

Frozen Transformers in Language Models Are Effective Visual Encoder Layers

Ziqi Pang Ziyang Xie* Yunze Man* Yu-Xiong Wang

University of Illinois Urbana-Champaign

{ziqip2, ziyang8, yunzem2, yxw}@illinois.edu

<https://github.com/ziqipang/LM4VisualEncoding>

Abstract

This paper reveals that large language models (LLMs), despite being trained solely on text data, are surprisingly strong encoders for purely visual tasks in the absence of language. Even more intriguingly, this can be achieved by a simple yet previously overlooked strategy – employing a frozen transformer block from pre-trained LLMs as a constituent encoder layer to directly process visual tokens. Our work pushes the boundaries of leveraging LLMs for computer vision tasks, significantly departing from conventional practices that typically necessitate a multi-modal vision-language setup with associated language prompts, inputs, or outputs. We demonstrate that our approach consistently enhances performance across a diverse range of tasks, encompassing pure 2D and 3D visual recognition tasks (e.g., image and point cloud classification), temporal modeling tasks (e.g., action recognition), non-semantic tasks (e.g., motion forecasting), and multi-modal tasks (e.g., 2D/3D visual question answering and image-text retrieval). Such improvements are a general phenomenon, applicable to various types of LLMs (e.g., LLaMA and OPT) and different LLM transformer blocks.

We additionally propose the “information filtering hypothesis” to explain the effectiveness of pre-trained LLMs in visual encoding – the pre-trained LLM transformer blocks discern informative visual tokens and further amplify their effect. This hypothesis is empirically supported by the observation that the feature activation, after training with LLM transformer blocks, exhibits a stronger focus on relevant regions. We hope that our work inspires new perspectives on utilizing LLMs and deepening our understanding of their underlying mechanisms.

1. Introduction

Large language models (LLMs), trained on massive amounts of text data, have recently demonstrated remarkable potential across various tasks, extending beyond their original linguistic domain. For example, in the field of computer vision, LLMs exhibit the ability to interact with visual tokens and *decode* them into tokenized output. This

is commonly achieved in a multi-modal vision-language framework that incorporates the language modality, as exemplified by either projecting visual tokens to LLMs via linear layers [23, 27, 33] or employing cross-attention mechanisms between visual and language tokens [1, 25]. As we explore the limits of utilizing LLMs for computer vision, an interesting question arises: can LLMs effectively handle *exclusively* visual tasks, without any reliance on language?

This paper provides a positive demonstration of feasibility in addressing this question, by introducing a straightforward yet previously overlooked approach: incorporating a *frozen* transformer block from a *pre-trained* LLM as a general-purpose visual *encoder* layer, directly processing the visual tokens. Specifically, as illustrated in Fig. 1a and Fig. 1b, our design involves the following steps: (1) extract a *frozen* LLM transformer block and append it on top of the original visual encoder; (2) insert *trainable* linear layers before and after the added LLM block to align the feature dimensions; and (3) *freeze* the LLM transformer while optimizing the other modules as usual during training.

Surprisingly, this simple design enhances performance across a *wide spectrum of tasks*, including 2D and 3D recognition (image and point cloud classification), video understanding (action recognition), and non-semantic (motion forecasting) tasks. In addition to these purely visual tasks, our approach is also effective in multi-modal tasks (2D/3D visual question answering and image-text retrieval).

Our discovery of using a pre-trained LLM transformer block as a visual encoder layer is intriguing, because it significantly deviates from the conventional designs of vision-language models (VLMs). In particular, our treatment of LLM transformers as *encoders* (1) operates independently of language prompts, inputs, or outputs; (2) allows for training from scratch without the need for pre-trained backbones like CLIP [37]; and (3) decouples and simplifies the usage of LLMs into separate transformer blocks.

However, one crucial question remains: *why are LLMs effective in visual encoding, given that they have been exclusively trained on text and have never encountered visual input?* To this end, we propose the *information filtering hypothesis*: the pre-trained LLM transformer blocks discern informative visual tokens and further amplify their contri-

*Equal contribution.

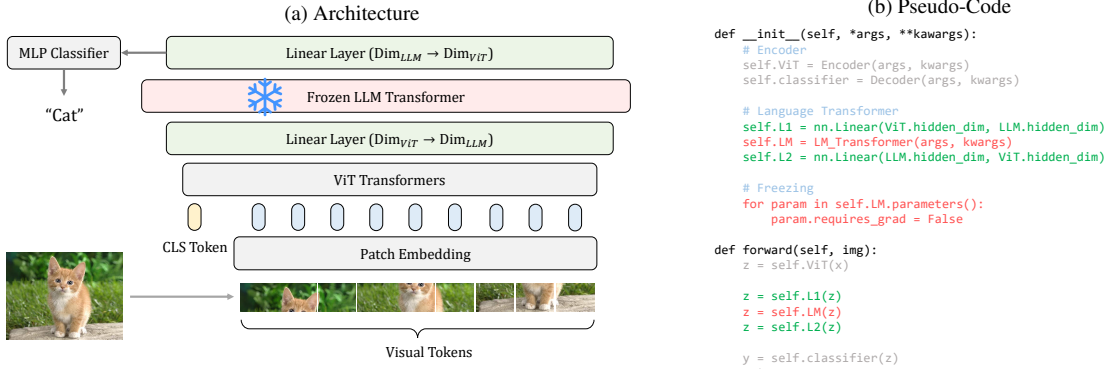


Figure 1. Our straightforward method of using a *frozen* transformer block from *pre-trained* LLMs as a visual encoder layer. Visualized with an example of ViT [8]. **(a)** Our design simply appends a frozen transformer block (pink) on top of the regular visual encoder (gray). Only two trainable linear layers (green) are added to align the feature dimensions. **(b)** Pytorch-style pseudo-code our simplicity.

bution to the latent representation. This hypothesis stems from our observation across multiple tasks, where the feature activation consistently exhibits a stronger focus on relevant regions, after integrating the frozen LLM transformer.

In summary, we have made the following contributions:

- We discover that *using a frozen transformer from LLMs as a visual encoder layer* facilitates a diverse range of tasks via a simple yet under-explored approach.
- We demonstrate that the benefits of frozen LLM transformers in visual encoding are a general phenomenon across diverse tasks.
- We propose the *information filtering hypothesis* to explain the effectiveness of frozen LLM transformers in processing visual tokens: the incorporated LLM blocks distinguish the informative tokens and amplify their effect.

2. Method

Framework design. We formally introduce using a pre-trained LLM transformer as a visual encoder layer shown in Fig. 1a. Without loss of generality, we consider a neural network that maps an input x to latent representation z and predicts labels y with an encoder \mathbf{F}_E and decoder \mathbf{F}_D ,

$$\mathbf{F}_E(x) \rightarrow z, \quad \mathbf{F}_D(z) \rightarrow y. \quad (1)$$

Then a *single pre-trained* transformer block from an LLM like LLaMA [42], denoted as \mathbf{F}_{LM} , is inserted between the encoder \mathbf{F}_E and decoder \mathbf{F}_D . As the feature dimensions are different between the encoder \mathbf{F}_E and the language transformer \mathbf{F}_{LM} , we employ two linear layers \mathbf{F}_L^1 and \mathbf{F}_L^2 before and after \mathbf{F}_{LM} to align the dimensionality. These modify the neural network into

$$\mathbf{F}_E(x) \rightarrow z, \quad \mathbf{F}_L^2 \cdot \mathbf{F}_{LM} \cdot \mathbf{F}_L^1(z) \rightarrow z', \quad \mathbf{F}_D(z') \rightarrow y. \quad (2)$$

In the training stage, the pretrained transformer \mathbf{F}_{LM} remains *frozen*, as in the pseudo-code of Fig. 1b, while all the other modules are trained normally, including \mathbf{F}_L^1 and \mathbf{F}_L^2 .

Comparison with vision-language models. Our approach appears similar to recent vision-language models (VLMs) at the first glance, such as FROMAGE [23], where linear layers directly project visual features to the input

Model	ImageNet	ImageNet-C	ImageNet-A	ImageNet-SK	ImageNet-R
ViT-T	72.1	43.9	7.7	19.6	32.3
ViT-T-LLaMA	73.2	45.8	8.7	20.6	33.8
ViT-S	80.1	57.2	20.5	28.9	42.1
ViT-S-LLaMA	80.7	58.7	22.7	30.5	42.8
ViT-B	80.6	60.5	23.4	31.9	43.5
ViT-B-LLaMA	81.7	62.1	26.9	33.2	44.3

Table 1. Incorporating the single transformer block from LLaMA to ViT models consistently improves both accuracy (ImageNet) and robustness (ImageNet-{C,A,SK,R}).

space of LLMs. However, our approach is different, because the linear layer \mathbf{F}_L^1 does not necessarily align the visual representation z into the language space. Concretely, this is reflected in three aspects: (1) **Independence of visual pre-training.** Our paradigm supports training-from-scratch without relying on pre-trained visual encoders like CLIP [37]. (2) **Independence of language.** Our framework can function without language-based input or prompts, and it is applicable for general visual representation learning instead of only vision-language tasks. (3) **Independence of transformer blocks.** Previous VLMs treat an LLM as a coherent module, while our framework separates each transformer block as an independent layer for visual encoding.

3. Applications to Diverse Visual Tasks

We instantiate our framework to various tasks and observe the wide applicability of pretrained LLM transformers. Our experiments cover 2D (image classification) and 3D (point cloud classification), single-frame and multi-frame (action recognition). The investigation of non-semantic motion forecasting and vision-language tasks are in . By default, we adopt the last transformer block from LLaMA-7B [42]. Our framework achieves consistent and significant improvements across these tasks. The experiments on motion forecasting, vision-language tasks, and ablation studies are in the supplementary materials.

3.1. Image Classification

Image classification is the most common challenge for representation learning. We conduct experiments on ImageNet1k [7], and additionally evaluate on robustness benchmarks: corrupted images from ImageNet-C [17], natu-

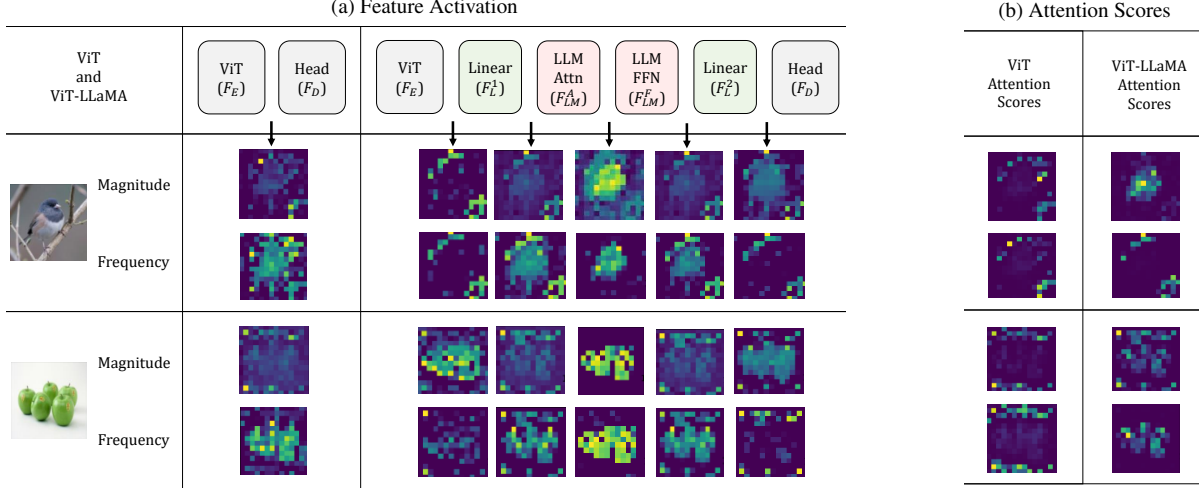


Figure 2. (a) Feature activation regarding both magnitudes and frequencies of features. We highlight that ViT-LLaMA demonstrates the emergent tendency of object segmentation compared with ViT, indicating its ability to select informative tokens. (b) Attention scores between the CLS and visual tokens. The attention from ViT is commonly noisy (left). Though ViT-LLaMA improves the concentration on a few heads, most of the attention heads are still noisy. Both good and bad attention from ViT-LLaMA are sampled for demonstration.

ral adversarial images from ImageNet-A [19], and out-of-distribution images from ImageNet-SK [44] and ImageNet-R [20]. We select ViT [8] as our baseline. Following the notation in Eqn. 2, the encoder \mathbf{F}_E is the set of self-attention transformer blocks and the decoder \mathbf{F}_D denotes the linear classifier, respectively. We train both the baseline ViT models and ViT-LLaMA from scratch following the same configuration of DeiT [41].

The accuracy of ViT models consistently improves after incorporating the frozen LLaMA transformer block as in Table 1, including both the accuracy on clean ImageNet images and robustness on corrupted or adversarial images.

3.2. Point Cloud Recognition

Point cloud classification handles a fundamentally different modality compared to images. The models predict the labels by processing unordered 3D points and understanding the geometry. Our experiments are on ScanObjectNN [43]. It contains three splits: background points (BG), foreground (OBJ), and clipped (T50) points. We adopt Point-BERT [46] and load its pretrained parameters on ShapeNet. Then we append the LLaMA transformer after its final attention block before finetuning on point cloud classification datasets.

As shown in Table 2a, our approach improves the accuracy for point cloud classification, further supporting the applicability of using a frozen LLM transformer as a visual encoding layer. This experiment also shows that our framework is compatible with finetuning setups, in addition to training-from-scratch scenarios in Sec. 3.1.

3.3. Action Recognition

For the video modality, we apply the pretrained LLM transformer block to action recognition, where the algorithm predicts the action labels of video clips. We choose the bench-

(a) Point Cloud Recognition				(b) Action Recognition		
Model	BG	OBJ	T50	Model	Acc1	Acc5
Point-BERT	87.4	88.1	83.1	ViT-S	64.71	89.15
+LLaMA	88.0	88.5	83.8	+LLaMA	65.89	89.93

Table 2. Frozen LLaMA transformer improves point cloud recognition on ScanObjectNN and action recognition on SSv2.

mark of “Something-something-v2” (SSv2) [14] for evaluation, because it highlights the challenge of understanding cross-frame movement, instead of relying on single-frame semantics. We follow VideoMAE [40] and adopt the simple yet effective ViT backbones. Different from patches of tokens in 2D images, the video tokens are cubes spanning both spatially and temporally. Our training setup also adopts the two-step practice in VideoMAE: (1) initialize ViT transformers from MAE [16] pre-training; (2) add the LLM transformer and then finetune on the SSv2 dataset using the same configuration of VideoMAE. The LLaMA transformer enhances the accuracy for both ViT-S and ViT-B in Table 2b, supporting our applicability for videos.

4. Information Filtering Hypothesis

This section aims to explain how a frozen and pre-trained LLM transformer benefits visual tasks. Intuitively, our hypothesis can be stated as:

Information filtering hypothesis. The pretrained LLM transformer functions as a “filter” that *distinguishes the informative tokens* and *amplifies their contribution for the prediction*, in the form of enlarged magnitudes or frequencies in the feature activation.

We have provided a sketch of our derivation below. More quantitative and qualitative evidence of our hypothesis are in the supplementary materials.

Emergent concentration on informative tokens. Our

hypothesis originates from the emergent behavior that *the feature activation highlights the informative tokens* after adding a pre-trained LLM transformer. In the analysis, we extract the activation of features after each layer as Fig. 2a, including the original ViT \mathbf{F}_E , the attention layer \mathbf{F}_{LM}^A and feedforward network \mathbf{F}_{LM}^F in the LLM transformer, and the linear layers \mathbf{F}_L^1 and \mathbf{F}_L^2 . Notably, the feature activation is calculated regarding both *magnitudes* (L2-norm after centering) and *frequencies* (L2-norm of angles after Fourier transformation). The different layers in Fig. 2a indeed show diverse preferences over magnitudes or frequencies.

As clearly shown in Fig. 2a, the token activation better captures the regions of target objects after adding the LLM transformer, especially the magnitudes of \mathbf{F}_L^2 and frequencies of \mathbf{F}_{LM}^A . Their tendency of segmentation is a surprising discovery because emergent segmentation only exists in self-supervised learned [4] or specially-designed [6] ViTs. More importantly, the activation’s concentration on the target object directly supports our hypothesis as evidence of selecting the informative tokens.

Noisy attention scores. In contrast to the feature activation, attention scores struggle to capture the relevant visual tokens for prediction. We investigate the attention scores between the CLS token and visual tokens in the last transformer block, which are the last self-attention block in \mathbf{F}_E for ViT and the transformer of \mathbf{F}_{LM} for ViT-LLaMA, respectively. Ideal attention scores that distinguish the target object should exhibit object segmentation patterns like DINO [4]. However, supervised ViT models commonly have noisy attention scores (left part in Fig. 2b). Although ViT-LLaMA illustrates the ability of emergent segmentation in a few attention heads, most of the attention scores also suffer from scattering and noisiness. These observations contrast the feature activation and indicate that the benefits of LLM transformers cannot be simply attributed to attention scores, since attention scores fail to reliably contribute correct visual tokens.

Deriving the amplification of informative tokens. Our derivation builds upon two observed pieces of evidence: (1) visual tokens concentrating on informative regions; and (2) noisy attention scores. These lead to our hypothesis and explain the benefits of frozen LLM transformers in amplifying the contribution of relevant visual tokens for the target task.

5. Conclusions

In this work, we explore the unexpected capability of large language models (LLMs) as encoders for visual tasks, a significant departure from their conventional text-based applications. By seamlessly integrating a frozen transformer block from pre-trained LLMs into visual encoders, we observe consistent performance enhancements across diverse visual challenges. This phenomenon, underpinned by our proposed information filtering hypothesis, highlights the inherent adaptability and versatility of LLMs

for more general representation learning. We hope that our insights will catalyze further exploration into the uncharted fields of LLM applications and foster innovative strategies to harness their potential in novel ways.

References

- [1] Jean-Baptiste Alayrac et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022. 1
- [2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3D question answering for spatial scene understanding. In *CVPR*, 2022. 6
- [3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 6
- [4] Mathilde Caron et al. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 4
- [5] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019. 6
- [6] Timothée Darcet et al. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 4
- [7] Jia Deng et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 10
- [8] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2, 3, 13
- [9] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *CVPR*, 2022. 6, 10, 12, 13
- [10] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *CVPR*, 2020. 6, 12, 13
- [11] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *TPAMI*, 2022. 8, 10
- [12] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *ICML*, 2021. 12
- [13] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large mini-batch SGD: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 12
- [14] Raghav Goyal et al. The” something something” video database for learning and evaluating visual common sense. In *ICCV*, 2017. 3
- [15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 6, 13
- [16] Kaiming He et al. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 3, 12

- [17] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2018. 2
- [18] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 6
- [19] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021. 3
- [20] Dan Hendrycks et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021. 3
- [21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 6
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12
- [23] Jing Yu Koh et al. Grounding language models to images for multimodal inputs and outputs. *ICML*, 2023. 1, 2
- [24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 13
- [25] Junnan Li et al. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1
- [26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 13
- [27] Xudong Lin et al. Towards fast adaptation of pretrained contrastive models for multi-channel video-language retrieval. In *CVPR*, 2023. 1
- [28] 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. *arXiv preprint arXiv:1907.11692*, 2019. 6, 13
- [29] Yicheng Liu, Jinghui Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021. 6, 12, 13
- [30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 12, 13
- [31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 12
- [32] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3D: Situated question answering in 3d scenes. In *ICLR*, 2023. 6, 10, 13
- [33] Jack Merullo et al. Linearly mapping from image to text space. *ICLR*, 2023. 1
- [34] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *NeurIPS*, 2011. 13
- [35] 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. In *ICCV*, 2015. 6, 13
- [36] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 6, 10
- [37] Alec Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 6, 10, 13
- [38] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 13
- [39] Baifeng Shi, Trevor Darrell, and Xin Wang. Top-down visual attention from analysis by synthesis. In *CVPR*, 2023. 6, 13
- [40] Zhan Tong et al. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NeurIPS*, 2022. 3, 12
- [41] Hugo Touvron et al. Training data-efficient image transformers and distillation through attention. In *ICML*, 2021. 3, 7, 12, 13
- [42] Hugo Touvron et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 7
- [43] Mikaela Angelina Uy et al. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*, 2019. 3, 12
- [44] Haohan Wang et al. Learning robust global representations by penalizing local predictive power. *NeurIPS*, 2019. 3
- [45] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *ICLR*, 2020. 11
- [46] Xumin Yu et al. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, 2021. 3, 12
- [47] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 7

Frozen Transformers in Language Models Are Effective Visual Encoder Layers

Supplementary Material

A. Applicability on Diverse Visual Tasks

In this section, we illustrate the applicability of frozen LLM transformers as visual encoder layers for non-semantic motion forecasting, 2D vision-language, and 3D vision-language tasks. This supplements the Sec. 3 of the main paper showing the applicability of our approach on image classification, point cloud recognition, and action recognition.

A.1. Motion Forecasting

We select motion forecasting as an example of a non-semantic task. It is safety-critical for autonomous driving and capitalizes on the understanding of dynamics, agent-agent interaction, and agent-lane relationship. The input commonly includes the historical trajectories of agents and way-points of lane segments, which are both represented in polylines on the bird’s-eye view (BEV). The desired output is a set of K most possible future trajectories. We conduct experiments on Argoverse [5]. The evaluation metrics are minimum average displacement (ADE), minimum final displacement (FDE), and miss rate (MR), which calculate the errors of predictions from different aspects and are better at lower values. We apply the frozen LLM transformer to VectorNet [10] and mmTransformer [29]. They first convert the agents and lanes into features, and then our LLaMA transformer block processes these agent and lane tokens. Demonstration and details are in Sec. D.4.

According to Table A, the models with LLaMA forecast better trajectories. However, we notice that the improvement is less significant compared with semantic tasks, which reflects the preference of LLM transformers for encoding rich semantics over object movements.

A.2. Vision-Language Tasks

2D Vision-Language tasks. The benefits of frozen LLM transformers for visual encoding are not limited to purely visual tasks. We experiment with 2D vision-language (VL) tasks, including visual question answering (VQA) on VQAv2 [15] and zero-shot image retrieval on Flickr30k [35]. We adopt the widely-used METER [9] as our baseline. It extracts uni-modal features for images and texts, fuses cross-modal features, and decodes the output from the cross-modal features. We insert the LLM transformer block after the cross-modal fusion. An intuitive illustration is in Fig. H. During training, our setup follows [39]: initialize the image encoder with CLIP-B/32 [37] and text encoder with RoBERTa [28], and then finetune on VQAv2 or Flickr30k. Details are in Sec. D.5. As in Table Ba, both of the 2D VL tasks are significantly enhanced with the LLaMA transformer. This evidence sup-

Model	ADE↓	FDE↓	MR↓
VectorNet	0.77	1.23	13.2
+LLaMA	0.76	1.20	12.7
mmTransformer	0.72	1.10	10.7
+LLaMA	0.71	1.08	10.5

Table A. LLM transformer layer is beneficial for motion forecasting.

ports the potential of a frozen LLM transformer for multi-modal tasks.

3D Visual Question Answering. We extend our proposed idea into 3D VQA, which requires comprehending an input 3D scene represented by point clouds or multi-view images and then answering questions. 3D VQA challenges the ability to ground language in 3D contexts. We conduct our experiments on the SQA3D [32] dataset, comparing with baseline methods and state of the arts [2, 32] on the exact match (EM) metric. We follow the baseline SQA3D to process the textual input with LSTM [21] and 3D point clouds with VoteNet [36]. Here, we add the LLM block after the VL fusion, which is consistent with our 2D VL design (Sec. A.2). More details are in Sec. D.6. According to Table Bb, adding a frozen LLM transformer effectively enhances the QA ability of models.

B. Ablation Studies

We justify our design choices (Sec. B.1) and illustrate the wide applicability of our framework to various LLMs and transformer layers (Sec. B.2). Our investigation also discovers that sufficiently large LLMs are the premise of benefiting visual encoding with a frozen transformer (Sec. B.3) and discusses the place to insert LLM blocks (Sec. B.5).

B.1. Ablation Study on Design Choices

Model Capacity. Regarding the wide applicability of frozen LLM transformers, we question if the improvement mainly comes from the increased capacity of the linear layers \mathbf{F}_L^1 and \mathbf{F}_L^2 , instead of the pre-trained weights in LLM transformers \mathbf{F}_{LM} . To analyze model capacity, we create ViT-S-MLP, which has identical trainable parameters compared with ViT-S-LLaMA. Concretely, ViT-S-MLP removes the LLM block \mathbf{F}_{LM} , and then inserts a GeLU activation [18] and layer normalization [3] between \mathbf{F}_L^1 and \mathbf{F}_L^2 . It also adopts the identical training procedure as ViT and ViT-LLaMA in Sec. 3.1 for a fair comparison. The results are summarized in Table C: the ViT-S-MLP has better performance than ViT-S due to its increased capacity, but the improvement is only about half of ViT-S-LLaMA, when fully trained. Therefore, the LLM transformer weights are crucial for the improvement and the observed benefits are

(a) 2D VQA and Image Retrieval							
Model	VQAv2 (T _{es} -dev)				Flickr30k (Val)		
	Overall	Yes/No	Number	Other	IR@1	IR@5	IR@10
METER	69.60	85.08	47.82	61.37	49.66	80.86	89.48
+ LLaMA	70.23	85.70	48.98	61.89	50.22	82.26	90.08

(b) 3D VQA		
Methods	EM@1	EM@10
ScanQA	46.58	85.97
SQA3D	47.20	86.82
SQA3D + LLaMA	48.09	89.03

Table B. Frozen LLaMA transformer enhances both 2D (a) and 3D (b) vision-language models.

Model	Acc
ViT-S	80.1
ViT-S-LLaMA	80.7
ViT-S-MLP	80.4
ViT-S-LLaMA-FT	78.9

Table C. Ablation study on model capacity and finetuning.

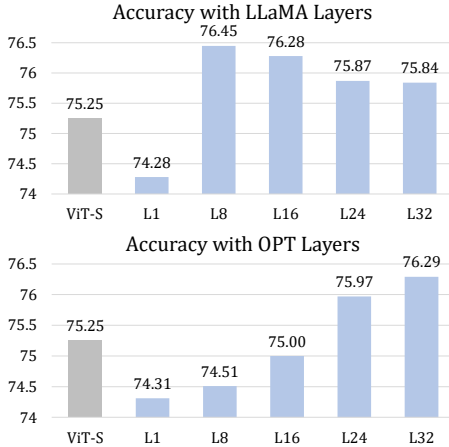


Figure A. Various LLM transformer layers improve the accuracy.

not mere consequences of an increased model capacity. Investigation with more tasks and baselines are in Sec. B.4.

Finetuning. We further verify whether finetuning the language transformer (ViT-S-LLaMA-FT) is better than freezing it. As in Table C, finetuning decreases the performance compared with ViT-S-LLaMA. This reveals the difficulty of training huge transformers.

B.2. Varying LLM Transformer Layers

We discover that LLM transformers influence visual representation learning significantly under our framework, even though they have identical capacities. Specifically, we use transformer blocks from diverse depths of LLaMA-7B [42] and OPT [47] onto ViT-S. The models are trained on the ablation study setting of 100 epochs. (More details in Sec. D.7.) As shown in Fig. A, the type of layer significantly changes the performance. These experiments also validate that *our framework is applicable to various LLMs and transformer layers*, and highlight the importance of selecting proper transformer layers. We additionally observe that the last LLM layers consistently improve the performance although they might not be optimal.

B.3. Varying LLM Transformer Scales

This section analyzes the influence of the scales of language transformers with OPT [47]. Our experiments incorporate the final transformer layers from OPT-{125M, 350M, 1.3B,

Model	Acc
ViT-S	75.25
+ OPT-125M	71.63
+ OPT-350M	71.56
+ OPT-1.3B	75.62
+ OPT-2.7B	75.74
+ OPT-6.7B	76.29

Table D. Accuracy improves with larger transformer.

2.7B, 6.7B}, into ViT-S for image classification. Our experiment setting builds upon DeiT [41] and trains for 100 epochs. Additionally, our experiments with small-scale OPT (OPT-{125M, 350M}) even yield the loss values of nan when trained with the original DeiT learning rate, so we decrease their learning rate by 1/5 for stable training. This reflects the importance of the LLMs’ scales indirectly.

As indicated by the results in Table D, the benefits of frozen language transformer grow with an increasing capacity of OPT transformers. The added transformers enhance the performance only with sufficient sizes (1.3B, 2.7B, 6.7B) and hurt the accuracy when the sizes are small (125M, 350M). Therefore, the phenomena of LLM transformers enhancing visual tasks only “emerges” at sufficient scales.

B.4. Whether Improvement Solely Originates from Increased Network Capacities

This section analyzes whether the improvement of our approach is the consequence of the increased capacities and supplements Sec. B.1. Specifically, we experiment with two sets of additional baselines: (1) *using additional MLPs*, aligning the number of trainable parameters; (2) *using randomly initialized LLM transformer blocks*, aligning the total number of parameters, to compare with our approach. Please note that the randomly initialized LLM transformer blocks are trained end-to-end with the visual encoders.

We conduct experiments across all the tasks covered in our experiments, as shown in Table E. We will include these results in the revision. The results in the tables show several important findings validating that our performance gains stem from our method rather than increased network capacity. More importantly, simply adding MLPs is not a consistently beneficial strategy for all visual tasks and can be detrimental, resulting in inferior performance on some tasks even compared to the plain baseline. This is because naively adding large MLPs or transformer blocks may lead to challenges in optimization and suffers from a small train-

ing dataset compared to LLM pertaining.

B.5. Layers to Insert the Frozen LLM Block(s)

Table F presents an ablation study on different architecture variants, including different places to insert the LLM blocks and the number of LLM blocks. Without loss of generality, we leverage the image classification with ViT-S for our ablation study. From the experiments, we mainly have the following discoveries:

LLM block location. Inserting the frozen LLaMA block at the beginning or the middle of the visual encoder performs worse than our inserting LLaMA at the end of the encoder. This supports our design choice and verify the intuition that LLM blocks are more suitable for high-level semantics instead of low-level visual patterns.

LLM block number. We also experiment with inserting the last 2 blocks from LLaMA, and find it better than our using a single LLaMA block. Due to computation constraints, we list extending this to diverse visual tasks as an interesting future direction.

C. Appendix for the Information Filtering Hypothesis

C.1. Detailed Derivation of the Hypothesis

We expand the details of several steps in our derivation in the main paper (Sec. 4) for better clarity. Beginning from the formula of the CLS token below,

$$z_L^2[\text{CLS}] = \mathbf{F}_L^2 \cdot \mathbf{F}_{LM}^F \cdot \mathbf{F}_{LM}^A \left(\sum_{v \in V} w_v z_L^1[v] \right), \quad (\text{A})$$

we further separate the visual tokens into the subset of informative tokens V_i and uninformative tokens V_u . This changes Eqn. A into,

$$z_L^2[\text{CLS}] = \mathbf{F}_L^2 \cdot \mathbf{F}_{LM}^F \cdot \mathbf{F}_{LM}^A \left(\sum_{i \in V_i} w_i z_L^1[i] + \sum_{u \in V_u} w_u z_L^1[u] \right). \quad (\text{B})$$

Using the same notation, our observation on the feature activation can be stated as: the attention weights for informative tokens $\{w_i, i \in V_i\}$ are still noisy after incorporating the frozen LLM transformer, while the final visual tokens shown as below have emergent concentration on target regions.

$$z_L^2[i] = \mathbf{F}_L^2 \cdot \mathbf{F}_{LM}^F \cdot \mathbf{F}_{LM}^A(z_L^1[i]), \text{ where } i \in V_i \quad (\text{C})$$

Combining Eqn. B and Eqn. C inspires us to express our hypothesis in terms of the connection between visual and CLS token with Eqn. D below, where the added modules $\mathbf{F}_L^2 \cdot \mathbf{F}_{LM}^F \cdot \mathbf{F}_{LM}^A$ augment the informative tokens V_i and

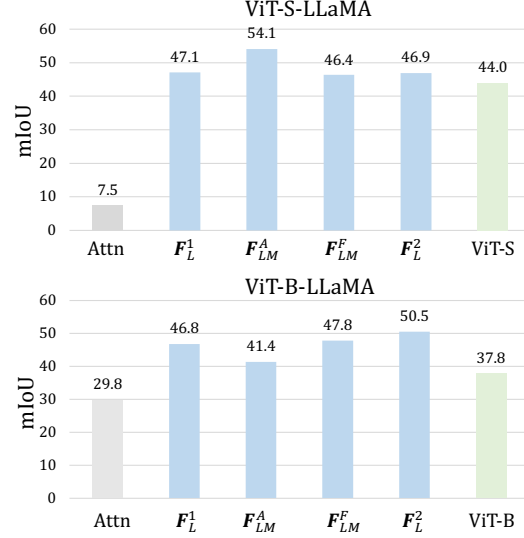


Figure B. Pseudo masks from ViT-LLaMA’s features (F_L^2) have larger mIoU than attention scores and ViT.

lead to better prediction:

$$z_L^2[\text{CLS}] \propto \sum_{i \in V_i} w_i \underbrace{(\mathbf{F}_L^2 \cdot \mathbf{F}_{LM}^F \cdot \mathbf{F}_{LM}^A(z_L^1[i]))}_{z_L^2[i]} + \sum_{u \in V_u} w_u \underbrace{(\mathbf{F}_L^2 \cdot \mathbf{F}_{LM}^F \cdot \mathbf{F}_{LM}^A(z_L^1[u]))}_{z_L^2[u]} \quad (\text{D})$$

This is a more thorough expression of our hypothesis in the main paper to better differentiate the role of informative tokens in our hypothesis.

C.2. Quantitative Evidence

The qualitative observation is further supported with quantitative evidence. Specifically, we use the ImageNet-S [11] dataset to provide the ground truth of “informative regions” from its annotation of semantic segmentation masks. To assess the fidelity of feature activation and attention scores, we first generate pseudo-masks highlighting their concentrating regions, *i.e.*, the tokens with larger activation or attention scores than the others on the same image. Then the quality of features and attention scores are reflected by the mIoU (mean intersection-over-union) between their pseudo-masks and ground truth segmentation masks. Implementation details are in Sec. C.4.

Finally, we summarize the mIoU statistics for feature activation and attention scores in Fig. B. As demonstrated, both ViT-S-LLaMA and ViT-B-LLaMA have better mIoU of pseudo-masks than attention scores. This directly supports our hypothesis that the features $\{z_L^2[v] | v \in V\}$ contribute more reliably than attention scores $\{w_v | v \in V\}$. We additionally notice that the pseudo-masks from ViT-LLaMA generally have larger mIoU compared with ViT, which reflects the benefits of training ViTs with a frozen LLM transformer. The advantage of the feature in the first

(a) Image Classification (ImageNet)			(b) Point Cloud Recognition (ScanObjectNN)			
Methods	Acc		Methods	BG	OBJ	T50
ViT-S	80.1		PointBert	87.4	88.1	83.1
+ LLaMA (Ours)	80.7		+ LLaMA (Ours)	88.0	88.5	83.8
+ MLP	80.4		+ MLP	86.5	87.3	83.4
+ Random LLM	76.9		+ Random LLM	87.2	88.0	82.6

(c) Action Recognition (SSv2)			(d) Motion Forecasting (Argoverse)			
Methods	Acc1	Acc5	Methods	ADE↓	FDE↓	MR↓
ViT-S	64.7	89.2	mmTransformer	0.72	1.10	10.7
+ LLaMA (Ours)	65.9	89.9	+ LLaMA (Ours)	0.71	1.08	10.5
+ MLP	63.8	88.9	+ MLP	0.74	1.16	11.8
+ Random LLM	64.1	88.8	+ Random LLM	0.74	1.15	11.5

(e) 2D Retrieval (Flickr30k)				(f) 3D VQA (SQA3D)		
Methods	EM1	EM5	EM10	Methods	EM1	EM10
METER	49.66	80.86	89.48	SQA3D	47.20	86.82
+ LLaMA (Ours)	50.22	82.26	90.08	+ LLaMA (Ours)	48.09	89.03
+ MLP	49.48	81.12	89.58	+ MLP	47.14	88.08
+ Random LLM	49.80	81.62	89.72	+ Random LLM	47.26	88.46

Table E. Addition comparisons for model capacities. We compare our approach of adding frozen LLM transformer with adding randomly initialized MLP (“+MLP”) or LLM blocks (“+Random LLM”) and training end-to-end. The results on diverse tasks uniformly support that our improvement is not merely the result of larger model capacities. Details in Sec. B.4.

Model	Configuration	Acc Top 1
ViT-S	Baseline ViT-S	75.32
+ LLaMA at Tail (Ours)	Single LLaMA block at the end of ViT	75.84
+ LLaMA at Middle	Single LLaMA block at the middle of ViT	75.55
+ LLaMA at Head	Single LLaMA block at the head of ViT	72.66
+ 2 LLaMA Blocks	2 LLaMA blocks at the end of ViT	77.10

Table F. Varying the position and number of LLaMA blocks.

linear layer F_L^1 also reveals that training with our framework is beneficial to even earlier stages of features. However, we would like to clarify that the pseudo-masks from either magnitude or frequency activation are intuitive but lossy measures to quantify feature quality because neural networks can encode information into other formats. Therefore, developing better measurements to analyze individual network layers will be meaningful for future work.

C.3. Information Filtering Hypothesis on Other Tasks

The previous sections mainly discuss our information filtering hypothesis in terms of image classification. Meanwhile, we also discover supportive evidence of our hypothesis on various other tasks.

Action Recognition

We mainly analyze the information filtering hypothesis in action recognition *qualitatively* because the ground truth of “relevant regions” is difficult to quantify for this task. In practice, we visualize the activation of video tokens in Fig. C. At a low activation threshold, we notice that the video tokens from ViT-S-LLaMA better capture the fore-

ground areas of hands and manipulated objects than ViT-S. With the video tokens in VideoMAE activated both spatially and temporally, we further increase the threshold to demonstrate its ability to select informative frames. As in the “high threshold” row of Fig. C, ViT-S-LLaMA more accurately focuses on the middle frames with actual human-object interaction. Therefore, we conclude that the informative video tokens are indeed distinguished and augmented in action recognition, which aligns with the information filtering hypothesis.

Point cloud recognition. We visualize the activation of the point tokens in point cloud classification before and after adding the frozen LLM transformer in Fig. D. In the examples of chairs and desks, we observe that “PointBERT+LLaMA” concentrates less on the background (chairs, indicated with red arrows) and more on the actual object surfaces (desks, indicated with red arrows). This demonstrates that the frozen LLM transformer learns to focus on the informative points, which is consistent with our hypothesis.

2D VQA. We investigate the activation of visual tokens in

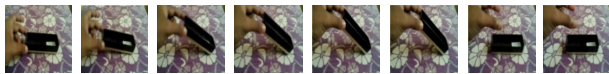




Model	Threshold								
ViT-S	Low								
	High								
ViT-S-LLaMA	Low								
	High								

Figure C. Token activation in action recognition. Video tokens are activated jointly in all the frames, and every video token is a cube with shape $2 \times 16 \times 16$. After adding the LLM transformer, the model better concentrates on the relevant objects and hands (“low threshold”) and more accurately focuses on frames with hand-object interaction (“high threshold”).

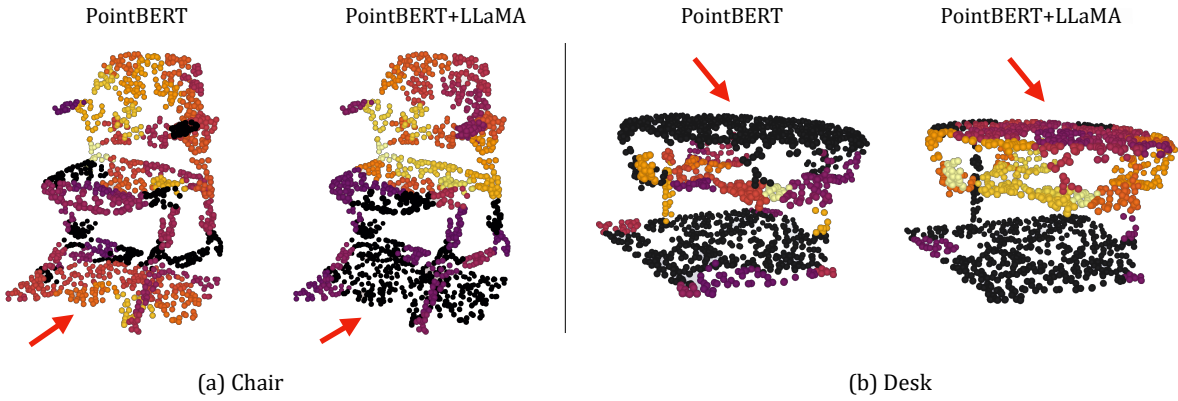


Figure D. Visualization of feature activation for point cloud classification. Brighter colors indicate higher activation values. To highlight the most salient activation, we apply a threshold to filter out points with low activation values. This visualization demonstrates that the model learns to focus on the most discriminative foreground object for classifying the point cloud after adding the frozen LLM transformer.

the 2D VQA task in Fig. E. With the METER [9] framework initializing the visual backbone from CLIP [37] weights, the quality of feature activation is reasonable and mostly concentrates on the target regions for both the baseline METER and our “METER+LLaMA.” However, we can still witness that feature activation better aligns with the images and questions than the noisy attention heads. Furthermore, the activation of “METER+LLaMA” is also slightly advantageous over METER with less scattering, such as better concentrating on the regions of light and leaves at high thresholds.

3D VQA. We analyze the effect of a frozen LLM transformer for 3D VQA and further confirm our hypothesis. As in Fig. F, we compare the activation of visual tokens, which are seed points in VoteNet [36], before and after incorporating the LLaMA transformer. To provide the context, the scenes in SQA3D [32] are projected onto the bird’s-eye-view (BEV). From the visualizations, we clearly observe that the feature activation exhibits sharper concentration to the directions guided by language after adding LLaMA,

such as the “table behind me” areas in the left figure and “to my left side” areas in the right figure. Therefore, these indicate that the added LLM transformer filters the informative points and augments them for downstream question answering.

C.4. Quantitative Evidence

In Sec. C.2, we generate the pseudo-masks from feature activation or attention scores and then evaluate their quality with mIoU. This section describes the details.

Dataset. We leverage ImageNet-S [11] because it provides semantic segmentation masks for ImageNet [7] images. Specifically, we adopt the ImageNet-S version with 50 categories and run our evaluation on its validation set, to avoid data leakage from the training set.

Definition of IoU. Our IoU calculates the alignment between the highly activated tokens and the ground truth mask. As tokens are sparse and in low resolution, we slightly change the calculation of IoU for our purpose. Specifically, we first project the ground truth mask to the

	Attn Scores	Threshold	METER	METER +LLaMA		Attn Scores	Threshold	METER	METER +LLaMA
Is the light on?		Low			How many leaves?		Low		
		High					High		

Figure E. Visualization of attention scores and feature activation for 2D VQA. We are able to visualize attention scores because METER uses CLS token. Both low and high thresholds for feature activation are displayed to illustrate the concentration on relevant regions.

SQA3D	SQA3D+LLaMA	SQA3D	SQA3D+LLaMA
Situation: I am standing in front of the table and facing trash can. Question: How many chairs are at the table behind me?		Situation: I am standing between the toilet on my right and sink on my left. Question: Is the door open or closed to my left side?	

Figure F. Analysis of feature activation for 3D VQA. The scenes are viewed from BEV. The green arrow marks the location and facing direction. The colors of points indicate their activation: a lighter color (yellow) represents a larger magnitude than a dark color (blue and green). We observe that “SQA3D+LLaMA” has sharper activation that is better related to the questions. Thus, it supports our information filtering hypothesis. (Best viewed in color and zooming in.)

resolution of tokens to acquire a binary mask of tokens, indicating whether they are related to the target object (with value 1) or not (with value 0), denoted as M_g . Then we compute the IoU between the pseudo-mask M_p generated from feature activation with M_g as follows:

$$TP = \text{sum}(M_g M_p), \quad (E)$$

$$FP = \text{sum}((1 - M_g) M_p), \quad (F)$$

$$FN = \text{sum}(M_g (1 - M_p)), \quad (G)$$

$$\widetilde{\text{IoU}}(M_g, M_p) = TP / (TP + FP + FN). \quad (H)$$

Pseudo-mask generation. To generate pseudo masks from feature activation, we treat the highly-activated regions as the final result. To generate pseudo masks with attention scores, we first sum the scores from all the attention heads and follow a similar procedure of treating highly-scored regions as pseudo masks. Concretely, given a feature activation z , generating a pseudo-mask is as straightforward as $M_p = (z > t)$, where t is a threshold between 0 and 1. Although the process is natural, we notice that selecting the thresholds for activation or score significantly affects the quality of pseudo masks. Therefore, we always automatically choose the threshold maximizing the mIoU between the pseudo and ground truth masks to avoid threshold tun-

ing and enable a fair comparison. The algorithm of choosing the best threshold for the activation or attention scores for each image is as below,

$$\text{IoU}(M_g, z) = \max_t \left(\widetilde{\text{IoU}}(M_g, M_p) \right), \quad (I)$$

where $M_p = z > t$ and $t \in \{0.1, 0.2, 0.3, \dots, 0.9\}$.

Finally, our mIoU in Sec. C.2 is the mean IoU on all the images.

C.5. Discussion and Limitations of the Hypothesis

Perspective of usable information. We supplement the opinions from [45] that greatly inspire our investigation of the information filtering hypothesis. [45] proposes that a well-trained neural network layer can be considered as a *decipher* adding usable information into the features and enabling subsequent modules to better infer the latent information. In our information filtering hypothesis, the incorporated modules are indeed acting as *deciphers* that enlarge the contribution of usable tokens and benefit downstream predictions.

Limitations. Though our information filtering hypothesis explains how the performance improves with frozen LLM transformers, we notice that several intriguing phenomena

are not yet covered. First, the current hypothesis is unable to analyze the utilities of different layers separately. Second, the hypothesis does not explain how the training dynamics facilitate the visual token features to cooperate with the frozen language transformer, which is interesting future work.

D. Details of Implementations

D.1. Image Classification

We follow the DeiT [41] in training the models of ViT- $\{T, S, B\}$ and ViT- $\{T, S, B\}$ -LLaMA in Sec. 3.1. Each visual token is a 16×16 patch on 224×224 images. The most important configurations include a total of 300 epochs, a base learning rate of $5e-4$, a cosine annealing learning rate schedule [30], and an AdamW optimizer [22, 31]. The total time for training lasts 4-6 days on $4 \times A100$ GPUs. The only change we adopt is the warm-up length of 20 epochs, compared to the original warm-up of 10 epochs in DeiT. A longer warm-up stabilizes the training of ViT models and also enables us to slightly outperform the original ViT-T and ViT-S performance.

D.2. Point Cloud Classification

In this section, we describe the implementation details of the pointcloud classification method. For optimization, we use AdamW [22, 31] optimizer and a cosine annealing learning rate schedule [30]. To map the dimension between the PointBERT and LLaMa transformer tokens, we add two linear layers with a learning rate of $5e-5$. The PointBERT backbone has a learning rate of $5e-4$, consistent with the original setting in [46]. We finetune our model for 300 epochs on both the ScanObjectNN [43] and ModelNet40 [12] datasets. The training takes around 6-10 hours on $4 \times A200$ GPUs.

D.3. Action Recognition

We investigate action recognition and provide more details of implementation here. Our setup strictly follows VideoMAE [40], where a ViT model is (1) pretrained by masked auto-encoding [16]; then (2) finetuned for additional epochs. As stated in Sec. 3.3, we directly begin from the second step and inherit the parameters of self-attention blocks in ViT from pretrained VideoMAE models. During the training process, VideoMAE trains ViT-S for 40 epochs (5 epochs of warm-up) and ViT-B for 30 epochs (5 epochs of warm-up), where we adopt the same length of training. The optimizer is AdamW [22, 31] with the cosine annealing learning rate schedule [30].

In Sec. 3.3, our ViT-S and ViT-B performance is lower than the reported numbers in VideoMAE. This is because VideoMAE uses $32 \sim 64$ GPUs during the finetuning stage and supports a much larger batch size compared to us,

Model	ADE \downarrow (k=1)	FDE \downarrow (k=1)
Paper	1.66	3.67
Ours	1.60	3.60

Table G. Our implementation of VectorNet is better than their paper. Please note that this table uses the same single-modal setting (k=1), as VectorNet for comparison.

Model	ADE \downarrow	FDE \downarrow	MR \downarrow
Paper	0.71	1.15	10.6
Ours	0.72	1.10	10.7

Table H. Our implementation of mmTransformer is comparable to the performance in their paper, with large advantages on the main metric of FDE.

though we scale the learning rate according to the batch size as [13]. Concretely, VideoMAE adopts the batch size of 384 video clips, while our computational resource only supports a batch size of 24 clips and 12 clips for ViT-S and ViT-B, respectively. However, we control the setup between ViT and ViT-LLaMA for a fair comparison. Finally, the ViT-S-LLaMA and ViT-B-LLaMA experiments take around 3-4 days on $4 \times A100$ GPUs.

D.4. Motion Forecasting

We evaluate the effects of frozen LLM transformers in Sec. A.1 with motion forecasting. Since motion forecasting is a less well-known task in computer vision, we intuitively demonstrate its problem setting and modular architecture in Fig. G, where VectorNet [10] and mmTransformer [29] encode each lane or agent trajectory into a token embedding, then our LLM blocks process these tokens and feed them into the regression decoder.

Since VectorNet and mmTransformer have not released their training code, we reproduce their results on our own and achieve better or similar results as reported in the paper. As in Table G and Table H, the baseline used in our paper (Table A) have comparable or even better performance compared to the original performance in the papers, which is critical for a fair and meaningful comparison.

During the training time, we separately train VectorNet or mmTransformer. VectorNet is a relatively simple architecture, so its training lasts 60 epochs, with cosine annealing learning rate schedule [30]. We use the AdamW optimizer [22, 31] with a learning rate of $5e-4$ and batch size equal to 32 samples. For mmTransformer, we train it with the same learning rate, batch size, and optimizer as VectorNet. The training lasts 32 epochs, where we drop the learning rate by 1/4 on epochs 20, 24, and 28. The training time for both models is around 2 days on one A100 GPU.

D.5. 2D Vision Language

This section provides more details on implementations and designs to supplement our discussion on 2D vision-language models in Sec. A.2. Our experiments adopt the widely used METER [9] as the baseline and incorporate

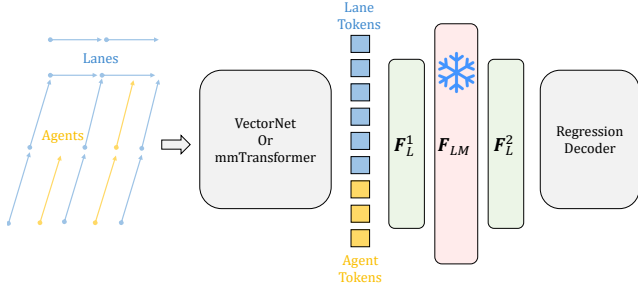


Figure G. Demonstration of typical motion forecasting design and our implementation. Motion forecasting models the trajectories of agents and lanes as polylines. Existing models [10, 29] using MLPs or transformers to convert the lanes and agent trajectories into token embeddings, then employ a decoder to regress the future trajectories. In our design, we treat either VectorNet [10] or mm-Transformer [29] as general encoder, then insert the frozen LLM blocks to process their embeddings.

pretrained LLM transformers after its vision-language fusion module. In Fig. H, we intuitively illustrate the modular design of METER and the special place to insert our frozen LLM transformer and linear layers.

Conventionally, METER follows a two-stage training strategy: (1) pretrains the whole vision-language model (VLM) on a large combination of vision-language datasets, including COCO [26], Conceptual Captions [38], SBU Captions [34], and Visual Genome [24]; (2) finetuning on downstream tasks like visual question answering (VQA) or image-text retrieval. However, the first step of pretraining is computationally extensive, so we adopt the setup in [39] by skipping the pretraining step and directly training on the target task. Specifically, we initialize the image encoder from CLIP-B/32 [37] and text encoder from RoBERTa [28], then finetune all the modules jointly expect for the LLM transformer F_{LM} . Because of the initialization from CLIP and RoBERTa, our model is capable of predicting reasonably.

During the training stage, we strictly follow the same hyper-parameters and configurations on VQAv2 [15] and Flickr30k [35] provided by METER. The most critical detail is that METER assigns different learning rates for each module. For example, the cross-modal fusion module and the decoder have larger learning rates compared to the pre-trained image and text encoders. Similarly, our experiments set the learning rates of linear layers (F_L^1 and F_L^2) $10\times$ the learning rate of the image encoder because they are randomly initialized. Finally, each training on VQAv2 and Flickr30k lasts for 10 epochs and around 1 day on $4\times$ A100 GPUs.

D.6. 3D Vision Language

This section provides more details of the dataset and training configurations of 3D vision language task. We conduct

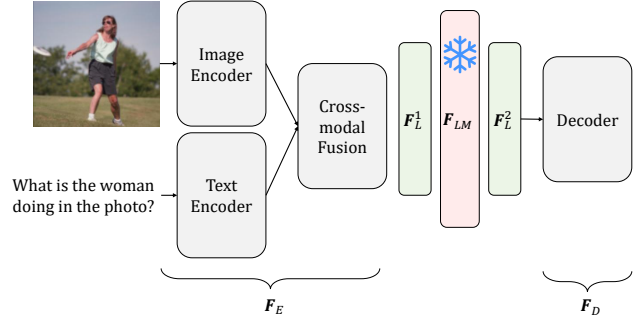


Figure H. Intuitive illustration of METER [9] framework and how to insert the frozen language transformer (pink F_{LM}) and linear layers (green F_L^1 and F_L^2) to process the visual tokens after vision-language fusion.

our experiments on SQA3D dataset [32], which contains $33.4k$ questions in 650 unique ScanNet scenes. In addition to question answering (QA), the benchmark also requires the model to understand its situation (position, orientation, etc.) in the 3D scene as described by text. Hence it is called situated question answering (SQA). We use batch size 32 during our model training, and AdamW as our optimizer. The hidden size for each embedding token is 768. We train all parameters from scratch for 30 epochs, and decrease the learning rate by 10 times at 10, 15, and 20-th epoch. The model is trained on a single A100 GPU.

D.7. Depths of LLM Layers

When varying the depth of transformer blocks in Sec. B.2 and Fig. A, we adopt the ablation setup of training for 100 epochs, compared to the full training of 300 epochs. The experiments are based on ViT-S/16 [8] in DeiT [41] with batch size 1024, which are also used in other ablation experiments in our project. The whole training process involves 20 epochs of warm-up and the remaining 80 epochs adopt the cosine annealing learning rate schedule [30]. Each experiment takes around 2 days on $4\times$ A100 GPUs.