Week 9

Foundation Models

[ECEA0649/ECE40049] Deep Learning for Image Processing | Spring 2026

Kihyun Na (Research Professor)
BK21 AI Education and Research Group &
Institute for Information and Communication Technology,
Handong Global University

HANDONG GLOBAL UNIVERSITY

Curriculum

** Adjusted based on survey results*

Wk 1	OT + Introduction	**Wk 9**	Foundation Models (CLIP, SAM) + Paper #1 (Today)
Wk 2	DL Fundamentals Review	**Wk 10**	Diffusion Models + Paper #2
Wk 3	CNN	**Wk 11**	Conditional Generation + Paper #3
Wk 4	From Sequence Modeling to Transformer	**Wk 12**	Vision-Language Models + Paper #4
Wk 5	Transformer in Vision	**Wk 13**	VLM Applications + Paper #5
Wk 6	Detection & Segmentation	**Wk 14**	Video Understanding + Paper #6
Wk 7	Self-Supervised Learning	**Wk 15**	Embodied AI & Robot Vision + Paper #7
Wk 8	Review Literacy + Role Explanation	**Wk 16**	Miniconference (Final Project)

Lecture Lecture + Paper Miniconference

Today's Agenda

01 **What is a Foundation Model?** — 10 min

Definition, conditions, and why they emerged now

02 **CLIP Deep Dive** — 20 min

From SimCLR to image-text contrastive learning at scale

03 **SAM Deep Dive** — 20 min

Data engine, promptable segmentation, and SAM 2, 3

04 **The Bigger Picture** — 10 min

CLIP vs DINO vs SAM; same backbone, different impacts, bridge to Perception Encoder

Part 1

What is a Foundation Model?

Definition, conditions, and why they emerged now

Foundation Models: Definition

General / Multi-task

One model, many downstream tasks.

Not trained for a single task — serves as the foundation for classification, detection, segmentation, captioning, etc.

Key: zero-shot or few-shot transfer without task-specific retraining.

Scale

Large models + large data.

Typically:

- 100M–600M+ parameters
- 100M–1B+ training samples
- Thousands of GPU-hours

Scale is a necessary condition, not a sufficient one.

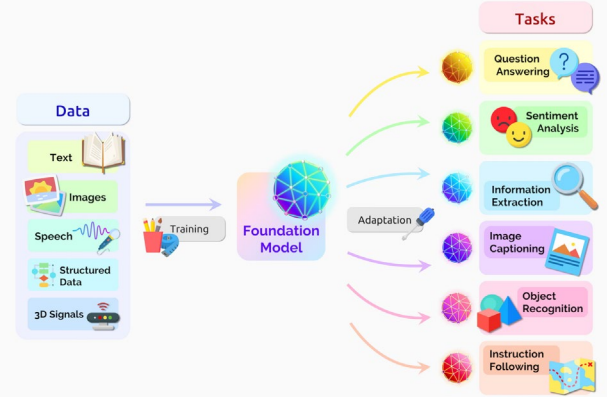
Self-Supervised Objective

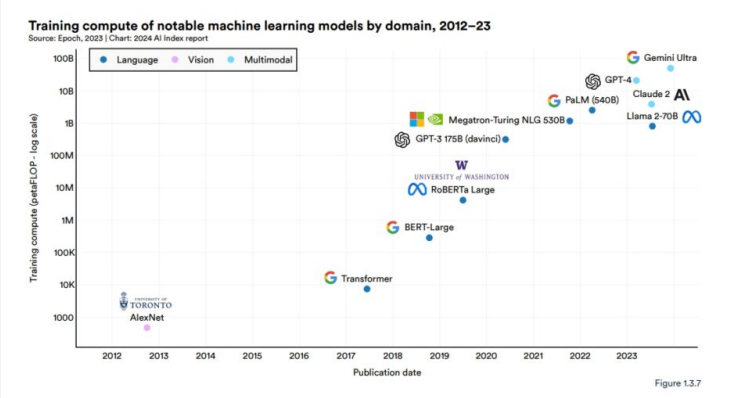
No manual labels at training time.

Instead:

- Contrastive (CLIP, SimCLR)
- Masking (MAE, BEiT)
- Self-distillation (DINO)
- Promptable (SAM)

This is why Wk 7 SSL matters here.



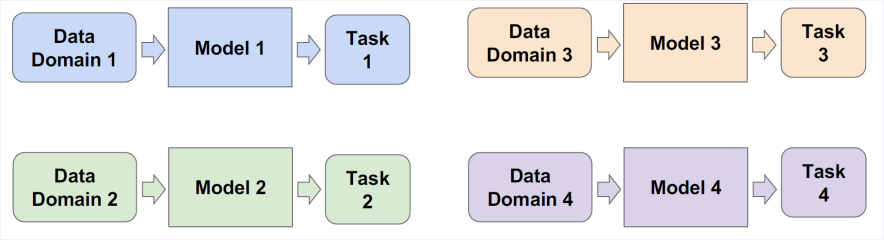


Source: https://blogs.nvidia.com/blog/what-are-foundation-models/

The Paradigm Shift

Before: Task-Specific Models

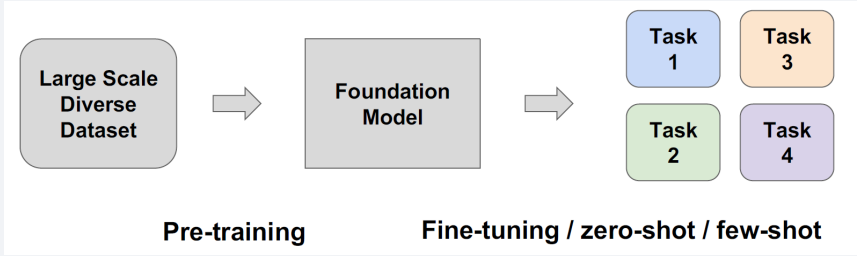
One model per task.
ImageNet classifier for classification.
Faster R-CNN for detection.
U-Net for segmentation.
Show-and-Tell for captioning.



Now: Foundation + Adaptation

One pretrained model, many tasks.
Adaptation methods:

- Zero-shot (CLIP text prompts)
- Linear probe (freeze + linear layer)
- Fine-tuning (full or LoRA)
- Prompting (SAM points/boxes)



Key Takeaway The encoder is trained once. Everything downstream is adaptation.

Why Did Foundation Models Emerge after 2020?

1	**Scale became feasible**	GPU clusters (A100/H200/B200), distributed training, mixed precision → training 1B+ param models became routine
2	**Web-scale data**	LAION-5B, WebImageText (400M), SA-1B (1.1B masks) — internet as the dataset
3	**SSL matured**	SimCLR → MoCo → DINO → MAE (Wk 7) proved labels are optional for learning good representations
4	**Transformer universality**	Same architecture (ViT, Wk 5) works for images, text, video, audio → modality-agnostic backbone
5	**Emergent capabilities**	Zero-shot transfer, compositional understanding, in-context learning — abilities not explicitly trained for

Key Takeaway Foundation models are the convergence of scale + data + SSL + Transformer — each necessary, none sufficient alone.

Part 2

CLIP Deep Dive

From SimCLR to image-text contrastive learning at scale

Wk 7 Recap: Contrastive Learning (SimCLR)

The foundation for understanding CLIP

SimCLR: Image ↔ Image

Same image, two augmentations → positive pair.
Different images → negative pairs.
InfoNCE loss:

- Pull positive pairs together
- Push negatives apart
- On a normalized hypersphere

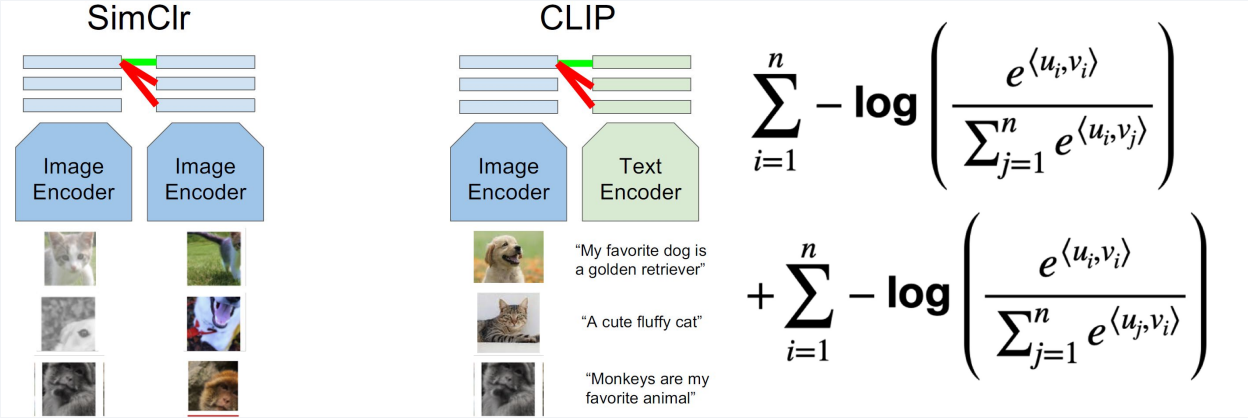
Result: encoder that maps semantically similar images to nearby embeddings.

CLIP: Image ↔ Text

Same concept, different modality → positive pair.
Image of a dog + "a photo of a dog" = positive.
Same InfoNCE loss, but across modalities:

- Image encoder (ViT-L/14)
- Text encoder (Transformer)
- Match image and text embeddings

Result: shared embedding space for vision and language.



CLIP: Training at Scale

Radford et al., ICML 2021

WebImageText (WIT)

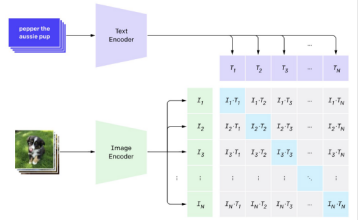
- 400 million image-text pairs
- Scraped from the internet (alt-text)
- No manual annotation
- ~30× larger than ImageNet (14M)
- Diverse: photos, illustrations, memes, diagrams

Cf. ImageNet: 14M images, 2 years of labeling
WIT: 400M pairs, automated scraping

Dual Encoder Architecture

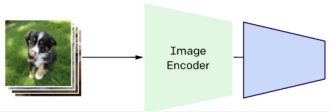
- Image encoder: ViT-L/14 or ResNet-50
  (Wk 5: /14 = patch size, L = 24 layers, 307M params)
- Text encoder: 12-layer Transformer, 63M params
- Both project to shared embedding space
- Training: contrastive loss on N×N similarity matrix
  (same as SimCLR, but image-text instead of image-image)
- Batch size: 32,768 (approx. in OpenCLIP)

Step 1: Pretrain a network on a pretext task that doesn't require supervision



Pre-training tasks:
Contrastive Objective

Step 2: Transfer encoder to downstream tasks via linear classifiers



Downstream tasks:
Image classification, object detection, semantic segmentation

CLIP: Prompt Engineering

Small text changes → big accuracy gains

Single Word

Prompt: "dog"

Baseline accuracy.

Problem: CLIP was trained on full sentences, not single words. "Dog" as a standalone word rarely appeared in training captions.

Mismatch between training and inference distribution.

Template Prompt

Prompt: "A photo of a dog"

+1.3% on ImageNet.

Better because training captions look like "A photo of..." — closer to the training distribution.

Simple but effective.

Ensemble Prompts

Prompts:
"A photo of a dog"
"A drawing of a dog"
"A blurry photo of a dog"
...

Average all text vectors → +5% on ImageNet.

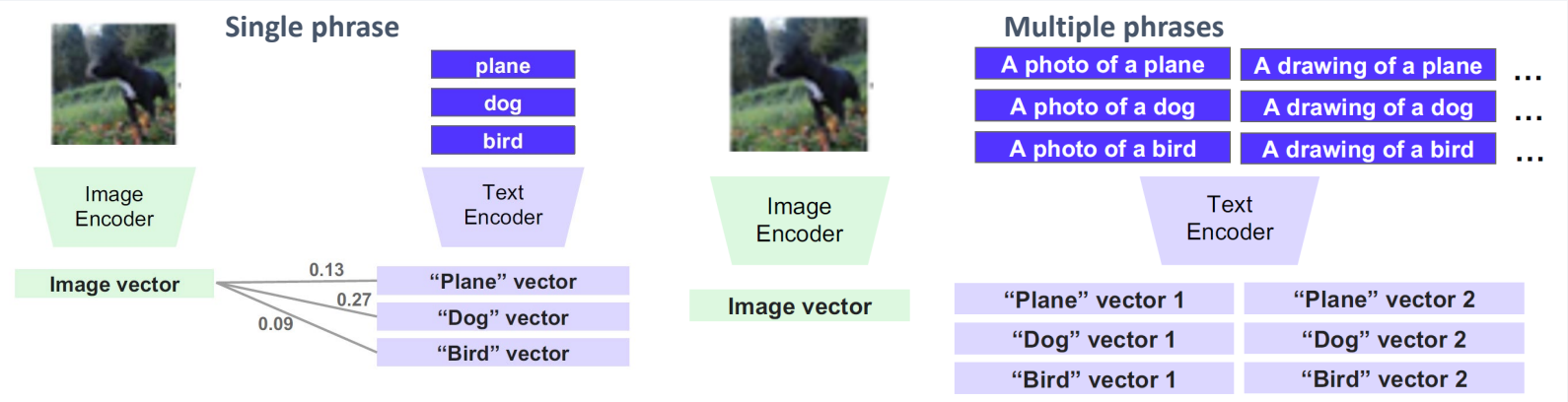
Reduces bias from any single template. 80 templates used in the original CLIP paper.

Key Takeaway Prompt engineering is the "hyperparameter tuning" of foundation models. Same model, different prompts, different performance.

CLIP: Zero-Shot Classification

No fine-tuning, no labels, no training on the target dataset

1	**Encode categories as text**	For each class, encode "A photo of a [category]" with the text encoder → text embedding vector
2	**Encode the test image**	Pass the image through the image encoder → image embedding vector
3	**Compute similarities**	Cosine similarity between image vector and each text vector → similarity scores
4	**Predict the class**	Highest similarity = predicted class. Effectively a 1-nearest-neighbor classifier in embedding space



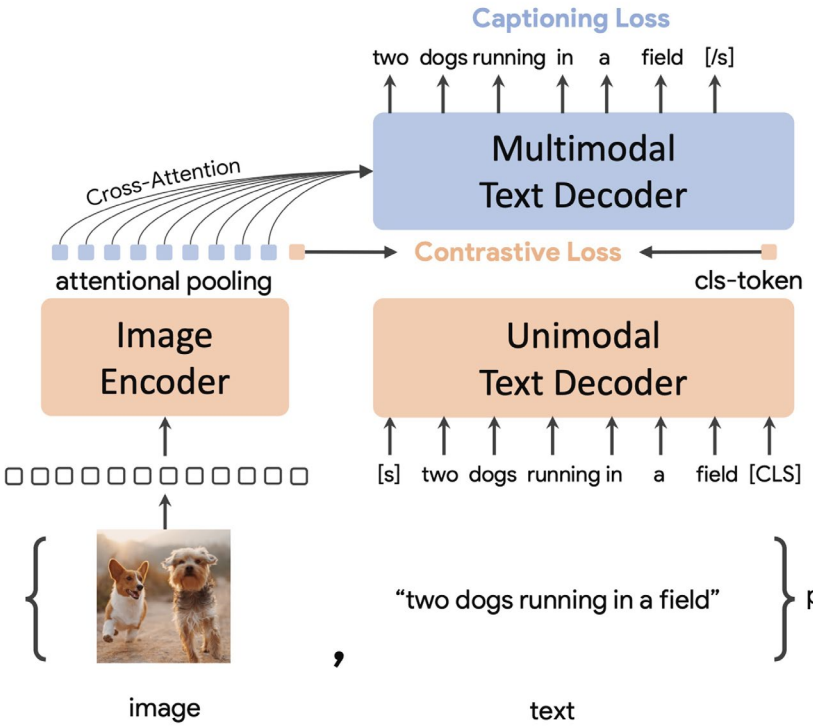
CLIP: Limitations

Understanding the boundaries is as important as understanding the capabilities

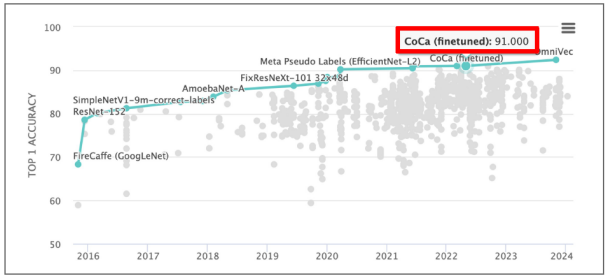
1	**Compositionality**	"A mug in some grass" vs "some grass in a mug" — CLIP cannot distinguish. Bag-of-words behavior
2	**Batch size dependency**	Fine-grained concepts require enormous batches (32K+). Batch=4: "animal". Batch=32K: "Welsh Corgi"
3	**Image-level captions only**	Alt-text describes the image globally, not regions → no spatial grounding
4	**Counting & spatial relations**	CLIP struggles with "three cats" vs "two cats" and spatial relationships (left/right/above)
5	**Data quality matters**	Web-scraped data includes noise, bias, and harmful content. Filtering is crucial but imperfect

Key Takeaway These limitations motivate SigLIP (batch size), RegionCLIP (spatial), CoCa (generation), and ultimately Perception Encoder.

CoCa: Contrastive Captioners



Model	ImageNet
ALIGN [13]	88.6
Florence [14]	90.1
MetaPseudoLabels [51]	90.2
CoAtNet [10]	90.9
ViT-G [21]	90.5
+ Model Soups [52]	90.9
CoCa (frozen)	90.6
CoCa (finetuned)	**91.0**



Model	ImageNet	ImageNet-A	ImageNet-R	ImageNet-V2	ImageNet-Sketch	ObjectNet	Average
CLIP [12]	76.2	77.2	88.9	70.1	60.2	72.3	74.3
ALIGN [13]	76.4	75.8	92.2	70.1	64.8	72.2	74.5
FILIP [61]	78.3	-	-	-	-	-	-
Florence [14]	83.7	-	-	-	-	-	-
LiT [32]	84.5	79.4	93.9	78.7	-	81.1	-
BASIC [33]	85.7	85.6	95.7	80.6	76.1	78.9	83.7
CoCa-Base	82.6	76.4	93.2	76.5	71.7	71.6	78.7
CoCa-Large	84.8	85.7	95.6	79.6	75.7	78.6	83.3
CoCa	**86.3**	**90.2**	**96.5**	**80.7**	**77.6**	**82.7**	**85.7**

Table 4: Zero-shot image classification results on ImageNet [9], ImageNet-A [64], ImageNet-R [65], ImageNet-V2 [66], ImageNet-Sketch [67] and ObjectNet [68].

Beyond CLIP: Variants and Successors

Wk 5 preview → now with context

Model	Key Change from CLIP	Why It Matters
OpenCLIP (2022~)	Open-source reproduction + LAION-2B training	Reproducibility + accessible to researchers
SigLIP (2023)	Sigmoid loss replaces softmax → no batch-size dependency	Solves CLIP limitation #2 (Batch size). Enables smaller-batch training
CoCa (2023)	Contrastive + captioning decoder	Combines CLIP's retrieval with generative capability
EVA-CLIP (2023)	Better ViT initialization (EVA pretraining)	Higher accuracy with same CLIP objective
SigLIP 2 (2025)	Improved recipe + multilingual + fairness	Powers Gemini, PaLI. Current Google standard
PE (2025)	Intermediate layer extraction + spatial alignment	SOTA on classification + detection + segmentation simultaneously

Key Takeaway PE (today's seminar paper) is the latest in this lineage. It asks: are the best embeddings at the output layer?

Part 3

SAM Deep Dive

Data engine, promptable segmentation, and SAM 2, 3

Wk 6 Recap: SAM Architecture

Kirillov et al., ICCV 2023

Image Encoder

ViT-H/14 (Wk 5)
632M parameters
Processes entire image once.
Outputs feature map.

Heavy but runs only once per image.
Pretrained with MAE (Wk 7).

Prompt Encoder

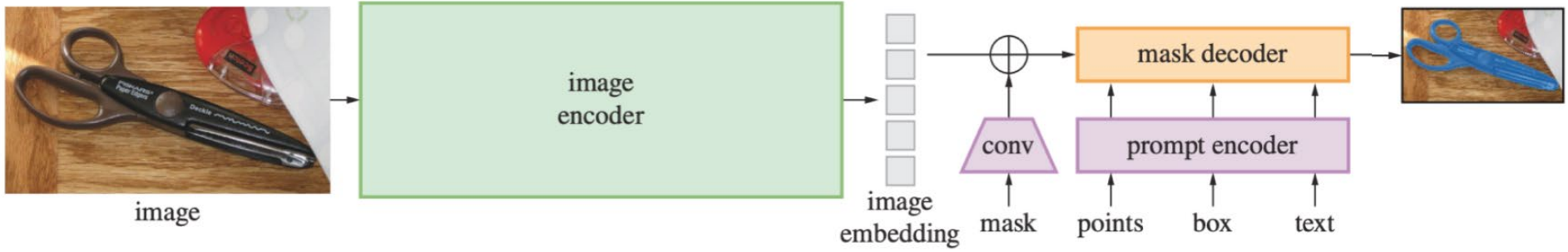
Encodes user prompts:
- Points (positive/negative)
- Bounding boxes
- Text (via CLIP embedding)
- Dense masks

Sparse prompts → positional encodings + learned embeddings.

Mask Decoder

Transformer decoder.
Cross-attends to image features + prompt tokens.
Outputs:
- 1~3 candidate masks
- Confidence scores

Lightweight (~4ms per prompt).
Real-time interactive segmentation.



SAM: The Data Engine

How do you get 1.1 billion masks without manually labeling them?

Stage 1: Assisted-Manual

Professional annotators use SAM interactively.

Annotator clicks points → SAM suggests masks → annotator corrects.

faster than manual polygon annotation.

4.3M masks from 120K images.

Bootstraps the first SAM model.

Stage 2: Semi-Automatic

SAM auto-detects confident objects.

Annotators focus on remaining objects SAM missed.

5.9M additional masks from 180K images.

Increases object diversity — catches things SAM initially couldn't segment.

Retrains SAM with combined data.

Stage 3: Fully Automatic

Grid of 32×32 point prompts per image.

SAM generates masks for every region.

Filtering: NMS, confidence threshold, stability.

1.1B masks from 11M images.

No human annotation at this stage.

= SA-1B dataset.

Key Takeaway The model improves the data, and the data improves the model. This loop is the core of foundation model development.

SAM: Promptable Segmentation as a Foundation Task

Why "Promptable"?

Traditional segmentation:
Trained on fixed classes (COCO: 80 categories).
Cannot segment novel objects.

SAM:
No class concept at all.
"Segment whatever I point to."

This is like CLIP's zero-shot:
CLIP classifies any category via text prompt.
SAM segments any object via spatial prompt.

Both achieve generality through prompting
instead of class-specific training.

Zero-Shot Transfer

SAM trained on SA-1B (web images).

Transfers without fine-tuning to:

- Medical imaging (X-rays, CT)
- Satellite imagery
- Microscopy
- Industrial inspection
- Agriculture

Not perfect on every domain —
but provides a strong starting point.

Fine-tuning (LoRA, adapter) bridges
the remaining domain gap.

SAM 2 & Beyond

From images to video to open-vocabulary

1	**SAM 2 (2024)**	Video extension. Memory mechanism tracks objects across frames. SA-V: 51K videos, 643K masklets
2	**Grounded SAM**	CLIP/GroundingDINO (text) + SAM (masks). "Segment the red car" → text finds it, SAM segments it (Wk 6 recall)
3	**SAM 3 / SA-Co (2025)**	Concept-based segmentation. 270K unique concepts (vs COCO's 80). Text or example-image prompting
4	**EfficientSAM / FastSAM**	Lightweight variants. Knowledge distillation from ViT-H to smaller backbones. Mobile deployment
5	**Domain-specific SAMs**	MedSAM (medical), RSPrompter (remote sensing), Track Anything (video). Fine-tuned for specific domains

Key Takeaway SAM's impact extends far beyond segmentation. It established 'promptable foundation model' as a paradigm.

Part 4

The Bigger Picture

CLIP vs DINO vs SAM; same backbone, different impacts, bridge to Perception Encoder

Three Foundation Models, One Backbone

Same ViT, trained differently → different capabilities

	CLIP	DINOv2	SAM
Backbone	ViT-L/14	ViT-g/14	ViT-H/14
Training	Contrastive (image-text pairs)	Self-distillation + iBOT (Wk 7)	Promptable seg. + data engine
Data	400M image-text	142M images (curated)	11M images 1.1B masks
Superpower	Semantic understanding (zero-shot cls)	Local features (unsupervised seg, depth, matching)	Spatial understanding (any-object seg)
Limitation	No spatial detail (image-level only)	No language understanding	No semantic concepts
Wk covered	Wk 5 preview + Today	Wk 7 (DINO/DINOv2)	Wk 6 + Today

Key Takeaway Can one model do all three? That's the question Perception Encoder tries to answer.

Bridge to Paper #1: Perception Encoder

The Problem

CLIP: semantic ✓, spatial ✗
DINO: local ✓, language ✗
SAM: spatial ✓, semantic ✗

Using only the output layer of any single model gives you one superpower but not the others.

Current solution: run multiple models and combine.
→ Expensive, complex, inelegant.

Is there a better way?

PE's Key Insight

"The best visual embeddings are not at the output of the network."

Intermediate layers contain different information:
- Early layers: texture, edges
- Middle layers: parts, spatial structure
- Late layers: semantic concepts

PE extracts from all layers using:
1. Language alignment (semantic)
2. Spatial alignment (local/dense)

One model, one training, all tasks.
→ Seminar session starts from here.

Summary & Resources

What We Covered

- Foundation model = general + scale + SSL
- CLIP: contrastive image-text training
  → zero-shot via text-as-classifier
  → prompt engineering (+5%)
  → limitations: compositionality, spatial
- SAM: promptable segmentation
  → data engine (3 stages → 1.1B masks)
  → SAM 2 (video), Grounded SAM
- CLIP vs DINO vs SAM:
  same ViT, different training, different powers
- PE: intermediate layers are the answer

Connections Across Weeks

Wk 3 (CNN) → ResNet as CLIP baseline
Wk 4 (Transformer) → Text encoder in CLIP
Wk 5 (ViT) → ViT-L/14, ViT-H/14 naming
Wk 6 (Det/Seg) → SAM architecture
Wk 7 (SSL) → SimCLR → CLIP objective
→ MAE → SAM pretraining
→ DINO → comparison with CLIP

Looking ahead:
Wk 10: Diffusion — another foundation paradigm
Wk 12: VLM — CLIP as the vision backbone
for LLaVA, GPT-4V, Gemini

Next Week Week 10: Diffusion models – Lecture + Paper #2 Seminar

Acknowledgments

Some slides and figures in this course are adapted from or inspired by the following open resources.

- **Stanford CS231n: Deep Learning for Computer Vision (Spring 2025)**
  Fei-Fei Li, Ehsan Adeli, et al. | cs231n.stanford.edu
- **UMich EECS 498/598: Deep Learning for Computer Vision (Winter 2022)**
  Justin Johnson | web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/
- **MIT 6.7960: Deep Learning (Fall 2024)**
  Phillip Isola, Sara Beery, Jeremy Bernstein | ocw.mit.edu/courses/6-7960-deep-learning-fall-2024/
- **CMU 16-824: Visual Learning and Recognition (Fall 2025)**
  Jun-Yan Zhu | visual-learning.cs.cmu.edu
- **Key papers: CLIP, SAM, CoCa, SigLIP, DINOv2, Perception Encoder**