Improving the Efficiency of Visually Augmented Language Models

Paula Ontalvilla **Aitor Ormazabal** **Gorka Azkune**
HiTZ Center - Ixa, University of the Basque Country (UPV/EHU)
{paula.ontalvilla, aitor.ormazabal, gorka.azcune}@ehu.eus

Abstract

Despite the impressive performance of autoregressive Language Models (LM) it has been shown that due to reporting bias, LMs lack visual knowledge, i.e. they do not know much about the visual world and its properties. To augment LMs with visual knowledge, existing solutions often rely on explicit images, requiring time-consuming retrieval or image generation systems. This paper shows that explicit images are not necessary to visually augment an LM. Instead, we use visually-grounded text representations obtained from the well-known CLIP multimodal system. For a fair comparison, we modify VALM, a visually-augmented LM which uses image retrieval and representation, to work directly with visually-grounded text representations. We name this new model BLIND-VALM. We show that BLIND-VALM performs on par with VALM for Visual Language Understanding (VLU), Natural Language Understanding (NLU) and Language Modeling tasks, despite being significantly more efficient and simpler. We also show that scaling up our model within the compute budget of VALM, either increasing the model or pre-training corpus size, we outperform VALM for all the evaluation tasks.

1 Introduction

Autoregressive Language Models, such as GPT-4 (Achiam et al., 2023) and Llama (Dubey et al., 2024), are the reference systems for Natural Language Understanding and Generation. However, due to reporting bias in textual corpora (Shwartz and Choi, 2020), LMs lack visual knowledge, which means that they do not know the visual properties of our world, struggling to predict the typical colors, sizes and shapes of real objects, for instance (Alper et al., 2023; Zhang et al., 2022; Liu et al., 2022). Several researchers tried to overcome those problems augmenting LMs with visual knowledge (Tan and Bansal, 2020; Tang et al., 2021; Yang et al., 2022; Lu et al., 2022), but focusing specially on Masked Language Models (MLM). MLMs are limited for text generation and are not as versatile as autoregressive LMs. A recent example of visually augmenting autoregressive LMs is VALM (Wang et al., 2022), which leverages image retrieval and representation using a pretrained CLIP multimodal model (Radford et al., 2021) to improve next token prediction. To effectively use visual information, they add a Fusion Layer to a base LM, allowing textual tokens to attend visual representations before next token prediction. They show that VALM improves significantly the performance for Visual Language Understanding (VLU), without degrading the NLU and text generation capabilities of the base LM.

But image retrieval and representation are very resource intensive, significantly impacting training and inference times. For a improved efficiency, we propose to directly use visually-grounded textual representations, obtained from the CLIP model. Based on the VALM architecture, we input visually-grounded textual representations to the Fusion Layer, avoiding image retrieval and representation. We name this new model BLIND-VALM. As the result of our experiments we show that: i) BLIND-VALM is orders of magnitude faster than VALM for both training and inference; ii) BLIND-VALM performs on par with VALM for VLU, NLU and LM tasks; iii) maintaining within the compute budget of VALM, but increasing the size of the pretraining corpus or the base LLM, BLIND-VALM improves the results of VALM for all the evaluation tasks. All the code is publicly available[1].

2 Related Work

There are several approaches in the literature to augment Language Models with visual knowledge. Most of them focus on Masked Language Mod-

[1] https://github.com/paulaonta/Blind-VaLM

Proceedings of the 31st International Conference on Computational Linguistics, pages 5115–5122
January 19–24, 2025.

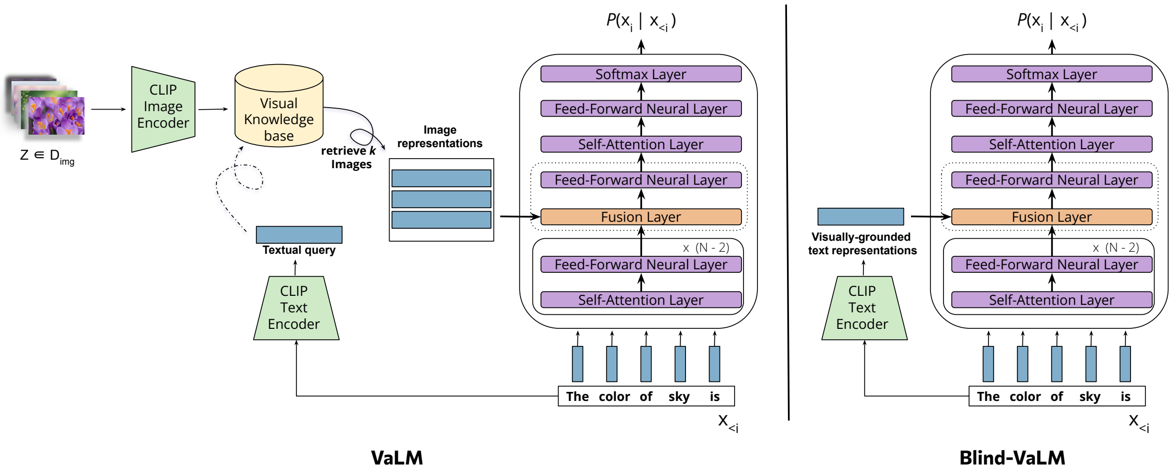

Figure 1: Architecture comparison of the original VALM (left) and our proposed BLIND-VALM (right).

els (MLM) such as BERT (Devlin et al., 2018) or RoBERTa (Liu et al., 2019), proposing different approaches: Vokenization (Tan and Bansal, 2020), VidLanKD (Tang et al., 2021), Z-LaVI (Yang et al., 2022) and iACE (Lu et al., 2022) are good examples. Given the limitations of MLMs for text generation, Tang et al. (2023) explore encoder-decoder architectures showing promising results for various generation tasks. But to the best of our knowledge, VALM (Wang et al., 2022) is the first work for augmenting decoder-only LMs with visual knowledge. More concretely, they augment a decoder-only LM with the so-called *Visual Knowledge Fusion Layer*, where contextual text representations generated by the LM are combined with the visual representations computed for retrieved images. VALM, similarly to previous approaches, uses images for training and inference, adding a significant overhead to the model. We use VALM as our base system, since it provides a solid framework to validate our hypothesis, i.e. LMs can be augmented without time-consuming image retrieval and representation. A similar idea is explored by concurrent work (Guo et al., 2023), but they do not cover decoder-only LMs.

3 BLIND-VALM architecture

The VALM architecture is composed of three main modules (Figure 1 left): 1) a backbone autoregressive LM (GPT2 (Radford et al., 2019)), 2) a text-to-image retrieval module based on CLIP (Radford et al., 2021), and 3) the Visual Knowledge Fusion Layer (Fusion Layer for short), to fuse the contextual text representations of the LM with the image representations retrieved for the input text. The intuition is that the retrieved visual representations should help to better predict the next token. For further details on the VALM architecture and Fusion Layer, see Wang et al. (2022).

To show that image retrieval and representation are not necessary to augment the backbone LM with visual knowledge, we make one modification to the VALM architecture: instead of using the CLIP image encoder representations of the retrieved images, BLIND-VALM directly uses CLIP text encoder representations of the text itself (see Figure 1 right).

More formally, given an input text sequence $\{x_i\}_{i=1}^{N}$, let H^0 denote its corresponding embedding sequence and let $H^l = LM_l(H^{l-1}), l \in [1, L-2]$ denote the contextual representations of the backbone LM for the first $L-2$ layers. Then, let $H_g = \text{CLIP}_{text}(\{x_i\}_{i=1}^{N})$ denote grounded representations over the same input text sequence obtained from CLIP. As in VALM, combine both representations of the input using the Fusion Layer, such that $H^{L-1} = FusionLayer(H^{L-2}, H_g)$, and apply a final transformer layer to obtain the final representation: $H^L = LM_L(H^{L-1})$.

In other words, we remove the image retrieval aspect from VALM, where $H^{L-1} = FusionLayer(H^{L-2}, \{\text{CLIP}_{img}(I_i)\}_{i=1}^{K})$, by replacing the retrieved image representations $\text{CLIP}_{img}(I_i)$ with a single textual CLIP representation, following the intuition that the latter already encodes relevant visual information as the product of the contrastive training process with images.

Model	Color (ACC ↑)			Shape (ACC ↑)	Size (ACC ↑)		AVG
	MemoryC	ColorTerms	VCT	ShapeITC	RelativeS	TNWT	
VALM	47.09	41.88	20.46	**40.45**	**26.03**	23.94	33.30
BLIND-VALM	**47.20**	**46.37**	**22.60**	40.07	25.43	**25.18**	**34.48**

Model	Wikitext-103	Lambada		SST-2	DBPedia	AGNews	MPQA
	PPL ↓	PPL ↓	ACC ↑	ACC ↑	ACC ↑	ACC ↑	ACC ↑
VALM	43.68	**45.69**	35.61	44.66	66.77	41.63	67.95
BLIND-VALM	**43.46**	45.78	**39.06**	**44.71**	**73.65**	**44.97**	**71.65**

Table 1: BLIND-VALM matches VALM on VLU, NLU and LM tasks when trained on the same setup, while being significantly more efficient to train.

Model	Color (ACC ↑)			Shape (ACC ↑)	Size (ACC ↑)		AVG
	MemoryC	CTerms	VCT	ShapeITC	RelativeS	TNWT	
VALM	47.09	41.88	20.46	40.45	**26.03**	23.94	33.30
BLIND-VALM	47.20	46.37	22.60	40.07	25.43	25.18	34.47
BLIND-VALM+	45.97	48.71	20.51	**43.64**	25.33	**25.40**	34.93
BLIND-VALM-M	**47.60**	**48.93**	**24.09**	43.33	24.36	24.93	**35.54**

Model	Wikitext-103	Lambada		SST-2	DBPedia	AGNews	MPQA
	PPL ↓	PPL ↓	ACC ↑	ACC ↑	ACC ↑	ACC ↑	ACC ↑
VALM	43.68	45.69	35.61	44.66	66.77	41.63	67.95
BLIND-VALM	43.46	45.78	39.06	44.71	73.65	44.97	71.65
BLIND-VALM+	40.95	43.83	39.05	44.66	**74.42**	**50.44**	**78.10**
BLIND-VALM-M	**33.95**	**37.63**	**45.08**	**59.00**	70.57	48.28	55.65

Table 2: When trained within the same compute budget by increasing either model size (BLIND-VALM-MEDIUM, referred to as BLIND-VALM-M in the table) or pre-training compute (BLIND-VALM+), our approach outperforms VALM on both VLU and NLU tasks.

4 Experimental setup

We want to show that actual retrieval and representation of images is not necessary to augment a LM with visual knowledge, and that directly using textual representations from a visually grounded text encoder instead is sufficient. To that end, we initially pre-train BLIND-VALM and VALM models in a comparable setting, which we now describe.

Text Corpus. Following the original VALM (Wang et al., 2022), we use the English corpus of CC-100 (Conneau et al., 2020) as the pre-training text corpus for all models. Due to limited access to compute, we only consume 10.5B tokens for pre-training, which accounts for around 19% of the English portion of the corpus.

Image data and retrieval module. VALM requires an image database and a vector retrieval module. We use a *FAISS* (Johnson et al., 2019) index, trained on the exact same setup as the original VALM, based on the GPT2-SMALL architecture, as detailed in Appendix A.

Pre-training hyperparameters. We train both models on the exact same setup, following a configuration similar to the original VALM. Details can be found in Appendix A.

Due to the increased efficiency of our architecture, our model employs significantly less compute than the original: BLIND-VALM was trained on 530 GPU-hours, while the VALM baseline required 1.2K GPU-hours, making our approach **2.2x faster to train**. We train all our models on a cluster of 8 A100 GPUs.

Evaluation. We evaluate our models on VLU, NLU and LM tasks (see Appendix B for details).

For **VLU**, we focus on three basic visual properties of objects: color, shape and size. We evaluate color knowlege on the following datasets: Memory Color (Norlund et al., 2021), Color Terms (Bruni et al., 2012) and ViComTe (color subset) (Zhang et al., 2022). We evaluate knowledge about shape on the ShapeITC (Alper et al., 2023) dataset, and knowledge of size on the RelativeSize (Bagherinezhad et al., 2016) and Things Not Written in Text (Liu et al., 2022) datasets.

We evaluate **NLU** capabilities on four downstream tasks: two sentiment analysis tasks on the SST-2 and MPQA datasets (Socher et al., 2013; Wiebe et al., 2005), and two topic classification tasks on the AGNews and DBPedia datasets (Auer et al., 2007; Zhang et al., 2015). Additionally, we evaluate pure language modeling ability by measuring perplexity on the Wikitext-103 and Lambada datasets (Merity et al., 2016; Paperno et al., 2016). For Lambada, we also report accuracy at predicting the last word of each sentence, following the original VALM work.

Scaling BLIND-VALM. It is important to note that, since BLIND-VALM does not require an actual image retrieval step, it is significantly more efficient at both training and inference time.[2] Taking advantage of this increased efficiency, we explore two ways of scaling up BLIND-VALM, within the compute budget of our VALM baseline.

(i) Scaling model size. We train **BLIND-VALM-MEDIUM**, by switching the LM backbone architecture to GPT2-MEDIUM instead of GPT2-SMALL. See Appendix A for details. This larger model was trained on 595 GPU-hours, still within the compute budget of the VALM baseline.

(ii) Scaling pre-training compute. We train **BLIND-VALM+**, by simply continuing pre-training the baseline BLIND-VALM for longer, until we reach a total of 88255 steps, which corresponds to 23.1B tokens (42% of CC-100). Again, thanks to the increased efficiency of our approach, this model is compute-matched[3] to the original VALM baseline, taking a total of 1.17K GPU-hours to train.

5 Results

We next describe our main results.

BLIND-VALM matches VALM on VLU, NLU and LM tasks. Table 1 shows results for BLIND-VALM and VALM trained on the same setup, as described in §4. We observe that our approach matches the original VALM on VLU tasks, as well as NLU and LM tasks: it achieves an average score 1.18 points higher in VLU tasks, and outperforms VALM on 6/7 NLU & LM tasks. This supports our hypothesis that **actually retrieving and encoding images is not required for visual augmentation**, since simply utilizing textual representations from an already visually grounded text encoder works equally well. Additionally, as described in section §4 **BLIND-VALM is 2.2x faster** to train, since it skips the time consuming vector retrieval step. Speedups are even more significant at inference time, since generation is not as compute-bound and retrieval latency plays a bigger role.

BLIND-VALM outperforms VALM, when trained within the same compute budget. Table 2 shows results for two scaled-up BLIND-VALM variants, obtained through scaling either pre-training compute or model size, as described in §4. We observe that both variants outperform VALM, while being trained within the same compute budget as it. For example, in the case of BLIND-VALM-MEDIUM, we outperform VALM by 2.2 points on average for VLU tasks, and outperform VALM on 6/7 NLU & LM tasks.

6 Conclusions

In this work we test the hypothesis that explicit image retrieval is not necessary to augment an LM with visual information. To that end, we train a modified variant of VALM (Wang et al., 2022), which we call BLIND-VALM, by replacing the retrieved image encoding vectors with textual embeddings obtained from the visually grounded CLIP encoder (Radford et al., 2021).

Our results show that BLIND-VALM matches VALM when trained on the same data, while being significantly more efficient to train. Additionally, scaling up our model within the compute budget of VALM, our approach outperforms VALM. Overall, these results show that simply leveraging the textual encoding from an already visually grounded CLIP encoder is enough to obtain the same gains on visual tasks as VALM, supporting our hypothesis that actual image retrieval is not essential.

These results open up new avenues in the line of work of visually augmenting language models,

[2]The efficiency gains are two-fold: 1) since we only use a single CLIP text representation in the Fusion Layer, this layer requires less floating-point operations (FLOPs), and 2) since we do away with the dense vector database retrieval, training and inference latency is significantly reduced, leading to improved wall-clock efficiency.

[3]Note that we use *compute-matched* in the wall-clock time sense here, not the FLOPs sense, since the time consuming dense vector retrieval step we remove is not reflected by FLOPs.

beyond the paradigm of image retrieval. Our findings allow for more efficient visual augmentations, which will result on broader exploration capacity for future works.

7 Limitations

Our work is focused on English only, due to the size and accessibility of the resources, i.e. text corpora, image-text datasets, different pre-trained models and evaluation benchmarks. However, it would be very interesting to extend the work to other languages.

The VLU evaluation is limited to visual object properties, such as color, shape and size. But, visual language is broader and extending evaluation benchmarks would allow to better understand how VLU evolves in pure and visually-augmented LMs.

Finally, we use the original CLIP model (Radford et al., 2021) to visually-augment LMs, as done in VaLM. Nowadays there are more powerful multimodal models and it would be very interesting to explore how those better models impact in the knowledge acquisition of visually-augmented LMs.

Acknowledgments

This work is partially supported by the Ministry of Science and Innovation of the Spanish Government (AWARE project TED2021-131617B-I00, DeepR3 project TED2021-130295B-C31), project funded by MCIN/AEI/10.13039/501100011033 and European Union NextGeneration EU/PRTR, and the Basque Government (IXA excellence research group IT1570-22 and IKER-GAITU project).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Morris Alper, Michael Fiman, and Hadar Averbuch-Elor. 2023. Is bert blind? exploring the effect of vision-and-language pretraining on visual language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer.

Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi. 2016. Are elephants bigger than butterflies? reasoning about sizes of objects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Hangyu Guo, Kun Zhou, Wayne Xin Zhao, Qinyu Zhang, and Ji-Rong Wen. 2023. Visually-augmented pretrained language models for nlp tasks without images. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14912–14929.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. 2022. Things not written in text: Exploring spatial commonsense from visual signals. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2365–2376.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yujie Lu, Wanrong Zhu, Xin Wang, Miguel Eckstein, and William Yang Wang. 2022. Imagination-augmented natural language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, pages 4392–4402.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Tobias Norlund, Lovisa Hagström, and Richard Johansson. 2021. Transferring knowledge from vision to language: How to achieve it and how to measure it? In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–162.

Denis Paperno, German David Kruszewski Martel, Angeliki Lazaridou, Ngoc Pham Quan, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda Torrent, Fernández Raquel, et al. 2016. The lambada dataset: Word prediction requiring a broad discourse context. In *The 54th Annual Meeting of the Association for Computational Linguistics Proceedings of the Conference: Vol. 1 Long Papers*, volume 3, pages 1525–1534. ACL.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6863–6870, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Hao Tan and Mohit Bansal. 2020. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2066–2080.

Tianyi Tang, Yushuo Chen, Yifan Du, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Learning to imagine: Visually-augmented natural language generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9468–9481.

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

Weizhi Wang, Li Dong, Hao Cheng, Haoyu Song, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2022. Visually-augmented language modeling. In *The Eleventh International Conference on Learning Representations*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39:165–210.

Yue Yang, Wenlin Yao, Hongming Zhang, Xiaoyang Wang, Dong Yu, and Jianshu Chen. 2022. Z-lavi: Zero-shot language solver fueled by visual imagination. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1186–1203.

Chenyu Zhang, Benjamin Van Durme, Zhuowan Li, and Elias Stengel-Eskin. 2022. Visual commonsense in pretrained unimodal and multimodal models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5321–5335.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

A Training hyper-parameters

We train all models using an inverse-square-root learning rate schedule, the ADAM optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, a peak LR of $2e-3$, a weight-decay of 0.01, dropout of 0.1, 4000 warmup steps, a global batch size of 512, and a sequence length of 512 tokens. We train both models for 40600 steps.

Since both BLIND-VALM and the original VALM share the limitation of needing tokenization to be compatible with the CLIP used for retrieval, use use the same BPE tokenizer as CLIP and the original GPT2.

VALM, BLIND-VALM and BLIND-VALM+ use the GPT2-SMALL architecture, comprising 124M parameters, for the LM backbone, and the CLIP-RN50X16 model for the visually grounded text-image encoder, with a total of 85M parameters on the text encoder.

For BLIND-VALM-MEDIUM, we switch the LM backbone architecture to GPT2-MEDIUM, with a total of 345M parameters. We switch the CLIP encoder to the CLIP-RN50x64 version with 151M parameters, which employs the hidden-size corresponding to GPT2-Medium, required by the Fusion Layer.

B Evaluation details

To evaluate VLU and NLU capabilities we follow the prompting approach designed by VaLM (Wang et al., 2022), with some slight modifications. Tables 3, 4 and 5 show the prompts we use for color, shape and size evaluation. Similarly, Table 6 defines the prompts for all the NLU tasks we consider. The reported results in the paper are always the average of the results obtained for all the prompts for a given task, with the objective of avoiding any bias towards different prompting choices.

Task	**Dataset**	**Prompt**	**Labels**
Object Color Reasoning	Memory Colors and Color Terms	Q: What is the color of [DESCRIPTOR] [ITEM]? A: It is [Label] Q: What is the colour of [DESCRIPTOR] [ITEM] ? A: It is [Label] What is the color of [DESCRIPTOR] [ITEM]? It is [Label] What is the colour of [DESCRIPTOR] [ITEM]? [Label] The color of [DESCRIPTOR] [ITEM] is [Label] The usual color of [DESCRIPTOR] [ITEM] is [Label] [DESCRIPTOR] [ITEM] usually has the color of [Label] What is the usual color of [DESCRIPTOR] [ITEM]? [Label] What is the typical color of [DESCRIPTOR] [ITEM]? [Label]	{red, white, orange, green, blue, yellow, purple, black, pink, grey, brown}
	ViComTe	[ITEM] can be of color [ITEM] has color The color of [ITEM] can be The color of the [ITEM] is [ITEM] is This [ITEM] is [ITEM] is of color	{red, white, orange, green, blue, yellow, purple, black, pink, grey, brown, silver}

Table 3: The prompts and prediction labels used in object color reasoning.

Task	**Prompt**	**Labels**
Object Shape Reasoning	[ITEM] can be shape of [ITEM] has shape of [ITEM] is of shape The shape of [ITEM] can be The shape of the [ITEM] is [ITEM] is This [ITEM] is [ITEM] can be shape [ITEM] has shape	{circle, rectangle, triangle}

Table 4: The prompts and prediction labels used in object shape reasoning.

Task		**Prompt**	**Labels**
Object Size Reasoning	Object prediction	Which is bigger? [ITEMA] or [ITEMB] Which is smaller? [ITEMA] or [ITEMB] Which is larger? [ITEMA] or [ITEMB] Which is tinier? [ITEMA] or [ITEMB] Which has a bigger size? [ITEMA] or [ITEMB] Which has a bigger size? [ITEMA] or [ITEMB] Which has a smaller size? [ITEMA] or [ITEMB] Which one is larger in size? [ITEMA] or [ITEMB] Which one is smaller in size? [ITEMA] or [ITEMB]	{ITEMA, ITEMB}
	Size prediction	[ITEMA] is larger or smaller than [ITEMB]? [ITEMB] is larger or smaller than [ITEMA]? The size of [ITEMA] is larger or smaller than [ITEMB]? The size of [ITEMB] is larger or smaller than [ITEMA]? [ITEMA] has a larger or smaller size than [ITEMB]? [ITEMB] has a larger or a smaller size than [ITEMA]?	{larger, smaller}

Table 5: The prompts and prediction labels used in object size reasoning.

Task	**Dataset**	**Prompt**	**Labels**
Natural Language Understanding	SST-2	Review: [Sentence] Sentiment: [Label]	{Positive, Negative}
	MPQA	Review: [Sentence] Sentiment: [Label]	{Positive, Negative}
	AGNews	input: [Sentence] type: [Label]	{world,. . . , technology}
	DBPedia	input: [Sentence] type: [Label]	{company, school,. . . , book}

Table 6: The prompts and prediction labels used in 4 natural language understanding datasets. The labels for AGNews are {world, sports, business, technology} and the labels for DBPedia are {company, school, artist, athlete, politics, transportation, building, nature, village, animal, plant, album, film, book}.