562

A Dataset Documentation and Additional Information

Below, we include all information required for dataset submissions to the NeurIPS Datasets and Benchmarks Track:

Dataset Documentation and Intended Uses The dataset documentation is provided at the Croissant and Huggingface URLs mentioned below. The dataset is mainly intended for evaluating foundational VLMs and their shape recognition abilities. The dataset can also be used for learning invariant representations using Domain Generalization techniques. Other uses may be possible.

Dataset URL Our datasets are available for viewing and full download at the following permanent link: `https://huggingface.co/datasets/arshiahemmat/IllusionBench`. The "dataset viewer" allows one to select a specific split (i.e., IllusionBench-IN, IllusionBench-LOGO, or IllusionBench-ICON). All images are provided in the `.png` format. The HuggingFace Datasets repository service (where our dataset is hosted) automatically generates structured Web standard metadata for dataset discovery.

Croissant Metadata URL Our Croissant metadata record is available at `https://huggingface.co/api/datasets/arshiahemmat/IllusionBench/croissant`.

Author Statement The authors have collected the conditioning images and generated this dataset for research purposes. For this reason, the data usage is allowed under the fair use law and is not intended to yield any copyright infringement. There is no warranty of fitness for a particular purpose or noninfringement. The authors remain available to edit the dataset to comply with the law. In no event shall the authors or the NeurIPS conference be liable for any claim, damages, or other liability arising from, out of, or in connection with the usage or release of this dataset.

Data License This work is openly licensed under CC BY-NC 4.0 (`https://creativecommons.org/licenses/by-nc/4.0/deed.en`).

Long-Term Hosting, Licensing, and Maintenance Plan We have uploaded our dataset to Hugging-Face Datasets (link above). The Licensing information and Croissant metadata URL are available above and also available in the HuggingFace URL. Regarding Maintenance of the dataset on the HuggingFace servers please refer to the `https://huggingface.co/content-guidelines`.

Reproducibility The code for generating the dataset and the experiments are publicly available in the following repository `https://github.com/arshiahemmat/IllusionBench`.

Haman Annotations We have provided screenshots of annotation forms which were distributed among participants in Appendix A.1.

Attributions This work utilizes stock images to condition generators (as described in Section 3). IllusionBench-ICON conditioning images are taken from `icons8.com`, which makes them freely available provided they are attributed using a link (as we do here).

A.1 Human Annotation Details

Subsampling for Annotation Given the size of our dataset (more than 32K samples) performing a complete annotation of it would be expensive. Furthermore, since the data is synthesized and we perfectly know the class of the shapes represented in each image, the purpose of the annotation is simply to verify that the generated images have shapes that are recognisable by humans. For this reason, we subsample the generated dataset by enforcing that, for each dataset (i.e., each of `IllusionBench-IN`, `IllusionBench-ICON`, and `IllusionBench-LOGO`), at least one conditioning image from each class and scene choice is annotated.

Furthermore, we observe that the difficulty in perceiving an object depends on the choice of the hyperparameters that control the diffusion process. For this reason, we additionally enforce that images are uniformly sampled from each hyperparameter setting so that annotators are exposed to images encompassing the full range of difficulty.

16
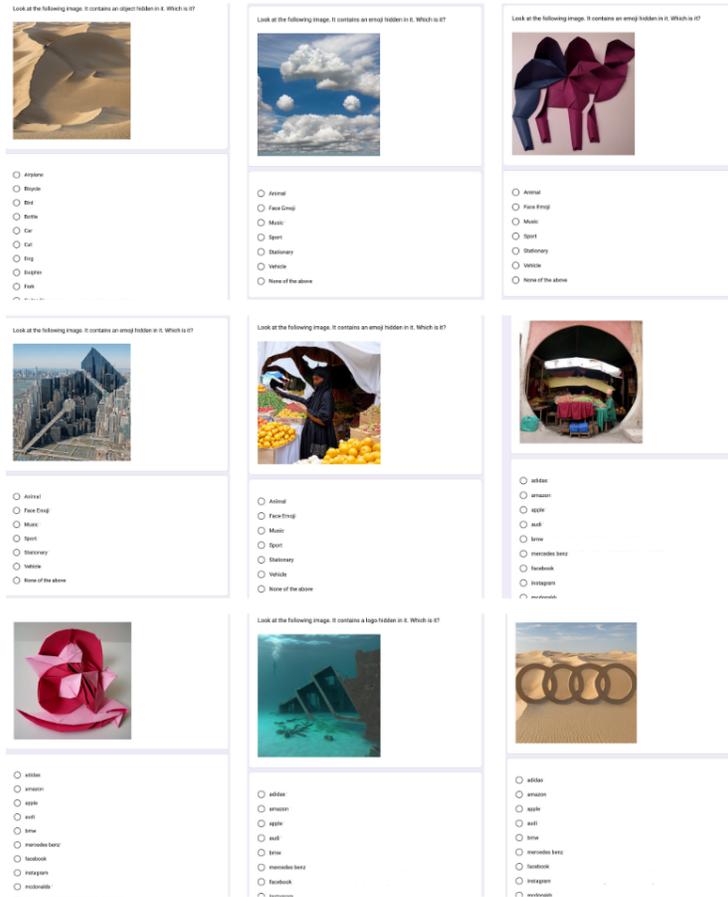
Figure 8: **Human Annotation** screenshots from GoogleForm through which the images are annotated by human annotators.

Participants Our human evaluation involved 106 participants. The annotators were first instructed about the task and required to perform a simple test on 10 images, in order to make sure they understood the task to be performed. Annotators participated on a purely volunteer basis and were awarded with in-course credit. Participation was not mandatory for any student or course. No risks were identified for the annotation process.

A.2 Generation Hyperparameters

Data Generator Hyperparameters For data generation, we focused on the Illusion Diffusion generative models (demo available here), containing three major components:

- ControlNet [Zhang et al., 2023a], specifically: controlv1p sd15 qrcode monster
- Base Model, specifically: RealisticVision V5.1 noVAE, built using Stable Diffusion [Rombach et al., 2022]
- Stable Diffusion-guided VAE, specifically: sd-vae-ft-mse

We used the following generation hyperparameters:

- Prompts were simply a single word corresponding to the scene types (e.g., "city" or "museum")
- Guidance-scale was always set to default value **7.5**
- Illusion_strength, which can be used to modulate the strength of abstract shape patterns, was selected based on our anecdotal observations regarding an appropriate difficulty level for each dataset (see below) and validated using human data annotation (as described above)
- Sampler was always set to default value **Euler**

17

The Illusion_strength for the different datasets are as follows:

- {Illusion_strength} of the `IllusionBench-LOGO` and `IllusionBench-IN`: [0.75, 0.80, 0.85, 0.90, 1.05, 1.10, 1.15, 1.20, 1.25, 1.35, 1.40, 1.50, 1.60]

- {Illusion_strength} of the `IllusionBench-ICON`: [0.85, 1.05, 1.25, 1.40]

A.3 Limitations

For future work, we will create more complex images and define more tasks in order to challenge models. We have also increased the size of our dataset so that we can train large models using our dataset. A current limitation is that we only hide a single shape in each image. Future work could extend this to incorporating several objects within the same background. Finally, we also plan to experiment with further tasks for compositional understanding and scene understanding of SOTA models. We leveraged prompt engineering to report the best possible performance of each model in the zero-shot case as described in Appendix B and Section 4, however, improvements may be possible. We describe several limitations of the methods explored in this work in Sections 4 and 5.

A.4 Data Samples

To illustrate the quality of abstract shape recognition images created for this dataset, we sample 1 image from each scene type in each dataset and display them in Fig. 9.
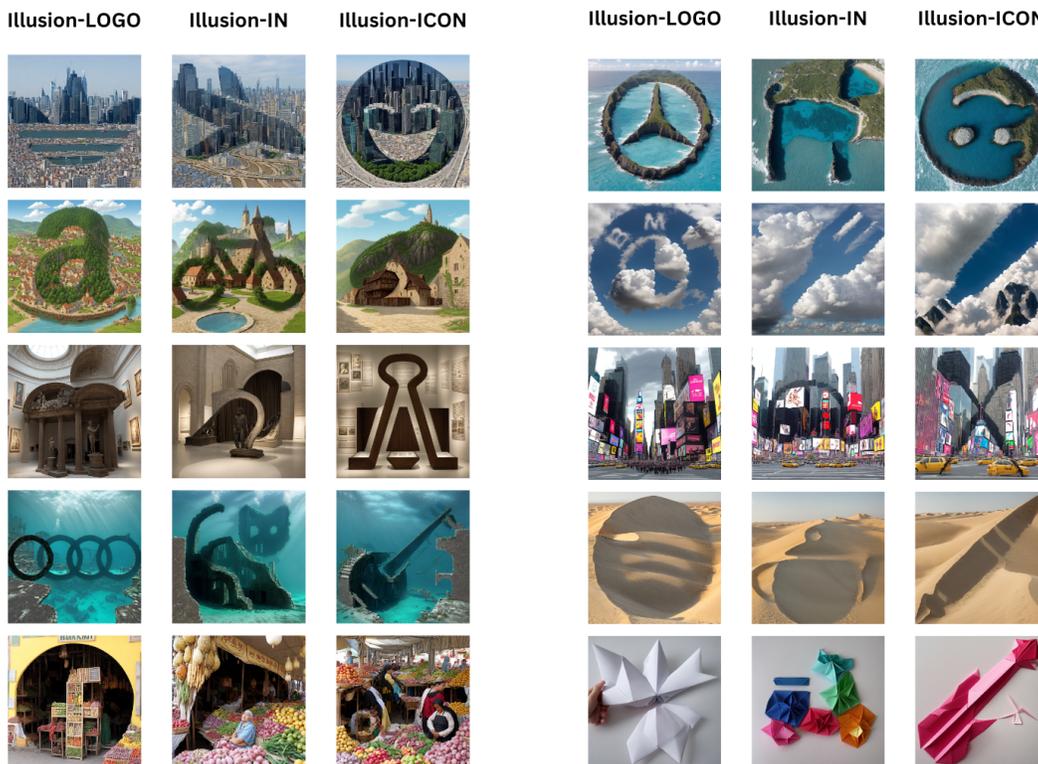
Figure 9: **Image Samples** from each dataset in our benchmark.

18

B   Zero-Shot Experiments Details

B.1   Zero-shot Experiments

We test our models zero-shot to evaluate their abstract shape recognition abilities. To leverage all capabilities of these models, we describe the conditions of our experiments in our prompt. The models are then asked to choose the correct shape type among a closed set of options, which include both shapes and scene names.

Let us focus on the predictive task τ_C. Analogous formulations hold for τ_S and $\tau_{C,S}$. Given that the model can correctly assign the class $y_{i_*}^C$ to the hidden shape in the scene $x_{i_*}^C$, we provide it with a set of options $\mathcal{O}$, which includes all the shapes and scene names considered in the dataset split. We then ask the model to predict the shape name from these options.

Define $\mathcal{O} = \{\text{shape}_1, \text{shape}_2, \dots, \text{scene}_1, \text{scene}_2, \dots\}$ as the set of possible options. The model's response is evaluated based on whether the correct shape name is present in its output.

B.2   Models

In our zero-shot experiments, we evaluate each of the following large vision language models (VLMs):

- BlipV2-T5 [Li et al., 2023c], a VLM utilizing the T5 architecture [Raffel et al., 2020] for text encoding and a state-of-the-art vision encoder, designed for high-performance multimodal tasks.
- CogVLM [Wang et al., 2024], an advanced VLM leveraging a Vision Transformer (ViT) [Dosovitskiy et al., 2021] and a powerful language model fine-tuned for vision-language reasoning tasks.
- InstructBlip-T5 [Dai et al., 2023], a model combining the T5 architecture [Raffel et al., 2020] for text processing with a highly efficient vision encoder, fine-tuned for instructional prompts and multimodal interactions.
- LLaVA-Next (Vicuna-7b) [Liu et al., 2024b], a VLM using Vicuna-7b-v1.5 [Zheng et al., 2024] and CLIP ViT-L/14 [Radford et al., 2021] as text and visual encoders, respectively. These are connected via simple projections.
- Qwen-VL-Chat [Bai et al., 2023], a 9B parameter model employing a cross-attention module to link an OpenClip ViT-bigG [Ilharco et al., 2021] vision encoder to a Qwen-7b [Bai et al., 2023] text backbone.
- Llava1.5-7b and 13-b [Liu et al., 2024a], a VLM employing a 7-billion parameter language model and advanced visual encoder, connected via efficient projections.
- InstructBlip-7b and 13b [Dai et al., 2023, 2024], a BLIP [Li et al., 2022] model fine-tuned using instruction tuning, using a 7-billion parameter language model and a high-resolution vision encoder for precise multimodal understanding.
- MoE-StableLM, MoE-Qwen, MoE-Phi2 [Lin et al., 2024], a mixture of experts (MoE) model combining StableLM architecture [Raffel et al., 2020] with multiple expert models for dynamic task specialization and improved performance.
- GPT-4o, a multimodal version of GPT-4 [OpenAI, 2023], incorporating optimized end-to-end multimodal encoding of images, text, and audio for improved multimodal task performance.
- Gemini-Flash [Gemini Team et al., 2023], a high-speed VLM combining the latest advancements in vision transformers [Dosovitskiy et al., 2021] and language models for rapid and accurate multimodal analysis.

Note that, for the last two models in this list, we are unable to provide any specific information regarding their respective architectures or training regimes, as this information has not been made publicly available.

B.3   Prompts

We use the following general prompt template for our ICL experiments:

19

- T1 Prompt: `This image contains a {shape} integrated into a background, where elements of the background contribute to forming the {shape}. Identify the {shape} that is represented in the image by choosing exclusively among the following options: {shape_options}, {background_classes}. Provide your response by stating only the single, most accurate class name that represents the {shape}. You have to respond with a single word.`
- Texture Question Bias: `This image contains a {shape} integrated into a background, where elements of the background contribute to forming the {shape}. Identify the background that is represented in the image by choosing exclusively among the following options: {shape_options}, {background_classes}. Provide your response by stating only the single, most accurate class name that represents the background. You have to respond with a single word.`

where `shape` $\in$ {logo, shape, icon} for the dataset IllusionBench-LOGO, IllusionBench-IN and IllusionBench-CI respectively.

20

693 **B.4 Individual Dataset Splits Zero-shot performance (compared Stylized ImageNet)**
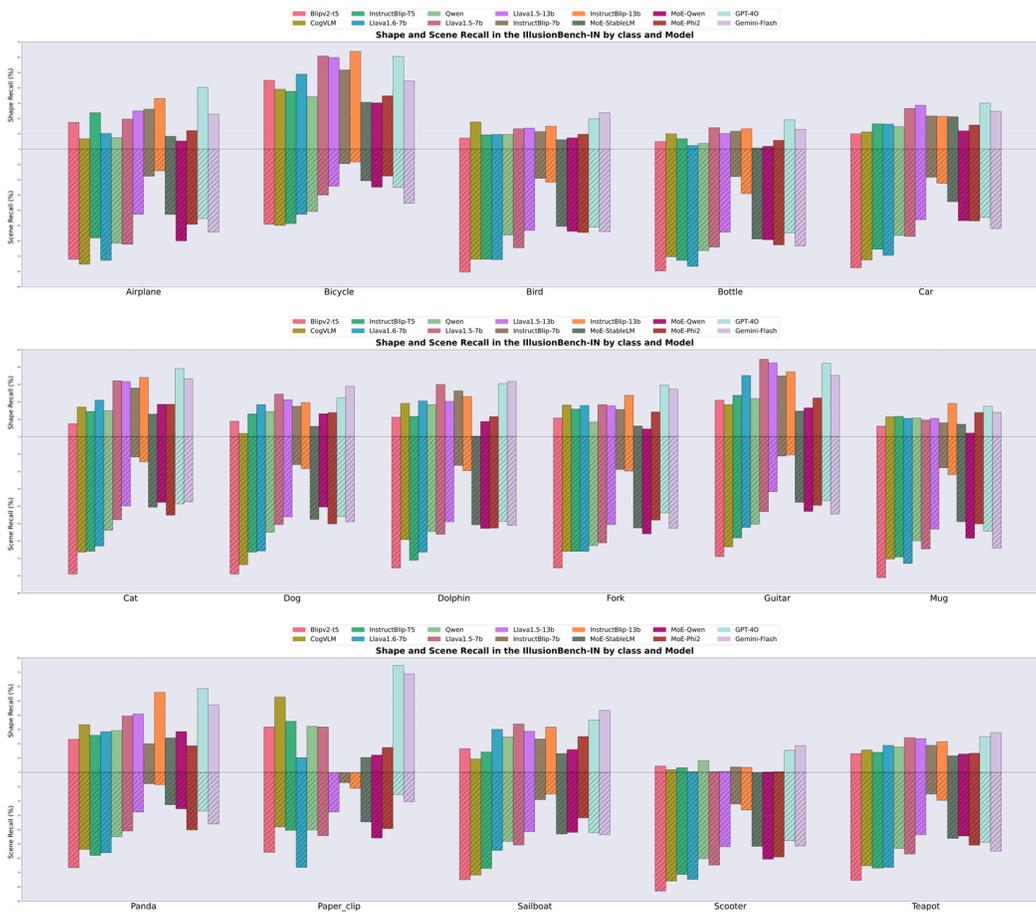
Figure 10: **Zero-shot Results on** IllusionBench-IN. zero-shot shape and scene recall of VLMs on the IllusionBench-IN dataset.

21
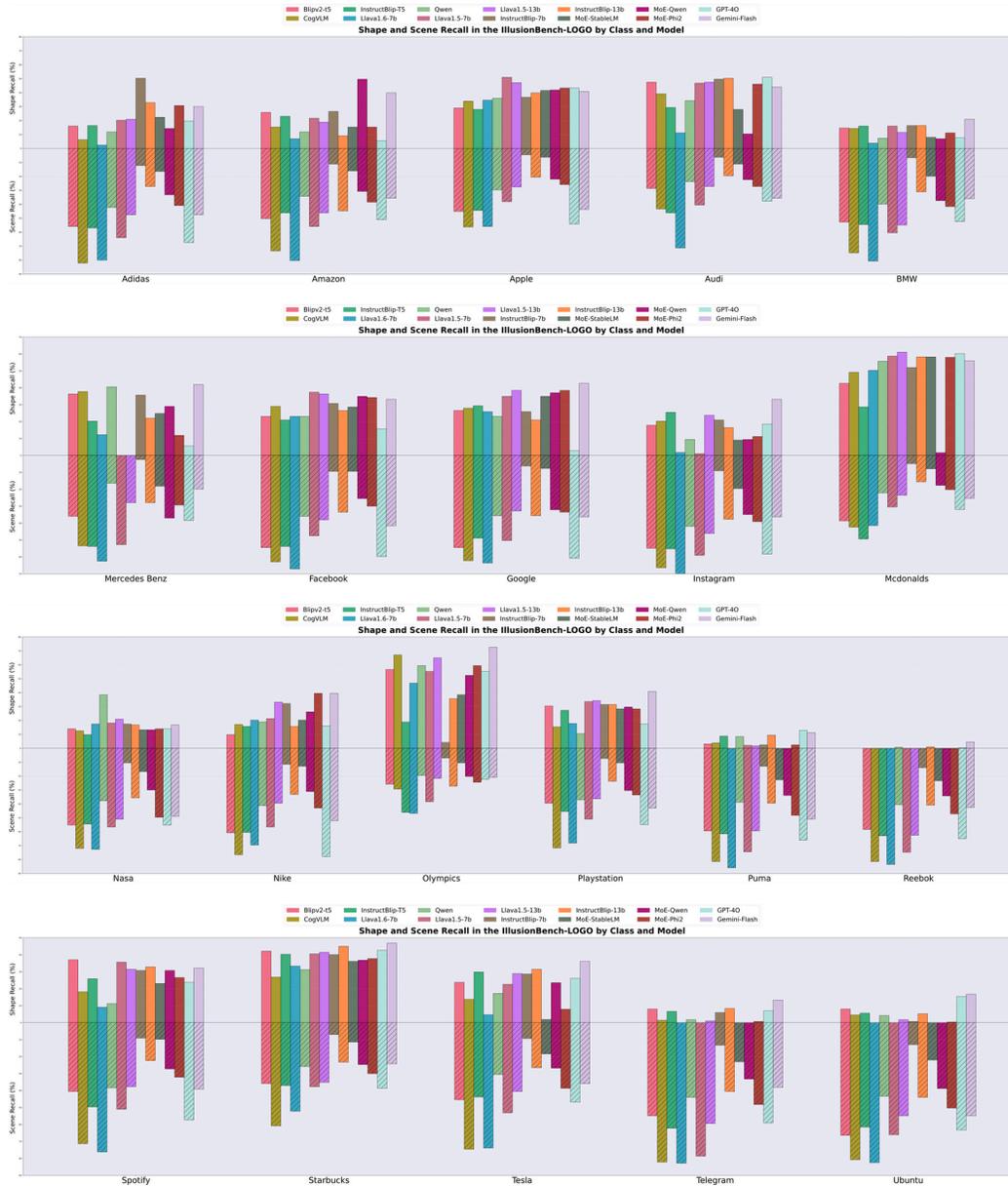
Figure 11: **Zero-shot Results on** `IllusionBench-LOGO`. zero-shot shape and scene recall of VLMs on the `IllusionBench-LOGO` dataset.
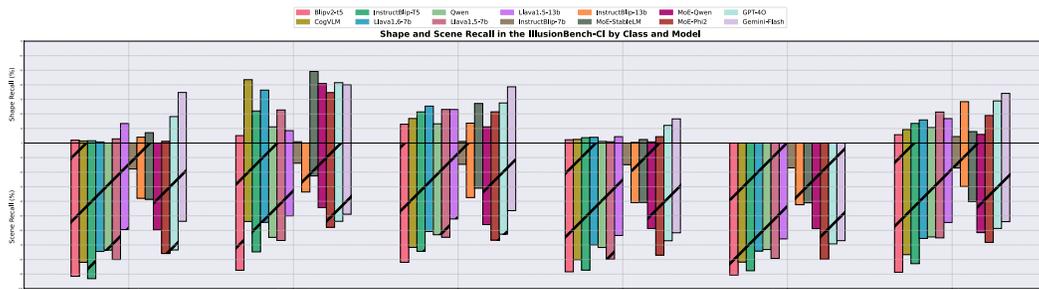
Figure 12: **Zero-shot Results on** `IllusionBench-ICON`. zero-shot shape and scene recall of VLMs on the `IllusionBench-ICON` dataset.

22
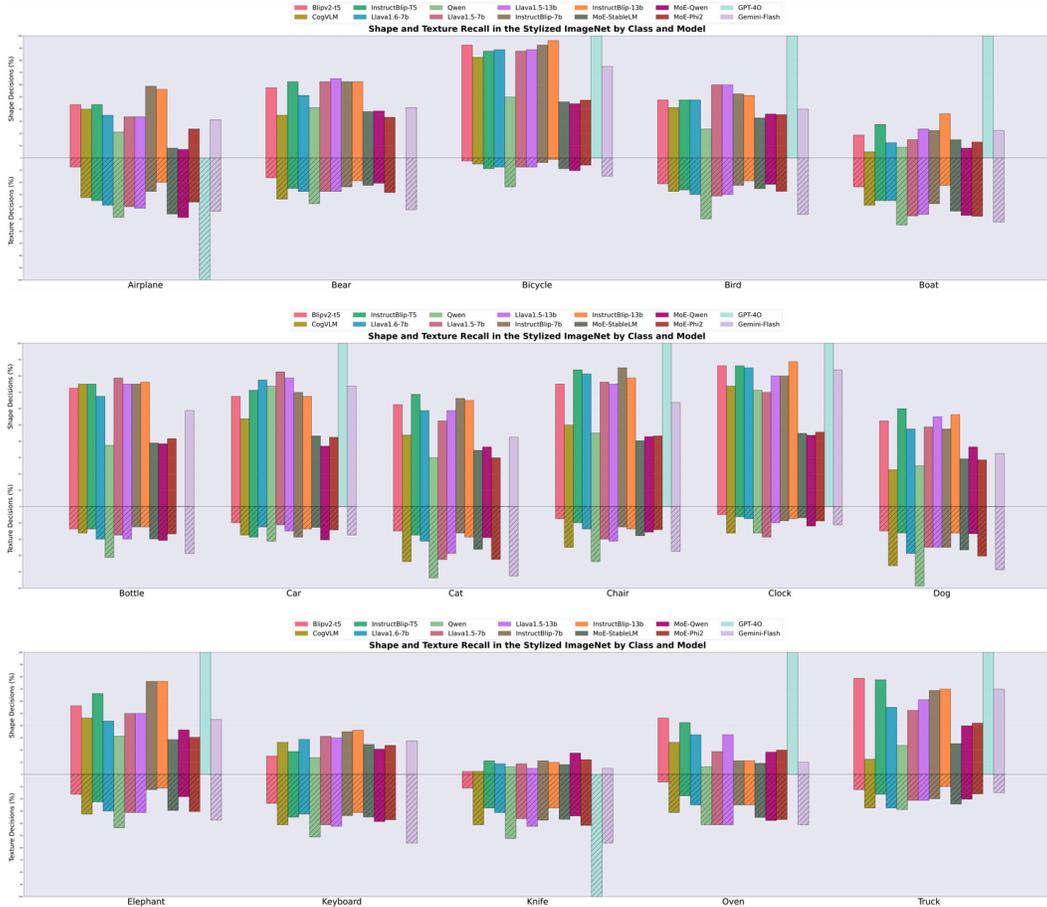
Figure 13: **Zero-shot Results on Stylized ImageNet**. zero-shot shape and texture decision of VLMs on the Stylized ImageNet dataset.

C In-Context Learning Experiments Details

C.1 In-Context Learning (ICL)

ICL is a method of adapting a model for an unseen task without any additional training or fine-tuning. Specifically, n-shot ICL consists of sequence of labelled demonstrations $\mathcal{C} = \{(x_{i_1}, y_{i_1}), \cdots, (x_{i_n}, y_{i_n})\}$. These are supplied to a model $p_{\boldsymbol{\theta}}(y|x)$ for an unseen task. The label corresponding to a test query x_* is predicted through the predictive distribution of the model conditioned on the demonstration set $\mathcal{C}$ alongside an instruction I for the new task:

$$p_{\boldsymbol{\theta}}(y|\mathcal{C}, I) = p_{\boldsymbol{\theta}}(y|x_{i_1}, y_{i_1}, \cdots x_{i_n}, y_{i_n}, I). \tag{1}$$

This learning method has proven to be an efficient and low-cost method for adapting LLMs to downstream tasks [Brown et al., 2020, Schick and Schütze, 2021, Winata et al., 2021, Liu et al., 2022]. The success of ICL for LLMs has led to recent research aiming to extend ICL to multi-modal models, where labeled demonstrations now contain interleaved image and text modalities [Alayrac et al., 2022, Bertini Baldassini et al., 2024, Zhao et al., 2023, Zong et al., 2024].

C.2 ICL Further Experimental Details

Considering we restrict evaluations to classes recognised in a zero-shot manner, we use the following class counts: 10 for the `IllusionBench-LOGO` split, 14 for the `IllusionBench-IN` split, and 6 for the icons split, utilizing all 11 scenes of the dataset. To overcome ICL biases like majority voting and recency bias, each shape and scene class is represented at most once within the context, with no repetitions, and new demonstrations are randomly sampled for each test sample.

23

C.3 Models Description

In our zero-shot experiments, we evaluate each of the following large vision language models (VLMs):

- LLaVA-Next (Vicuna-7b) [Liu et al., 2024b], a VLM operating at an input image resolution of 336^2, using Vicuna-7b-v1.5 [Zheng et al., 2024] and CLIP ViT-L/14 [Radford et al., 2021] as text and visual encoders, respectively. These are connected via simple projections.
- Qwen-VL-Chat [Bai et al., 2023], a 9B parameter model with an input resolution of 448^2, employing a cross-attention module to link an OpenClip ViT-bigG [Ilharco et al., 2021] vision encoder to a Qwen-7b [Bai et al., 2023] text backbone.
- Otter-MPT [Li et al., 2023a], a 9B parameter VLM based on the OpenFlamingo architecture [Awadalla et al., 2023], featuring an input image resolution of 224^2 and utilizing LLaMA-7B [Touvron et al., 2023] and CLIP-ViT-L/14 as text and image backbones, respectively, connected through cross-attention.
- IDEFICS-9B-Instruct [Laurençon et al., 2024], an open-source reproduction of Flamingo [Alayrac et al., 2022], with an input image resolution of 224^2, using cross-attention transformer blocks to connect LLaMA and OpenClip text and image backbones.
- MMICL-T5-XXL [Zhao et al., 2023], a 12B parameter model that employs a Q-former [Li et al., 2023b] to integrate language and image components within an InstructBlip-FLANT5-XXL [Dai et al., 2024] backbone. This model can handle complex prompts with interleaved text and images, allowing for text-image references through dummy demonstration tokens, and operates at an input image resolution of 224^2.

C.4 Prompts

We use the following general prompt template for our ICL experiments:

```
{TASK_INSTRUCTION}
{demonstration_image_1}
Answer: {demonstration_label_1}
{demonstration_image_2}
Answer: {demonstration_label_2}
:
{demonstration_image_n}
Answer: {demonstration_label_n}
{query_image}
Answer:
```

where `demonstration_image_i` and `demonstration_label_i` refer to the image and label for the ith demonstration used as the context for predicting the answer for the query image `query_image`. `TASK_INSTRUCTION` is the instruction used based on the prediction target and the dataset. We used the following `TASK_INSTRUCTION` prompts for predicting the shape, texture, and both the texture and shape simultaneously respectively:

24

```
# Predict shape
TASK_INSTRUCTION ='This image contains a {shape} integrated into a
background, where elements of the background contribute to forming
the image.
background options:  [{BG_OPTIONS}]
{shape} options:  [{SHAPE_OPTIONS}]
Identify the {shape} that is represented in the image by choosing
among the provided options.  Provide your response by stating only
the single, most accurate option that represents the {shape} in the
image.  You have to respond with a single word.'

# Predict texture
TASK_INSTRUCTION = 'This image contains a {shape} integrated into a
background, where elements of the background contribute to forming
the image.
background options:  [{BG_OPTIONS}]
{shape} options:  [{SHAPE_OPTIONS}]
Identify the background that is represented in the image by choosing
among the provided options.  Provide your response by stating only
the single, most accurate option that represents the background in
the image.  You have to respond with a single word.'

# Predict both texture and shape
TASK_INSTRUCTION = 'This image contains a {shape} integrated into a
background, where elements of the background contribute to forming
the image.
background options:  [{BG_OPTIONS}]
{shape} options:  [{SHAPE_OPTIONS}]
Identify BOTH the background AND the {shape} that are represented
in the image by choosing among the provided options.  Provide your
response by stating only the single, most accurate options that
represent the background and the {shape} in the image respectively.
You have to respond with two words, one predicting the background and
one predicting the {shape}'
```

where shape $\in$ {logo, object, icon}for the dataset IllusionBench-LOGO, IllusionBench-IN and
IllusionBench-CI respectively.

C.5 ICL Results: Exceptions

We list the exceptions to the general treneds reported in Section 5. We maintain the key takeaway
headings and format in Section 5 and discuss key exceptions.

- *ICL does not mitigate tendency to predict scene over shape.* LLaVA on the task τ_C (along
  the first row) stands as an exception, where the model demonstrates low scene prediction
  accuracy and non-trivial performance shape accuracy on ICL2 and ICL4.
- *Context selection strategy effects prediction tasks differently.*
  - τ_C : For LLaVA, including the shape through ICL2 or ICL4 for 1 or 2 shots leads to a
    significant performance increase over all other models. This is especially evident for
    1-shot, where we see high shape accuracy values of ICL2: 97.9% and ICL4: 99.9%.
    These high accuracy values indicate that the model exhibits a copying phenomenon
    [Bertini Baldassini et al., 2024], where for 1-shot, it simply copies the label from the
    ICL demonstration, which will have the same test label.
  - τ_S: QWEN shows an improvement in scene accuracy (for 4-shot, scene accuracies are
    ICL1: 51.4% and ICL3: 88.1%) when the scene is included in the context. Additionally,
    LLaVA exhibits a similar copying phenomenon for scene prediction in ICL3 and ICL4
    as discussed for τ_C but also shows some improvements over zero-shot for 2-shots.

25

$\tau_{C,S}$: As an exception, OTTER and QWEN show a general increase in scene accuracy on $\tau_{C,S}$ compared to τ_S, while their shape accuracy remains similar to τ_C. This suggests that predicting both shape and scene and including demonstrations with such predictions can help these models better disentangle scene from shape. Again, we observe the copying mechanisms in LLaVA described for τ_C and τ_S.

C.6 Individual Dataset Splits ICL Results
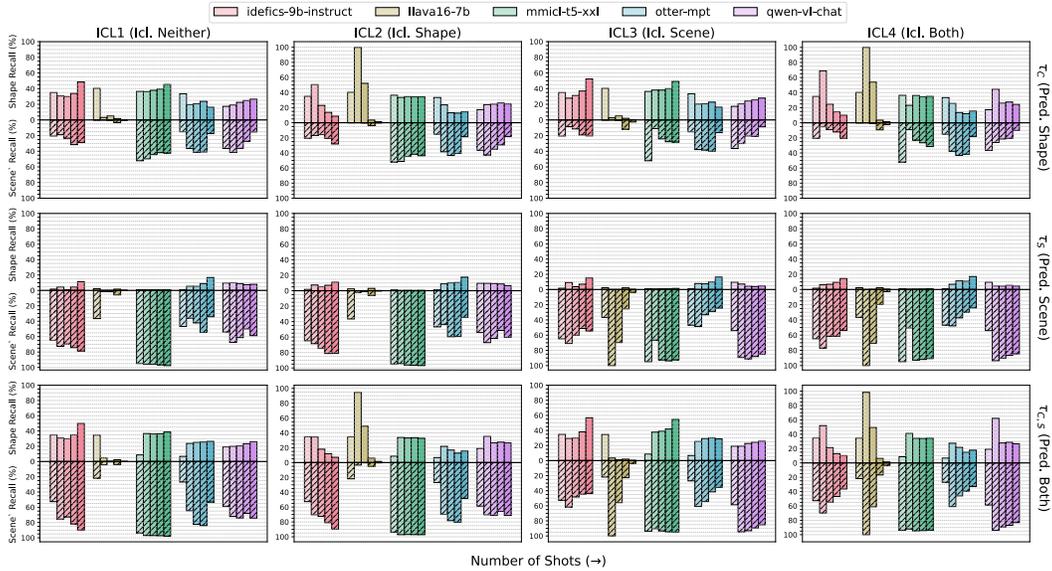
Figure 14: **ICL Results on IllusionBench-LOGO.** Few-shot shape and texture accuracy of VLMs on the IllusionBench-LOGO dataset across the different ICL learning tasks and the different prediction tasks.
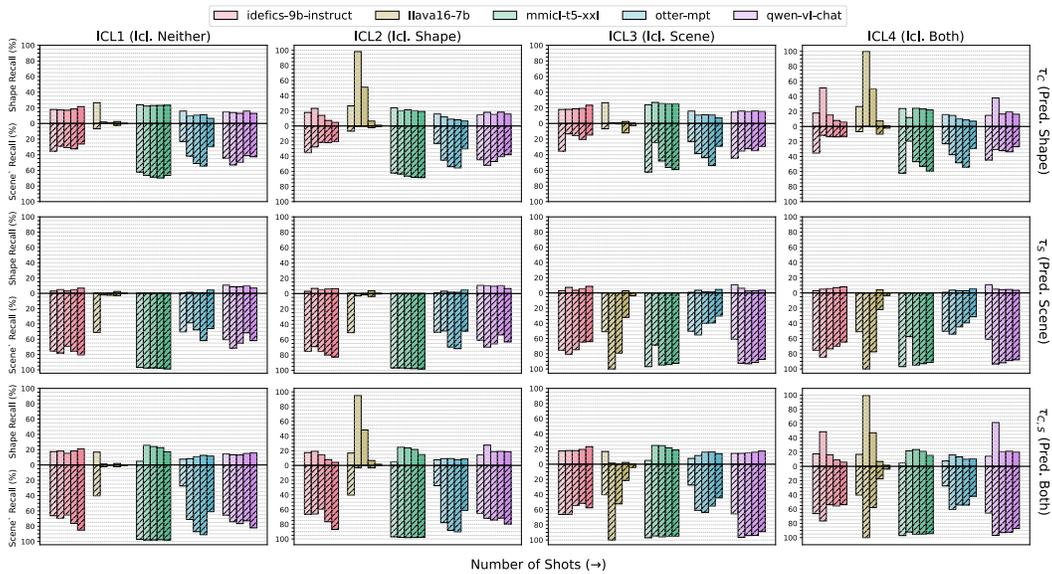
Figure 15: **ICL Results on IllusionBench-IN.** Few-shot shape and texture accuracy of VLMs on the IllusionBench-IN dataset across the different ICL learning tasks and the different prediction tasks.

26
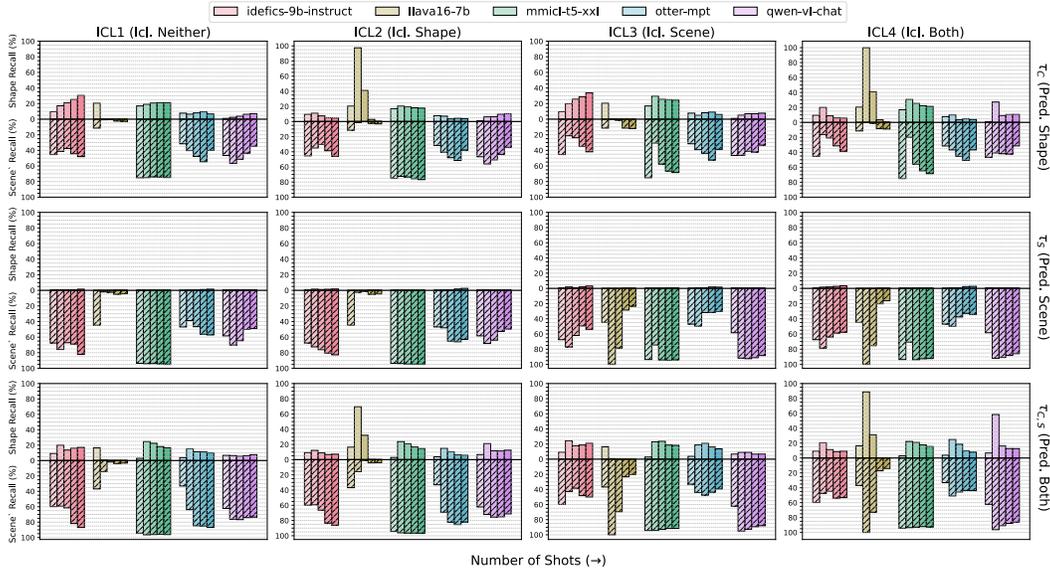
Figure 16: **ICL Results on `IllusionBench-ICON`.** Few-shot shape and texture accuracy of VLMs on the `IllusionBench-ICON` dataset across the different ICL learning tasks and the different prediction tasks.

C.7 Responses From Low Performing Models

We often observe close to 0% shape accuracy of the `LLaVA` model on shape prediction tasks across all four ICL-constrained ICL prediction tasks when using a higher number of ICL demonstrations. Figure 17 illustrates three example responses from the `LLaVA` model using 4-shot ICL for ICL3, which includes the test query background in the ICL demonstrations. From the example model responses R1, R2, and R3, it is evident that the `LLaVA` model tends to produce descriptive and verbose responses. Specifically, it fails to be concise and accurate, unlike the other models we investigate that usually respond with a single class prediction even with more shots. This verbosity leads to poor accuracy as the model fails to adhere to the prompt instructions of predicting a single class, resulting in the test class rarely being included in the model's responses.

> - **R1**: `The image shows a paper sculpture that resembles a stylized`
> - **R2**: `The image shows a logo integrated into a background that features a mountainous landscape`
> - **R3**: `The image shows a beautiful natural scene with a large rock formation in the ocean`

Figure 17: `LLaVA` **verbose responses.** Example responses from the `LLaVA` model for 4-shot shape prediction (T1) on the ICL3 learning task.

However, Fig. 18 shows example responses from the `LLaVA` model on the same task and for the same test queries as in Fig. 17 but using 2-shots. Observations from responses R1', R2', and R3' indicate that with fewer shots, the model is much more likely to produce single-class predictions or responses that are generally more concise and less descriptive. The differences observed with increasing numbers of shots suggest that `LLaVA`'s ability to correctly process and learn both the expected answer format and the task diminishes with a greater number of shots, highlighting its limitation as an in-context learner.

27

```
• R1': The logo in the image is Tesla.
• R2': The logo in the image is Starbucks.
• R3': Audi
```

Figure 18: LLaVA **concise responses.** Example responses from the LLaVA model for 2-shot shape prediction (T1) on the ICL3 learning task for the same test query as in Fig. 17.

D Domain Generalisation Experiments Details

D.1 Background Details

Domain generalisation has been a challenging task for image recognition. Several methods have been developed to improve training strategies for better generalisability of early specialist visual models, which are also applicable to CLIP models. Data augmentation strategies such as MixUp [Yan et al., 2020] and RegMixUp [Pinto et al., 2022b] are known to improve generalisation capacity through interpolation or extrapolation of data samples outside the training domain for diversity. GroupDRO [Sagawa et al., 2019] performs ERM with a re-weighting of classes with larger errors, making them more significant. VREx [Krueger et al., 2021] reduces differences in risk across training domains, which can decrease a model's sensitivity. Additionally, prompt learning, a promising approach for CLIP-style models, can also be leveraged for domain generalisation. Specifically, we adopt DPLCLIP [Zhang et al., 2023b], which trains a prompt generator during the training phase and infers unseen domains.

D.2 Further Experiment Details

CLIP Model For all experiments, the image encoder backbone of CLIP model is a ResNet50 [He et al., 2016]. For full-parameter fine-tuning, we train the whole image encoder, whereas for linear probing we only train the projection layer. The inferent prompt template for all methods is ``A photo of [Class name]''.

Training Hyperparameters For all experiments, we use a batch size of 32 and the Adam optimiser [Kingma and Ba, 2014] with a learning rate of 5e-5. For full parameter fine-tuning, we train the model for 1000 steps, and for linear probing, we train the model for 800 steps. For MixUp [Yan et al., 2020] and RegMixUp [Pinto et al., 2022b], the alpha and beta are both set to 0.2. For GroupDRO [Sagawa et al., 2019], the eta is set to 1e-2. For VREx [Krueger et al., 2021], the penalty weight is set to 1.0. For DPLCLIP [Zhang et al., 2023b], the number of domain tokens is 16.

E Compute Resources

All experiments are performed on our internal cluster.

Resources for image generation For the Image generation, we used three A40 GPUs with 45 GB RAM with around 65h to generate all of the images in the dataset.

Resources for zero-shot experiments For the zero-shot experiments, we used eight A40 GPUs with 45 GB RAM for around 250h total to cover all Zero-shot experiments experiments.

Resources for in-context learning experiments We perform ICL inference using 8 A40 GPUs with 45GB RAM for around 168h total to cover all ICL experimental settings.

Resources for domain generalisation experiments For each fine-tuning CLIP we use a single A40 GPUs with 45GB RAM for an hour on average for full parameter fine-tuning and half an hour for linear probing.

28