Foundation Models: Modern Advances and Applications

Naeemullah Khan
naeemullah.khan@kaust.edu.sa



جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

KAUST Academy
King Abdullah University of Science and Technology

October 7, 2026

Table of Contents

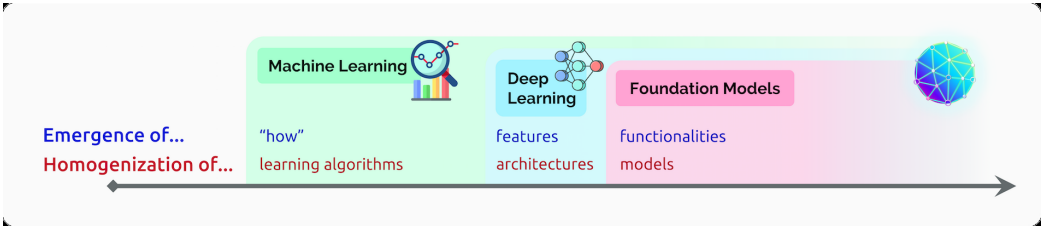
1. Recap: from self-supervised pretraining to foundation models
2. Foundation models: definition, impact, types, and adaptation (probes, LoRA, prompting)
3. Foundation models vs. LLMs and ecosystem implications (Bommasani et al.)
4. Historical arc and vision FM taxonomy (representation vs. generative)
5. CLIP: joint image–text representations and zero-shot transfer
6. Grounding DINO and DINOv2: open-vocabulary detection and representation learning
7. SAM 1, SAM 2, and SAM 3: promptable segmentation and data engines
8. SEEM: segmentation from multimodal prompts
9. Depth Anything and SuperGlue: monocular depth and geometric matching
10. TabPFN and TRIBE v2: tabular and brain-response foundation models
11. InternVideo2: large-scale video foundation modeling
12. Limitations, summary, hands-on resources, and references

Recap: From Pretraining Methods to Foundation Models

- **Covered earlier:** pretext tasks, contrastive learning (SimCLR, MoCo), and self-distillation (BYOL, DINO, DINOv2) learn features from unlabeled data.
- **Also covered earlier:** latent prediction (I-JEPA, V-JEPA 2) trains encoders that serve many tasks from frozen features; V-JEPA 2 already behaves like a video foundation model.
- Those lectures answered *how to pretrain*. Today we look at the pretrained models the field actually builds on: what they are, what data they use, and how they are adapted.
- Working definition (Bommasani et al., 2021): a **foundation model** is trained on broad data at scale and can be **adapted** to a wide range of downstream tasks.

What are Foundation Models?

- **Foundation models** are large, general AI models pretrained on massive, diverse datasets.
- They provide a universal base that can be adapted (via fine-tuning or prompting) across many domains and tasks.
- The key shift: build powerful, flexible models for reuse, rather than training a new model for each task from scratch.

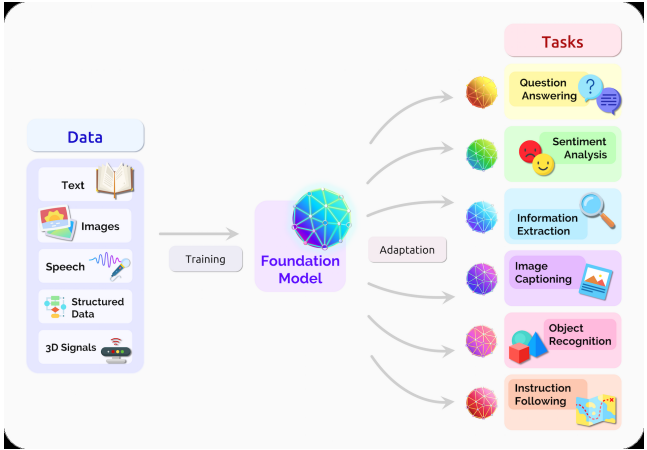


Emergence and homogenization from ML → deep learning → foundation models (Bommasani et al., arXiv:2108.07258).

Why Do Foundation Models Matter?

- **Accelerate innovation:** Dramatically reduce time and resources needed to create high-performing AI applications.
- **Expand accessibility:** Lower the expertise required for deploying AI, empowering domain experts (not just AI researchers).
- **Enable new capabilities:** Support advanced "few-shot" and "zero-shot" learning, allowing for rapid adaptation to new tasks.
- **Transform industries:** Fuel advances in medicine, science, robotics, business, and beyond.
- **Introduce new challenges:** Require careful management of data/computational resources, fairness, safety, and responsible use.

One Foundation Model, Many Tasks



Diverse modalities feed training; one foundation model supports many downstream tasks via adaptation (Bommasani et al., arXiv:2108.07258).

Types of Foundation Models

- **Language models:** Trained on text data, these models can generate, summarize, and understand language.
  Examples: BERT, GPT
- **Vision models:** Trained on images/videos, they can recognize objects, segment scenes, and more.
  Examples: CLIP, SAM
- **Multi-modal models:** Learn to connect information across text, images, audio, or video.
  Examples: CLIP (text-image), Flamingo, GPT-4V, Gemini
- **Other domains:** Foundation models exist for structured (tabular) data, medical data, code, and scientific domains.
  Examples: TabPFN (tabular), TRIBE (brain responses); both appear later in this lecture

How Do Foundation Models Work?

- **Pretraining at scale:** Train on enormous datasets (text, images, etc.) using self-supervision; learn generalized representations.
- **Adaptation:** Foundation models can be quickly adapted to new tasks using:
  - Fine-tuning on small amounts of labeled data
  - Prompting (providing instructions or context)
  - In-context learning (few-shot or zero-shot)
- **Transfer learning:** Reuse knowledge across tasks and domains, reducing the need for bespoke model training.
- **Unified architectures:** Many foundation models use architectures like transformers, enabling processing across modalities.

Adaptation in Practice: From Frozen Probes to LoRA

- **Frozen probes:** keep the backbone fixed and train a small head on top (linear or attentive probe, as covered earlier). Cheapest option; also the standard way to *evaluate* a foundation model.
- **Full fine-tuning:** update all weights on the downstream task. Strongest task performance, highest cost, and the model can forget general capabilities.
- **Parameter-efficient fine-tuning (PEFT):** train small added modules while the backbone stays frozen. **LoRA** (Hu et al., 2022) learns a low-rank update $\Delta W = BA$ next to each frozen weight matrix W, typically well under 1% of the parameters.
- **Prompting and in-context learning:** no weight updates at all; instructions or examples are given at inference time. TabPFN (later this lecture) classifies whole tabular datasets this way.

Key Differences: Modalities, Tasks, and Adaptation

Foundation models

- Trained on **heterogeneous** data: text, images, speech, structured data, 3D/signals, and more.
- **Homogenization:** one pretrained model is adapted to many applications; **emergence:** capabilities (e.g. in-context learning) from scale were not explicitly programmed.
- Objectives: self-supervised, contrastive, masked, or alignment losses, not only next-token prediction.

LLMs

- **Language-first:** data are primarily text; core loss is usually autoregressive or masked LM.
- **Downstream:** generation, QA, summarization, code; multimodal **variants** pair an LLM with other encoders.
- Same "pretrain then adapt" pattern, but the **scope of pretraining data** is narrower than that of full multi-modal foundation models.

Implications: Ecosystem and Deployment

- **Foundation models** enable transfer across domains (vision–language, robotics, biomedicine, . . .) but concentrate capability in a few shared pretrained artifacts.
- **LLMs** excel at language and code; for vision, video, or tabular data they are usually **one component** of a larger system unless trained multi-modally from the start.
- **Limitations:** compute and data scale, inherited biases from pretraining, evaluation and safety gaps as the same "foundation" is reused widely.

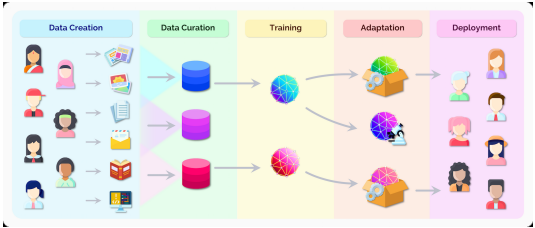
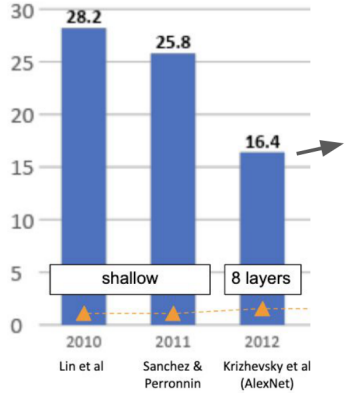
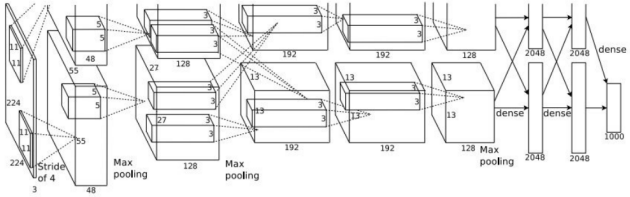


Fig. 3: From people and data through training and adaptation to deployment (Bommasani et al., arXiv:2108.07258).

2012: Deep learning emerges in AI





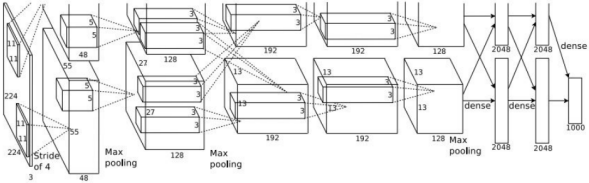
Krizhevsky et al. 2012. 8-layer “AlexNet”.

ImageNet Visual Recognition Challenge results.

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

Convergence of key ingredients of deep learning

Algorithms



Compute

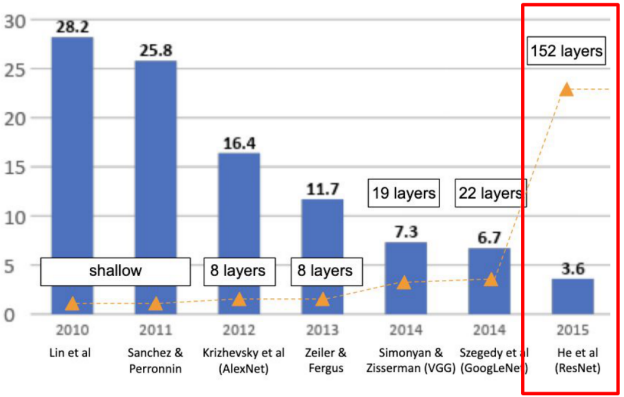


Data



Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

2015: Very deep convnets and challenging vision tasks



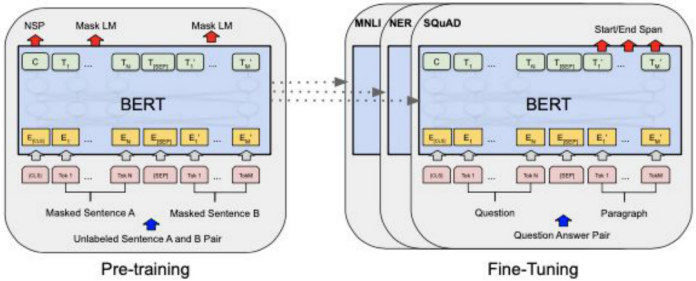


He et al. 2015. ResNet.

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

2018: Breakthroughs in deep learning for NLP (sequences)

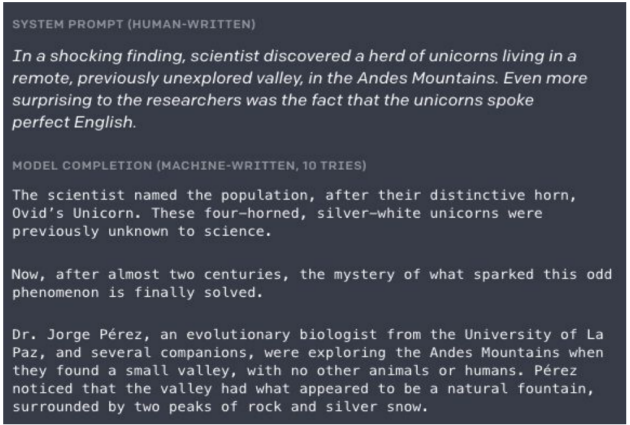
Transformer neural network architectures (containing self-attention mechanism) and pre-training -> fine-tuning. State-of-the-art on 11 NLP benchmarks.



Vaswani et al. 2017. “Attention is all you need.”
Devlin et al. 2018. BERT.

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

2020: Very large-scale text generation models



Transformer architectures become ubiquitous

Training these models on very large amounts of data -> rise of "foundation models"

OpenAI GPT-3 (figure from GPT-2). Brown et al. 2020.

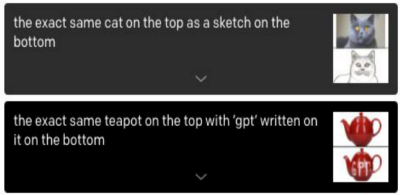
Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

2021: Transformers & foundation models reach computer vision

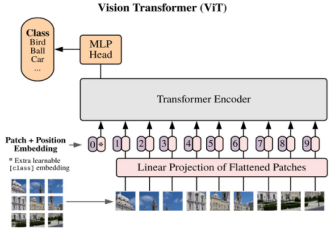
Transformer architecture can be applied to images as well!

Key idea: Convert image into sequence of patches (i.e. image tokens).

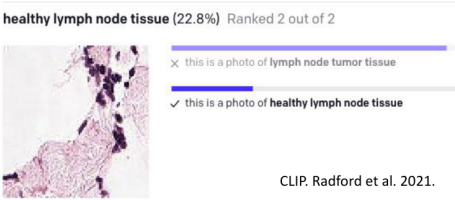
Can then benefit from Transformer architecture and its self-attention mechanism. Allows process of token encoding to incorporate contextual understanding (attention) from other tokens of the same (self) sequence.



DALL-E. Ramesh et al. 2021.



ViT. Dosovitsky et al. 2021



CLIP. Radford et al. 2021.

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

2022: Major improvements in text-to-image generation models

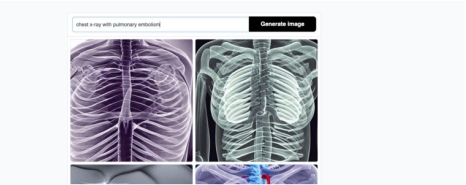
DALL-E 2
Ramesh et al. 2022.



Meta Make-a-Video.
Singer et al. 2022



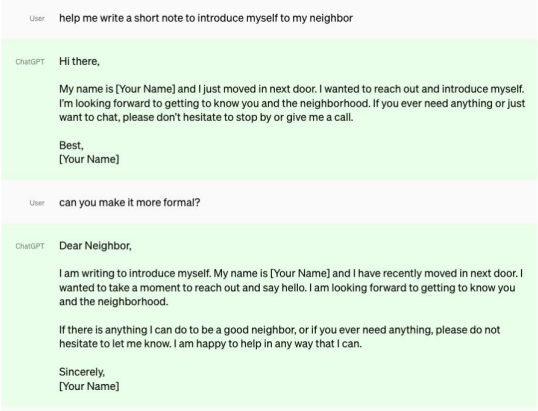




Stable Diffusion.
https://stablediffusionweb.com/

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

Late 2022: ChatGPT reaches public mainstream



OpenAI ChatGPT
https://openai.com/blog/chatgpt

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy).

Two major classes of vision foundation models

Representation Learners

Train on huge amounts of data

Objective: learn powerful generalist feature extractors that can be used for downstream computer vision tasks.

- Vision Only (e.g. DINOv2)
- Vision-Language (e.g. CLIP)

Generative Models

Train on huge amounts of data

Objective: learn to generate images and/or text.

- Text $\rightarrow$ Image (e.g. Stable Diffusion, DALL-E)
- Text, Image $\rightarrow$ Text (e.g. Flamingo, GPT-4V)

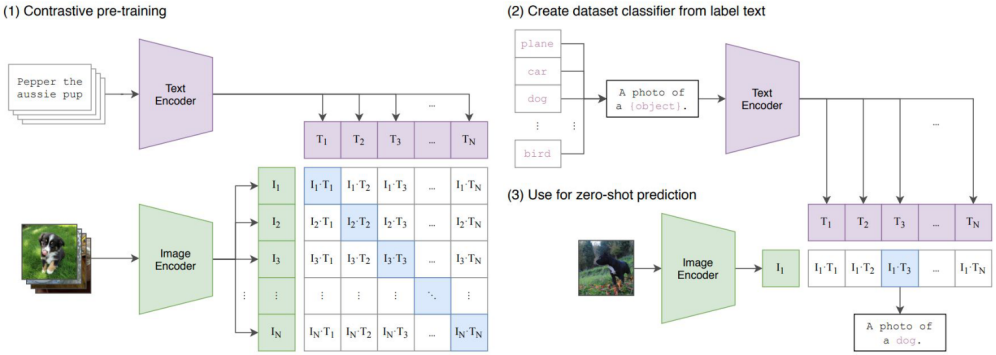
The Generative Class in Brief

- **Text-to-image diffusion** (Stable Diffusion, DALL-E): start from random noise and iteratively **denoise** it, conditioned on a text prompt. The denoiser is trained on billions of captioned images; latent diffusion (Rombach et al., 2022) runs the process in a compressed latent space to cut cost.
- **Vision-language generative models** (Flamingo, GPT-4V, LLaVA): a vision encoder's output reaches an **LLM** either as visual tokens through a small projector (LLaVA, the same pattern as the V-JEPA 2 language alignment covered earlier) or through cross-attention layers inserted in the LLM (Flamingo). The LLM then generates text about the image.
- Text conditioning in both cases relies on strong text encoders; Stable Diffusion uses **CLIP's** text encoder, which we cover next.
- This lecture focuses on the **representation-learning** class; generative modeling is covered in its own lectures elsewhere in the course.

CLIP: Learning Vision from Natural Language

- **CLIP** (Radford et al., 2021; covered earlier): trained on **400 million** image–text pairs collected from the web; no class labels at all. Here we look at how it works and why it became infrastructure.
- Two encoders: an **image encoder** (ResNet or ViT) and a **text encoder** (transformer), each mapping its input to a shared embedding space.
- Training pulls each image toward *its own caption* and away from the other captions in the batch.
- Same contrastive idea as SimCLR (covered earlier), but the two "views" come from **different modalities**: an image and its natural-language description.

CLIP: Contrastive Pre-training



Radford et al. 2021.

Radford et al., "Learning Transferable Visual Models From Natural Language Supervision," ICML 2021, Fig. 1.

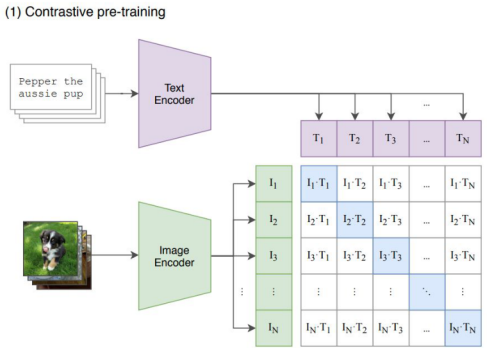
CLIP: The Objective

For a batch of N pairs, compute the $N \times N$ matrix of cosine similarities s_{ij} between image embeddings I_i and text embeddings T_j. The loss is a symmetric cross-entropy that treats the diagonal as correct:

$$\mathcal{L} = -\frac{1}{2N} \sum_{i=1}^{N} \left[\log \frac{e^{s_{ii}/\tau}}{\sum_j e^{s_{ij}/\tau}} + \log \frac{e^{s_{ii}/\tau}}{\sum_j e^{s_{ji}/\tau}} \right]$$

- First term: match each **image** to the right **caption**; second term: match each **caption** to the right **image**.
- τ is a learned **temperature**; large batches (up to 32,768 pairs) supply many negatives, as in the contrastive methods seen earlier.
- The result is a **joint embedding space** where semantically matching images and texts are close.

CLIP: Objective in Practice



Radford et al. 2021.

Learn text and image encoders (i.e. feature extractors) through a self-supervised "contrastive" objective: images and their matching text captions should produce more similar features than non-matching ones

Trained on 400M image-text pairs: generated by searching for image-text pairs on the web, where text comes from a base query list of 500,000 queries comprising all words occurring at least 100 times in the English version of Wikipedia

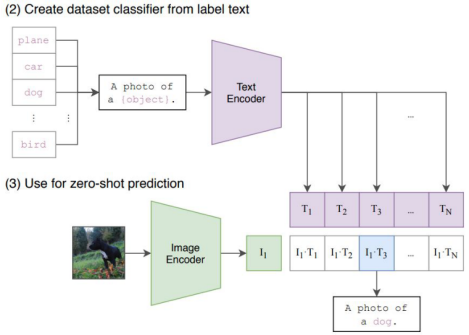
Stanford BIODS 276, Lecture 1 (S. Yeung-Levy); diagram from Radford et al., ICML 2021, Fig. 1.

CLIP: Zero-Shot Classification via Prompts

- A classifier can be built **from label names alone**: embed prompts such as "a photo of a {label}" with the text encoder, embed the image, and pick the label with the highest similarity.
- No training on the target dataset: this is **zero-shot transfer**. Prompt wording matters, and ensembling several templates improves accuracy.
- The largest CLIP matches the accuracy of the original supervised ResNet-50 on ImageNet **without seeing a single labeled ImageNet example**, and is notably more robust to distribution shift.
- Known weaknesses: fine-grained categories, counting objects, and reading text in images; the web training data also carries social biases.

CLIP: Zero-Shot Classification

Due to the contrastive training objective, image and text encoders output features in a shared representation space and the trained CLIP model can be used for zero-shot classification, i.e., perform N-way classification without showing the model paired examples of (input, class) for any of the N classes.



Radford et al. 2021.

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy); diagram from Radford et al., ICML 2021, Fig. 1.

CLIP as Infrastructure

- CLIP's encoders became **building blocks** across the field:
  - **Stable Diffusion** conditions image generation on CLIP text embeddings.
  - **Open-vocabulary detection** (Grounding DINO, next section) grounds detections in text.
  - **SAM 3** segments "any concept" described by a short text prompt.
  - Vision-language assistants (LLaVA and similar) use a CLIP-style vision tower.
- **OpenCLIP** and the **LAION** datasets reproduce CLIP openly, so the recipe is fully public.
- In practice: when a task can be phrased in language, a CLIP-family model often gives a strong zero-shot baseline before any training.

Detection Foundation Model: Grounding DINO

- Open-vocabulary detection with natural-language grounding.
- Uses transformer-based backbone for both image and text encoding.
- Matches object proposals in an image with user-provided text prompts (e.g. “a red backpack”).
- Produces bounding boxes, labels, and confidence scores for detected objects.
- **Key architectural mechanisms**:
  - **Feature enhancer neck** with deep image–text cross-attention (not only late fusion).
  - **Language-guided query selection** to initialize decoder queries on relevant regions.
  - **Cross-modality decoder head** for iterative region-text refinement.
- **Result:** strong transfer to unseen categories and prompts; the text branch is a BERT encoder trained with GLIP-style grounded pre-training (region-word alignment), a different route from CLIP’s contrastive image-text objective.

Grounding DINO: How It Works

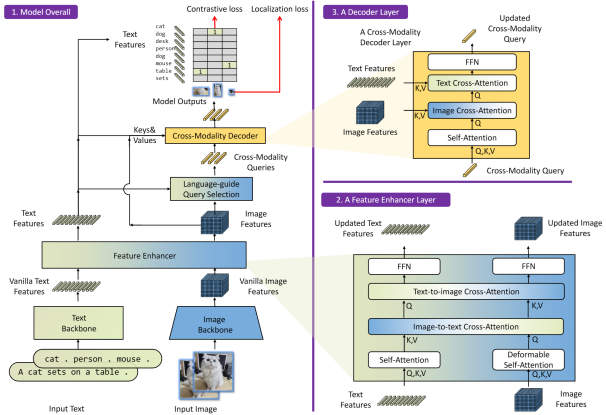


Figure 3. The framework of Grounding DINO. We present the overall framework, a feature enhancer layer, and a decoder layer in block 1, block 2, and block 3, respectively.

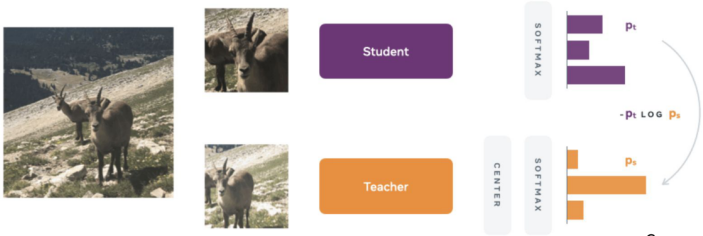
Liu et al., "Grounding DINO," ECCV 2024, Fig. 3.

DINOv2: A Vision-Only Foundation Backbone

- Scales the **self-distillation** recipe covered earlier (DINO image-level loss + iBOT masked-token loss) into a foundation model.
- **Curated data instead of raw web data:** LVD-142M is built by retrieving images similar to curated seed datasets from a large web pool, with deduplication. Data curation is a core part of the recipe.
- Train a large ViT ($\sim$1.1B parameters), then **distill** it into smaller ViTs for deployment.
- **Frozen features** transfer to classification, segmentation, depth, and retrieval without fine-tuning; Depth Anything (later this lecture) uses a DINOv2 encoder.
- **DINOv3** (2025) scales further (7B-parameter ViT) and adds a gram-anchoring loss to keep **dense** patch features clean at long training schedules.

DINOv2: Self-Supervised Training at Scale

- DINO (self-Distillation with NO labels) is trained through self-supervised learning (no external label curation needed)
- Student-teacher self-supervised training objective: At a high level, images (teacher) should produce similar feature representations to randomly cropped portions of the image (student)

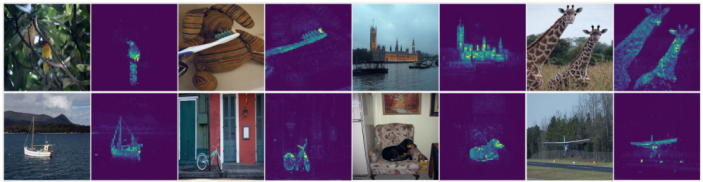


Caron et al. 2021, Oquab et al. 2023

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy); diagram from Caron et al. 2021 / Oquab et al. 2023.

DINOv2: One Frozen Backbone, Many Downstream Tasks

- Once trained, the DINO neural network backbone can be used as the “pre-trained” feature extractor for many downstream tasks



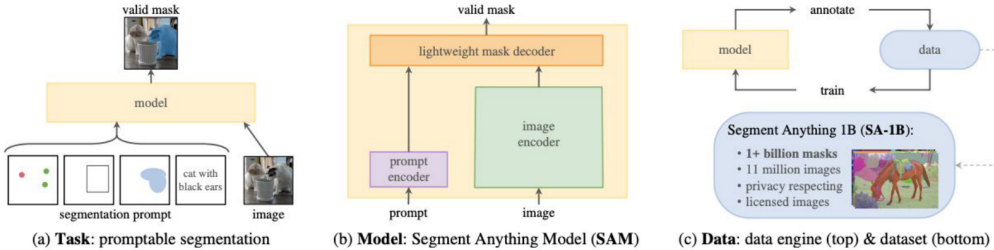
- DINOv2 made further implementation refinements and trained on 142 million images
- An example of a vision “**foundation model**”: a model trained on a very large amount of data that can be subsequently used or tuned for diverse tasks. DINOv2 was demonstrated to be a state-of-the-art backbone for many tasks.

Caron et al. 2021, Oquab et al. 2023

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy); visualizations from Caron et al. 2021 / Oquab et al. 2023.

Segment Anything Model (SAM): Promptable Segmentation

- Foundation Model for promptable segmentation, based on a Transformer encoder-decoder architecture
- Trained on 1 billion masks from 11 million images, using the model in a data collection loop



Kirrilov et al. 2023.

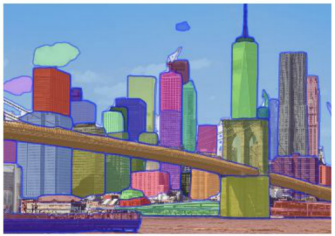
Stanford BIODS 276, Lecture 1 (S. Yeung-Levy); diagram from Kirillov et al., "Segment Anything," ICCV 2023, Fig. 1.

SAM: Zero-Shot Segmentation Examples

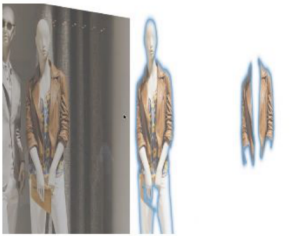
- Allows segmentation based only from prompts, without needing to provide training examples of target classes (zero-shot segmentation)



Prompt it with interactive points and boxes.



Automatically segment everything in an image.



Generate multiple valid masks for ambiguous prompts.

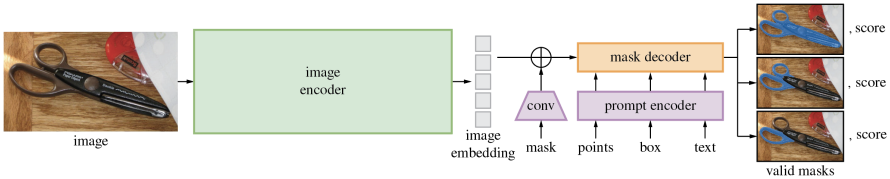
Meta AI Segment Anything Model (SAM)
https://segment-anything.com/

Kirrilov et al. 2023.

Stanford BIODS 276, Lecture 1 (S. Yeung-Levy); examples from segment-anything.com.

How Does SAM Work?

- A heavy **image encoder** (MAE-pretrained ViT) embeds the image once; a light **prompt encoder** and **mask decoder** then run interactively in milliseconds.
- User or downstream model provides prompts (points, boxes, masks).
- The decoder predicts **several candidate masks** to resolve ambiguous prompts (a click on a shirt could mean the shirt or the person).
- Produces high-quality object masks across a wide range of settings without fine-tuning; masks are class-agnostic (no labels attached).



Kirillov et al., "Segment Anything," ICCV 2023, Fig. 4.

SAM 1: The Data Engine Behind SA-1B

- There was no billion-mask dataset to train on, so Meta built one **with the model in the loop**, in three stages:
  1. **Assisted-manual:** annotators refine masks proposed by an early SAM.
  2. **Semi-automatic:** SAM proposes confident masks; annotators add what it missed.
  3. **Fully automatic:** SAM is prompted with a grid of points and annotates images alone.
- Result: **SA-1B** with 11M licensed images and **1.1B masks**, orders of magnitude larger than prior segmentation datasets.
- The pattern (model annotates, humans verify, retrain, repeat) recurs in SAM 2 and SAM 3; for foundation models, the **data engine** is as important as the architecture.

SAM 2: Promptable Segmentation in Video

- Extends promptable segmentation to **video**: prompt on any frame, and the model propagates the object mask (a *masklet*) through the clip.
- **Streaming memory:** a memory bank stores features and object pointers from past frames; each new frame **cross-attends** to this memory before decoding its mask.
- Corrections are interactive: extra clicks on any frame refine the whole track.
- Trained with its own data engine, producing the **SA-V** dataset ($\sim$51K videos, $\sim$600K masklets); also faster than SAM 1 on still images.

SAM 1, SAM 2, and SAM 3: What's the Difference?

SAM 1

- Segments any object in any image.
- Requires as little as a single click to segment an object.
- Refine prediction with a few clicks.

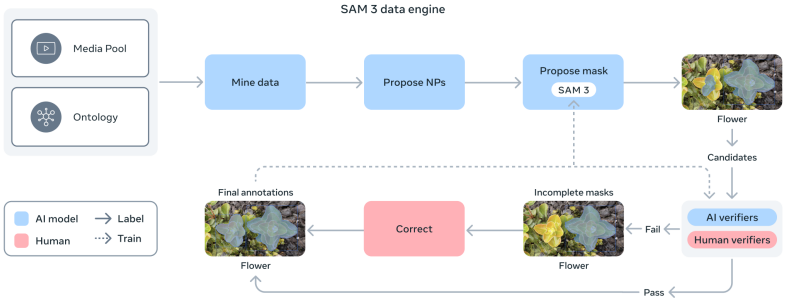
SAM 2

- Segments and tracks any object in any image or video.
- Uses click, box, or mask prompts.
- Tracks segment predictions across video frames.
- Refine prediction with a few clicks.

SAM 3

- Detect, segment, and track every instance of any object category in an image or video.
- Uses text or examples as prompts.
- Segments and tracks objects from a click.
- Tracks segment predictions across frames.
- Detects and segments matching instances.
- Refine detection with visual examples.

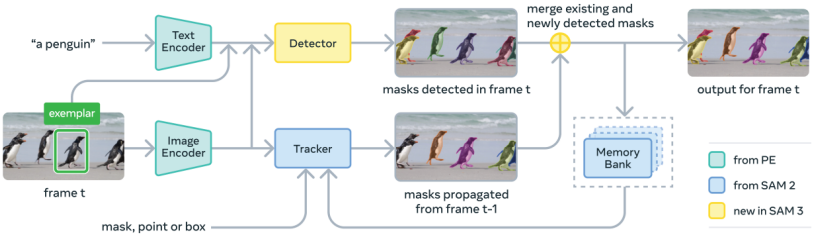
SAM 3 Architecture: Data Engine



SAM 3 data-engine loop from the Meta AI blog, "SAM 3.1" (https://ai.meta.com/blog/segment-anything-model-3/).

- Getting masks *with text labels* across many categories and domains is the hard part; SAM 3 scales the SAM data-engine pattern to concepts.
- AI models and annotators (human and Llama-based) collaborate in a feedback loop to mine, annotate, and verify data, yielding **4M+** annotated concepts across diverse domains.

SAM 3 Architecture: Core Model



SAM 3 model summary and detector/tracker architecture. [Carion et al., "SAM 3: Segment Anything with Concepts," 2025, Fig. 4]

A dual encoder-decoder transformer serves as the **detector** for image-level capabilities and is used together with a **tracker** and memory for video. Both receive vision-language features from an aligned Perception Encoder (PE) backbone.

SAM 3.1 Inference Architecture: Object Multiplexing



SAM 3.1 multiplexed inference versus per-object computation. [Source: Meta AI blog, "SAM 3.1," https://ai.meta.com/blog/segment-anything-model-3/]

- Process multiple tracked objects in one forward pass (*multiplexing*) instead of one pass per object.
- Shared object-level context reduces redundant compute and helps crowded-scene tracking.

Segment Everything Everywhere All at Once

Xueyan Zou*[1§], Jianwei Yang*[2‡], Hao Zhang*[♯], Feng Li*[♯], Linjie Li[†], Jianfeng Gao[¶‡], Yong Jae Lee[¶§]
[§] University of Wisconsin-Madison [‡] Microsoft Research at Redmond [♯] HKUST [†] Microsoft Cloud & AI
*Equal Contribution ¶Equal Advisory Contribution 1. Main Technical Contribution 2. Project Lead
{xueyan,yongjaelee}@cs.wisc.edu {jianwyan,jfgao,linjli}@microsoft.com {hzhangcx,fliay}@connect.ust.edu

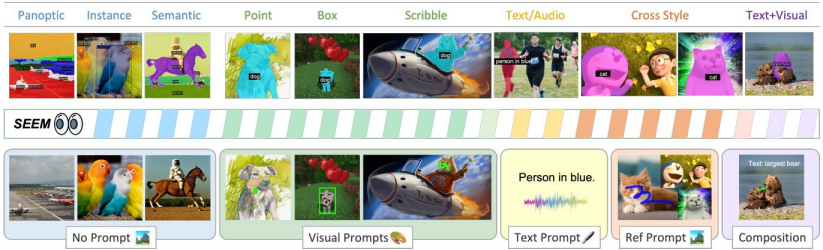


Figure 1: Our model, *SEEM*, can perform any segmentation task. When no prompt is given, it can do classic segmentation tasks such as semantic, instance, and panoptic segmentation in open-set scenarios. *SEEM* also supports various types of prompts and any combination of these prompts.

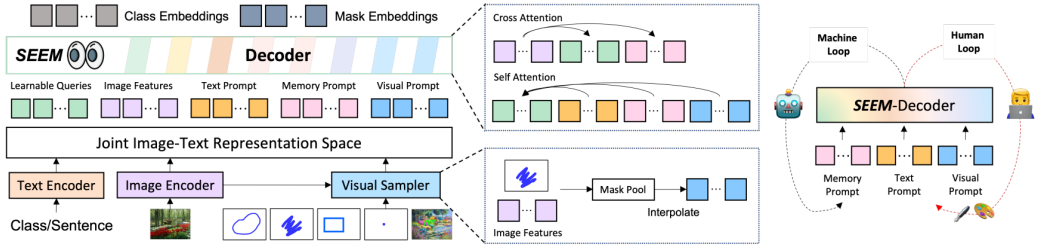
Segment Everything Everywhere All at Once (SEEM) (cont.)

Proposed by Zou et al., UW-Madison and Microsoft Research, NeurIPS 2023 (arXiv:2304.06718)

SEEM is a promptable panoptic segmentation model designed to segment any region, object, or stuff in an image using various input modalities such as text, points, boxes, or masks.

- **Unified Segmentation:** Supports instance, semantic, and panoptic segmentation.
- **Versatility:** Works with any image, any resolution, and enables zero-shot segmentation.
- **Multimodal Prompts:** Accepts prompts in multiple forms, including text, points, boxes, and masks.
- **Video and Open Vocabulary:** Capable of segmenting video frames and handling open vocabulary tasks.
- **Foundation for Interactive Systems:** Serves as a foundation for real-time, interactive vision systems.

Segment Everything Everywhere All at Once (SEEM) (cont.)

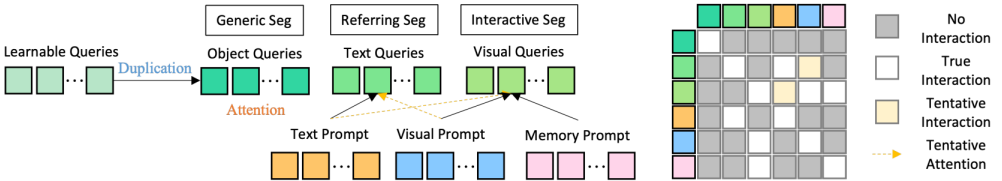


(a) Model Architecture (b) Human-Model Interaction

Figure 2: Overview of *SEEM*- Decoder. (a) *SEEM* encodes image, text, and human inputs into *joint visual-semantic space* as queries, features, and prompts, and then decodes queries to class and mask embeddings. (b) With the benefit of *SEEM* decoder, the machine loop enables memorizing history mask information, and the human loop provides new corrections to the next round.

Zou et al., "Segment Everything Everywhere All at Once," NeurIPS 2023.

Segment Everything Everywhere All at Once (SEEM) (cont.)



(a) Queries and Prompt Interaction

(b) Self-Attention Mask

Figure 3: Queries and prompt interaction during training and evaluation. (a) Learnable queries are duplicated as object, grounding, and visual queries with the same set of weights for each task. (b) Attention mask between any two kinds of tokens (denoted as qpm in Algorithm. 1). Tentative means the interaction is not trained but able to do inference without any modification.

Zou et al., "Segment Everything Everywhere All at Once," NeurIPS 2023.

Segment Everything Everywhere All at Once (SEEM) (cont.)

Algorithm 1: Pseudo code for SEEM.

```
# Inputs: Image(img)[B,3,H,W]; Pos_Mask(pm), Neg_Mask(nm)[B,1,H,W]; Text(txt)[abc...];
# Variables: Learnable Queries(Q_h); Attention Masks between Q and P(qpm)
# Functions: Img_Encoder(),Text_Encoder(),Visual_Sampler(),feature_attn(),prompt_attn(),output();

def init( ):
    Q_o, Q_t, Q_v = Q_h.copy();# Initialize object, text and visual queries.
    F_v, P_t = Img_Encoder(img), Text_Encoder(txt);# F_v and P_t denote image feature, text
      prompt.
    P_v = Visual_Sampler(F_v, pm, nm);# Sample visual prompt from image feature, pos/neg
      mask.
def SEEM_Decoder(F_v, Q_o, Q_t, Q_v, P_v, P_t, P_m):
    Q_o, Q_t, Q_v = feature_attn(F_v, Q_o, Q_t, Q_v);# Cross attend queries with image features.
    Q_o, Q_t, Q_v = prompt_attn(qpm, Q_o, Q_t, Q_v, P_v, P_t, P_m);# Self attend queries and prompts.
    O_m, O_c, P_m = output(F_v, Q_o, Q_t, Q_v);# Compute mask and class outputs.
def forward(img,pm,nm,txt):
    F_v, Q_o, Q_t, Q_v, P_v, P_t = init(); P_m = None;# Initialize variables.
    for i in range(max_iter):
        O_m, O_c, P_m = SEEM_Decoder(F_v, Q_o, Q_t, Q_v, P_v, P_t, P_m)
```

Zou et al., "Segment Everything Everywhere All at Once," NeurIPS 2023.



Figure 4: Click/scribble-based segmentation. *SEEM* supports arbitrary formats of clicks or scribbles by users. Moreover, it simultaneously gives the semantic label for the segmented mask, which is not possible in SAM [36].

Zou et al., "Segment Everything Everywhere All at Once," NeurIPS 2023.

SEEM - Results (cont.)



Figure 5: Text to mask or text referring segmentation. The referred text is shown on the masks. *SEEM* adapts to various types of input images in the domain of cartoons, movies, and games.

Zou et al., "Segment Everything Everywhere All at Once," NeurIPS 2023.

SEEM - Results (cont.)

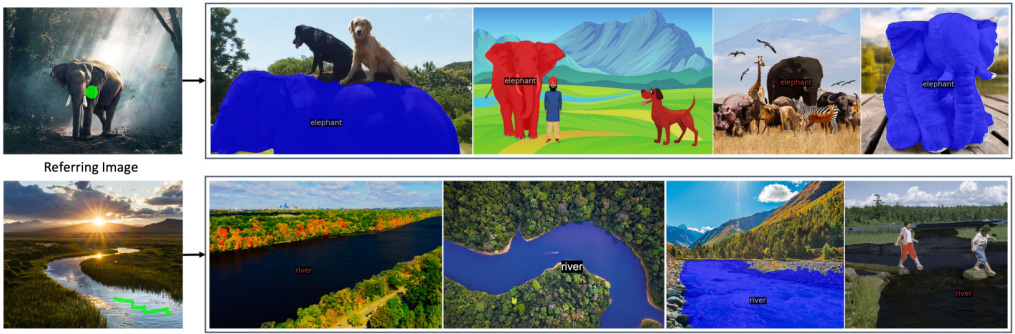


Figure 6: **Zero-shot** visual referring segmentation with *SEEM*. Given a referring image with simple spatial hints, *SEEM* can segment the regions which are semantically similar in different target images.

Zou et al., "Segment Everything Everywhere All at Once," NeurIPS 2023.

SEEM - Results (cont.)

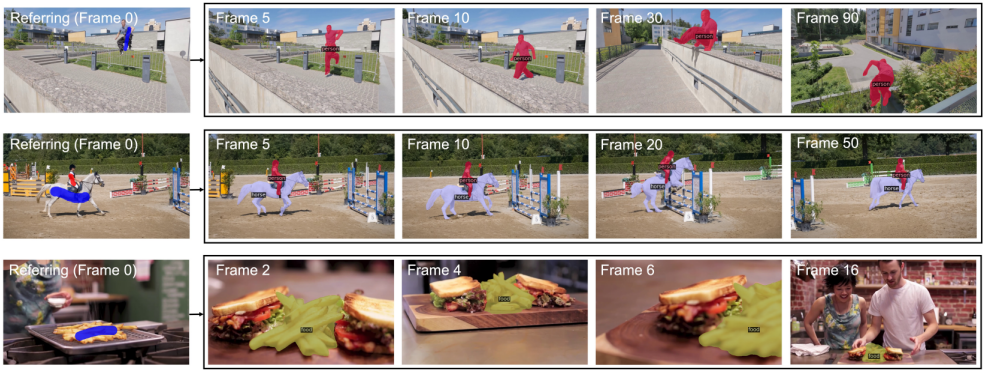


Figure 7: **Zero-shot** video object segmentation using the first frame plus one stroke. From top to bottom, the videos are “parkour" and “horsejump-low” from DAVIS [73], and video 101 from YouCook2 [74]. *SEEM* precisely segments referred objects even with significant appearance changes caused by blurring or intensive deformations.

Zou et al., “Segment Everything Everywhere All at Once,” NeurIPS 2023.

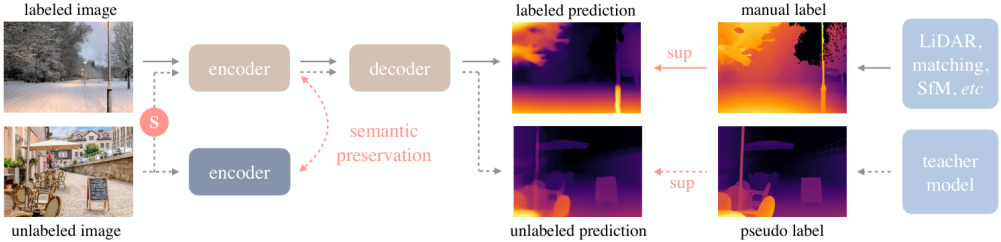
SEEM - Results (cont.)

Depth Estimation Foundation Model: Depth Anything

- Foundation model for single image depth estimation.
- Trained with a large-scale teacher–student pipeline:
  - $\sim$1.5M labeled images + $\sim$62M unlabeled images
- Encoder priors from DINOv2 improve cross-domain geometric understanding.
- Robust transfer across indoor/outdoor domains with limited task-specific supervision.
- Depth Anything V2 (2024) trains the teacher on synthetic images with exact labels, improving fine detail and robustness.
- Applications:
  - Robotics and autonomous driving
  - Virtual/augmented reality
  - 3D scene understanding and reconstruction

How Depth Anything Works

- **Teacher–student pipeline** on labeled depth plus large-scale unlabeled RGB: pseudo-labels from the teacher.
- DINOv2-style encoder + prediction head maps RGB to per-pixel depth.
- Outputs dense depth maps that transfer to segmentation and other geometric downstream tasks.



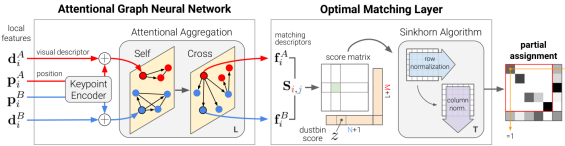
Yang et al., "Depth Anything," CVPR 2024 (OpenAccess).

SuperGlue: Reusable Matching for Geometric Vision

- Combines graph neural networks and attention to find matches between sets of keypoints in different images.
- Generalizes across scenes and keypoint detector types.
- Applications: Visual localization and mapping, 3D reconstruction, Object pose estimation.
- A caveat on terminology: SuperGlue (2020) predates the foundation-model era and is task-specific; it appears here as the standard *reusable component* for matching, a role foundation models now play for other tasks.

SuperGlue: Model Architecture

- Two main components: an attentional graph neural network (GNN) and an optimal matching layer.
- GNN uses keypoint encoder to combine each keypoint's position and visual descriptor into single vector.
- Alternating self-attention and cross-attention layers create powerful feature representations.
- The optimal matching layer builds an $M \times N$ score matrix between keypoints of both images, appends additional rows/columns ("dustbins") for potential unmatched keypoints.
- Uses Sinkhorn algorithm for T iterations to find optimal partial matching assignment.

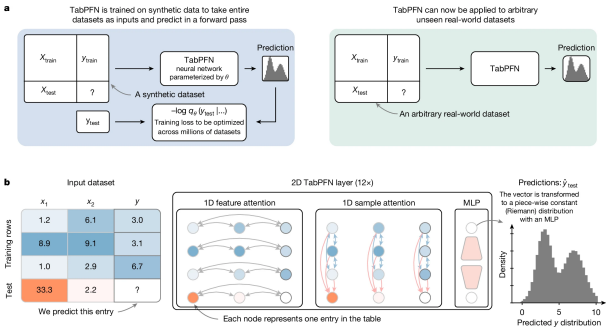


Sarlin et al., "SuperGlue: Learning Feature Matching with Graph Neural Networks," CVPR 2020 (OpenAccess).

TabPFN: Prior-Data Fitted Network

- Trained to "meta-learn" classification solutions for tabular datasets; pretraining uses millions of **synthetic** datasets, never real data.
- A new dataset is handled by **in-context learning**: the training rows go in as context, and predictions come out in one forward pass, in under a second.
- Zero-shot predictions for new, small tabular datasets.
- TabPFN v2 (published in *Nature*, 2025) extends this to regression, missing values, and larger tables.
- Applications: Science data analysis, health records, finance.

TabPFN: Architecture and Capabilities



Hollmann et al., "Accurate Predictions on Small Data with a Tabular Foundation Model," *Nature*, 2025, Fig. 1 (doi:10.1038/s41586-024-08328-6)

- Learns a prior over tabular tasks by pretraining on large synthetic task distributions.
- Performs in-context learning on a new dataset in one forward pass.
- Two-way attention captures dependencies across both rows and features.

Medical Foundation Models: TRIBE v2

- Model of brain-response dynamics.
- Frozen encoders for video/audio/text + transformer integration backbone.
- Supports zero-shot transfer to unseen subjects/tasks at population level.

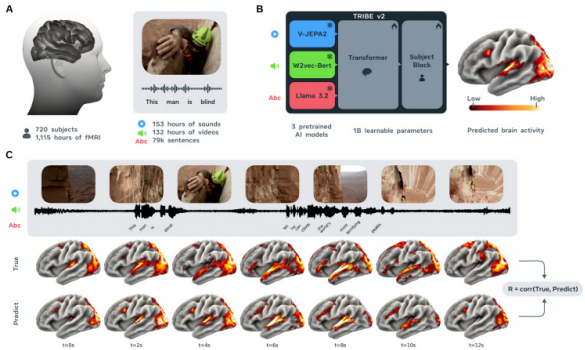


Figure from Meta AI's TRIBE v2 announcement: data → pretrained embeddings → predicted cortical response vs. measured mean response

TRIBE v2: Encoding Performance (Results)

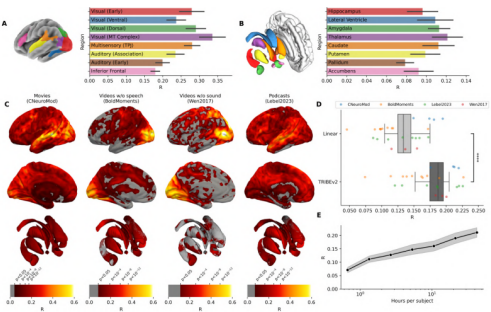


Figure from Meta AI's TRIBE v2 announcement: Cortical/subcortical encoding scores; vs. linear baseline; scaling with training data.

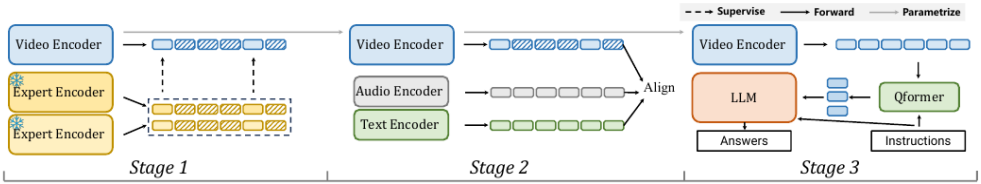
- **Accurate whole-brain prediction:** TRIBE v2 accurately predicts fMRI responses across the entire brain.
- **Cortical encoding scores:** Achieves high encoding scores in key cortical regions.
- **Subcortical encoding performance:** Robust encoding in 8 subcortical regions.
- **Per-subject overview:** Individual subject encoding scores, with dots for TRIBE v2.
- **Scaling laws:** Demonstrates average encoding scores on the Algonauts dataset improve as more training data hours per participant are used.

Video Foundation Model: InternVideo2

- Large-scale pre-trained video foundation model (InternVideo family; InternVideo2, Wang et al., 2024).
- Developed to generalize across diverse video understanding tasks by learning rich spatio-temporal features adaptable to unseen video domains.
- Capabilities: Recognition, localization, captioning, segmentation, search, QA, and multi-modal video reasoning.
- Complements V-JEPA 2 (covered earlier): same goal of general video representations, but built from masked reconstruction plus contrastive alignment rather than latent prediction.

InternVideo2: How It Works

- Stage 1: Spatiotemporal perception (masked reconstruction)
- Stage 2: Semantic alignment (video-text/audio contrastive learning)
- Stage 3: World-modeling style reasoning (next-token objectives)



Wang et al., InternVideo2 (arXiv:2403.15377), Fig. 2: the three consecutive training stages.

Limitations and a Critical Perspective

- **Homogenization risk:** when many applications share one backbone, they inherit the same failure modes and biases (Bommasani et al.).
- **Data provenance:** web-scale corpora raise licensing, privacy, and bias questions; curation (DINOv2's LVD-142M) and licensed collection (SA-1B) are partial answers, not full solutions.
- **Evaluation:** zero-shot numbers depend on prompt engineering, and at web scale it is hard to rule out that benchmark images appeared in training data.
- **Compute concentration:** pretraining budgets limit who can build these models; most practitioners work at the **adaptation** layer (probes, LoRA, prompting).
- **Task coverage:** strong global features do not guarantee strong *dense* features (segmentation, depth); DINOv3's gram anchoring targets exactly this gap.
- **Source bias:** most of today's headline vision foundation models come from a handful of industry labs; treat their benchmark selections critically.

Summary and Discussion

- Foundation models are reshaping AI across all domains (vision, language, tabular, medicine, video).
- Key trends: Scalability, data diversity, task generalization, multi-modal grounding.
- Challenges: Compute costs, ethical issues, responsible deployment, domain adaptation.
- **Discussion:**
  - How will foundation models change your research?
  - What domains remain out-of-reach for today's foundation models?

Hands-On: Code, Checkpoints, and Exercise

- **Hugging Face `transformers`:** CLIP (`openai/clip-vit-base-patch32`, zero-shot classification pipeline), SAM (`facebook/sam-vit-huge`), DINOv2 (`facebook/dinov2-base`), Depth Anything V2, and Grounding DINO (`IDEA-Research/grounding-dino-tiny`) all load in a few lines.
- **Official repos:** facebookresearch/segment-anything, facebookresearch/sam2, IDEA-Research/GroundingDINO, PriorLabs/TabPFN.
- **Suggested exercise:** build an open-vocabulary segmentation pipeline: Grounding DINO turns a text prompt into boxes, SAM turns the boxes into masks. Compare against a DINOv2 encoder with a linear head on the same images.

References

1. Bommasani et al., “On the Opportunities and Risks of Foundation Models.” arXiv:2108.07258.
2. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision (CLIP),” ICML 2021. arXiv:2103.00020.
3. Oquab et al., “DINOv2: Learning Robust Visual Features without Supervision,” TMLR 2024. arXiv:2304.07193.
4. Siméoni et al., “DINOv3,” 2025. arXiv:2508.10104.
5. Liu et al., “Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection,” ECCV 2024. arXiv:2303.05499.
6. Kirillov et al., “Segment Anything,” ICCV 2023. arXiv:2304.02643.
7. Ravi et al., “SAM 2: Segment Anything in Images and Videos,” ICLR 2025. arXiv:2408.00714.
8. Carion et al. (Meta), “SAM 3: Segment Anything with Concepts,” 2025. arXiv:2511.16719 (SAM 3.1 update: Meta AI blog).
9. Zou et al., “Segment Everything Everywhere All at Once” (SEEM), NeurIPS 2023. arXiv:2304.06718.
10. Yang et al., “Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data,” CVPR 2024. arXiv:2401.10891.
11. Yang et al., “Depth Anything V2,” NeurIPS 2024. arXiv:2406.09414.
12. Sarlin et al., “SuperGlue: Learning Feature Matching with Graph Neural Networks,” CVPR 2020.

References (cont.)

13. Hollmann et al., "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second," ICLR 2023. arXiv:2207.01848.
14. Hollmann et al., "Accurate Predictions on Small Data with a Tabular Foundation Model," *Nature* 637:319–326, 2025. doi:10.1038/s41586-024-08328-6 (source of the TabPFN architecture figure).
15. Prior Labs, TabPFN resources. https://github.com/PriorLabs/TabPFN.
16. Meta AI, "Introducing TRIBE v2: A Predictive Foundation Model Trained to Understand How the Human Brain Processes Complex Stimuli" (blog). https://ai.meta.com/blog/tribe-v2-brain-predictive-foundation-model/; d'Ascoli et al., "TRIBE: TRImodal Brain Encoder for Whole-Brain fMRI Response Prediction," 2025. arXiv:2507.22229.
17. Wang et al., "InternVideo2: Scaling Foundation Models for Multimodal Video Understanding," ECCV 2024. arXiv:2403.15377.
18. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," ICLR 2022. arXiv:2106.09685.
19. Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models," CVPR 2022.
20. Alayrac et al., "Flamingo: a Visual Language Model for Few-Shot Learning," NeurIPS 2022.
21. Liu et al., "Visual Instruction Tuning (LLaVA)," NeurIPS 2023.
22. Ramesh et al., "Zero-Shot Text-to-Image Generation (DALL-E)," ICML 2021.
23. Schuhmann et al., "LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models," NeurIPS 2022. OpenCLIP: https://github.com/mlfoundations/open_clip.

References (cont.)

24. Yeung-Levy, S., Stanford BIODS 276: Advanced Topics in Computer Vision and Biomedicine, Lecture 1 (source of the history-arc and taxonomy figures).
https://web.stanford.edu/class/biods276/lectures/lecture1.pdf