# Presentation: Multimodality

Weizhi Wang

# Vision, Language, Vision-Language Tasks

| Corpus | \|Train\| | \|Test\| | Task | Metrics | Domain |
|---|---|---|---|---|---|
| | | | Single-Sentence Tasks | | |
| CoLA | 8.5k | **1k** | acceptability | Matthews corr. | misc. |
| SST-2 | 67k | 1.8k | sentiment | acc. | movie reviews |
| | | | Similarity and Paraphrase Tasks | | |
| MRPC | 3.7k | 1.7k | paraphrase | acc./F1 | news |
| STS-B | 7k | 1.4k | sentence similarity | Pearson/Spearman corr. | misc. |
| QQP | 364k | **391k** | paraphrase | acc./F1 | social QA questions |
| | | | Inference Tasks | | |
| MNLI | 393k | **20k** | NLI | matched acc./mismatched acc. | misc. |
| QNLI | 105k | 5.4k | QA/NLI | acc. | Wikipedia |
| RTE | 2.5k | 3k | NLI | acc. | news, Wikipedia |
| WNLI | 634 | **146** | coreference/NLI | acc. | fiction books |

Table 1: Task descriptions and statistics. All tasks are single sentence or sentence pair classification, except STS-B, which is a regression task. MNLI has three classes; all other classification tasks have two. Test sets shown in bold use labels that have never been made public in any form.

# Vision, Language, Vision-Language Tasks

| Category | Task | Dataset | Metric | Previous SOTA | BEIT-3 |
|---|---|---|---|---|---|
| Vision | Semantic Segmentation | ADE20K | mIoU | 61.4 (FD-SwinV2) | **62.8** (+1.4) |
| | Object Detection | COCO | AP | 63.3 (DINO) | **63.7** (+0.4) |
| | Instance Segmentation | COCO | AP | 54.7 (Mask DINO) | **54.8** (+0.1) |
| | Image Classification | ImageNet† | Top-1 acc. | 89.0 (FD-CLIP) | **89.6** (+0.6) |
| Vision-Language | Visual Reasoning | NLVR2 | Acc. | 87.0 (CoCa) | **92.6** (+5.6) |
| | Visual QA | VQAv2 | VQA acc. | 82.3 (CoCa) | **84.0** (+1.7) |
| | Image Captioning | COCO‡ | CIDEr | 145.3 (OFA) | **147.6** (+2.3) |
| | Finetuned Retrieval | COCO | R@1 | 72.5 (Florence) | **76.0** (+3.5) |
| | | Flickr30K | | 92.6 (Florence) | **94.2** (+1.6) |
| | Zero-shot Retrieval | Flickr30K | R@1 | 86.5 (CoCa) | **88.2** (+1.7) |

Table 1: Overview of BEIT-3 results on various vision and vision-language benchmarks. We compare with previous state-of-the-art models, including FD-SwinV2 [WHX+22], DINO [ZLL+22], Mask DINO [ZLL+22], FD-CLIP [WHX+22], CoCa [YWV+22], OFA [WYM+22], Florence [YCC+21]. We report the average of top-1 image-to-text and text-to-image results for retrieval tasks. "†" indicates ImageNet results only using publicly accessible resources. "‡" indicates image captioning results without CIDEr optimization.

# Connecting Images and Texts

- Pathway 1: task-agnostic joint representations of image and text
  - Two-Stream model: VL-BERT
  - One-Stream model: OSCAR

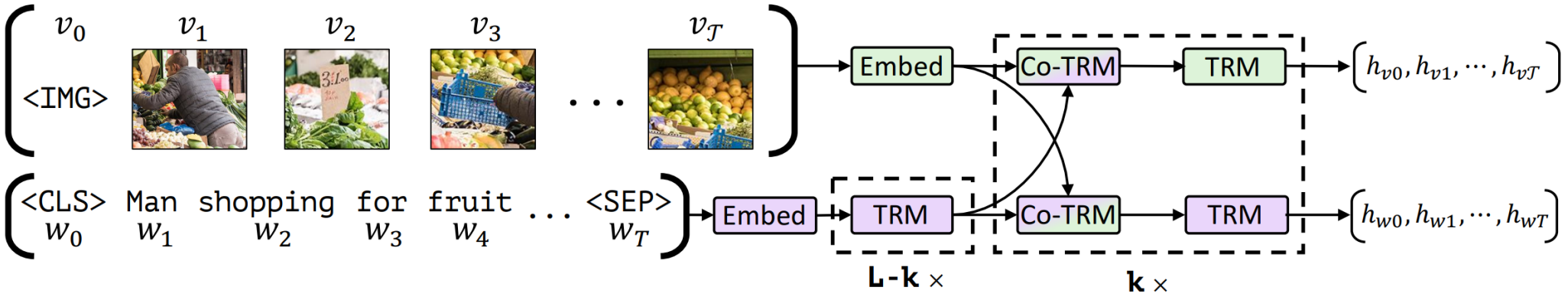



Figure 1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers.

- Pathway 2: contrastive learning
  - CLIP
  - ALIGN

# CLIP: Connecting Images and Texts

- ViLBERT: Connecting Regions of Patches with Tokens at different length
- Data: Unsupervised, Supervised (ImageNet->JFT300M, LAION400M)
- CLIP: Connecting Sentences with Images

- Goal: Unifying Text and Image into one embedding space
- Architecture:
  - Transformer Decoder for encoding text at sentence level
  - ResNet/ViT for encoding image at image level
  - Matching the multi-modal embedding space via simple dot-production

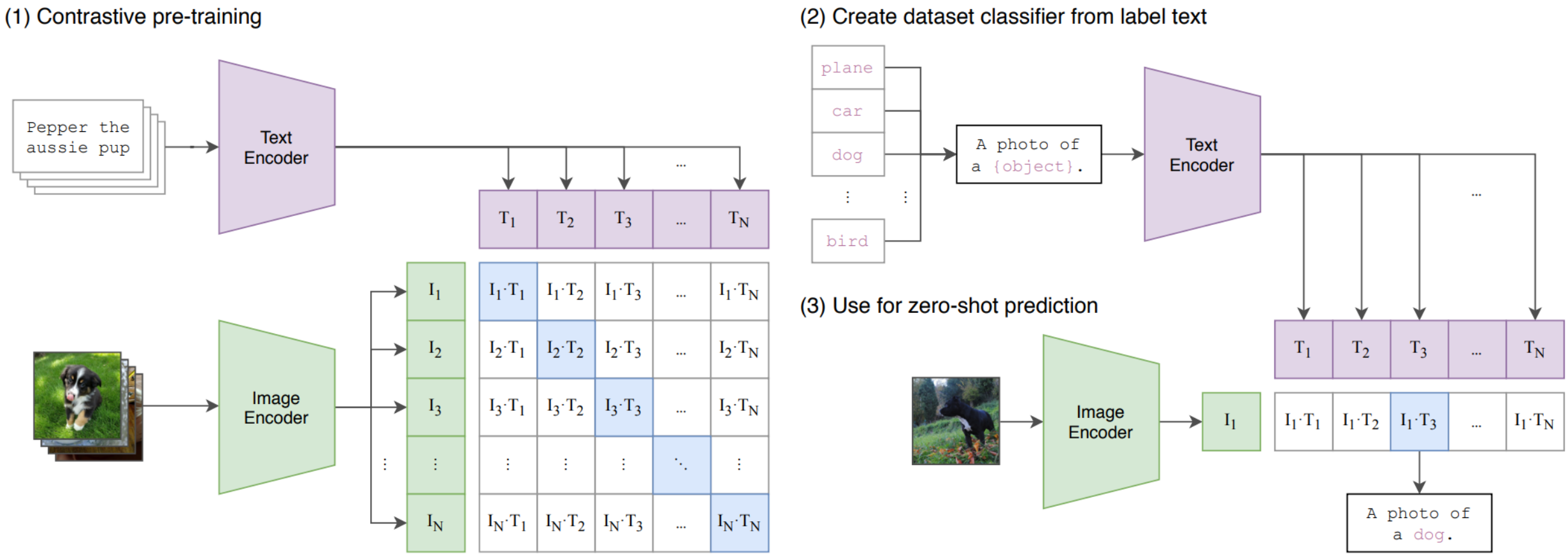



*Figure 1.* Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

*Figure 3.* Numpy-like pseudocode for the core of an implementation of CLIP.

# Training Loss

- n : batch size
- l : sequence length
- d_i : image embedding dim
- d_e : text embedding dim
- T_f : output[:, -1, :]

# CLIP: Training Details

- The short length image captions input with the limitation of 75 tokens
- Very large minibatch size of 32,768
- The largest ResNet model, RN50x64, took 18 days to train on 592 V100 GPUs
- The largest Vision Transformer took 12 days on 256 V100 GPUs
- ViT-L/14@336px (ViT-L/14 with higher 336pixel resolution) performs best, denoted as CLIP later
- LAION dataset pre-training, the largest image-text pair dataset at that time with 400M pairs. Now LAION comes to 5B

# Zero-shot Comparison

| | aYahoo | ImageNet | SUN |
|---|---|---|---|
| Visual N-Grams | 72.4 | 11.5 | 23.0 |
| CLIP | **98.4** | **76.2** | **58.5** |

*Table 1.* Comparing CLIP to prior zero-shot transfer image classification results. CLIP improves performance on all three datasets by a large amount. This improvement reflects many differences in the 4 years since the development of Visual N-Grams (Li et al., 2017).

Zero-shot Obstacles for CV:

- Limited Size of ImageNet
- Feature to Inference

# Results on 27 datasets

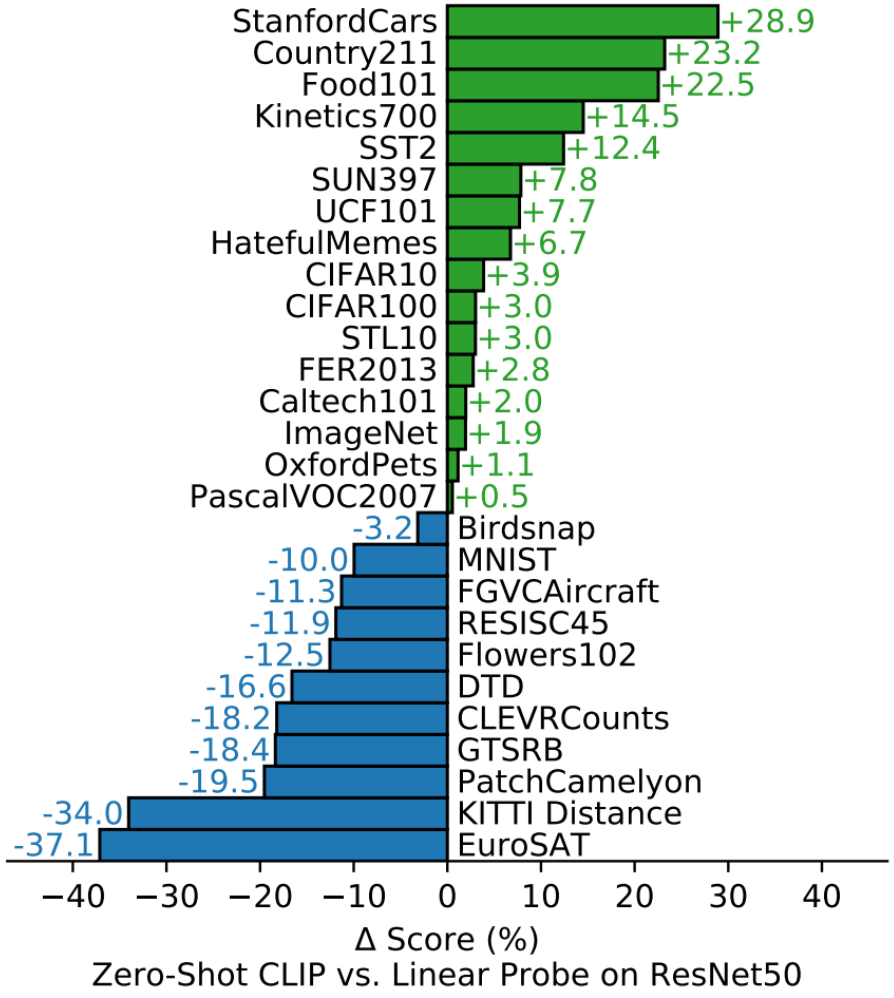



*Figure 5.* **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

Linear Probe baseline:

- Pre-train ResNet-50 on ImageNet
- Throw away the last linear layer mapping E to N_classes
- Fix pre-train parameters
- Fully supervised, regularized, logistic regression classifier train on each downstream task

# Supervised Linear Probing Results

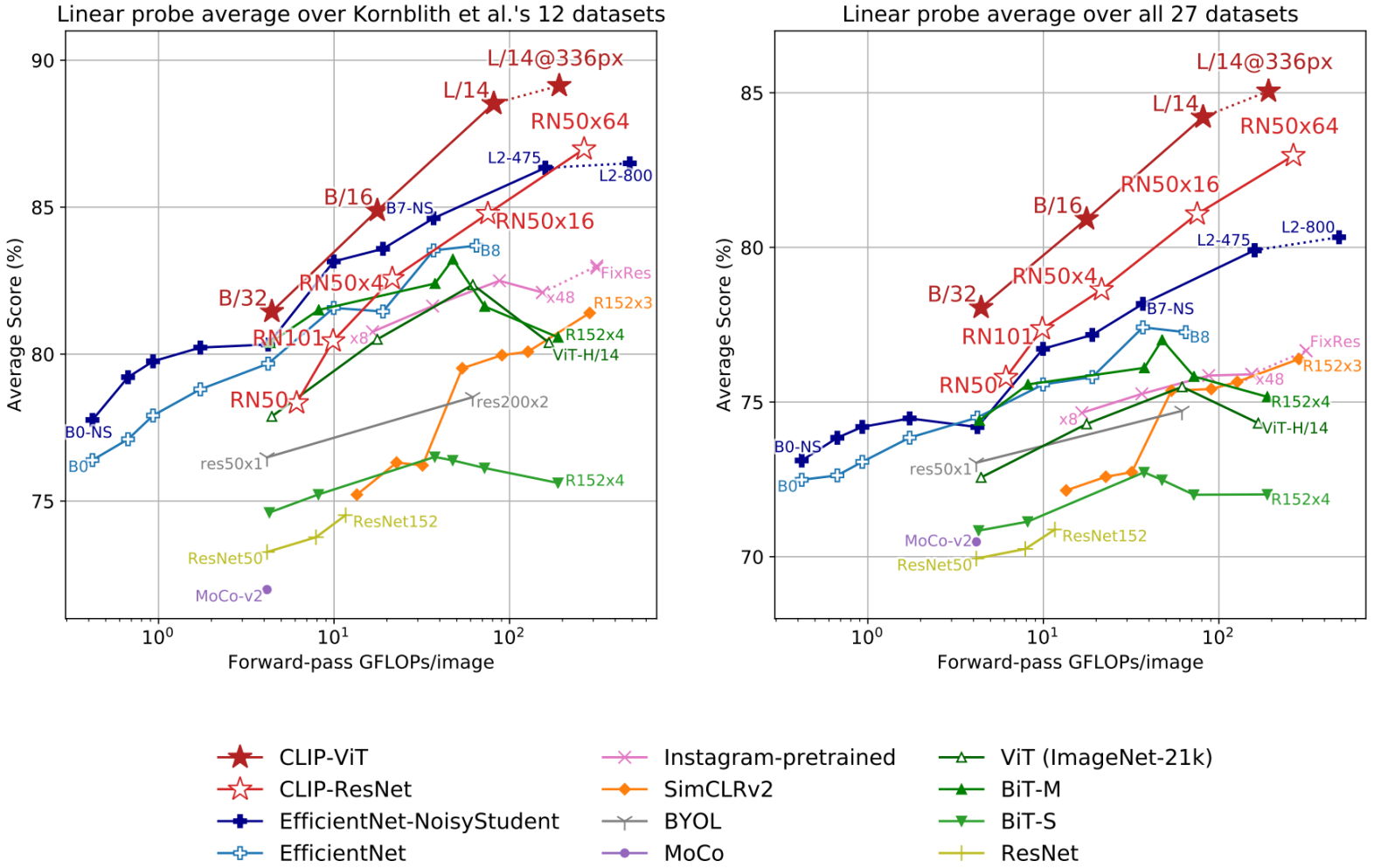



*Figure 10.* **Linear probe performance of CLIP models in comparison with state-of-the-art computer vision models**, including EfficientNet (Tan & Le, 2019; Xie et al., 2020), MoCo (Chen et al., 2020d), Instagram-pretrained ResNeXt models (Mahajan et al., 2018; Touvron et al., 2019), BiT (Kolesnikov et al., 2019), ViT (Dosovitskiy et al., 2020), SimCLRv2 (Chen et al., 2020c), BYOL (Grill et al., 2020), and the original ResNet models (He et al., 2016b). (Left) Scores are averaged over 12 datasets studied by Kornblith et al. (2019). (Right) Scores are averaged over 27 datasets that contain a wider variety of distributions. Dotted lines indicate models fine-tuned or evaluated on images at a higher-resolution than pre-training. See Table 10 for individual scores and Figure 20 for plots for each dataset.

# Supervised Linear Probing Results

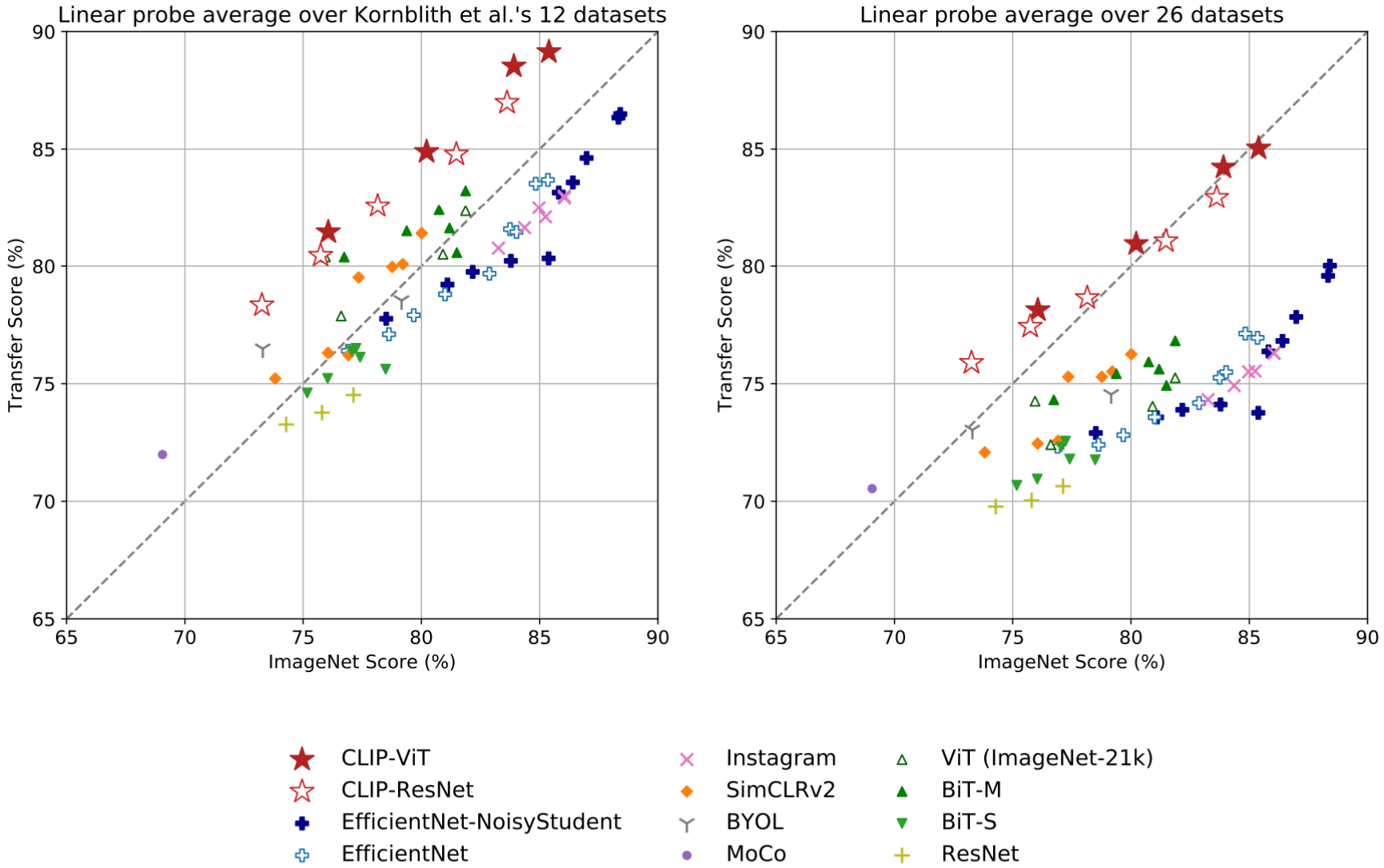



*Figure 12.* **CLIP's features are more robust to task shift when compared to models pre-trained on ImageNet.** For both dataset splits, the transfer scores of linear probes trained on the representations of CLIP models are higher than other models with similar ImageNet performance. This suggests that the representations of models trained on ImageNet are somewhat overfit to their task.

# Big Convergence of Vision and Language

Language Helps Vision:

Architecture Perspective: Vision Transformer->BEiT/Masked AutoEncoder

Pre-training Objective Perspective: MLM->Masked Image Modeling

Vision-Language Joint Modeling:

VLBERT, VisualBERT, CLIP, ALIGN

**Big question: Can vision or visual knowledge help language?**

# Big Convergence of Vision and Language

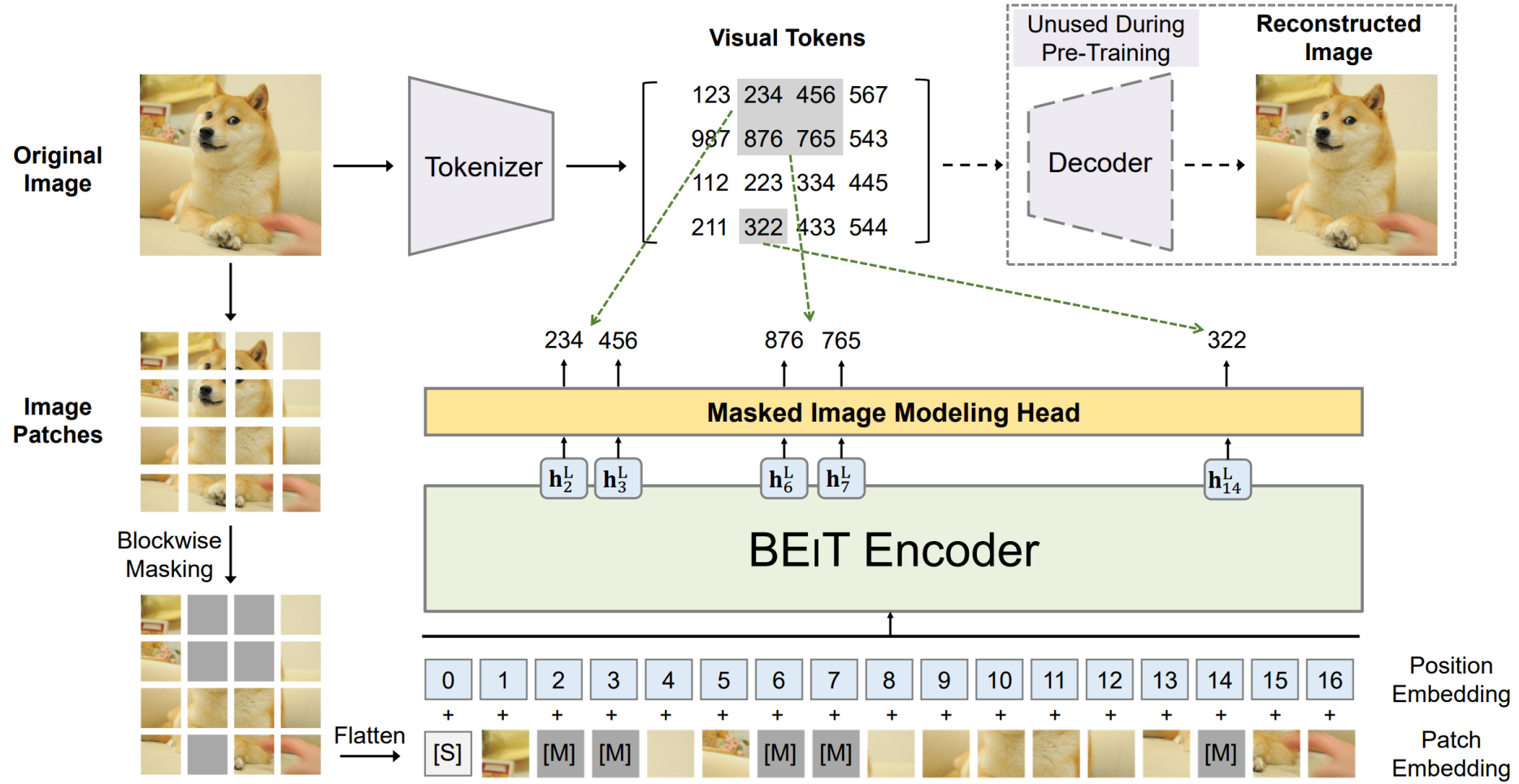
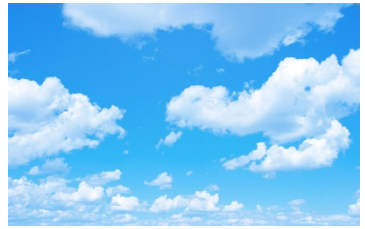



Figure 1: Overview of BEIT pre-training. Before pre-training, we learn an "image tokenizer" via autoencoding-style reconstruction, where an image is tokenized into discrete visual tokens according to the learned vocabulary. During pre-training, each image has two views, i.e., image patches, and visual tokens. We randomly mask some proportion of image patches (gray patches in the figure) and

# Simple Questions Which Requires Visual Knowledge

What is the color of the sky?



Is sofa larger than a cat?

What is the shape of an apple?

# Visually-Augmented Language Modeling

Motivation:

- Demonstrate that vision can also help language modeling
- Propose a novel area of Vision for NLP

Prerequisite:

- CLIP provides a good mapping between a sentence and an image
- LAION is constructed with 400M image-text pairs
- GPU-enabled Dense Retrieval with Inner Product Scoring is available

# VaLM Architecture

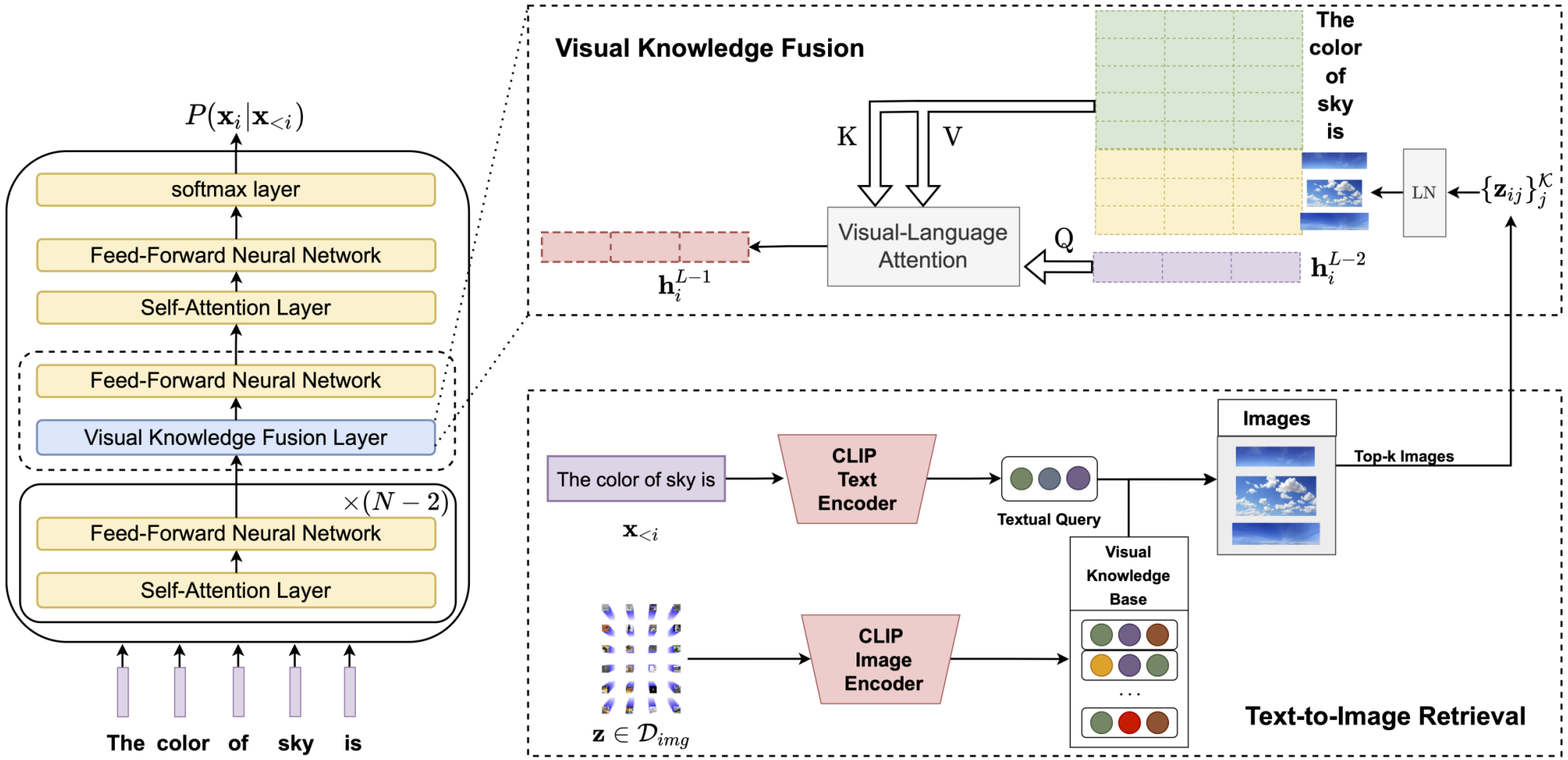



Figure 1: Overview of visually-augmented language modeling (VALM). We conduct dense retrieval to get top-$k$ images for the input context at each time step. Then the visual knowledge fusion layer attends to both text tokens and retrieved images. The vision-language fused representation is fed back to Transformer for language modeling.

# Training Objective

## Training Objective: Causal Language Modeling

Maximize the likelihood on the text corpus:

$$\mathbf{H}^l = \text{Layer}_l(\mathbf{H}^{l-1}), l \in [1, L] \qquad \mathbf{H}^{L-1} = \text{VisualLayer}(\{\mathbf{H}_i^{L-2}, \{\mathbf{z}_{ij}\}_{j=1}^{\mathcal{K}}\}_{i=1}^{N})$$

$$P(\mathbf{x}_i|\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}) = \text{softmax}(W\mathbf{H}^L + b)$$

We conduct *generative unsupervised pre-training* (Radford et al., 2019) for VALM on a large-scale text corpus. The training objective of VALM is the standard left-to-right language modeling objective, which maximizes the likelihood of the next word token based on the left context:

$$\max \sum_{x \in \mathcal{D}} \sum_{i=1}^{|\mathbf{x}|} \log P(\mathbf{x}_i|\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}), \tag{1}$$

where $\mathbf{x}$ represents a sentence randomly sampled from the large-scale pre-training text corpus $\mathcal{D}$.

# Text-Image Retriever

Revisit: the text input length to CLIP text encoder is 75

Image Knowledge Base Creation:
Input each image in LAION to get a E-dim image encoding, Now we have 200M * IMG-E dimensional KB vector

Textual Query:
In order to map each token to K images, truncate the sentence with sliding window

Nearest Neighbor Retrieval:
Retrieve the top-k images in KB given the token encoding query w.r.t. IP score

# Visual Knowledge Fusion Layer

the hidden state output for $i$-th token is $\mathbf{h}_i$ and the corresponding retrieved images are $\{\mathbf{z}_{ij}\}_{j=1}^{\mathcal{K}}$, the hidden state $\mathbf{H}_i^{L-1}$ is computed as:

$$\mathbf{Q} = \mathbf{H}^{L-2}W^Q + b^Q, \mathbf{K} = \mathbf{H}^{L-2}W^K + b^K, \mathbf{V} = \mathbf{H}^{L-2}W^V + b^V, \tag{3}$$

$$\dot{\mathbf{k}}_{ik} = \mathrm{LN}_{img}(\mathbf{z}_{ik})W^K + b^K_{img}, \dot{\mathbf{v}}_{ik} = \mathrm{LN}_{img}(\mathbf{z}_{ik})W^V + b^V_{img}, \tag{4}$$

$$e_i = \frac{\mathbf{Q}_i\mathbf{K}^T}{\sqrt{d}}, a_i = \frac{\exp(e_i)}{\sum_{j=1}^{\mathcal{L}}\exp(e_{ij}) + \sum_{k=1}^{\mathcal{K}}\exp(e_{ik})}, \tag{5}$$

$$e_{ik} = \frac{\mathbf{Q}_i\dot{\mathbf{k}}_{ik}^T}{\sqrt{d}}, a_{ik} = \frac{\exp(e_{ik})}{\sum_{j=1}^{\mathcal{L}}\exp(e_{ij}) + \sum_{k=1}^{\mathcal{K}}\exp(e_{ik})}, \tag{6}$$

$$\mathbf{H}_i^{L-1} = a_i\mathbf{V} + \sum_k a_{ik}\dot{\mathbf{v}}_{ik}, \tag{7}$$

where $\mathbf{Q}_i, \dot{\mathbf{k}}_{ik}, \dot{\mathbf{v}}_{ik} \in \mathcal{R}^{\mathrm{E}}$, $\mathbf{K}, \mathbf{V} \in \mathcal{R}^{|\mathbf{x}|\times\mathrm{E}}$, $e_i, a_i \in \mathcal{R}^{|\mathbf{x}|}$. The hidden state output from previous layer $\mathbf{H}_\mathbf{i}^{\mathbf{L-1}}$ is linearly projected into contextual queries, keys, and values $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ separately. $\mathcal{K}$ is the number of retrieved images for each token, and E is the embedding dimension for both context and image representations. In order to generate image-specific attention keys and values, we adopt image-specific bias $b^K_{img}, b^V_{img}$ in linear projections and reuse the contextual projection weights $W^K, W^V$ to generate image-specific attention keys and values. Moreover, it is vital to mention that the image-specific attention keys and values are distinct for each query token, which is highly different from self-attention where the contextual keys and values are kept the same for each token. A secondary subscript $k$ is used to denote different image representations for the $i$-th token.

# Training and Evaluation Details

Pre-trained Text corpus: CC-100, 400GB raw text

Image Data: LAION, half, 200M

Model Architecture: GPT2 Base, 117M

Evaluation Tasks:

- Visual- knowledge intensive tasks:

| Task | Example Prompt | Object / Pair | Answer |
|---|---|---|---|
| Object Color Reasoning | *The color of* [object] *is* [answer] | *the sky* | *blue* |
| Object Shape Reasoning | *The shape of* [object] *is* [answer] | *apple* | *round* |
| Object Size Reasoning | *Is* [Item1] *larger than* [Item2]*?* [answer] | (*sofa*, *cat*) | *Yes* |

Table 1: Evaluation examples of object color, shape, and size reasoning.

- Natural Language Processing Tasks

# Results on Visual-Knowledge Intensive Tasks

| Model | $\mathcal{K}$ | Color (ACC↑) MEMORYCOLOR | Color (ACC↑) COLORTERMS | Shape (ACC↑) OBJECTSHAPE | Size (ACC↑) RELATIVESIZE |
|---|---|---|---|---|---|
| GPT-2* | N/A | 44.14% | 39.10% | 51.09% | 47.22% |
| BERT | N/A | 24.34% | 26.33% | 31.86% | 34.78% |
| CaptionBERT | N/A | 24.84% | 28.40% | 38.14% | 66.05% |
| CLIP | N/A | 26.25% | 23.08% | 13.66% | 47.99% |
| OSCAR | N/A | 20.32% | 16.86% | 33.14% | 50.14% |
| VisualBERT | N/A | 26.68% | 38.02% | 11.14% | 67.23% |
| VALM | 4 | 53.99% | **52.66%** | **62.77%** | **85.03%** |
| VALM | 8 | **58.64%** | 50.19% | 59.41% | 62.35% |

Table 2: Accuracy on object commonsense reasoning datasets. GPT-2* is the re-implemented model with identical pre-training data and hyper-parameter settings to VALM. $\mathcal{K}$ represents for the number of images augmented to each token. Best performance is marked with bold.

# Results on NLP tasks

| Model | $\mathcal{K}$ | SST-2 ACC↑ | MPQA ACC↑ | DBPedia ACC↑ | AGNews ACC↑ |
|---|---|---|---|---|---|
| Majority | N/A | 50.90% | 50.00% | 9.4% | 25.0% |
| GPT-2* | N/A | 68.04% | 71.25% | 67.20% | 53.51% |
| VALM | 4 | 70.12% | 78.70% | 72.27% | 53.81% |
| VALM | 8 | 67.33% | 77.35% | 68.48% | 59.77% |

Table 4: Zero-shot evaluation results on natural language understanding tasks (SST-2, MPQA, DBPedia, AGNews).

| Model | $\mathcal{K}$ | Wikitext-103 PPL↓ | Lambada PPL↓ | Lambada ACC↑ |
|---|---|---|---|---|
| GPT-2* | N/A | 36.44 | 42.46 | 42.17% |
| VALM | 4 | 35.78 | 42.51 | 42.65% |
| VALM | 8 | 35.76 | 42.38 | 42.87% |

Table 5: Zero-shot evaluation results on language modeling tasks. We report perplexity (PPL) on Wikitext-103 and Lambada and final word prediction accuracy (ACC) on Lambada.

# Retrieval Cost

| Image Size | Color (ACC↑) MEMORYCOLOR | Color (ACC↑) COLORTERMS | Shape (ACC↑) OBJECTSHAPE | Size (ACC↑) RELATIVESIZE | Timecost (GPT2* as 1x) |
|---|---|---|---|---|---|
| 200M | 53.99% | 52.66% | 62.77% | 85.03% | 2.06x |
| 100M | 53.50% | 49.71% | 61.39% | 81.84% | 1.91x |
| 10M | 51.79% | 47.49% | 62.18% | 82.15% | 1.79x |
| 1M | 51.87% | 46.31% | 48.51% | 82.35% | 1.74x |

Table 7: Accuracy on object commonsense reasoning datasets of VALM ($\mathcal{K} = 4$) with variants of retrieval imageset size. $\mathcal{K}$ represents for the number of images augmented to each token.

# Case Study

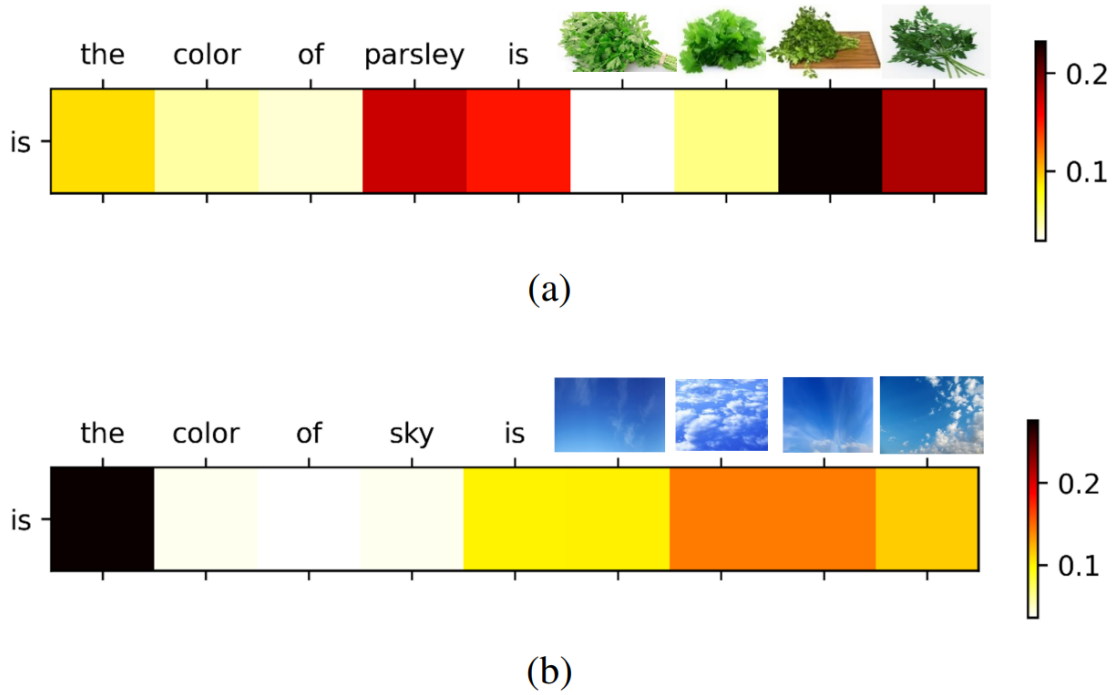
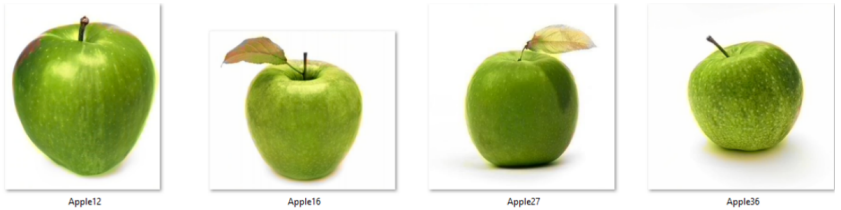
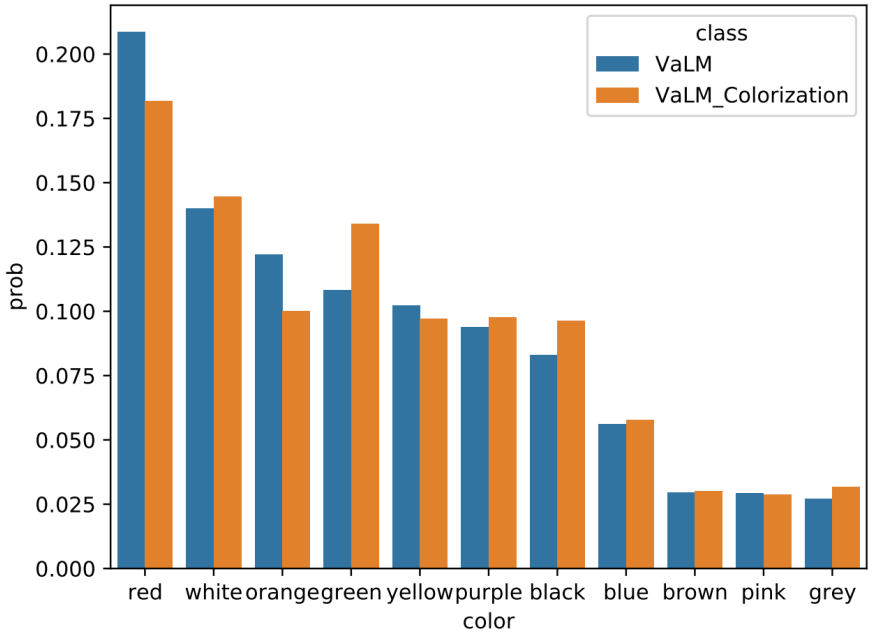



Figure 2: The attention matrix visualization given the query prompt "the color of [object] is" for VALM. VALM achieves accurate image retrieval of top-4 images corresponding to the objects of sky and parsley as augmented images, shown in the horizontal index of each subfigure.

# Case Study




(a) Images for green apples in OBJECTCOLORIZATION dataset as replacements.




(b) Probability Distribution Visualization for retrieved images and colorization images.

Figure 3: The visualization of the predicted probability distribution on 11 object color types with retrieved images and colorization images, respectively. The adopted prompt for reasoning the object color of an apple is “the color of [object] is”.