CLIP → SigLIP 2

Vision-language alignment & open-vocabulary representations

Seminal anchor: CLIP — Radford et al., ICML 2021
Frontier anchor: SigLIP 2 — Tschannen et al., arXiv 2025

Bin Xia · Jingyu Liu
CS 8803-VLM · Sep 3, 2026

Georgia Tech

Roadmap

- Problem: closed-set visual labels are too narrow
- CLIP: natural-language supervision, WIT data, contrastive loss, and zero-shot transfer
- CLIP evidence: prompt sensitivity, broad-but-uneven transfer, and robustness
- CLIP variants: richer captions, longer text encoders, and dense prediction
- Bridge to SigLIP: sigmoid loss, batch-size behavior, and why SigLIP 2 inherits this direction
- SigLIP 2: a modern encoder recipe built on the same alignment idea

Running question: what had to stay from CLIP, and what had to change for the VLM era?

Problem statement: class labels are too narrow

Classic supervised vision

- Predict one class ID from a fixed vocabulary
- Works when the target labels are known in advance
- Adding a new concept usually means collecting new labeled data

Open-vocabulary vision

- Use text as the interface to visual concepts
- Specify categories at test time with prompts
- Reuse the same model for retrieval, classification, and grounding

The main shift is from a closed classifier head to an image-text embedding space.

Before CLIP: natural-language supervision had a history

CLIP did not invent image-text learning; its contribution was to simplify and scale it.

Earlier image-text supervision

- Mori et al. 1999: predict nouns/adjectives from paired text documents
- Joulin et al. 2016: bag-of-words prediction from Flickr metadata
- Li et al. 2017: visual n-grams and zero-shot transfer
- VirTex / ICMLM 2020: caption LM and masked LM objectives

What CLIP keeps and scales

- ConVIRT 2020: contrastive image-text representation learning
- CLIP: a simplified ConVIRT-style objective trained from scratch at web scale
- Main simplification: match the correct image-text pair, not generate the whole caption

This slide sets up the key design choice: use natural language as supervision, but choose an objective that can scale.

Radford et al. 2021, Sec. 1–2.1; related work includes Mori 1999, Joulin 2016, Li 2017, VirTex, ICMLM, and ConVIRT.

Building WIT: data collection is part of the method

CLIP's 400M pairs were not just "more data"; the collection process encouraged broad concept coverage.

Existing caption data
COCO / Visual Genome:
~100K training images each

YFCC100M issue
100M images, but
natural-language metadata is
sparse and noisy

After filtering
English natural-language
titles/descriptions leave ~15M
photos

WIT solution
Collect 400M image-text pairs
from public Internet sources

Query vocabulary

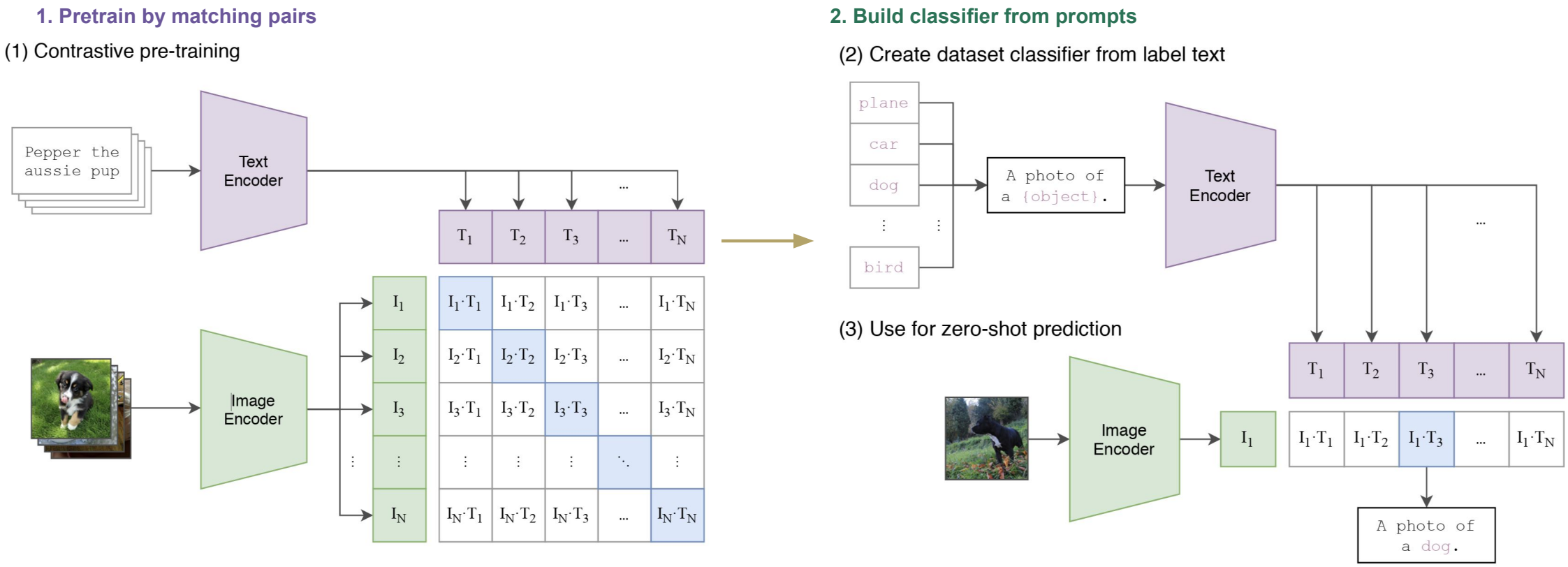
- Use about 500K text queries to cover broad visual concepts
- Queries include frequent Wikipedia words, bigrams, article names, and WordNet synsets

Balancing heuristic

- Cap at roughly 20K pairs per query
- Approximate balancing prevents common concepts from dominating the dataset

Radford et al. 2021, Sec. 2.2: Creating a Sufficiently Large Dataset.

CLIP approach: one model, labels written as text



Train image and text encoders together; at test time, the text encoder builds the classifier from prompts.

Radford et al. 2021, Fig. 1.

The CLIP loss: learn the diagonal of the batch

For a batch of image-text pairs:

S_{ij} = cosine($image_i$, $text_j$) / τ,
exp(t) = 1/τ in the implementation

maximize S_{ii}; suppress S_{ij} for i ≠ j

correct pairs

distractors

- Symmetric contrastive learning over the similarity matrix
- No task-specific classifier is trained during pretraining
- The learned space supports retrieval and zero-shot prediction

Why large batch matters
N = 32,768 ⇒ many in-batch negatives
(~N^2 mismatch pairs)

CLIP loss in code: the Figure 3 implementation

```
# 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
```

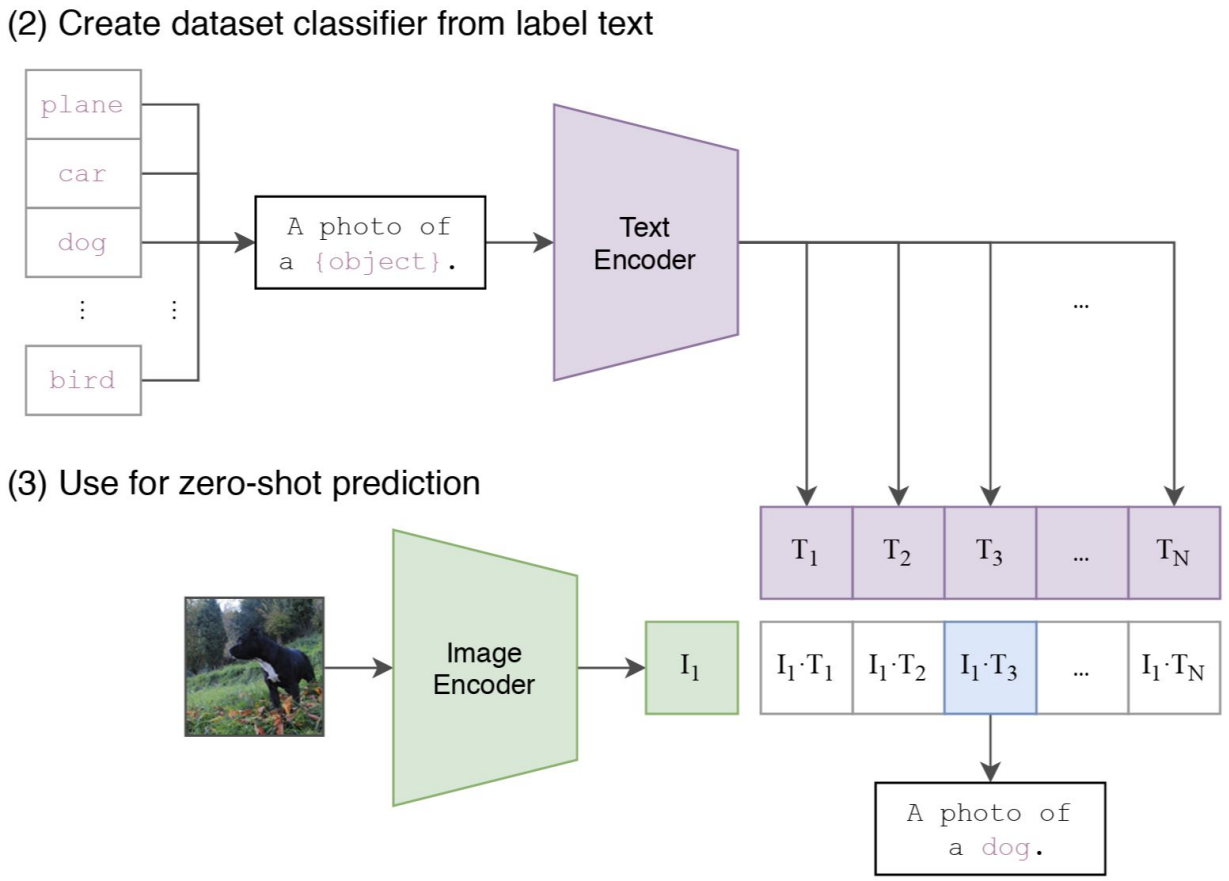
Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

How this matches the diagonal slide

- Encode images and texts separately.
- Project both into the same multimodal embedding space.
- L2-normalize so dot product becomes cosine similarity.
- Build the full N×N similarity matrix.
- Apply cross-entropy in both directions: image→text and text→image.

Figure 3 makes the loss concrete: the "diagonal" is just the target labels arange(n).

Zero-shot prediction: the classifier is generated by text



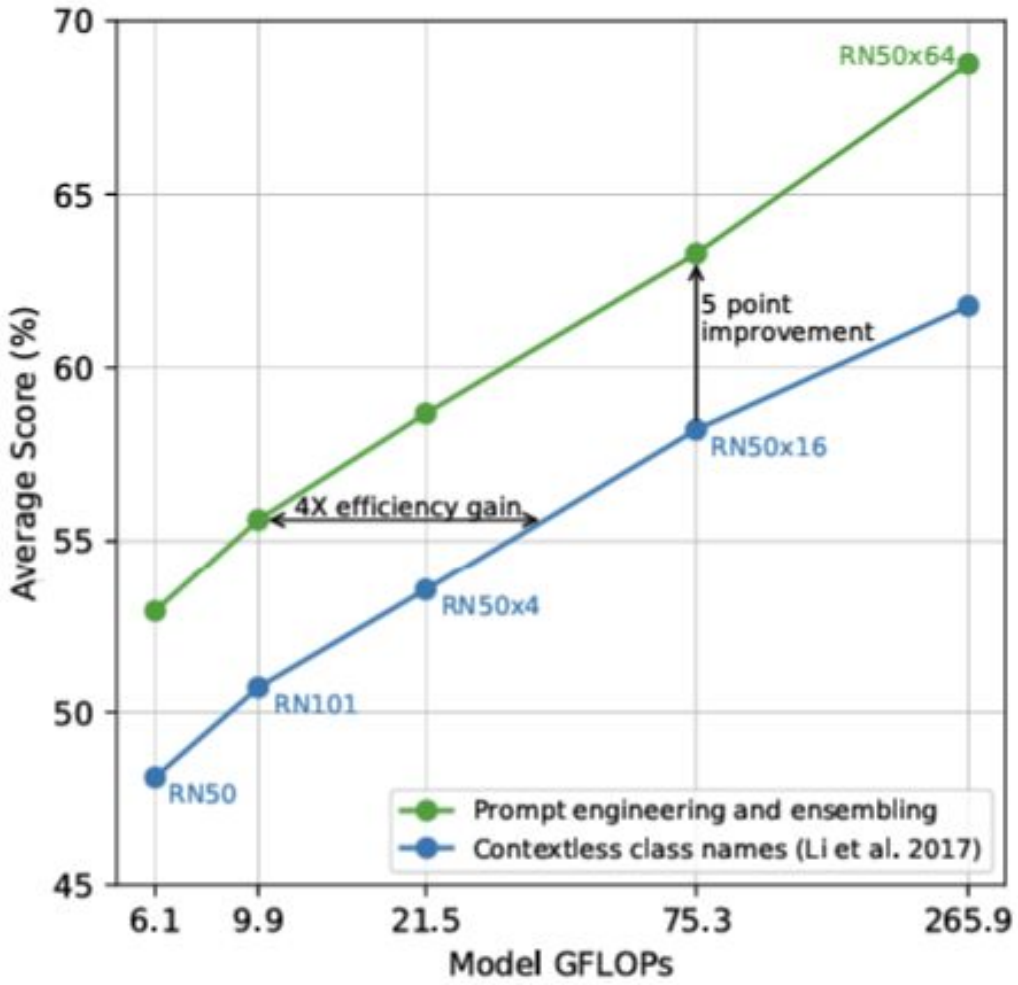
- Write candidate labels as prompts: “a photo of a {object}”
- Encode all labels with the text encoder
- Encode the test image with the image encoder
- Pick the text with highest similarity

This is why CLIP is not just a classifier: the output vocabulary can be edited at inference time.

Prompt wording matters:
"a photo of a dog" works better than just "dog".

Radford et al. 2021, zero-shot classifier construction.

Prompt engineering shows the text side matters



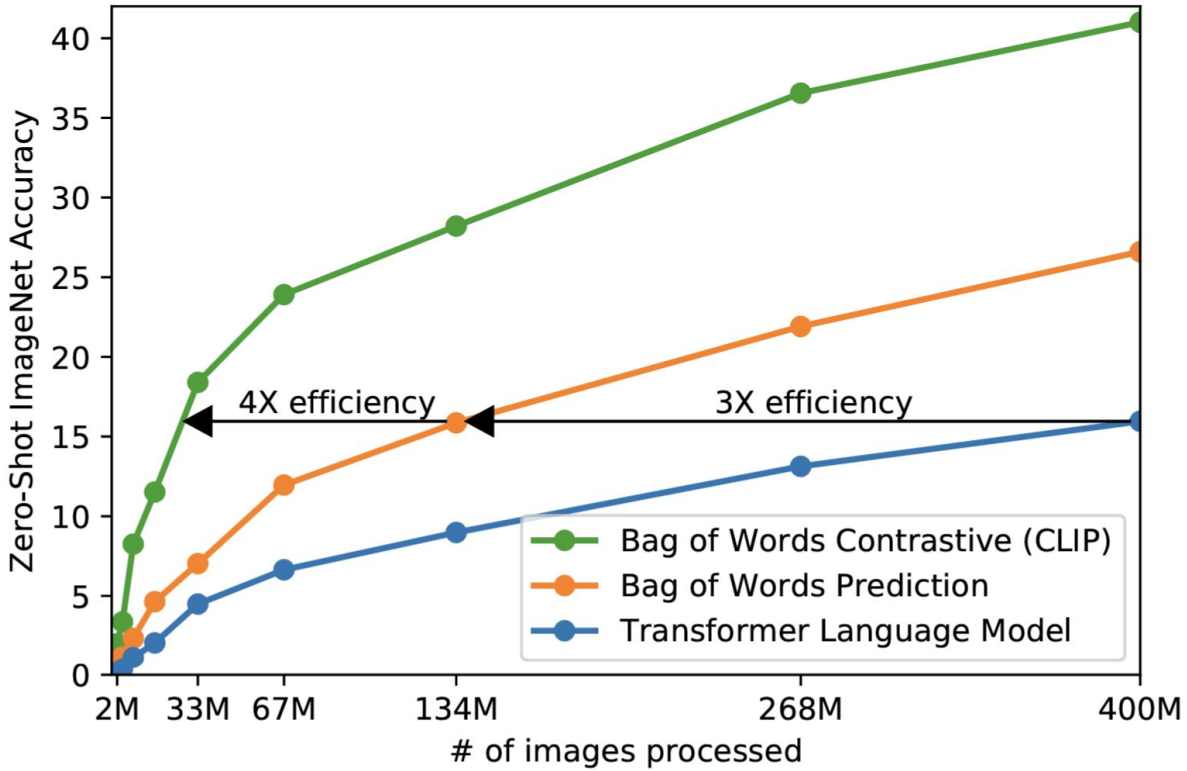
Why this is not a footnote

- Class names are often ambiguous: “crane” can mean a bird or a machine.
- Templates add semantic context, e.g. “a photo of a {label}”.
- Prompt ensembling averages multiple text embeddings into a better zero-shot classifier.
- The paper reports an average gain of almost 5 points across 36 datasets.

Takeaway: zero-shot classification is generated by language, so language formulation directly changes the classifier.

Radford et al. 2021, Fig. 4: prompt engineering and ensembling improve zero-shot performance.

CLIP result: contrastive matching is much more efficient



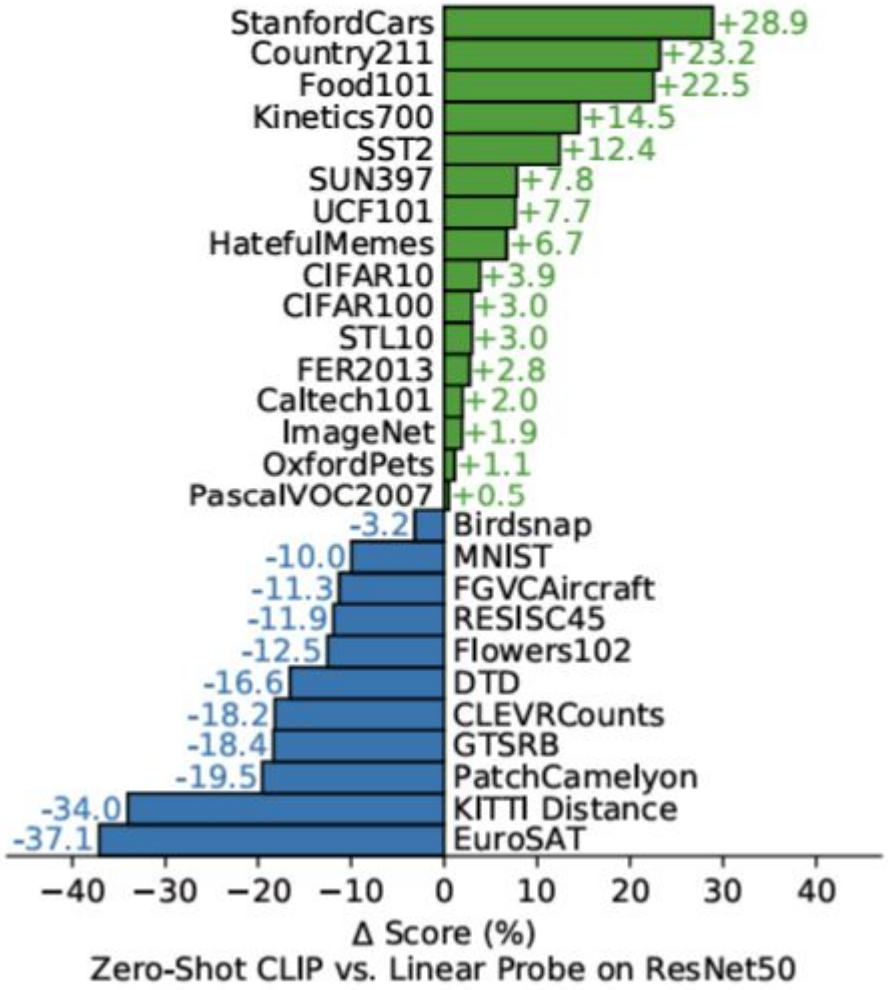
- Caption language modeling is expressive but slow for zero-shot transfer
- Bag-of-words prediction learns roughly 3× faster than a caption LM
- Switching to CLIP contrastive objective gives another roughly 4× gain over BoW

Headline result from the paper
Best CLIP reaches 76.2% zero-shot on ImageNet and transfers across 30+ datasets.

Interpretation: the two efficiency gains compound — BoW improves over caption LM, and contrastive matching improves again over BoW.

Radford et al. 2021, Fig. 2 caption: BoW is ~3× faster than caption LM; CLIP adds another ~4× over BoW.

Zero-shot CLIP transfers broadly — but unevenly



How to read Figure 5

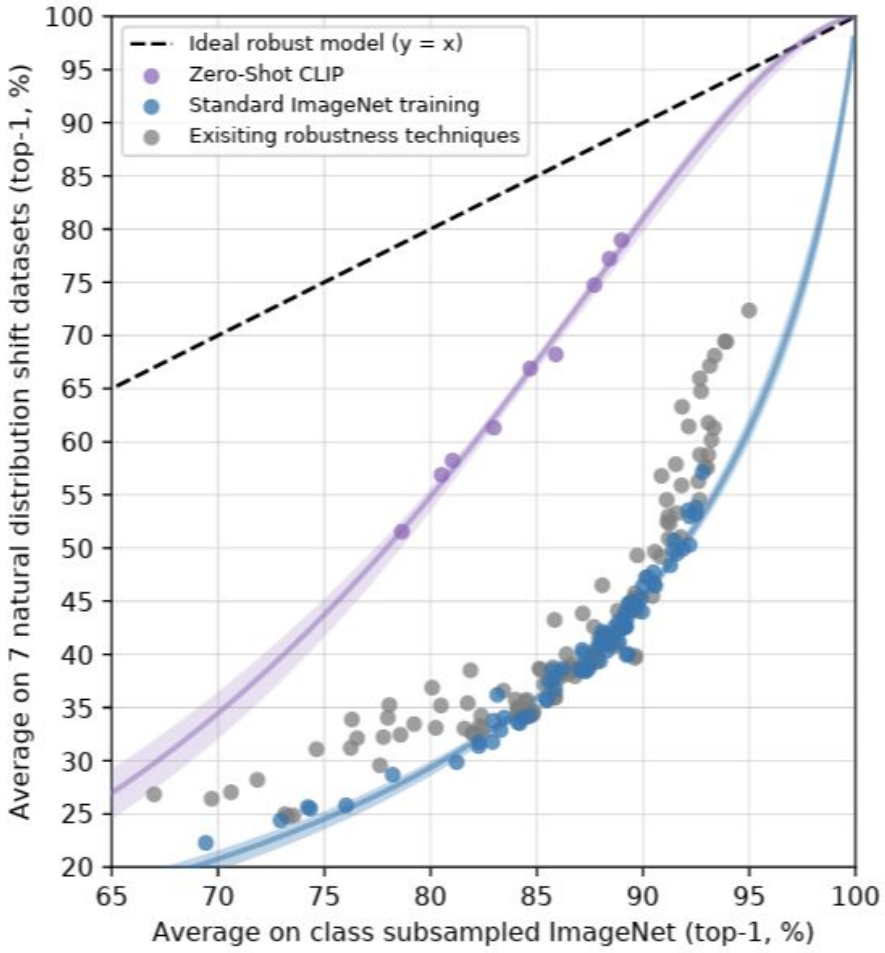
- CLIP beats a supervised RN50 linear probe on 16 of 27 datasets.
- It is very strong on some fine-grained or web-like recognition tasks.
- It is weak on specialized or abstract tasks: satellite scenes, medical images, counting, and traffic signs.
- This is a better framing than saying CLIP simply “solves” zero-shot vision.

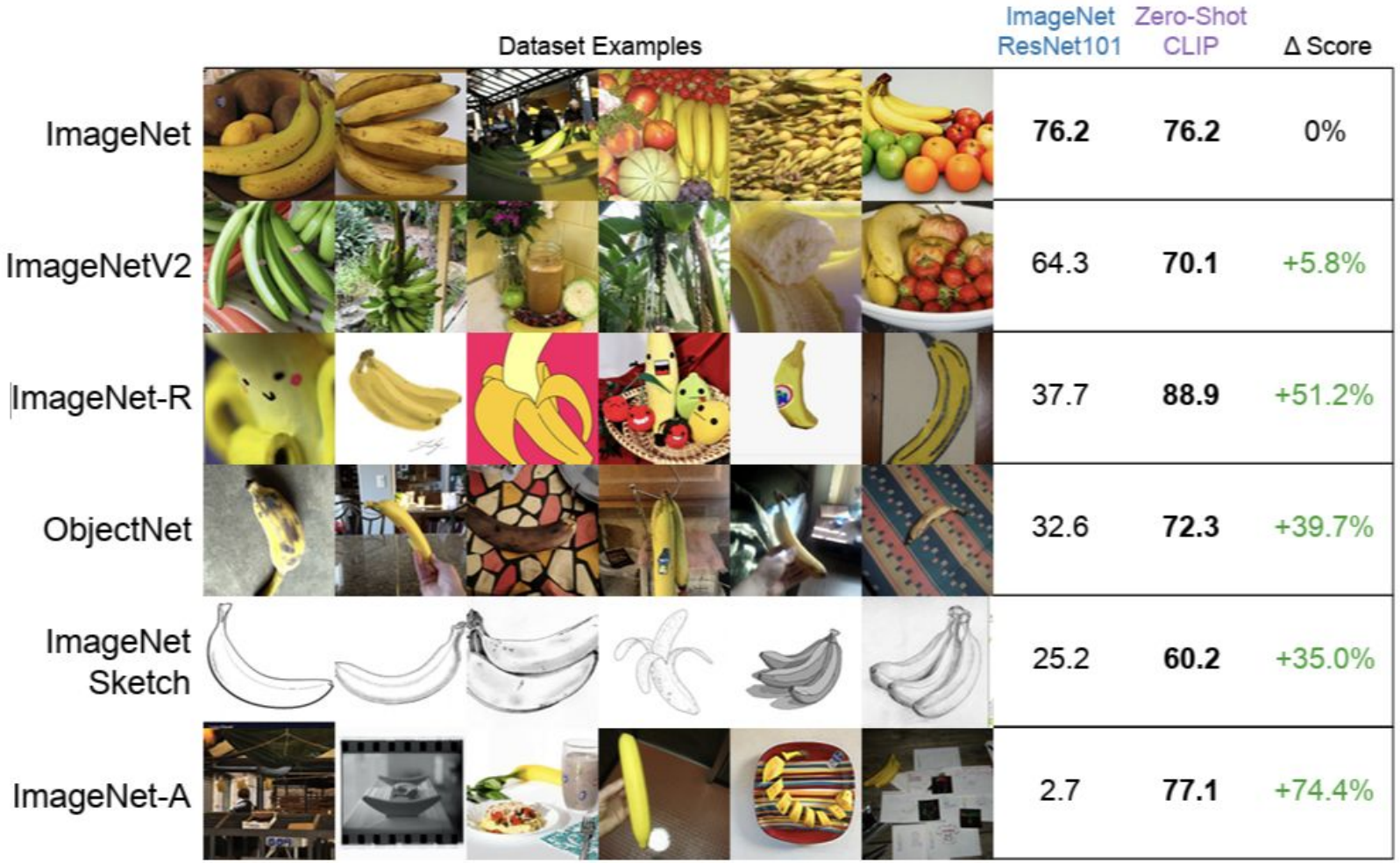
Strength and limitation appear in the same plot.

Radford et al. 2021, Fig. 5: zero-shot CLIP vs. supervised linear probe on ResNet-50 features.

Zero-shot CLIP is less tied to the ImageNet distribution

Original CLIP Fig. 13 shows a robustness gap: matched ImageNet accuracy does not imply matched distribution-shift performance.





Two kinds of CLIP limitations

Directly visible in CLIP results

- prompt sensitivity
- uneven transfer across datasets
- weak specialized/abstract tasks
- web-supervision noise and bias

More visible from later VLM needs

- short captions miss fine detail
- 77-token text context bottleneck
- image-level embeddings are coarse
- dense grounding needs local features

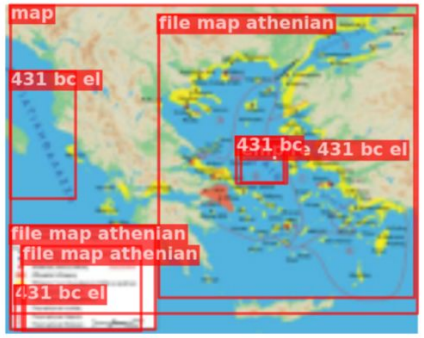
This distinction is useful: CLIP is not “wrong”; it is optimized for global image-text alignment, while later VLMs need richer local and textual supervision.

Radford et al. 2021; later open-vocabulary detection and dense prediction literature.

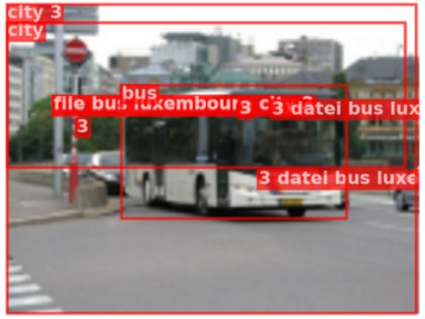
Open vocabulary moves from classification toward grounding

file 2005 citroën c3 interior 8th july 2015...

αρχε o map athenian empire 431 bc el svg



file bus luxembourg city 3 jpg

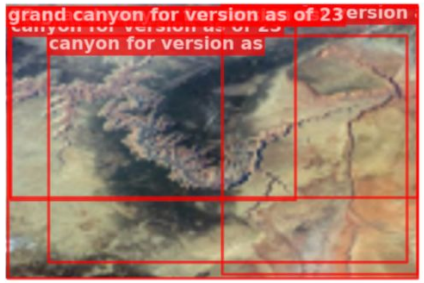


file patrick henry bruce landscape google art...

file fresco showing a silver tray containing...

file monumento a josé maria dos santos pinhal...

file iss 39 grand canyon jpg

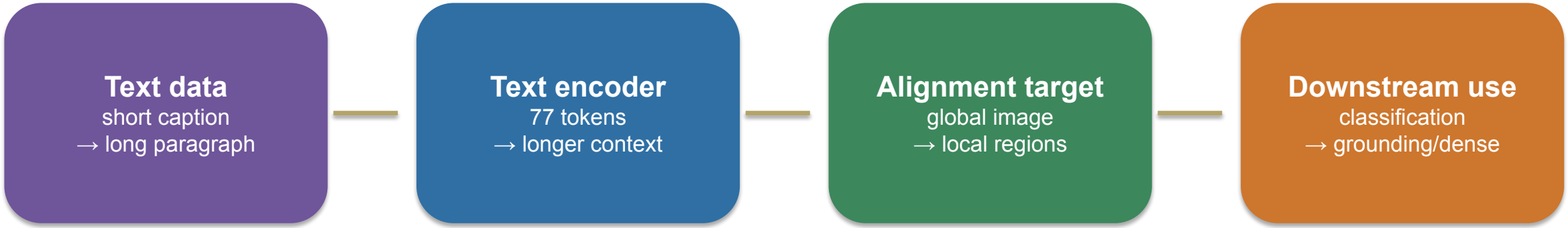


file kalisz bazylika wniebowzi cia nmp...

The same text interface can guide boxes and local regions, but this requires local visual features rather than only image-level embeddings.

A useful taxonomy of CLIP variants

Think of variants as modifying one of three bottlenecks in the CLIP pipeline.



These variants keep CLIP's image-text interface, but change the granularity of the supervision and representation.

Selected CLIP variants used here: DreamLIP, Long-CLIP, and DenseCLIP.

DreamLIP: replace short captions with long descriptions

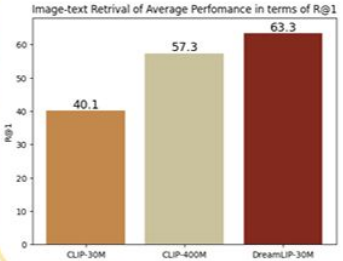
MLLM

Generated Long Caption

Long Caption:

- The image captures a serene bedroom scene.
- Dominating the center of the room is a bed, neatly made with a white comforter and a gray blanket.
- The bed is flanked by two nightstands, each hosting a lamp, casting a warm glow in the room.
- The room itself is characterized by a wooden floor, adding a touch of rustic charm to the space.
- A large window punctuates one wall, draped with white curtains that filter the outside light.
- The walls are painted in a soothing shade of gray, providing a neutral backdrop that enhances the room's tranquil ambiance.
- Above the bed, a large abstract painting hangs, its vibrant colors contrasting with the room's muted tones.
- The bed, being the largest piece of furniture, serves as the focal point of the room.
- The nightstands, lamps, and painting are arranged around it, creating a balanced composition.

Image-Text Retrieval



CLIP-30M:

~~Kite surfer in the air on top of a red board.~~

A large orange and white kite flying in a blue sky.

~~A large kite with three orange, white and black fish.~~

~~A picture of a very large kite flying in the air.~~

CLIP-400M:

A large white and red parasail in the blue sky.

A red and white parachute is in the sky.

~~A red, 163.jpgflying among the sky.~~

~~Several para-gliders can be seen in the blue sky.~~

DreamLIP-30M:

A large orange and white kite flying in a blue sky.

A red and white parachute is in the sky.

A large white and red parasail in the blue sky.

~~A large kite with three orange, white and black fish.~~

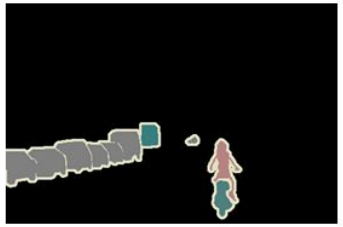
Core idea: if the text supervision is richer, contrastive pretraining can learn more than the most salient object label.

Semantic Segmentation

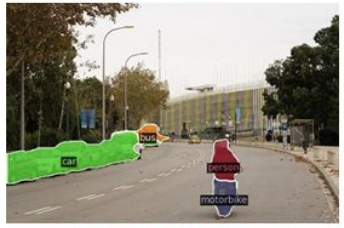
Image

CLIP-30M

CLIP-400M



GT



DreamLIP-30M

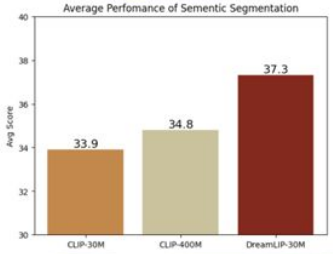
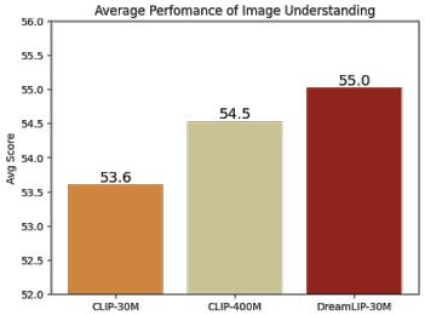


Image Understanding in MLLM



Is the Christmas tree positioned in front of a large window or in front of a wall between two windows?

CLIP-30M: ☹

(a) In front of a large window. ✓ (b) In front of a large window. ×

CLIP-400M: ☹

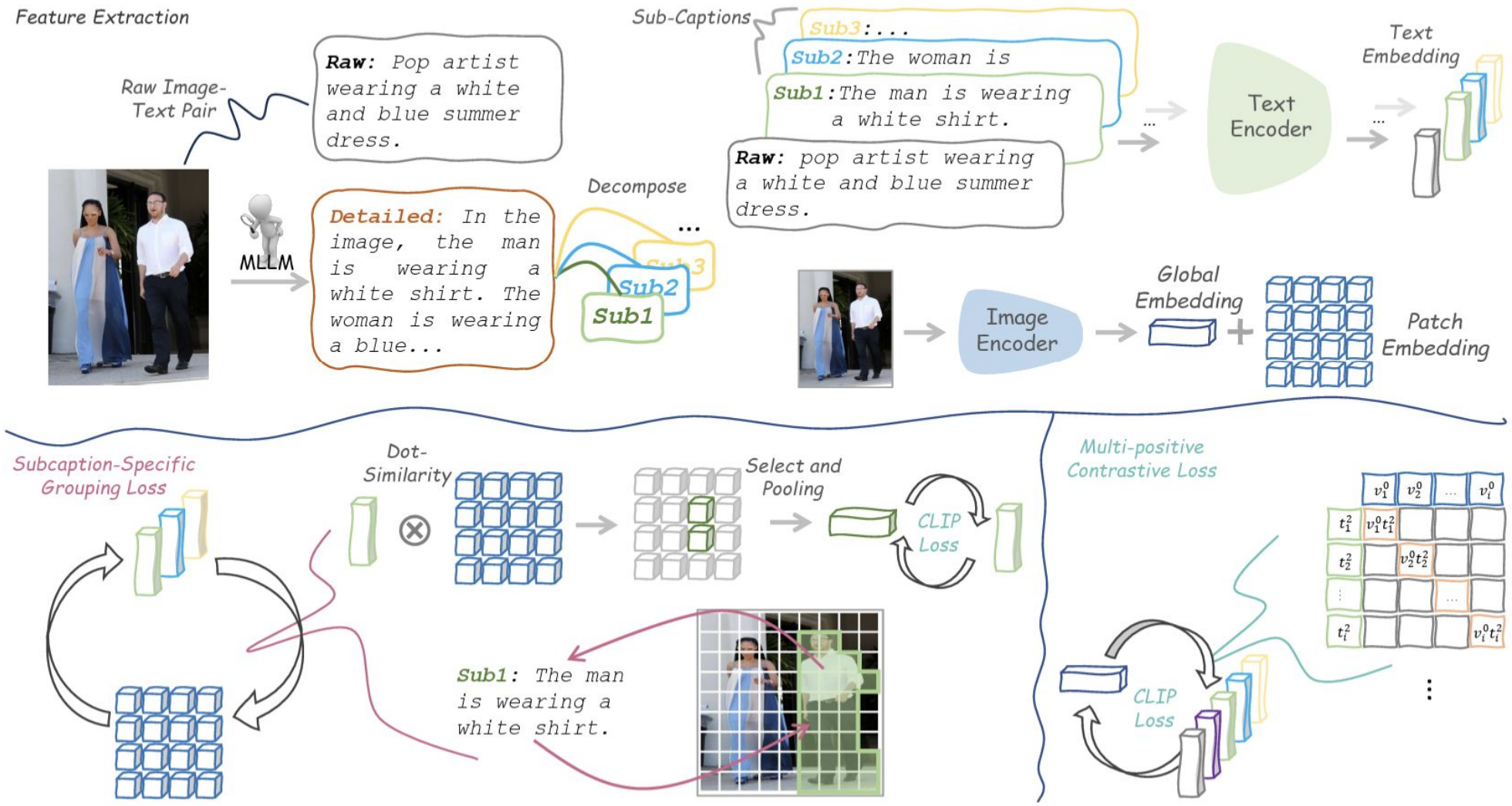
(a) In front of a large window. ✓ (b) In front of a large window. ×

DreamLIP-30M: ☺

(a) In front of a large window. ✓ (b) In front of a wall between two windows. ✓

DreamLIP: sub-captions become multiple positives

A long caption is not used as one monolithic string; its sentences often describe different regions or objects.

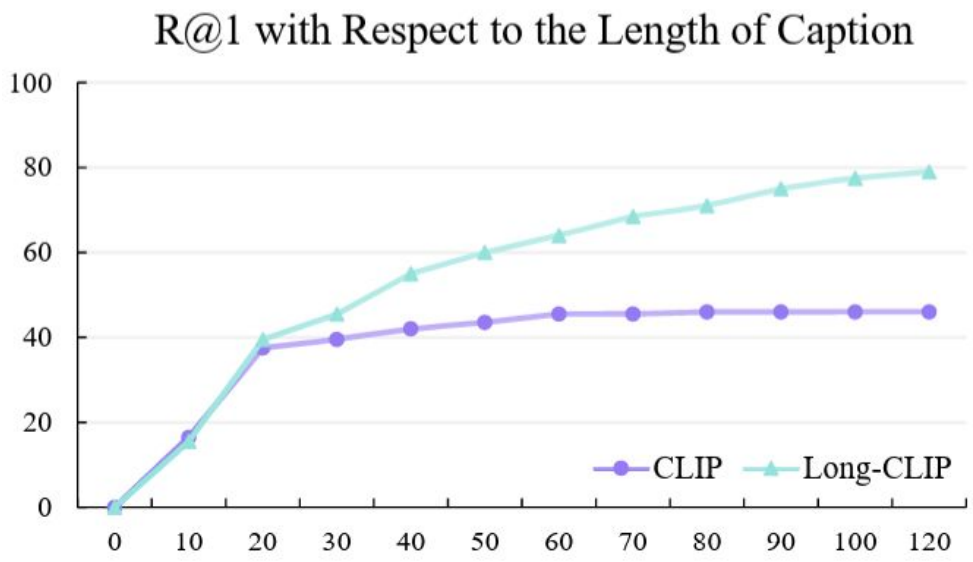


- Global contrastive loss still aligns whole images with whole descriptions.
- Sub-caption-specific grouping adds pressure for local image patches to match local textual details.
- This directly targets the gap between “image-level CLIP” and fine-grained grounding.

Zheng et al., 2024, DreamLIP, Fig. 2.

Long-CLIP: CLIP has limited effective context and weak relation modeling

Longer captions only help if the text encoder can actually read and represent them.



a) Low effective token length

A. The lemon on the **left** is **yellow** and the eggplant on the **right** is **purple**

B. The lemon on the **left** is **purple** and the eggplant on the **right** is **yellow**

C. The lemon on the **right** is **yellow** and the eggplant on the **left** is **purple**

D. The lemon on the **right** is **purple** and the eggplant on the **left** is **yellow**

CLIP: D > C > B > A Long-CLIP: A > C > B > D

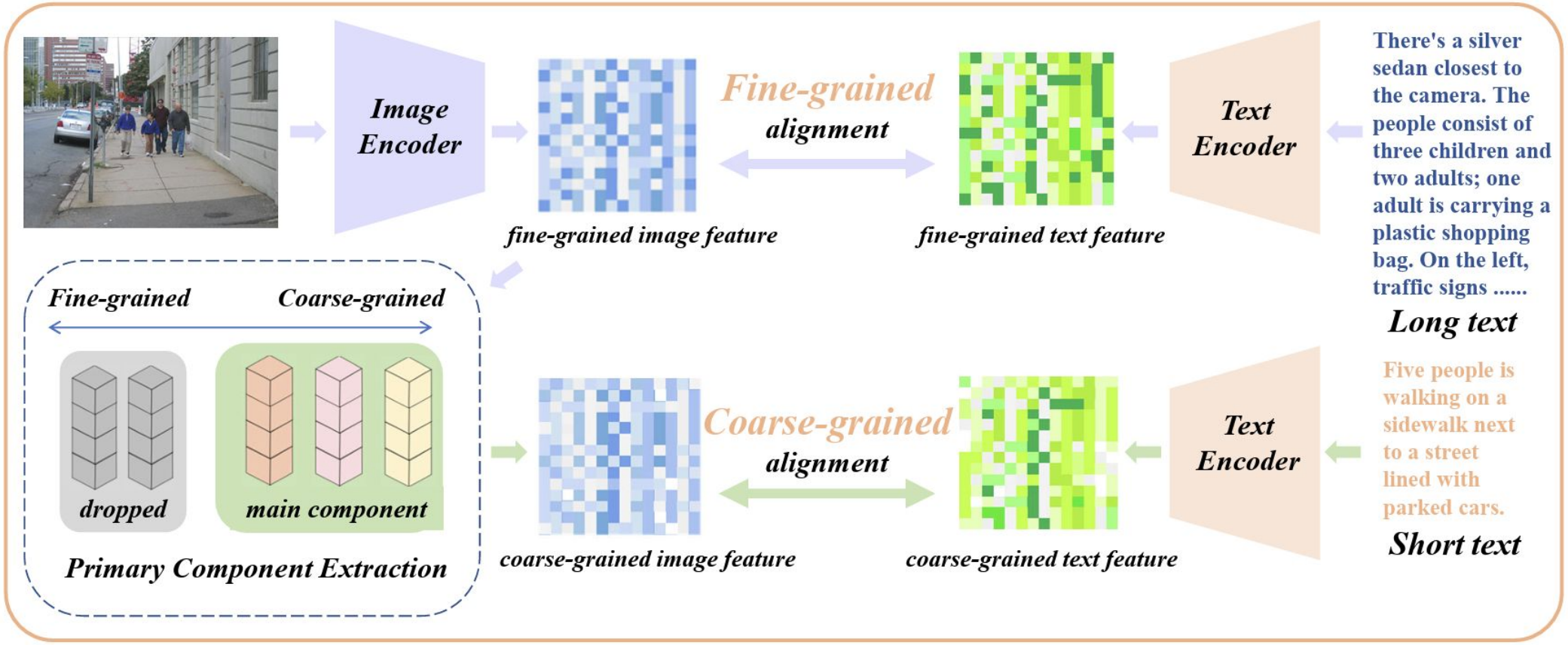
b) Fail in modeling interrelationship

Long-CLIP keeps the CLIP-style embedding interface, but extends the text side so detailed descriptions can influence the image-text similarity.

Zhang et al., 2024, Long-CLIP, Fig. 2.

Long-CLIP: preserve short-text ability while adding long-text ability

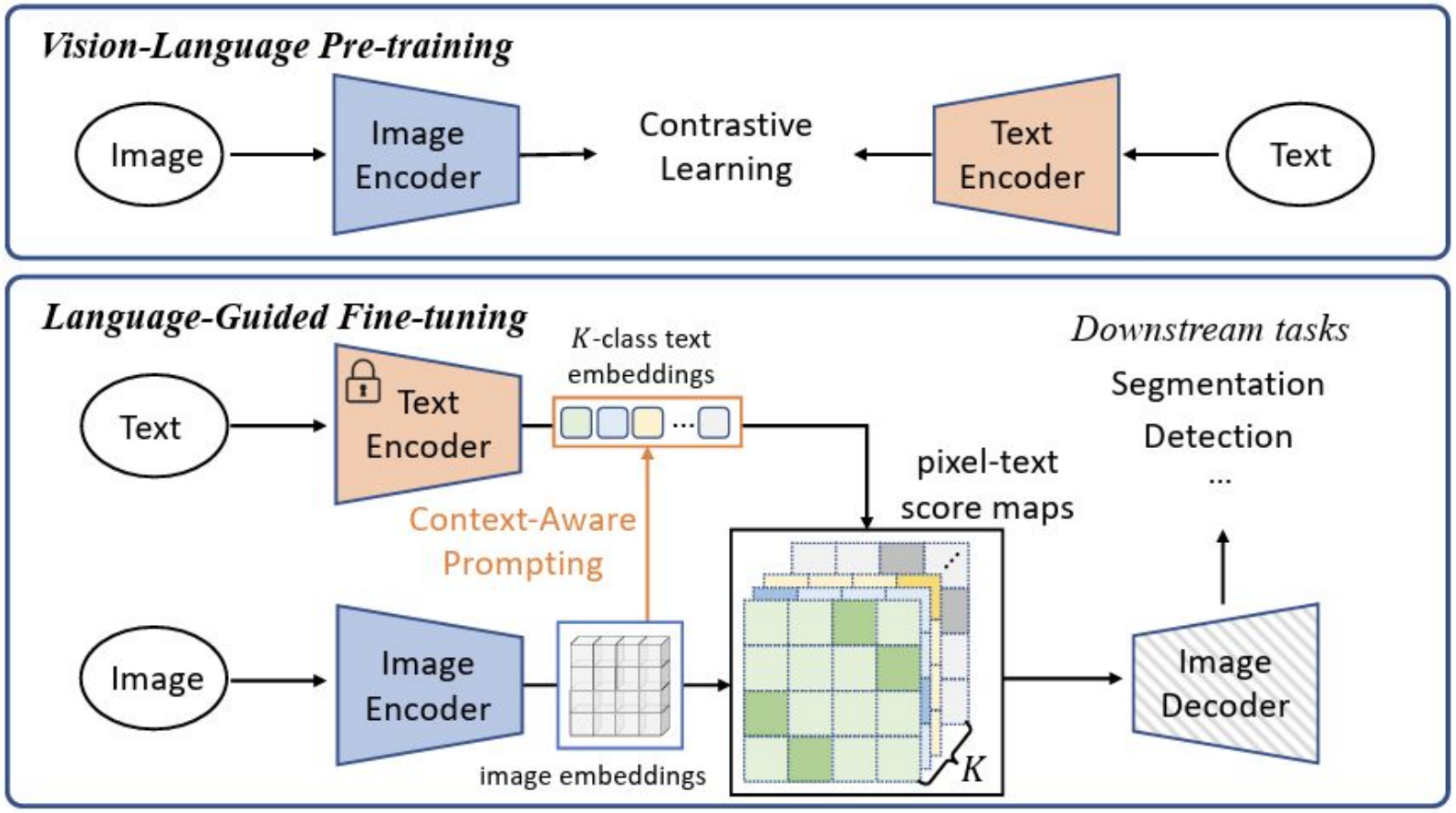
A naive longer text encoder can damage CLIP's existing short-text zero-shot behavior.



Takeaway: this variant mainly improves the text encoder side of CLIP, not the basic dual-encoder alignment paradigm.

DenseCLIP: turn image-text matching into pixel-text matching

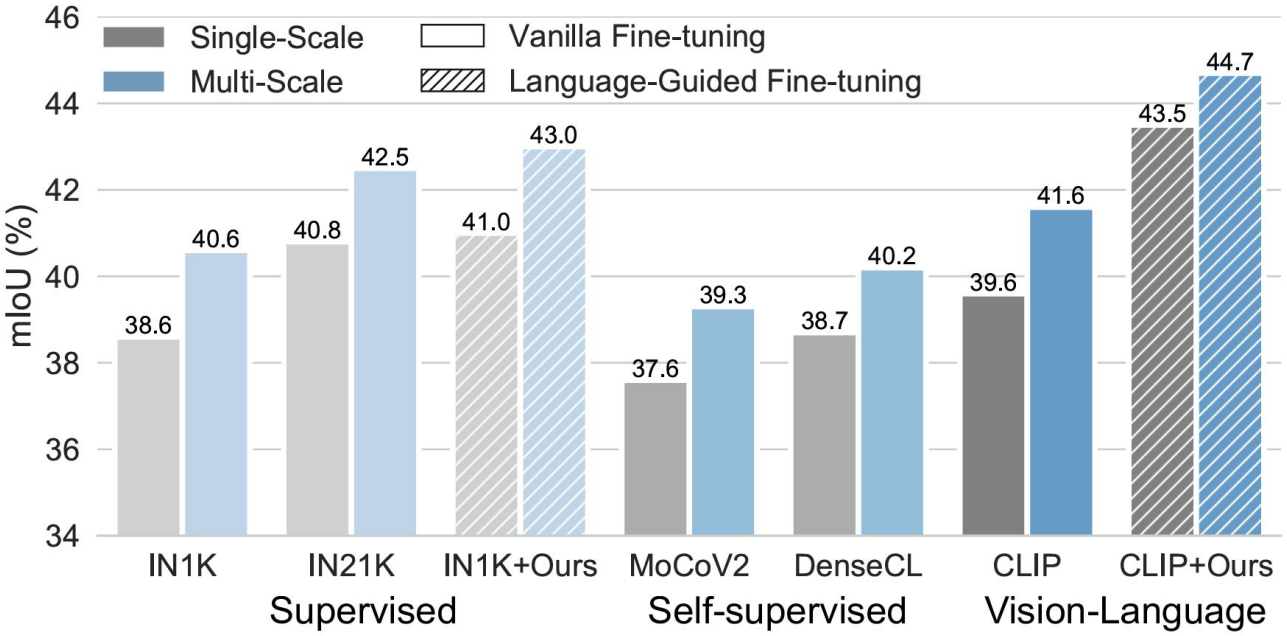
Dense prediction needs a score at each location, not one score for the whole image.



Key bridge: CLIP's language-compatible features can guide semantic segmentation, detection, and instance segmentation.

Rao et al., 2022, DenseCLIP, Fig. 1(b).

DenseCLIP: useful adaptation, but not the same as pretraining dense features



This is the clean bridge to SigLIP 2: keep image-text alignment, but train patch-level and localization behavior more explicitly.

What it adds

- DenseCLIP exploits pretrained CLIP knowledge for dense downstream tasks.
- It narrows the gap between image-level contrastive pretraining and pixel-level prediction.
- It uses language-guided score maps and context-aware prompting.

What remains

- But original CLIP was still not trained primarily for dense patch-level objectives.
- Dense performance depends heavily on the downstream dense prediction framework.
- This motivates newer encoders that train local/dense representations more directly.

Rao et al., 2022, DenseCLIP, Fig. 2.

CLIP-side takeaway: same interface, richer granularity

The post-CLIP direction is not to abandon contrastive alignment, but to make it less coarse.

Data granularity
short captions
→ paragraph descriptions

Text capacity
77-token prompts
→ long-context text encoding

Visual granularity
image-level embedding
→ patch / region semantics

Tasks
classify
→ ground

So the high-level lesson is: CLIP changed the interface of vision models from fixed labels to language. Later work asks whether that language interface can support detailed descriptions, long-context retrieval, and local grounding.

Bridge: from CLIP softmax to SigLIP sigmoid

SigLIP keeps CLIP's dual-encoder image-text interface, but changes the contrastive loss.

CLIP softmax contrastive loss

- Each image must identify its correct text among all texts in the batch.
- Positive pairs compete against many in-batch negatives.
- Requires softmax normalization over the similarity matrix.
- Larger batches provide more in-batch negatives under the softmax objective.

SigLIP sigmoid loss

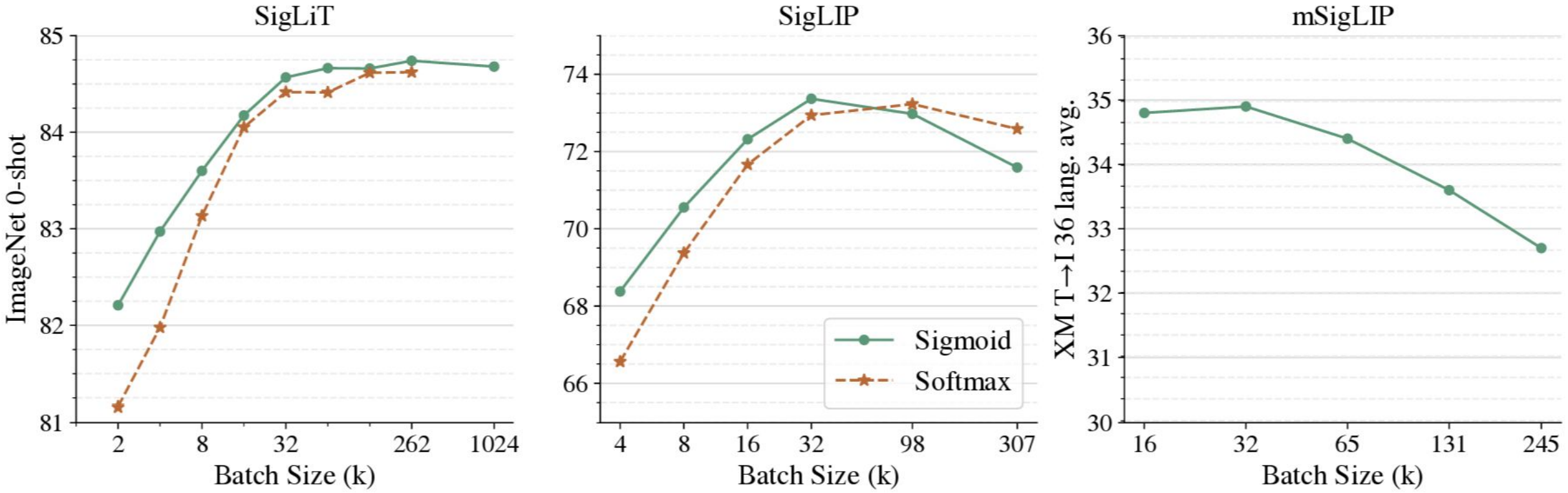
- Each image-text pair is treated as an independent binary classification problem.
- Pairs are labeled positive or negative.
- No global softmax normalization is required.
- The loss is less tied to a global batch view.

This is why the SigLIP idea is a natural bridge: it preserves CLIP-style alignment, but changes the competition structure.

Zhai et al. 2023, Sigmoid Loss for Language Image Pre-Training; Radford et al. 2021.

SigLIP insight: bigger batches are not the whole story

CLIP motivates large batches; SigLIP asks whether the softmax formulation is the bottleneck.



Why this matters for SigLIP 2

- Sigmoid loss is especially competitive at smaller practical batches.
- Both losses show diminishing returns as batch size grows.
- The paper reports that a reasonable batch size around 32K is sufficient.
- This motivates SigLIP 2's starting point: inherit the sigmoid alignment loss, then add local/dense objectives.

Zhai et al. 2023, Fig. 2 and Abstract: sigmoid loss performs well at practical batch sizes; ~32K is sufficient.

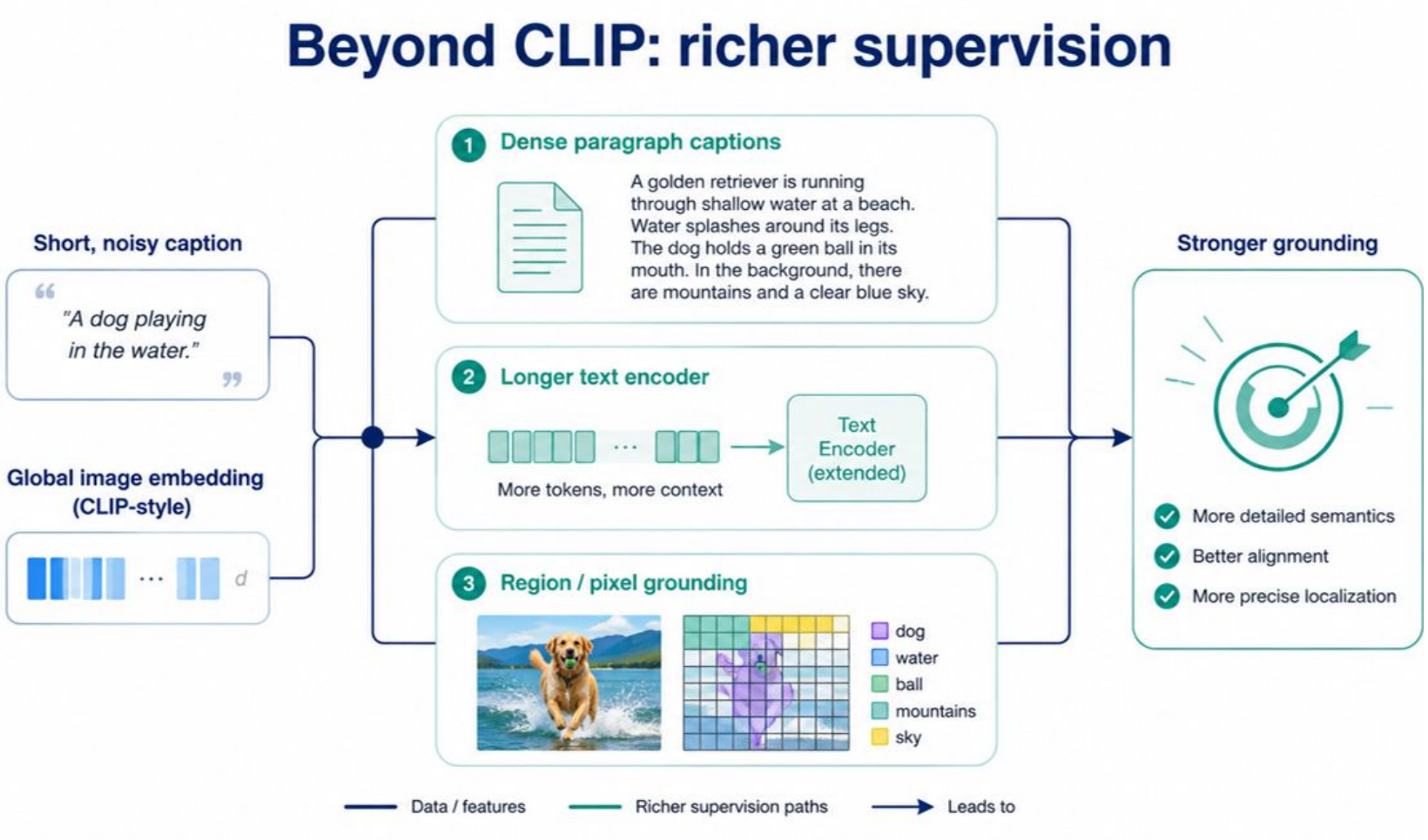
CLIP Variants: Beyond Clip

Gap in current story

- CLIP uses short/noisy captions
- Global embeddings weakly locate objects
- Text encoder may miss long descriptions

What this mini-section adds

- Longer captions: richer supervision
- Longer context: better text encoder
- Local matching: better grounding



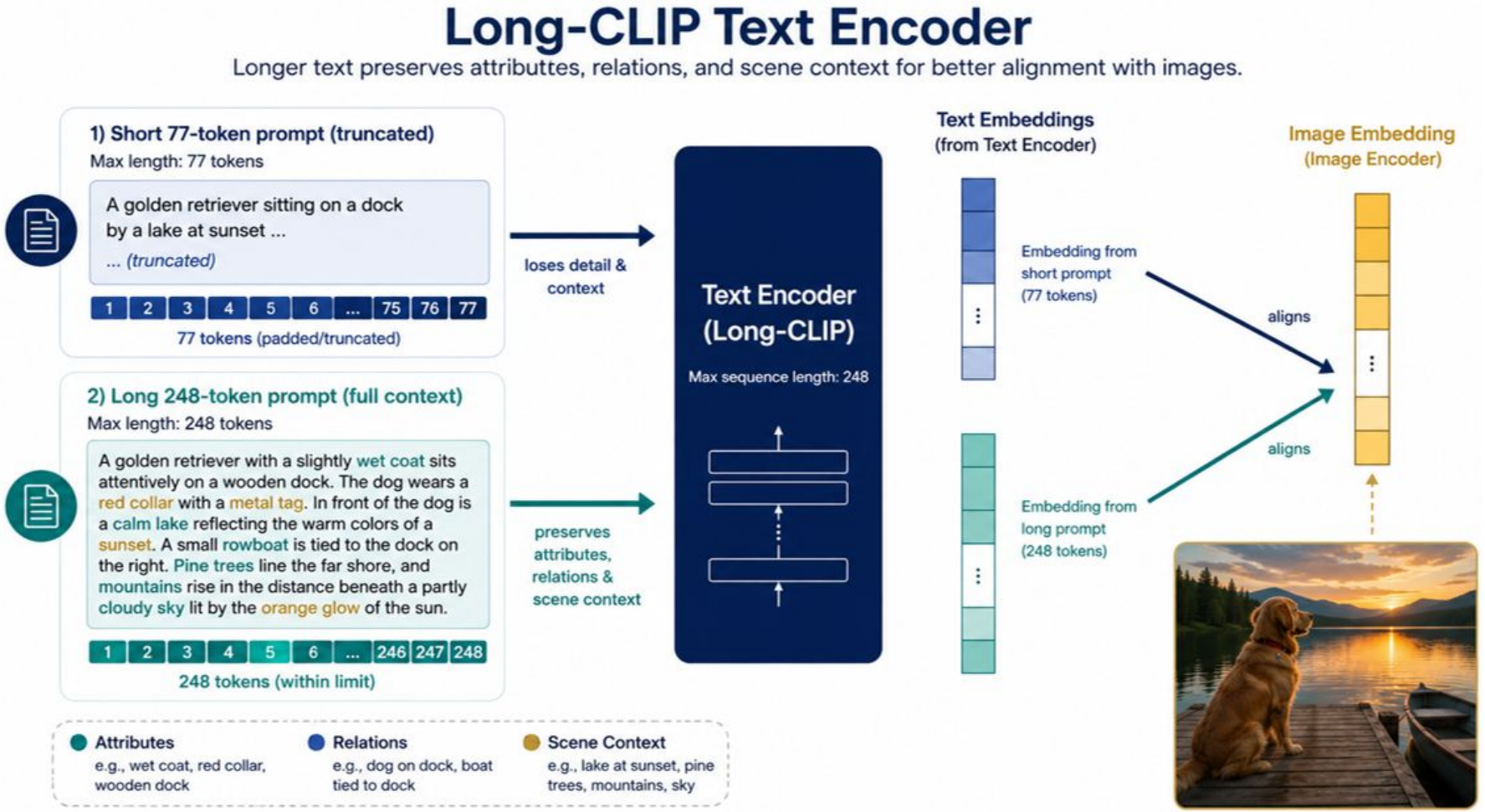
CLIP Variants: Long Clip

CLIP bottleneck

- Default CLIP text input is limited to 77 tokens
- Effective useful length can be much shorter

Long-CLIP upgrade

- Extends text input to 248 tokens
- Preserves CLIP latent space and zero-shot ability
- Fine-tunes with long text–image pairs



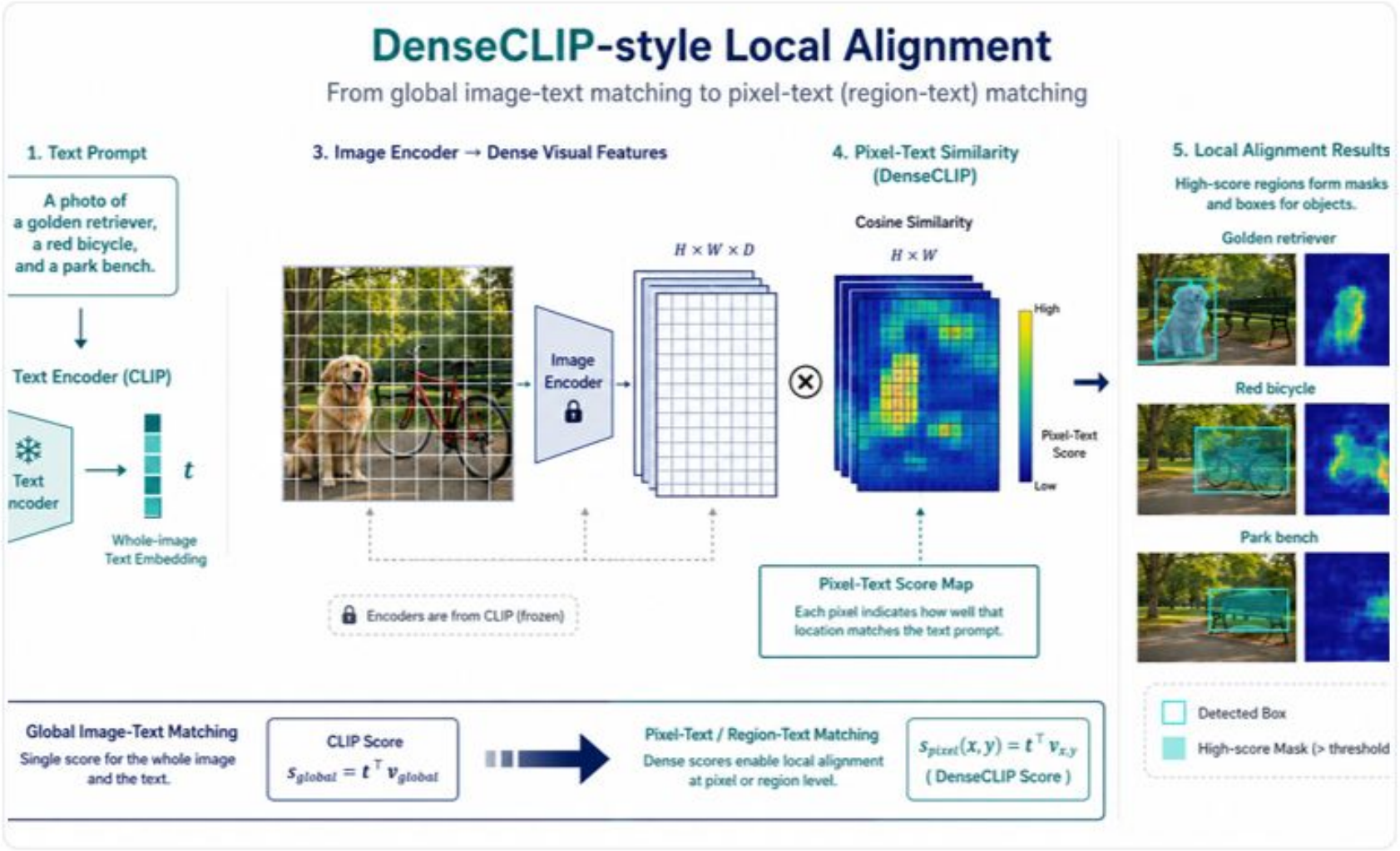
CLIP Variants: Dense Clip

Vanilla CLIP

- Global image vector ↔ global text vector
- Good for retrieval/classification, weak for dense prediction

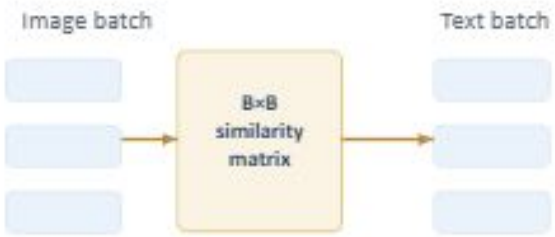
DenseCLIP idea

- Convert image-text matching into pixel-text matching
- Use pixel-text score maps to guide segmentation / detection
- Prompting can be context-aware



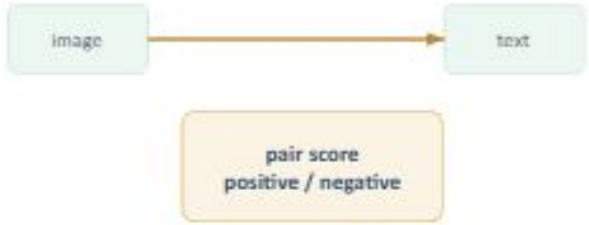
CLIP -> SigLIP -> SigLIP2

CLIP: batch-level softmax



Global normalization couples every pair in the batch.

SigLIP: pairwise sigmoid



No global softmax: each image–text pair can be optimized more independently.

Batch-size insight

Sigmoid is stronger at smaller batches; returns diminish beyond ~32K.

Takeaway: SigLIP removes global softmax normalization → less dependence on huge batches + simpler distributed training.

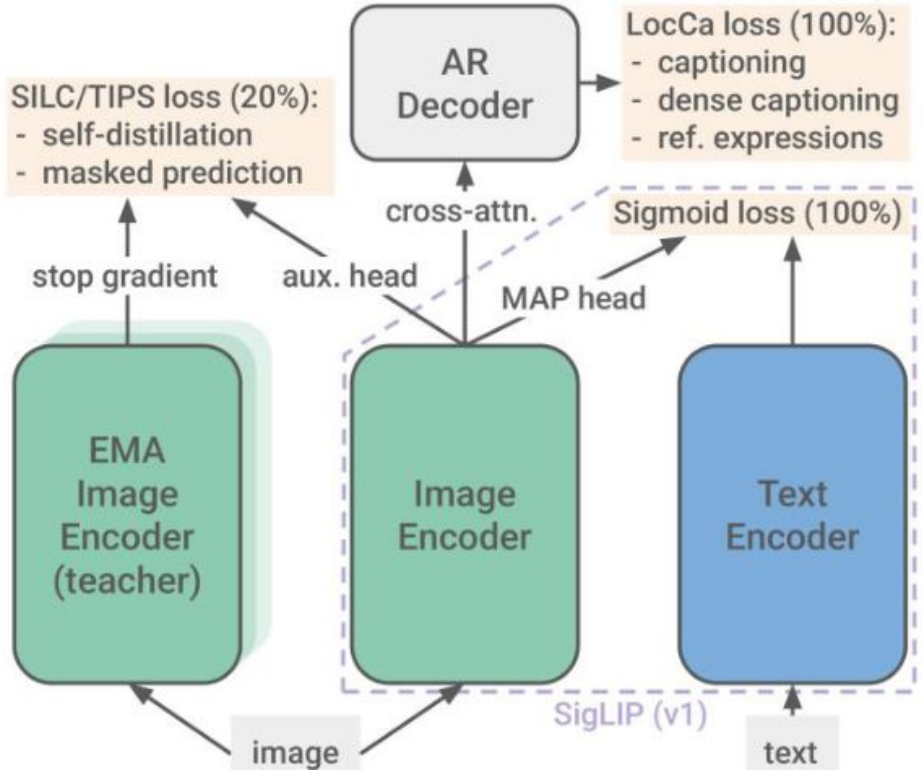
Transition to SigLIP 2

SigLIP improves how global image–text alignment is trained. SigLIP 2 keeps this foundation, then makes the visual representation richer with:

LocCa · self-distillation · masked prediction · multilingual WebLI · NaFlex

Tschannen et al. 2025, Fig. 1 and Abstract.

SigLIP 2: CLIP-style alignment, modernized for VLMs



1. Global image–text alignment

- Inherits the **sigmoid contrastive loss** from SigLIP
- Keeps the dual-encoder image–text representation space

2. Local & dense visual learning

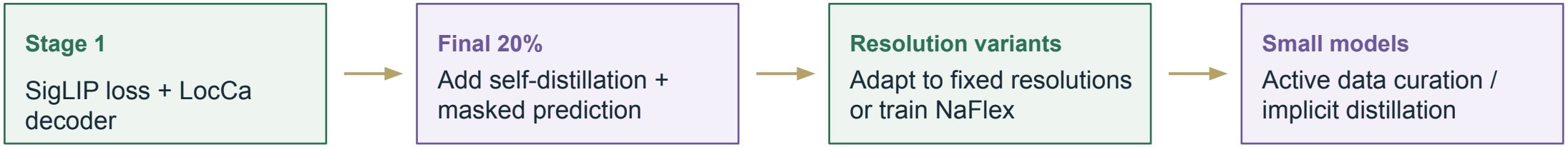
- Adds **LocCa captioning and localization objectives**
- Adds **self-distillation + masked prediction** late in training

3. Better data & input handling

- Multilingual training data
- **NaFlex** for variable image aspect ratios
- Active data curation for smaller models

Tschannen et al. 2025, Fig. 1 and Abstract.

SigLIP 2 is a staged training recipe



Stage 1 — Global + local alignment (0–80%)

- Train with **SigLIP loss + LocCa decoder**
- Learn image–text alignment and localization-aware patch features

Stage 2 — Strengthen dense features (final 20%)

- Add **self-distillation**
- Add **masked prediction**
- Improve unpooled patch representations

Variant-specific training

- **NaFlex:** trained separately without the two image-only losses
- **B/16 & B/32:** additional active data curation / implicit distillation

Ingredient: Data for training

Tschannen et al. 2025, Sec. 2.2; SigLIP loss originally from Zhai et al. 2023.

Ingredient inherited from SigLIP: sigmoid loss

CLIP softmax contrastive loss

- Each image must identify its **correct text among all texts in the batch**
- Positive pairs **compete against batch negatives**
- Requires normalization across the similarity matrix

Sigmoid image-text loss

- Treats **every image-text pair independently**
- Each pair is classified as **positive or negative**
- No softmax normalization across the full batch

Takeaway: the inherited sigmoid loss changes the competition structure; SigLIP 2's novelty is the combined training recipe.

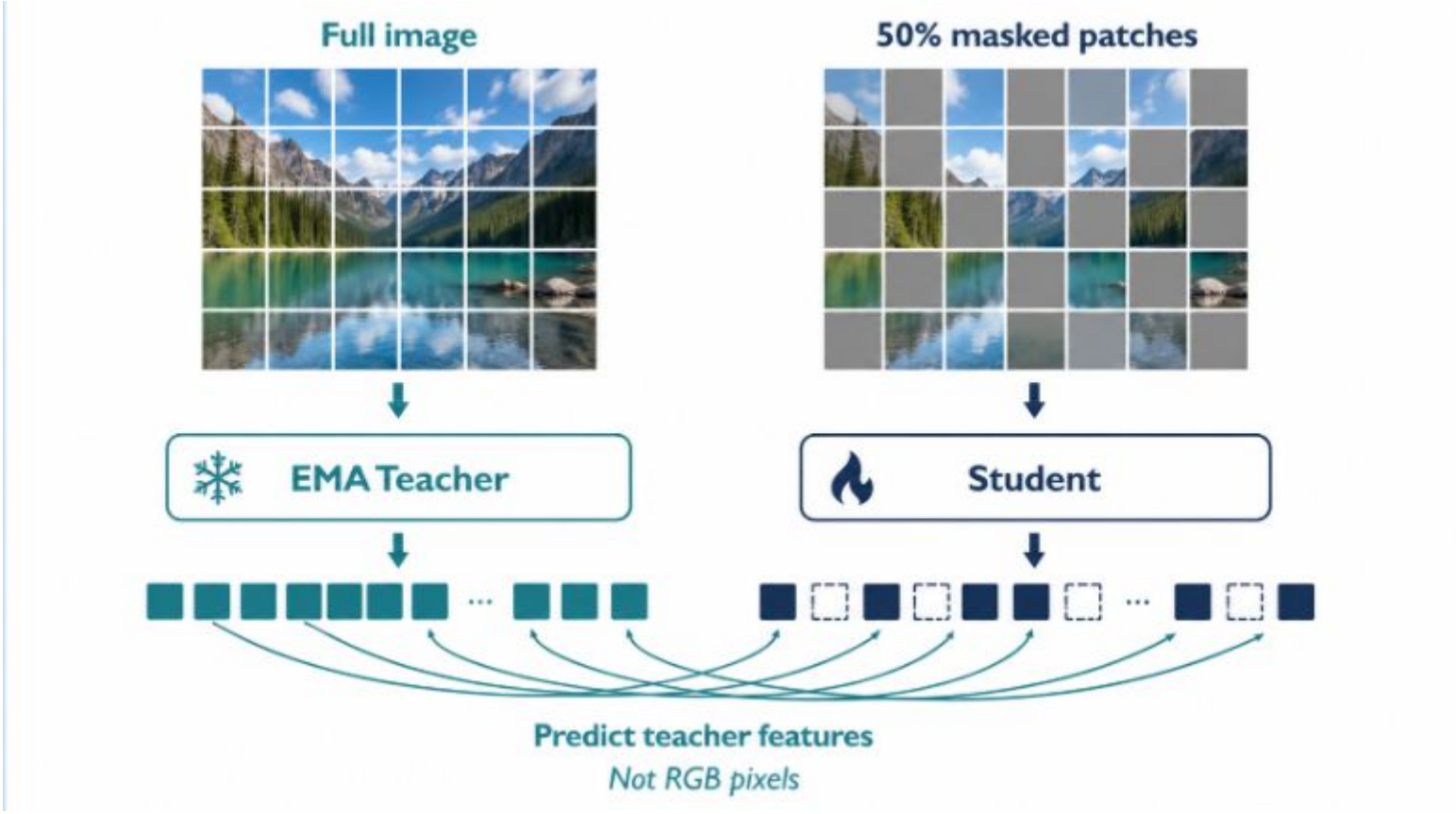
Tschannen et al. 2025, Sec. 2.2; SigLIP loss originally from Zhai et al. 2023.

Ingredient: Masked Prediction

Predict the teacher representation of masked patches.

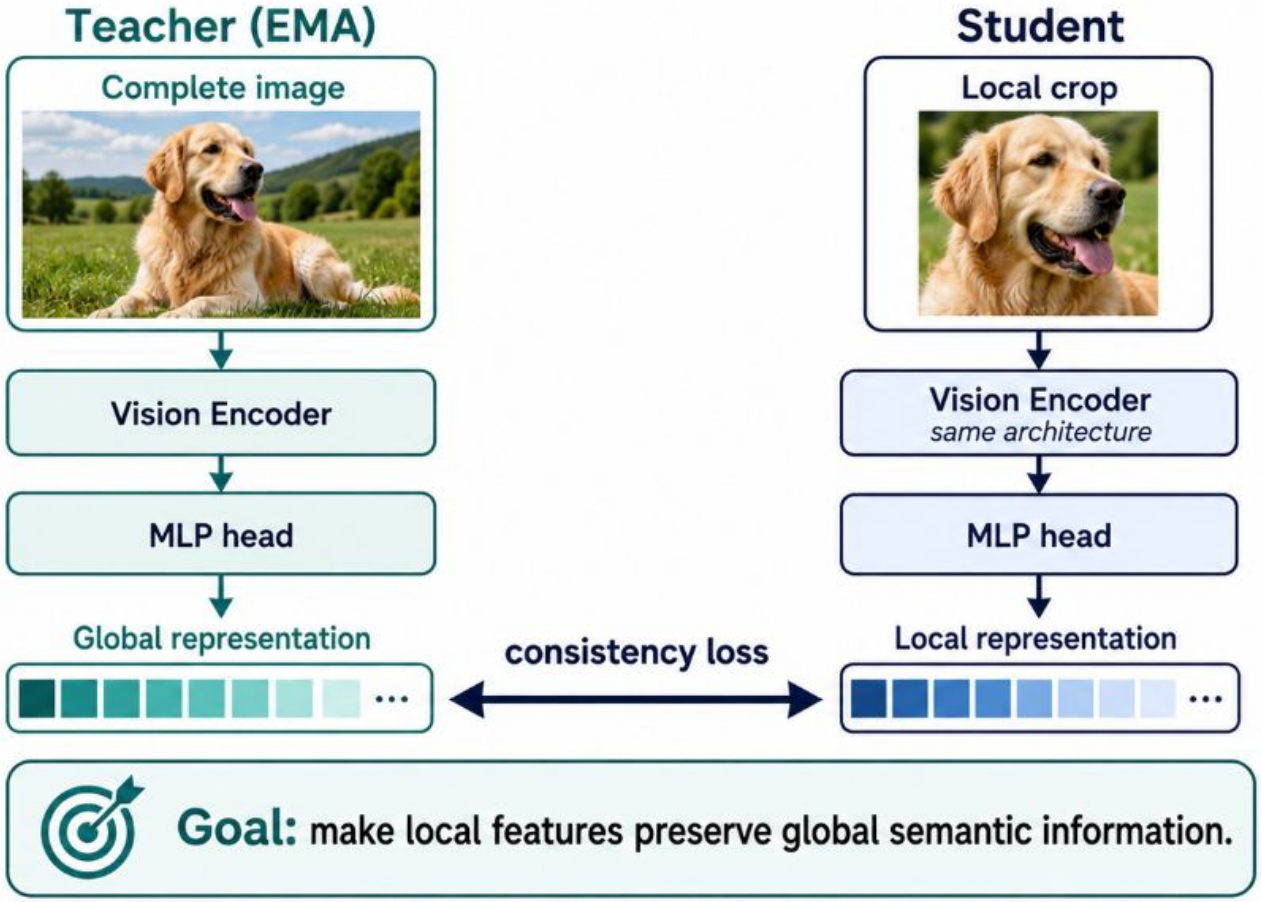
- Mask **50% of student image patches** during training
- Teacher sees the **complete global image**
- Student predicts teacher features at **masked locations**
- Loss is applied to **un-pooled patch features**, not the pooled image embedding

- Forces patches to preserve semantic context
- Improves local features used by dense tasks
- Connects masked-token learning to segmentation, depth, and localization



Ingredient: Self-Distillation

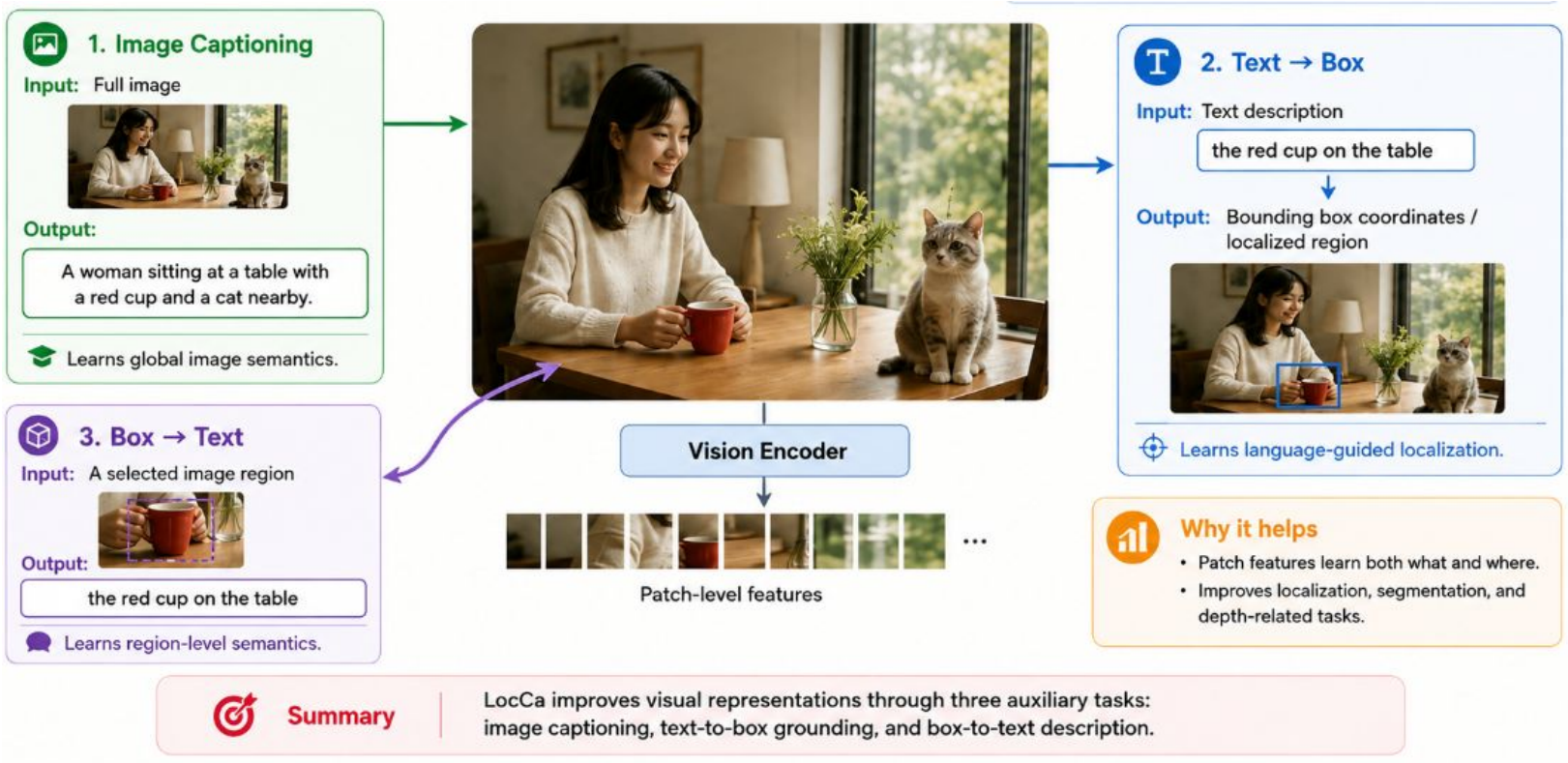
Model	architecture	patch_size
B/16	ViT-Base	16 × 16
B/32	ViT-Base	32 × 32
L/16	ViT-large	16 × 16
SO/14	Larger SigLIP 2 vision encoder variant	14 × 14



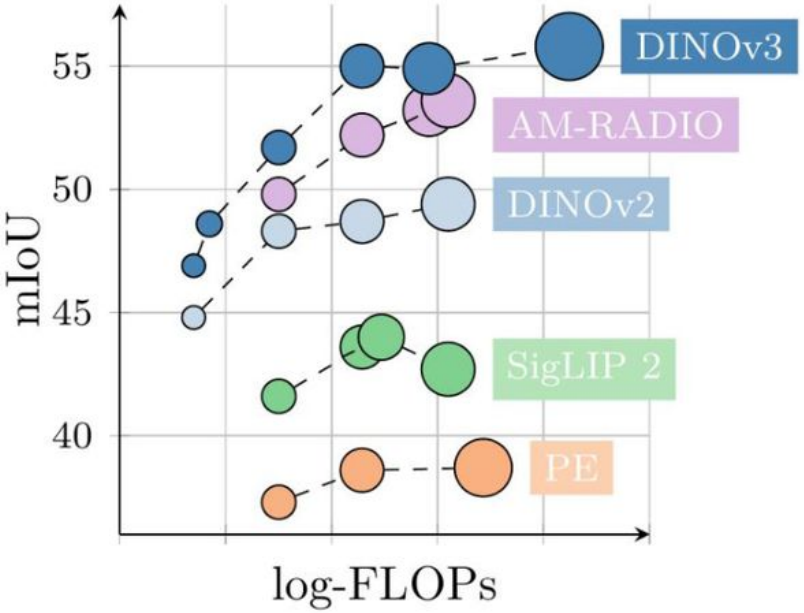
Ingredient: LocCa decoder adds localization pressure

Decoder tasks used for representation learning

- Image captioning
- Automatic referring expression prediction: text description → box coordinates
- Grounded captioning: box coordinates → region-specific caption
- The decoder is training-time scaffolding, not part of the released encoder



Ingredient: Language alignment alone is not enough for dense vision



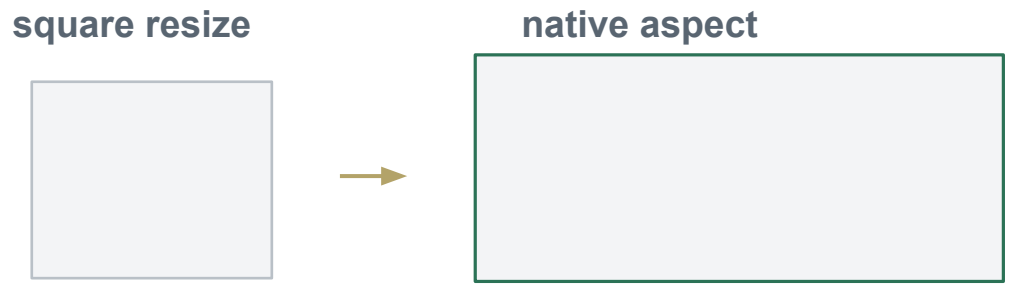
- This figure shows that dense performance is not determined by compute alone. Image-only objectives such as self-distillation and masked prediction provide direct supervision to patch-level representations, which helps close the gap between global image-text alignment and dense visual understanding.

Tschannen et al. 2025, Sec. 2.3; dense-feature context from course slides.

Ingredient: NaFlex keeps images closer to their native layout

Why fixed square resizing is a problem

- Forces different images into the same aspect ratio
- Can distort **text, charts, screenshots, and documents**
- Layout-sensitive visual information may be lost



NaFlex

1. Preserves the **native aspect ratio** as much as possible
2. Supports **variable visual sequence lengths**
3. Masks padding tokens when batching different image shapes
4. Uses **one checkpoint** across multiple image layouts

Square resizing hurts most for layout-sensitive inputs: screenshots, scanned documents, charts, and OCR-heavy pages.

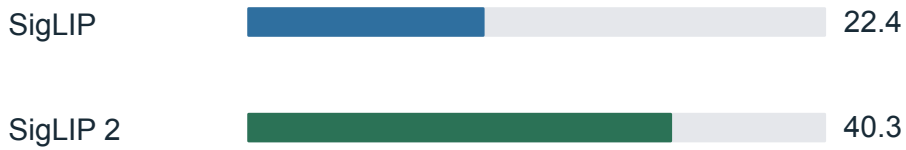
Tschannen et al. 2025, Sec. 2.4.2 and Fig. 3 caption.

Results: fair selected comparisons from Table 1

ImageNet val, B/16 @ 224px

image classify task

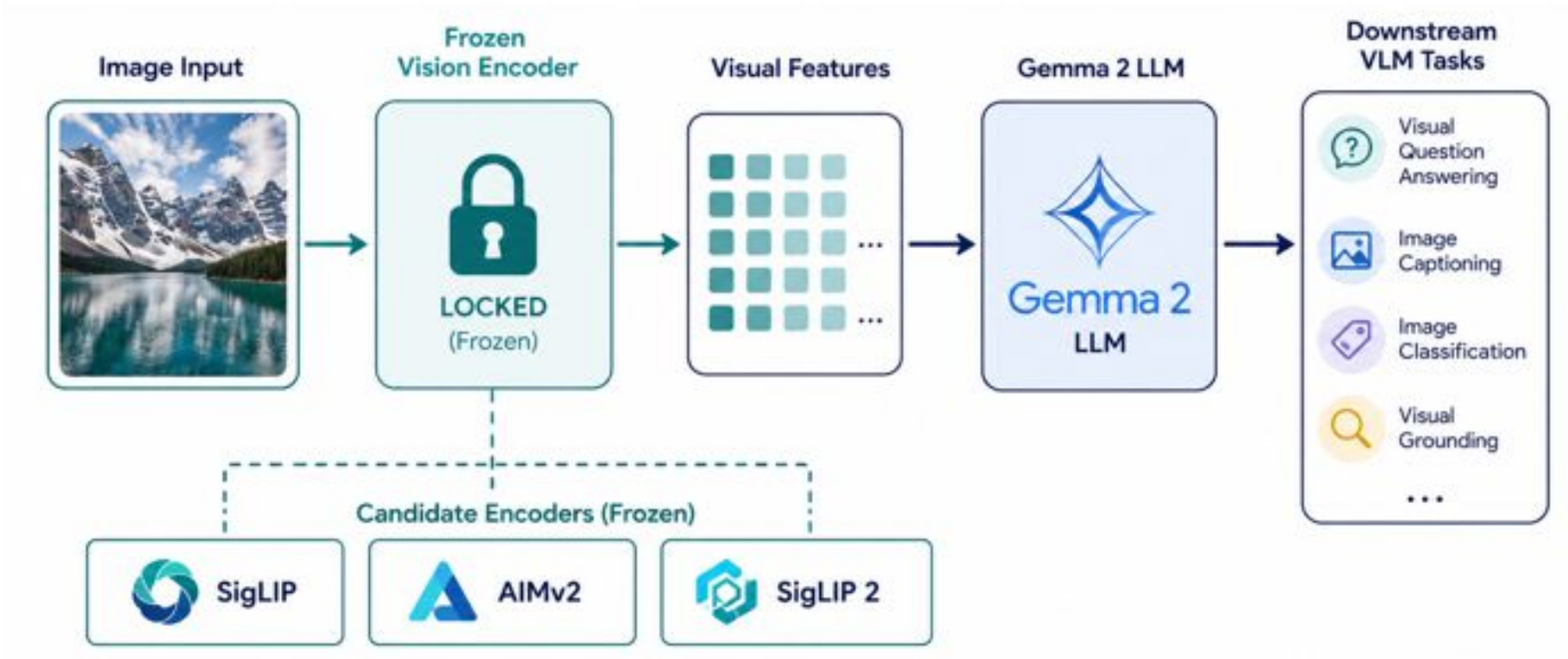
XM3600 text→image R@1, B/16 @ 224px



Multilingual image-text retrieval task

Apples-to-apples: both charts use B/16 @ 224px; mSigLIP is intentionally excluded because it is reported at a different model size.

Results : SigLIP 2 is evaluated as a frozen vision encoder in a Gemma 2-based VLM.



Tschannen et al. 2025, Sec. 2.4.2 and Fig. 3 caption.

Results: SigLIP 2 as a frozen VLM vision encoder

- Evaluation recipe: frozen vision encoder + Gemma 2 LLM, then VLM fine-tuning
- The comparison is against SigLIP and AIMv2 under the same VLM recipe
- The reported result: SigLIP 2 performs better across model sizes and resolutions
- Interpretation: the encoder is useful not just for retrieval, but as a VLM front end

Figure 4 takeaway

schematic summary from the paper's VLM-transfer experiment

Takeaway: the representation gains survive when the encoder is frozen inside a downstream VLM pipeline.

Results: the biggest gains are local and dense

Dense probing examples

- So/14, 224px: PASCAL mIoU 72.0 → 77.1
- ADE20k mIoU 37.6 → 41.8
- Depth RMSE improves 0.576 → 0.493

Localization examples

- RefCOCO val, L/16 seq 256: 67.33 → 86.04
- RefCOCO+ val: 59.57 → 77.29
- RefCOCOg val-u: 61.89 → 80.11

This is where SigLIP 2 most clearly addresses CLIP's weak point: text-to-region grounding and patch-level semantics.

Tschannen et al. 2025, Tables 2 and 4/5; arrows denote SigLIP → SigLIP 2.

Limitations and implications

Limitations

- CLIP remains vulnerable to prompt sensitivity and dataset bias
- SigLIP 2 bundles many ingredients, so attribution is harder
- Better benchmark results do not fully prove robust open-world understanding
- Dense features still depend on downstream probes and evaluation design

Implications

- The vision encoder controls what a VLM can see
- Local grounding matters for boxes, masks, GUI clicks, and embodied action
- Frontier encoders are likely trained with mixed global/local/multilingual objectives
- Evaluation should test both semantic alignment and grounded use

Discussion questions

1. If language supervision is so powerful, why do SigLIP 2 and DINO-style methods still need image-only dense losses?

Discussion questions

2 Is zero-shot classification a good proxy for visual understanding, or mostly a retrieval-style test?

Discussion questions

3 For future VLMs, should the vision encoder optimize global alignment, local grounding, or downstream agent success?

Discussion questions

4 When a frontier paper bundles many training improvements together, how should we evaluate what really caused the gain?