanqing Cui, Keping Bi, Jiafeng Guo, and Xueqi Cheng. More: Multi-modal retrieval augmented generative commonsense reasoning, 2024. URL `https://arxiv.org/abs/2402.13625`.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Hangyu Guo, Kun Zhou, Wayne Xin Zhao, Qinyu Zhang, and Ji-Rong Wen. Visually-augmented pretrained language models for nlp tasks without images, 2023.

Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation, 2019.

Woojeong Jin, Tejas Srinivasan, Jesse Thomason, and Xiang Ren. Winoviz: Probing visual properties of objects under different states, 2024.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023a.

Lei Li, Jingjing Xu, Qingxiu Dong, Ce Zheng, Qi Liu, Lingpeng Kong, and Xu Sun. Can language models understand physical concepts?, 2023b. URL `https://arxiv.org/abs/2305.14057`.

Lei Li, Jingjing Xu, Qingxiu Dong, Ce Zheng, Qi Liu, Lingpeng Kong, and Xu Sun. Can language models understand physical concepts? *arXiv preprint arXiv:2305.14057*, 2023c.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL `https://arxiv.org/abs/1405.0312`.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024. URL `https://arxiv.org/abs/2403.05525`.

Yujie Lu, Wanrong Zhu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. Imagination-augmented natural language understanding, 2022.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. Mm1: Methods, analysis & insights from multimodal llm pre-training, 2024. URL `https://arxiv.org/abs/2403.09611`.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Tobias Norlund, Lovisa Hagström, and Richard Johansson. Transferring knowledge from vision to language: How to achieve it and how to measure it?, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL `https://arxiv.org/abs/2304.07193`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, 2016.

Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. Strength in numbers: Estimating confidence of large language models by prompt agreement. In Anaelia Ovalle, Kai-Wei Chang, Ninareh Mehrabi, Yada Pruksachatkun, Aram Galystan, Jwala Dhamala, Apurv Verma, Trista Cao, Anoop Kumar, and Rahul Gupta (eds.), *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pp. 326–362, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.trustnlp-1.28. URL `https://aclanthology.org/2023.trustnlp-1.28`.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL `https://api.semanticscholar.org/CorpusID:160025533`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29, 2016.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. URL `https://arxiv.org/abs/2311.17042`.

Christoph Schuhmann and Peter Bevan. 220k-gpt4vision-captions-from-lvis. `https://huggingface.co/datasets/laion/220k-GPT4Vision-captions-from-LIVIS`, 2023.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL `https://aclanthology.org/P18-1238`.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Unsupervised commonsense question answering with self-talk. *arXiv preprint arXiv:2004.05483*, 2020.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL `https://arxiv.org/abs/2408.03314`.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

Hao Tan and Mohit Bansal. Vokenization: Improving language understanding with contextualized, visual-grounded supervision, 2020.

Tianyi Tang, Yushuo Chen, Yifan Du, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Learning to imagine: Visually-augmented natural language generation, 2023.

Zineng Tang, Jaemin Cho, Hao Tan, and Mohit Bansal. Vidlankd: Improving language understanding via video-distilled knowledge transfer. *Advances in Neural Information Processing Systems*, 34:24468–24481, 2021.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024.

MosaicML NLP Team et al. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. *URL www. mosaicml. com/blog/mpt-7b. Accessed*, pp. 05–05, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Weizhi Wang, Li Dong, Hao Cheng, Haoyu Song, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Visually-augmented language modeling, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL `https://arxiv.org/abs/2201.11903`.

Heming Xia, Qingxiu Dong, Lei Li, Jingjing Xu, Tianyu Liu, Ziwei Qin, and Zhifang Sui. Imagenetvc: Zero- and few-shot visual commonsense evaluation on 1000 imagenet categories, 2023.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

Yue Yang, Wenlin Yao, Hongming Zhang, Xiaoyang Wang, Dong Yu, and Jianshu Chen. Z-lavi: Zero-shot language solver fueled by visual imagination, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL `https://arxiv.org/abs/2305.10601`.

Tian Yun, Chen Sun, and Ellie Pavlick. Does vision-and-language pretraining improve lexical grounding? *arXiv preprint arXiv:2109.10246*, 2021.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1476–1485, 2019.

Chenyu Zhang, Benjamin Van Durme, Zhuowan Li, and Elias Stengel-Eskin. Visual commonsense in pretrained unimodal and multimodal models, 2022a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022b.

Xinyun Zhang, Haochen Tan, Han Wu, and Bei Yu. Towards versatile and efficient visual knowledge integration into pre-trained language models with cross-modal adapters, 2024. URL `https://arxiv.org/abs/2305.07358`.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL `https://arxiv.org/abs/2306.05685`.

Wanrong Zhu, An Yan, Yujie Lu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. Visualize before you write: Imagination-guided open-ended text generation, 2023.

A Comparison to baselines vision-text models

Baseline methods have various setups. For a fair comparison, we limit the training of our method to the Visual Genome dataset Krishna et al. (2016). Expect VaLM and iLNG, all other baselines were trained on the VG dataset during pretraining or retrieved images from Visual Genome during inference. Specifically, Vokenization and X-adapter rely on COCO (Lin et al., 2015) and VG, while LIVE incorporates COCO, VG, CC3M (Sharma et al., 2018), and Flick30k (Plummer et al., 2016).

Z-LaVI and MORE are zero-shot models; we employed them with the VG and Bing Image Search image collections.

VaLM, uses a different setup from ours, as it trains GPT-2 from scratch. Furthermore, since VaLM's weights are not publicly available, we could not fine-tune it on Visual Genome and report results from their paper instead. To ensure a fair comparison with iNLG, we also provide an additional evaluation, comparing our method with iNLG under the same training settings, i.e., trained on MS COCO.

In the Relative Size test, which involves a binary decision (yes/no questions), GPT-2, BERT, Vokenization, MORE, LIVE, and iNLG models exhibited a strong bias toward either "yes" or "no," often resulting in consistently answering either "yes" or "no." To address this, we fine-tuned the open-weight models (all models except VaLM and X-Adapter, which do not have open weights) using the proposed method, with 3,200 yes/no questions about object sizes from the ViComTe size dataset (Zhang et al., 2022a), over three epochs with a learning rate of $5e^{-5}$.

B Additional results on object commonsense tasks

Retrieval mechanism. For an equivalent comparison with X-adapter (Guo et al., 2023), we adopted the VaLM (Wang et al., 2023) image retrieval method, using retrieval instead of our image generation mechanism. Like X-adapter, we utilized Visual Genome as the image collection for retrieval and used MS COCO as our dataset for pretraining. As shown in Table 7, this approach yielded scores of 65.5 on Memory Color and 62.8 on Color Terms, both of which are higher than X-adapter's results (64.1 and 60.0, respectively). Since X-adapter's weights are not publicly available, we refer to the results reported in the paper and are unable to conduct evaluations for additional tasks.

Table 7: Performance comparison of our method with X-adapter on Memory Color and Color Terms.

Method	Memory Color	Color Terms
X-adapter	64.1	60.0
vLMIG (retrieval)	**65.5**	**62.8**

Equal number of images involved. We explored the impact of varying the number of images k used during inference on the performance of our method compared to the baseline LIVE (Tang et al., 2023). For both LIVE and our method, we applied our CLIP-based fusion strategy to fuse multiple images and generate the final predictions.

Table 8 presents the accuracy scores for object commonsense tasks across different values of k. Our method consistently outperforms LIVE for all values of k. Notably, even with $k = 1$, our approach achieves superior results. As k increases, both methods benefit from our fusion strategy, but our method continues to outperform LIVE across all tasks.

Table 8: Comparison of our method with LIVE (Tang et al., 2023) using our CLIP-based fusion and different numbers of images k. Accuracy scores are reported for object commonsense tasks.

Method	k	Memory Color	Color Terms	Object Shape	Relative Size
LIVE	1	49.6	46.7	41.5	66.7
LIVE	4	46.8	42.1	36.8	75.6
LIVE	8	47.5	42.7	37.1	76.1
vLMIG	1	65.1	62.2	63.5	70.2
vLMIG	4	70.2	67.6	66.0	83.6
vLMIG	8	**72.1**	**68.2**	**66.5**	**85.0**

Additional object commonsense tasks. We evaluate vLMIG on the VEC benchmark (Li et al., 2023b), which measures object visual commonsense across additional tasks involving Embodied Concepts, including temperature, mass, hardness, and height. Each sample in the test set contains a sentence, a positive word (correct), and a negative word (incorrect). A response is correct if the model assigns lower perplexity to the sentence with the positive word compared to the one with the negative word. For example, in “Deep red fire is hotter than melted steel” (positive: hotter) versus “Deep red fire is colder than melted steel” (negative:

colder), the model succeeds if it assigns lower perplexity to the first sentence. We report the accuracy for each dataset independently, evaluating vLMIG (Ours) based on GPT-2, GPT-2, LiVE, iLNG, MORE, and Z-LaVI, the open weights baselines. As shown in the Table 9, vLMIG consistently outperforms all baselines.

Table 9: VEC benchmark results. We report accuracy on the Height, Mass, Temperature, and Hardness test sets.

Model	Base Model	Height	Mass	Temperature	Hardness
GPT2	-	0.54	0.49	0.52	0.54
iLNG	BART	0.58	0.49	0.51	0.56
LIVE	BART	0.61	0.50	0.50	0.56
LIVE	T5	0.59	0.49	0.53	0.55
MORE	T5	0.52	0.50	0.52	0.55
Z-LaVI	GPT-Neo-1.3B	0.66	0.53	0.57	0.56
vLMIG (Ours)	GPT2	**0.71**	**0.66**	**0.60**	**0.58**

C Additional ablation study

Test-time scaling. We align Gemma-2B's inference runtime with that of our method configured with ten generated images by applying Best-of-N sampling. This procedure creates N independent candidate completions, scores each by average token log-likelihood, and selects the sequence with the highest score. As shown in Table 10, Best-of-N narrows the gap in commonsense reasoning and reading comprehension, but it is still not enough for visual commonsense where VLMIG keeps a clear lead.

Table 10: Effect of Best-of-N test-time scaling on Gemma-2B.

Approach	Visual Commonsense	Commonsense Reasoning	Reading Comprehension
Gemma-2B	45.6	63.8	48.8
Gemma-2B Best-of-N	47.8	64.8	49.0
vLMIG	**50.1**	**65.1**	**48.9**

Effect of scaling in fusion strategy on performance. We evaluate the impact of increasing the number of transformer layers in the fusion mechanism. Specifically, we compare variants of our late fusion approach and an early fusion baseline, each with one or three transformer layers. The results in Table 11 show that increasing the number of layers in late fusion leads to further improvements in visual commonsense reasoning, while early fusion remains less effective even with additional layers. These findings suggest that deeper late fusion enhances the integration of semantic text and visual representations, reinforcing its advantage over early fusion.

Table 11: Effect of scaling transformer layers in fusion strategy.

Approach	Visual Commonsense	Commonsense Reasoning	Reading Comprehension
Llama3-8B	52.0	72.0	57.9
Late fusion (1 transformer layer)	55.0	**72.9**	**58.0**
Late fusion (3 transformer layers)	**56.3**	72.7	57.7
Early fusion (1 transformer layer)	53.3	69.9	56.7
Early fusion (3 transformer layers)	53.6	69.5	55.9

Inference time comparison. In some cases where inference speed is a critical factor, faster alternatives to image generation can be employed. For instance, using CLIP embeddings instead of generating images provides a significant reduction in running time while still leveraging visual information. Additionally, retrieval-based methods can also offer efficient alternatives when image generation is computationally expensive.

In Table 12, we summarize the inference times for (1) our method using different image generation settings, (2) using retrieval instead of generation, (3) using CLIP embeddings only (no generation), and (4) other baseline approaches. The experiment was conducted using the GPT-2 model, with average inference times computed over the Color Memory test-set predictions. The image generation and retrieval methods involve different configurations, such as varying the number of generated images (k) and the size of the image collections for retrieval.

Table 12: Inference time comparison for various methods and configurations, measured in milliseconds (ms). Our methods are indicated as vLMIG .

Method	Inference time (ms)
No visual involved (GPT-2)	12
vLMIG (Image generation, $k = 1$)	229
vLMIG (Image generation, $k = 5$)	496
vLMIG (Image generation, $k = 10$)	750
vLMIG (CLIP embeddings)	50
vLMIG (Retrieval, $k = 4$, 6M images)	33
Vokenization	105
Z-Lavi	355
iNLG	235
LIVE	240
MORE	215
VaLM (retrieval), $k = 4$, (400M images)	51

The effect of the visual encoder. Our model employs the CLIP (Radford et al., 2021) visual encoder to handle image features, leveraging its multimodal training with text. We evaluate its effectiveness against a unimodal image encoder, DINOv2 (Oquab et al., 2024), across the same tasks: the Color test, PIQA, and BoolQ. Results are summarized in Table 13. Although DINOv2 provides comparable or superior performance to the baseline methods, results suggest that CLIP still outperforms DINOv2, particularly in tasks requiring nuanced visual comprehension, validating our choice of CLIP for enhanced multimodal learning.

Table 13: Experiment results using different visual encoders on *Visual Commonsense (includes the Color testset (Xia et al., 2023)), *Commonsense Reasoning (includes the PIQA testset (Bisk et al., 2019)), and *Reading Comprehension (includes the BoolQ testset (Clark et al., 2019)).

Visual encoder	*Visual Commonsense	*Commonsense Reasoning	*Reading Comprehension
DINOv2	43.9	77.0	66.6
CLIP	**45.4**	**77.7**	**67.0**

The effect of the image generation model. To explore the impact of image fidelity on reasoning capabilities, we evaluate two text-to-image models: SDXL-turbo and SD-turbo. These experiments were conducted on the same tasks and datasets as the previous ablation. As shown in Table 14, SDXL-turbo significantly outperforms SD-turbo in the Color task, indicating that superior image quality directly contributes to better performance in visual commonsense reasoning. While improvements in PIQA and BoolQ are less pronounced, they underscore the importance of high-quality image generation in our model. These results imply that advancements in text-to-image research will additionally improve our method.

Impact of image source on visual commonsense performance. We tested our model under three different variations (using the number of images k as 10): (i) Replacing the generated images with images

Table 14: Experiment results using different text-to-image models on *Visual Commonsense (includes the Color testset (Xia et al., 2023)), *Commonsense Reasoning (includes the PIQA testset (Bisk et al., 2019)), and *Reading Comprehension (includes the BoolQ testset (Clark et al., 2019)).

T2I model	*Visual Commonsense	*Commonsense Reasoning	*Reading Comprehension
SD-turbo	41.9	76.9	66.7
SDXL-turbo	**45.4**	**77.7**	**67.0**

representing different prompts from the dataset, (ii) Using k-1 images representing different prompts from the dataset and a single generated image using the correct prompt, and (iii) Generating k images from correct prompts as default. The results in Table **??**, based on our Gemma-2B model, report accuracy on our visual commonsense benchmark. They indicate that even when generated images are unrelated to the text context, our method performs comparably to the backbone, with further improvements using one generated image and the best performance achieved with k generated images.

Table 15: Accuracy on the visual commonsense task using different image sources.

Approach	Accuracy
Gemma-2B	33.4
Images representing different prompts	33.0
k-1 images representing different prompts	38.4
vLMIG (k generated images)	**45.4**

Scale on longer text inputs. We evaluated the effect of increasing the context length from a single sentence to ten sentences while keeping the number of generated images, k, at 10. Unlike early fusion methods, our late fusion approach is computationally efficient as it adds only a single layer. Given a backbone with K attention layers and a text context of size L, the backbone's complexity scales as $\mathcal{O}(KL^2)$, while vLMIG introduces an additional late fusion layer, resulting in an overall complexity of $\mathcal{O}((K+1)L^2)$. This means our approach incurs only a marginal increase in computational cost compared to the base model. As shown in Table 16, increasing the input length in the backbone model resulted in a proportional increase in inference running time. Notably, vLMIG exhibited a similar relative increase, suggesting that its computational cost is primarily driven by image processing rather than text length.

Table 16: Inference running time (ms) for different text input lengths.

Approach	Inference running time (ms)
Gemma-2B - single sentence	20.82
Gemma-2B - ten sentences	46.69
vLMIG - single sentence	750.71
vLMIG - ten sentences	779.34

Generated image prompting strategy. To determine the most effective image generation strategy for enhancing our model's interpretative and reasoning capabilities, we compared three methods: generating images from the last sentence, the entire textual context, and the latest k sentences. These strategies were evaluated across the same benchmarks: the Color task from ImageNetVC, PIQA, and BoolQ. Results, detailed in Table 17, show that generating images from the latest k sentences consistently leads to the best performance in PIQA and BoolQ test, providing a dynamic and contextually relevant visual representation. In the Color test, since all the questions include a single sentence, the results are the same.

The effect of the CLIP-fusion mechanism. To determine the effectiveness of our suggested CLIP-score fusion mechanism, we compare our method with and without CLIP-score fusion. Specifically, we consider

Table 17: Experiment results comparing different image generation strategies on *Visual Commonsense (includes the Color testset (Xia et al., 2023)), *Commonsense Reasoning (includes the PIQA testset (Bisk et al., 2019)), and *Reading Comprehension (includes the BoolQ testset (Clark et al., 2019)).

Style	*Visual Commonsense	*Commonsense Reasoning	*Reading Comprehension
Last Sentence	45.4	75.9	66.1
Full Context	45.4	76.6	66.4
K Latest Sentences	**45.4**	**77.7**	**67.0**

(i) generating a single image ($k = 1$), (ii) averaging the logits over ten different image generations and outputting the highest scoring token (no CLIP-fusion), (iii) as in (ii), but using max instead of average, and (iv) our method (which uses CLIP-fusion) with $k = 10$.

Results are presented in Table 18, showing that our CLIP-fusion approach consistently yields the best performance across all tasks. While averaging and maximizing logits from multiple image generations improve results over generating a single image, CLIP-fusion further boosts performance by effectively integrating visual representations.

Table 18: Experiment results comparing different strategies with and without CLIP-score fusion on *Visual Commonsense (includes the Color testset (Xia et al., 2023)), *Commonsense Reasoning (includes the PIQA testset (Bisk et al., 2019)), and *Reading Comprehension (includes the BoolQ testset (Clark et al., 2019)).

Method	*Visual Commonsense	*Commonsense Reasoning	*Reading Comprehension
Generating single image (i)	40.8	76.1	66.1
Average logits (ii)	44.6	76.5	66.5
Maximum (iii)	43.0	76.1	66.9
vLMIG with CLIP-fusion (iv)	**45.4**	**77.7**	**67.0**

Effect of attention mechanism in fusion layer. We compare the use of cross-attention and joint self-attention in the late fusion layer. As shown in Table 19, cross-attention performs similarly to joint self-attention in visual commonsense reasoning but leads to lower performance in commonsense reasoning and reading comprehension. This suggests that while cross-attention is effective for vision-related tasks, joint self-attention better integrates multimodal information for text-based reasoning.

Table 19: Comparison of cross-attention and joint self-attention in late fusion.

Approach	Visual Commonsense	Commonsense Reasoning	Reading Comprehension
Cross-attention	49.8	62.8	45.2
Joint self-attention	**50.1**	**65.1**	**48.9**

C.1 Inference Illustration

We present illustrative examples that highlight our method's inference process in C.1

C.2 Detailed Results

We present comprehensive results for all benchmarks discussed. First, the visual commonsense benchmark results are detailed in Table 20. Second, results for commonsense reasoning are provided in Table 21, and third, results for reading comprehension are provided in Table 22.

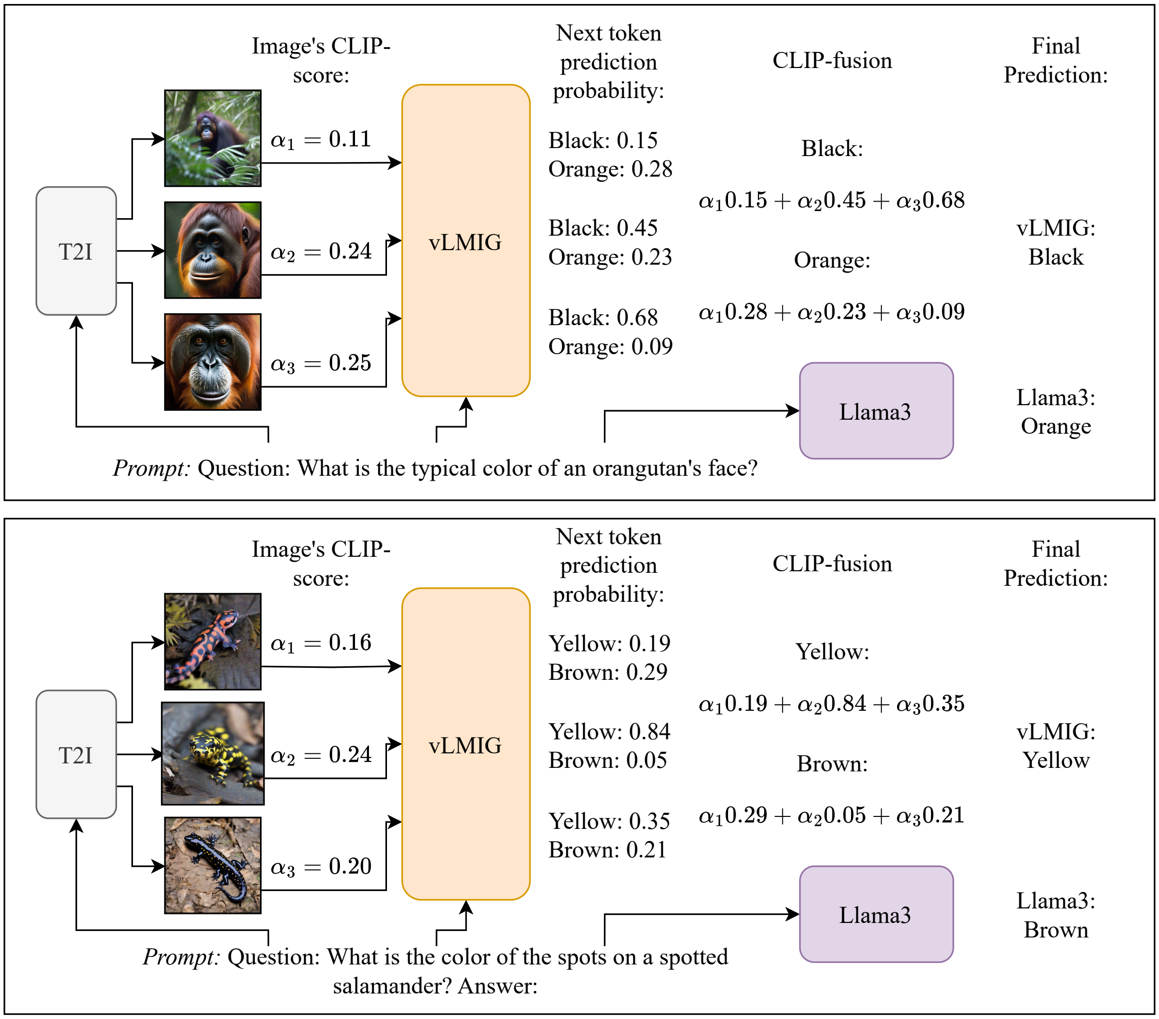


Figure 4: An illustration of our method's inference process, showcasing two examples. In the first example, we address the prompt: "What is the typical color of an orangutan's face?" vLMIG's final prediction correctly selects "Black," leveraging three generated images, each with their respective CLIP-scores influencing the CLIP-fusion result. vLMIG weights the images such that higher scores align with the black prediction, while Llama3 incorrectly chooses "Orange." Similarly, in the second example with the prompt "What is the color of the spots on a spotted salamander?" vLMIG correctly predicts "Yellow" by assigning the highest weight to the second image, whereas Llama3 selects "Brown."

Furthermore, Table 23 presents the complete results of our experiment investigating the impact of the number of images generated per inference on performance, as discussed in Figure 3.

Table 20: Visual commonsense performance per subtask, corresponding to Tab. 2.

		Tasks					
Model	Base Model	Color	Shape	Material	Component	Others	Avg.
Random	-	7.7	9.9	6.1	49.8	24.3	19.4
		Small-Scale Models					
GPT-2	-	17.1	21.8	27.1	**50.4**	35.1	30.3
vLMIG (ours)	GPT-2	**44.8**	**29.2**	**32.8**	49.9	**36.5**	**38.6**
		Mid-Scale Models					
Gemma-2B	-	33.4	34.1	52.3	59.5	49.0	45.6
vLMIG (ours)	Gemma-2B	**45.4**	**36.8**	**57.7**	**59.6**	**51.2**	**50.1**
Opt 2.7B	-	25.7	39.9	40.2	51.3	48.1	41.0
BLIP-2	Opt 2.7B	**37.8**	38.7	**53.1**	51.7	48.5	**46.0**
vLMIG (ours)	Opt 2.7B	35.5	**40.8**	48.5	**51.9**	**50.2**	45.4
		Large-Scale Models					
Llama3-8B	-	40.2	39.6	57.6	67.8	55.0	52.0
vLMIG (ours)	Llama3-8B	**48.0**	**40.9**	**60.4**	**69.7**	**56.0**	**55.0**

Table 21: Commonsense reasoning performance per subtask, corresponding to Tab. 2.

		Tasks							
Model	Base Model	PIQA	SIQA	HS	WG	ARC	OBQA	CQA	Avg.
		Small-Scale Models							
GPT-2	-	**62.6**	38.4	31.8	50.8	**34.8**	25.6	32.8	46.1
vLMIG (ours)	GPT-2	62.2	**38.9**	**31.9**	**51.5**	33.7	**27.4**	**34.0**	**46.7**
		Mid-Scale Models							
Gemma-2B	-	77.0	42.1	66.6	62.2	47.7	40.2	46.8	63.8
vLMIG (ours)	Gemma-2B	**77.7**	**44.0**	**67.0**	**62.5**	**49.1**	**40.3**	**50.6**	**65.1**
OPT-2.7B	-	73.4	42.4	**55.2**	**57.3**	47.0	**34.8**	46.5	50.9
BLIP-2	OPT-2.7B	68.8	40.0	54.2	53.8	40.3	33.0	38.8	46.9
vLMIG (ours)	OPT-2.7B	**73.8**	**43.8**	55.0	57.2	**48.5**	34.3	**49.1**	**51.6**
		Large-Scale Models							
Llama3-8B	-	80.3	46.1	**77.1**	**71.0**	**60.0**	44.6	54.8	72.0
vLMIG (ours)	Llama3-8B	**81.4**	**46.6**	76.5	70.8	59.8	**46.0**	**56.3**	**72.9**

Table 22: Reading comprehension performance per subtask, corresponding to Tab. 2.

		Tasks			
Model	Base Model	Boolq	SQuAD	QuAC	Avg.
Small-Scale Models					
GPT-2	-	47.7	27.4	16.6	30.5
vLMIG (ours)	GPT-2	**48.7**	**29.3**	**18.8**	**32.2**
Mid-Scale Models					
Gemma-2B	-	66.8	**57.4**	22.4	48.8
vLMIG (ours)	Gemma-2B	**67.0**	57.3	22.4	**48.9**
Opt 2.7B	-	**63.1**	50.5	**20.4**	44.6
BLIP-2	Opt 2.7B	59.9	40.4	16.5	38.9
vLMIG (ours)	Opt 2.7B	63.0	**51.5**	19.8	**44.7**
Large-Scale Models					
Llama3-8B	-	**79.3**	**69.2**	29.1	57.9
vLMIG (ours)	Llama3-8B	79.0	69.1	**29.3**	**58.0**

Table 23: Impact of the number of generated images per inference on performance per task, corresponding to Figure 3.

	Tasks			
Number of Images	Color	PIQA	BoolQ	Avg.
1	40.8	76.1	66.1	60.3
2	41.8	76.7	66.4	61.2
3	42.6	77.1	66.5	62.0
4	43.5	76.9	66.6	62.3
5	43.8	77.3	66.8	62.6
6	45.1	77.6	66.6	63.1
7	44.8	77.4	66.8	63.1
8	45.4	77.7	67.0	63.3
9	45.2	77.7	66.8	63.2
10	**45.4**	**77.7**	**67.0**	**63.3**