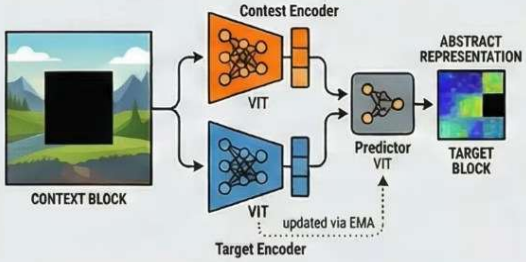
The JEPA Evolution: From Images to Multimodal AI

I-JEPA: Mastering Static Images

Learns Semantic Features from Images

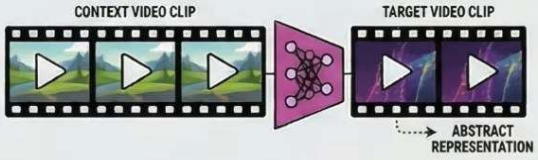


Predicting Abstract Representations: The model sees a large part of an image (context block) and is trained to predict the abstract representations of several missing parts (target blocks).

Prediction in Embedding Space, Not Pixel Space

V-JEPA: Stepping into Video

Extending Prediction to the Time Dimension



Context-to-Target Video Prediction: Similar to its image-based predecessor, V-JEPA uses a context video clip to predict the representations of a target video clip.

A Foundational Component for Multimodality

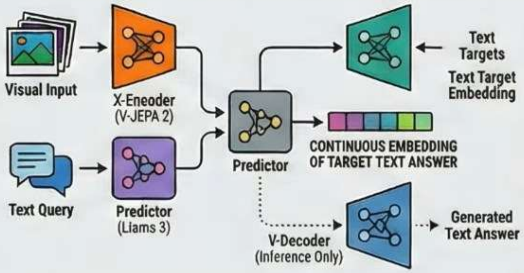
A specialized version, V-JEPA 2, serves as the powerful visual encoder within the more advanced VL-JEPA system.

COMPARISON ACROSS ARCHITECTURES

Feature	I-JEPA (Images)	V-JEPA (Video)	VL-JEPA (Multimodal)
Primary Bomaia	Static Images	Video	Visioo < Language
Input	Image Context Block	Video Context	Visual Input > Text Query
Prediction Target	Image Target Block Representation	Video Target Representation	Text Target Embedding
Core Goal	Semantly image Representation	Video Representation Learning	Multimodal Understanding & Generation
Generative Training?	No (Pore Representation Learning)	No (Pore Representation Learning)	No (Predicts Embeddings)
Loss Function	L2 Distance	Embedding Space	InfoNCE (Contrastive)
Key Efficiency	No data aogmentations needed	Efficient video encoding	Selective decoding (2.85x fewer operations)

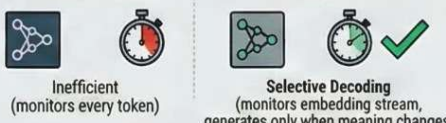
VL-JEPA: Unifying Vision and Language

A Multimodal Architecture for Vision-Language Tasks



A Modular, Four-Part System: Consists of an X-Encoder for vision, a Y-Encoder for text targets, a Predictor for the core task, and a Y-Decoder used only at inference to generate text.

Efficiency Through Selective Decoding



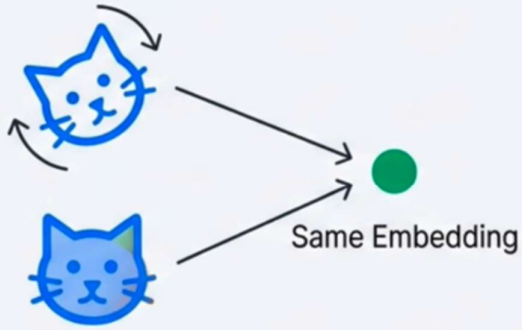
An Architectural Evolution: Tracing the JEPA Trajectory

A comparative analysis of I-JEPA, V-JEPA, and VL-JEPA, exploring the shift from pixel reconstruction to semantic prediction in self-supervised learning.

I-JEPA (Image) V-JEPA (Video) VL-JEPA (Vision + Language)

Self-Supervised Learning at a Crossroads

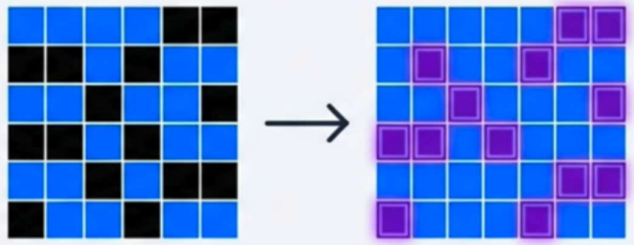
The Path of Invariance



Core Idea: Learn by producing similar embeddings for different 'views' of the same image (e.g., crops, color jitter).

- **Pro:** Learns high-level semantic features.
- **Con:** Relies on hand-crafted data augmentations that introduce strong biases and don't generalize well across modalities.

The Path of Reconstruction



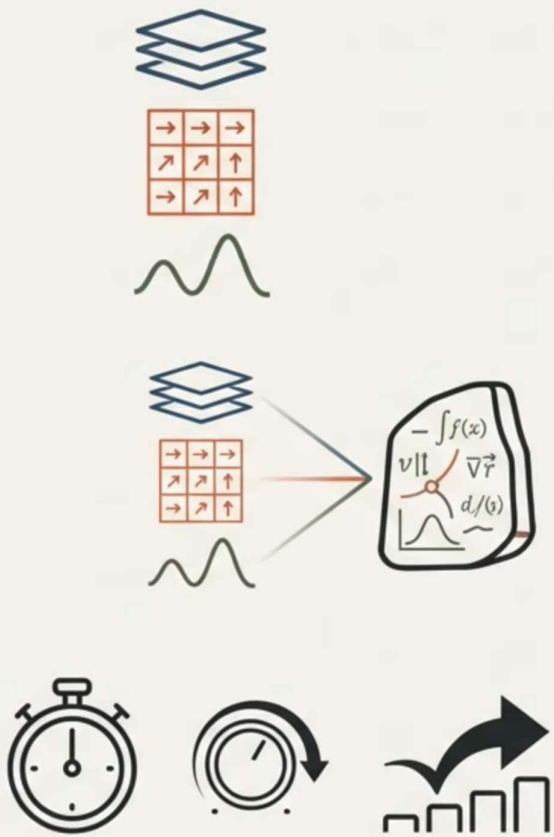
Pixel-Level Inpainting

Core Idea: Learn by reconstructing missing pixels or tokens from a corrupted input (e.g., Masked Autoencoders).

- **Pro:** Requires less prior knowledge and generalizes easily.
- **Con:** Focuses on low-level, pixel-perfect details, often missing the semantic forest for the trees. Computationally expensive.

Must we choose between semantic brittleness and pixel-level obsession?

Our Journey to a Unified Theory



The Three Paths

We will explore the origins and mechanics of the three primary formulations of diffusion:

- **The Variational View**: Generation as layered denoising (DDPM).
- **The Score-Based View**: Guiding samples through a probability landscape (Score SDE).
- **The Flow-Based View**: Transporting distributions along a deterministic path (Flow Matching).

The Grand Unification

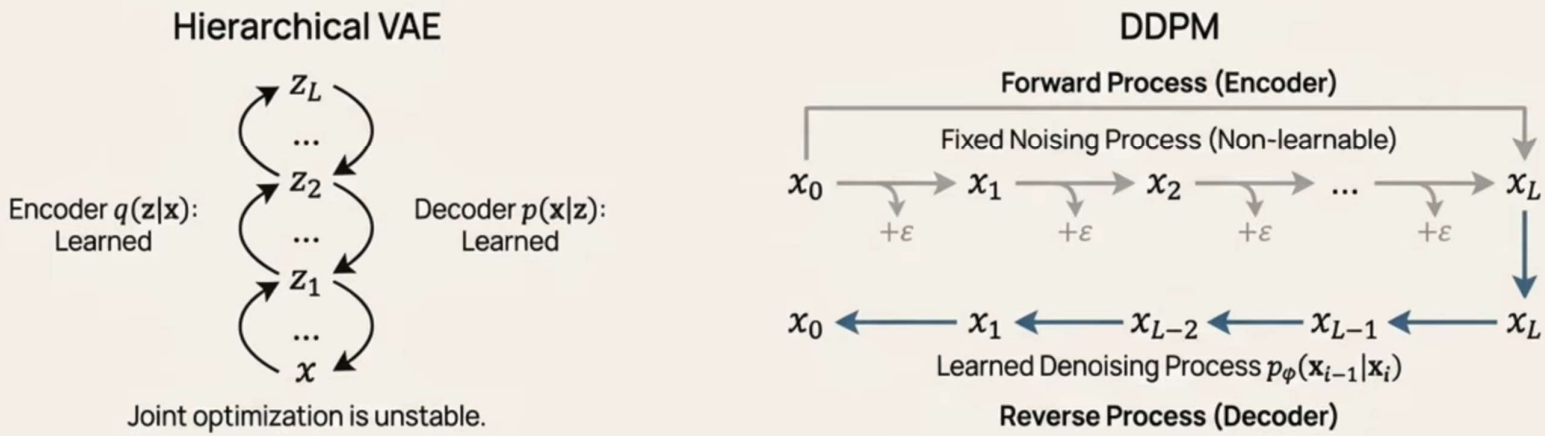
We will reveal the deep **mathematical** equivalence connecting these views, showing they all learn the same underlying object: a time-dependent vector field.

The Power of Unity

We will demonstrate how this unified understanding leads to practical breakthroughs, from accelerating inference to creating next-generation, 'from-scratch' generative models.

Path 1: The Variational View – Generation as Layered Denoising

This perspective, rooted in Variational Autoencoders (VAEs), models generation as a sequence of denoising steps. DDPMs refine this by fixing the encoder to a simple noising process and focusing all learning on a powerful, iterative decoder.



- **The Goal:** Maximize the Evidence Lower Bound (ELBO) on the data log-likelihood, just like a VAE.

$$-\log p_\varphi(\mathbf{x}_0) \leq -L_{\text{LBO}}$$

- **The Objective:** The DDPM objective decomposes the ELBO into a series of simple denoising tasks.

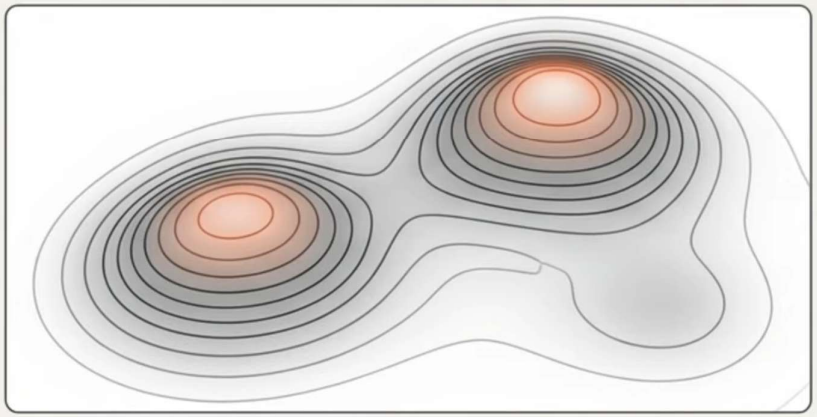
$$\mathcal{L}_{\text{diffusion}}(\mathbf{x}_0;\phi) = \sum \mathbb{E}_{p(\mathbf{x}_i|\mathbf{x}_0)}\left[D_{\text{KL}}\left(p(\mathbf{x}_{i-1}|\mathbf{x}_i,\mathbf{x}_0) \parallel p_\varphi(\mathbf{x}_{i-1}|\mathbf{x}_i)\right)\right]$$

- **The Core Mechanism:** The model learns to predict the noise ϵ added at each step, $\epsilon_\varphi(\mathbf{x}_i, i)$, allowing it to reverse the process. The training objective simplifies to a mean squared error on this noise prediction.

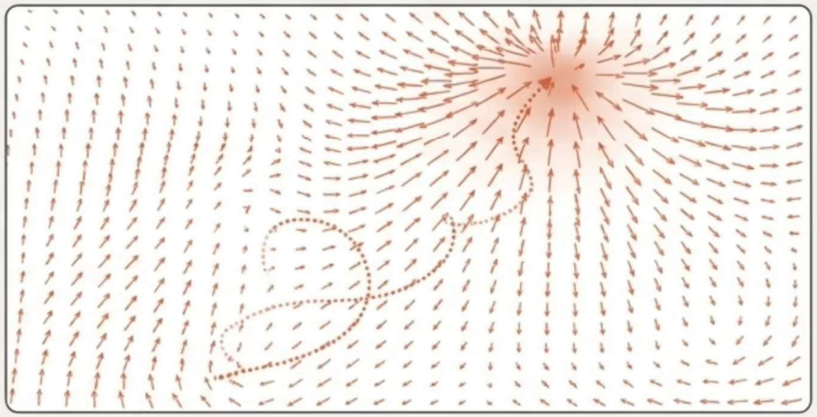
Path 2: The Score-Based View – Navigating a Probability Landscape

Core Idea: Originating from Energy-Based Models (EBMs), this view defines a probability distribution by its gradient field, $\nabla_x \log p(\mathbf{x})$, known as the "score." The model learns this score field, which points towards regions of higher data density, and generation is framed as "climbing" the probability landscape.

Probability Density $p(\mathbf{x})$



Score Field $\nabla_x \log p(\mathbf{x})$

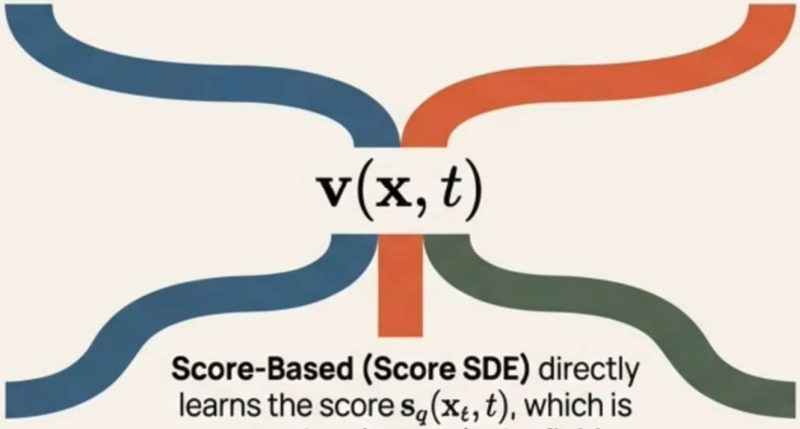


Key Technical Details:

- The Goal: Match the model's score function $s_q(x,t)$ to the true data score $\nabla_x \log p_t(x_t)$.
- The Challenge: The true score is intractable. The solution is **Denoising Score Matching (DSM)**, which perturbs data with noise $p_o(\tilde{x}|x)$ and learns the score of the noised distribution $\nabla_{\tilde{x}} \log p_o(\tilde{x})$.
- The Core Mechanism: The process is formalized as a Stochastic Differential Equation (SDE). Generation involves reversing this SDE, using the learned score s_q to guide the drift term.
  - Forward SDE: $dx = f(x, t)dt + g(t)dw$ (noises data)
  - Reverse SDE: $dx = [f(x, t) - g^2(t)s_q(x, t)]dt + g(t)d\bar{w}$ (generates data)

The Grand Unification: Three Languages, One Underlying Principle

Despite their different origins and mathematical formulations, all three perspectives converge on a single task: **learning a time-dependent vector field to reverse a forward noising process.**



Variational (DDPM) learns $p_q(\mathbf{x}_{i-1}|\mathbf{x}_i)$, which implicitly defines a denoising vector field.

Score-Based (Score SDE) directly learns the score $\mathbf{s}_q(\mathbf{x}_t, t)$, which is proportional to a velocity field.

Flow-Based (Flow Matching) directly learns the velocity field $\mathbf{v}_q(\mathbf{x}_t, t)$.

The Shared Challenge: Intractable Marginal Targets

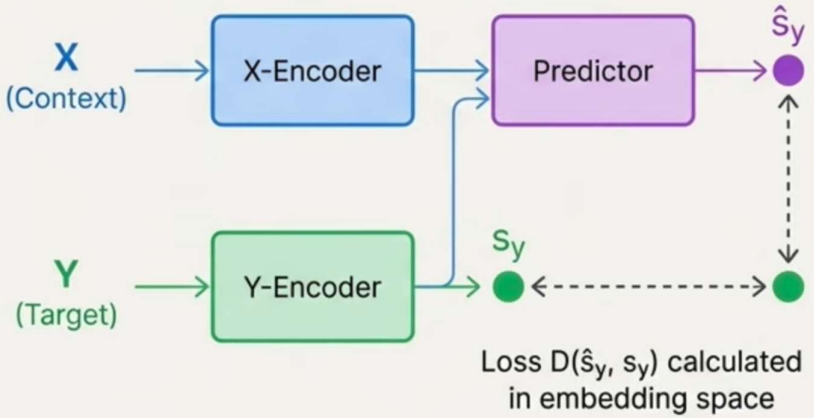
The 'oracle' training objective for each view requires access to the marginal distribution $p_t(\mathbf{x}_t)$, which is an intractable integral over the unknown data distribution p_{data}.

- **JKL (Variational)**: $D_{\text{KL}}(p(\mathbf{x}_{t-1}|\mathbf{x}_t) \,||\, p_q(\mathbf{x}_{t-1}|\mathbf{x}_t))$ requires $p(\mathbf{x}_t)$.
- **JSM (Score)**: $||\mathbf{s}_q(\mathbf{x}_t, t) - \nabla \log p_t(\mathbf{x}_t)||^2$ requires $\nabla \log p_t(\mathbf{x}_t)$.
- **JFM (Flow)**: $||\mathbf{v}_q(\mathbf{x}_t, t) - \mathbf{v}_t(\mathbf{x}_t)||^2$ requires $\mathbf{v}_t(\mathbf{x}_t)$.

How can three seemingly different frameworks all overcome the same fundamental obstacle?

A Third Way: Predicting in an Abstract World

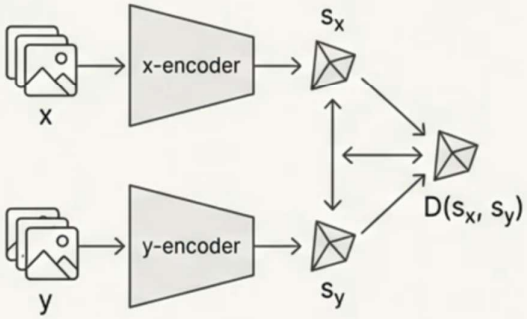
The Joint-Embedding Predictive Architecture (JEPA) learns by predicting the representations of missing data, not the raw data itself.



- **Eliminates Waste**: By ignoring pixel-level details, the model focuses only on learning high-level, semantic features.
- **More Efficient**: Prediction in a compact representation space is far less computationally demanding than reconstructing high-dimensional pixel space.
- **No Hand-Crafted Bias**: Learns powerful representations without relying on brittle, modality-specific data augmentations.

The Landscape of Self-Supervised Learning

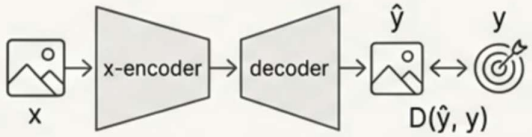
Invariance-Based (e.g., DINO)



Learns by producing similar embeddings for augmented views.

Limitation: Relies on hand-crafted augmentations that introduce strong biases.

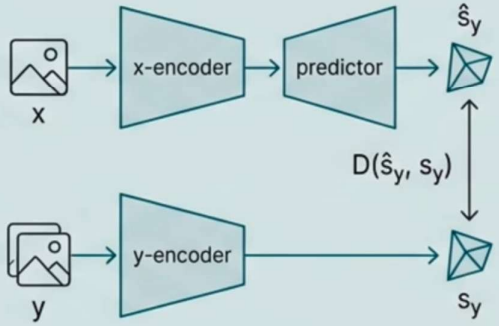
Generative (e.g., MAE)



Learns by reconstructing masked portions of an input (pixels).

Limitation: Computationally expensive, predicting every low-level, semantically irrelevant detail.

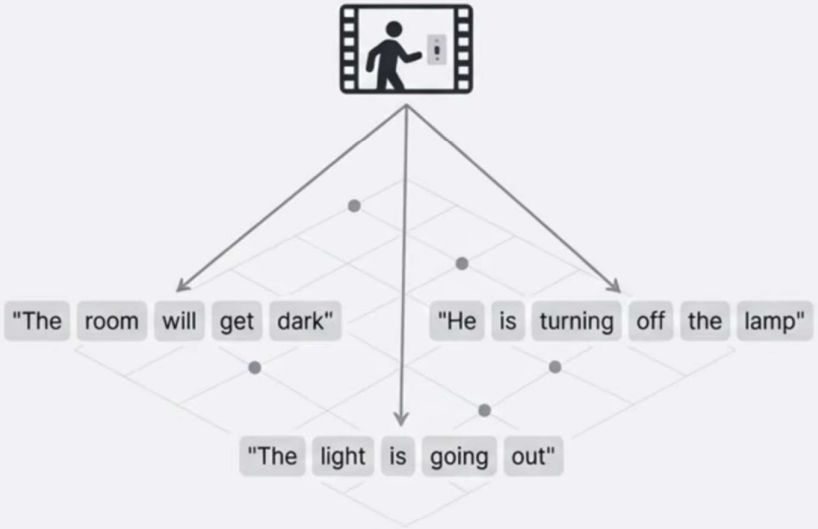
JEPA: The Third Way



Learns by predicting the **representations** of missing data in an abstract embedding space, not the raw pixels.

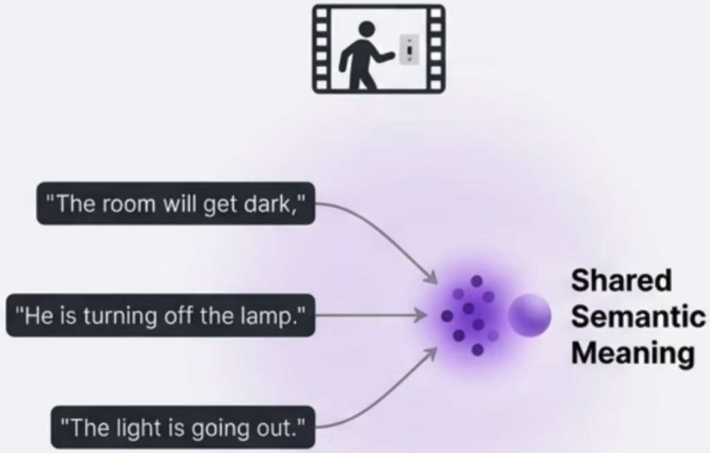
Predicting the Idea, Not the Words

Token Prediction (VLM)



Models must learn to predict many different, almost orthogonal token sequences for the same semantic meaning. This is inefficient.

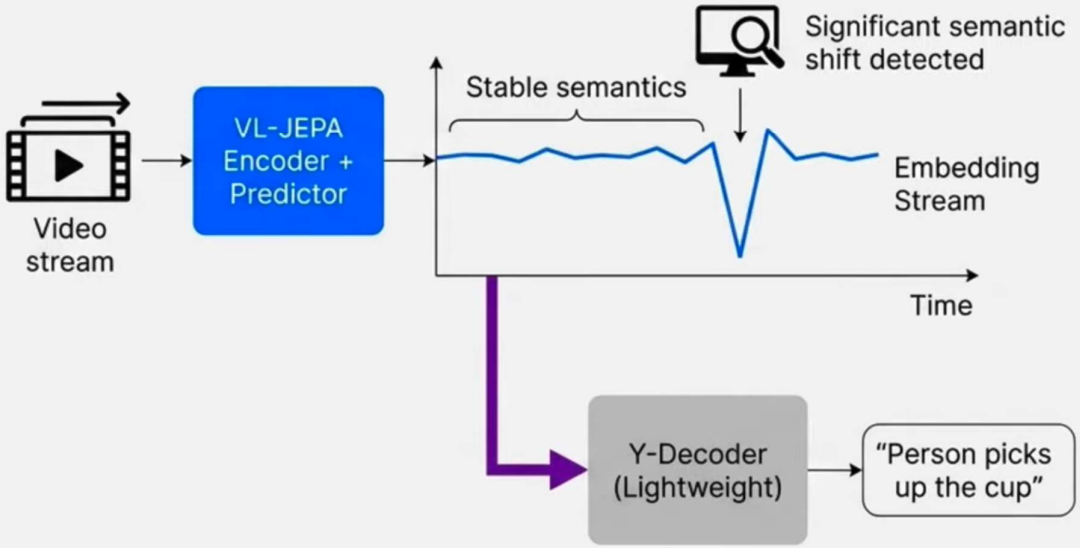
Embedding Prediction (VL-JEPA)



VL-JEPA learns a simpler target. All plausible answers map to a nearby point in embedding space, making learning more efficient and robust.

A Breakthrough in Efficiency: Selective Decoding

Real-time video applications require continuous understanding, but constant text generation is too slow and expensive.



2.85x

Fewer Decoding Operations

VL-JEPA achieves the same performance as uniform decoding with nearly 3x less compute by only generating text when the meaning of the scene actually changes.

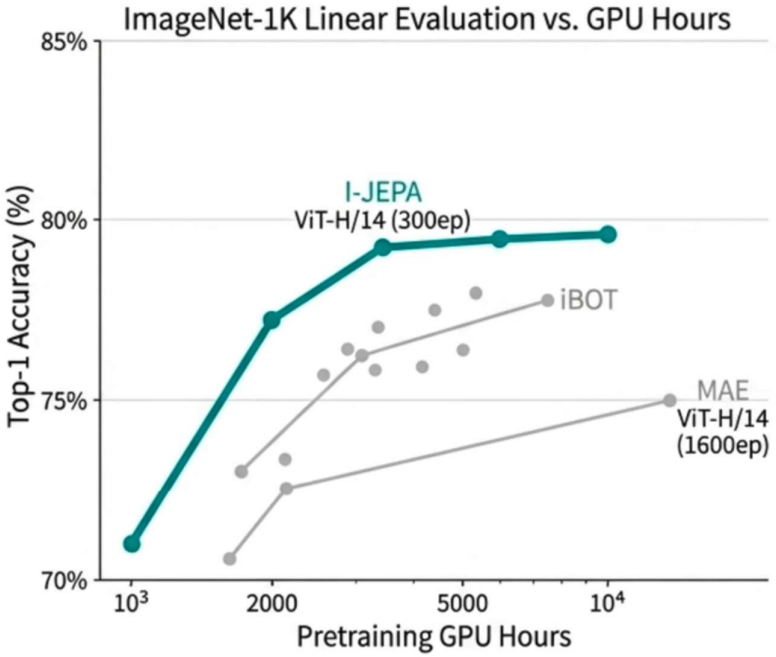
An Efficient Architecture for Semantic Feature Learning

Architectural Deep Dive

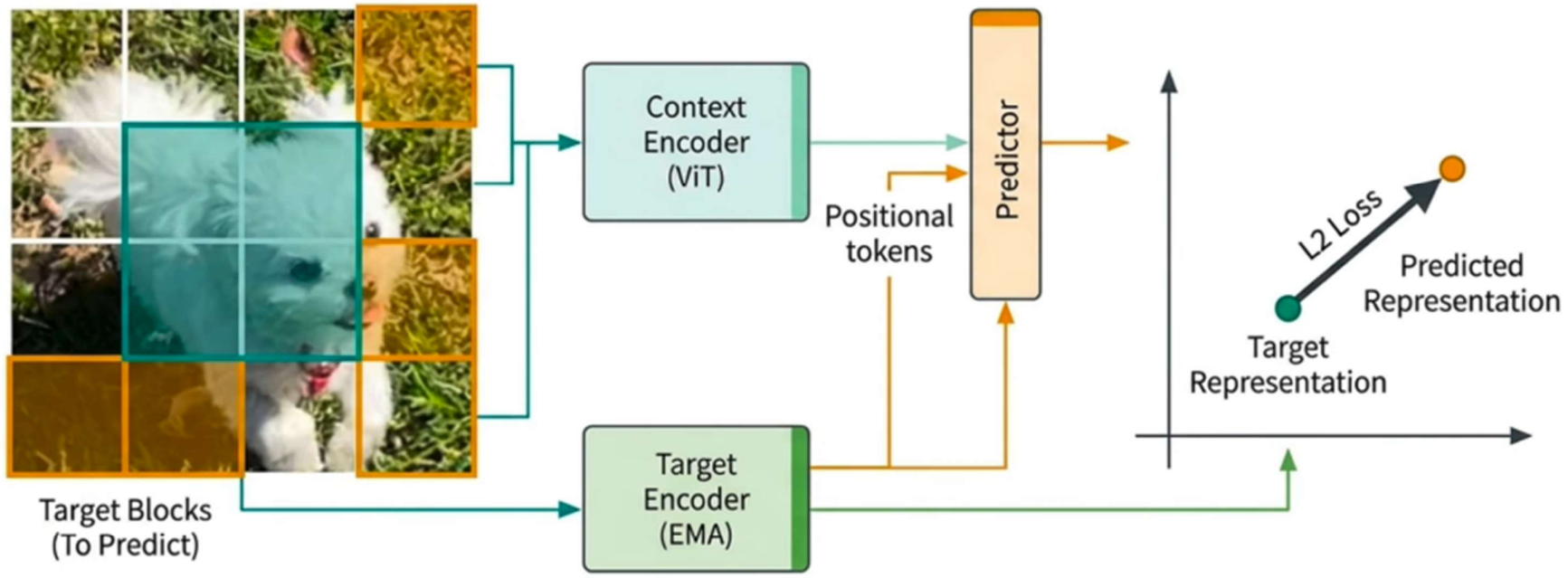
- **Encoders & Predictor**: Both context and target encoders are Vision Transformers (Vit). The predictor is a smaller, lightweight ViT.
- **Target Stability**: The target encoder's weights are an exponential moving average (EMA) of the context encoder's weights, providing a stable prediction target.
- **Loss Function**: A simple L2 distance between predicted and target representations.

The Importance of Masking

Efficiency and Performance



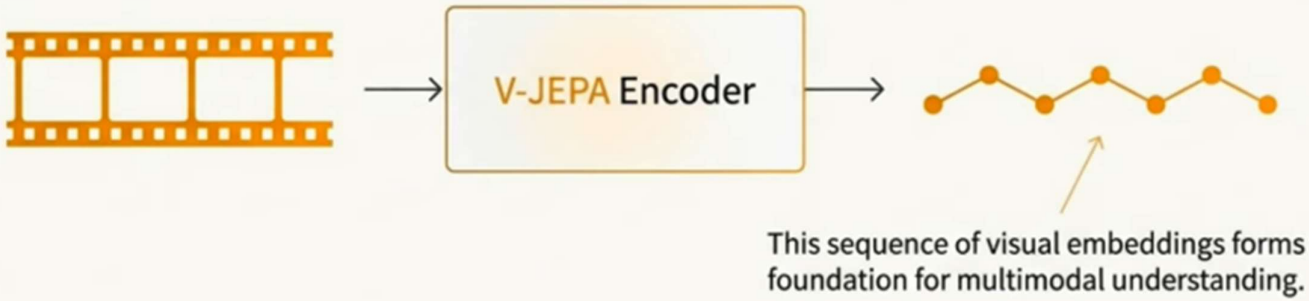
I-JEPA: Predicting Semantic Content in a Static Image



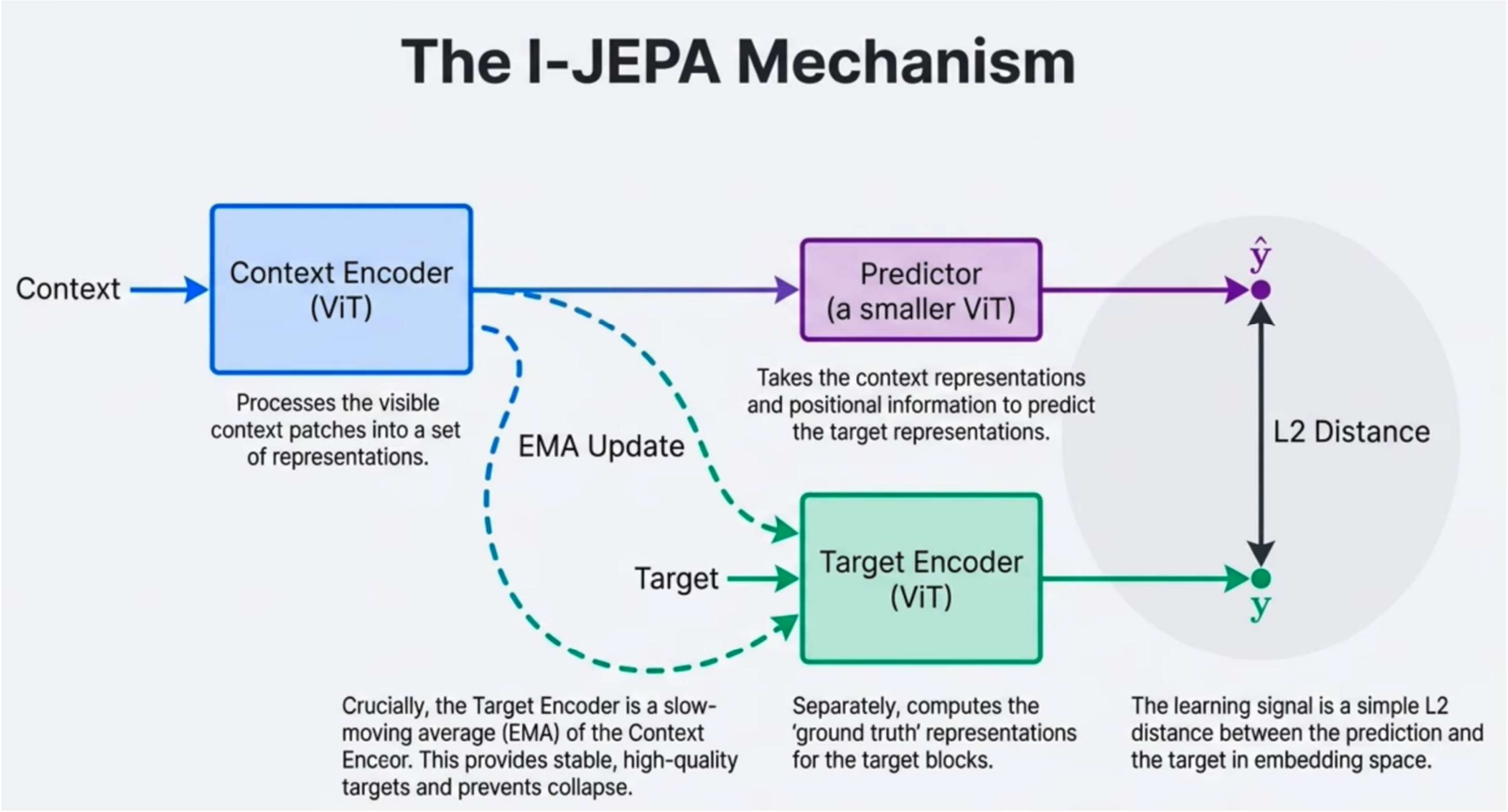
By predicting in **representation space**, the model is forced to learn **semantic features** and avoids the computational waste of pixel-level reconstruction.

V-JEPA: Extending Prediction into the Temporal Domain

V-JEPA is the natural extension of the JEPA framework to video. It learns powerful video representations by predicting the latent representations of future or masked video clips from a given context clip.

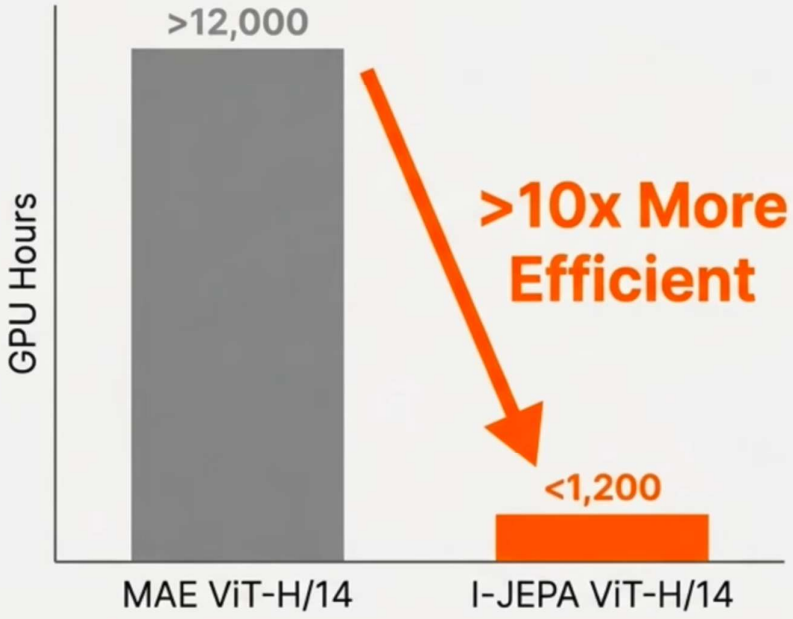


The Bridge to Multimodality: A specific version, **V-JEPA 2**, is used as the frozen vision backbone (X-Encoder) in the VL-JEPA architecture.


The I-JEPA Mechanism
Context
Context Encoder (ViT)
Processes the visible context patches into a set of representations.
Predictor (a smaller ViT)
Takes the context representations and positional information to predict the target representations.
EMA Update
Target
Target Encoder (ViT)
Crucially, the Target Encoder is a slow-moving average (EMA) of the Context Enceor. This provides stable, high-quality targets and prevents collapse.
Separately, computes the 'ground truth' representations for the target blocks.
L2 Distance
The learning signal is a simple L2 distance between the prediction and the target in embedding space.

More Semantic, Less Compute

Pretraining GPU Hours (ImageNet)



Stronger Off-the-Shelf Features

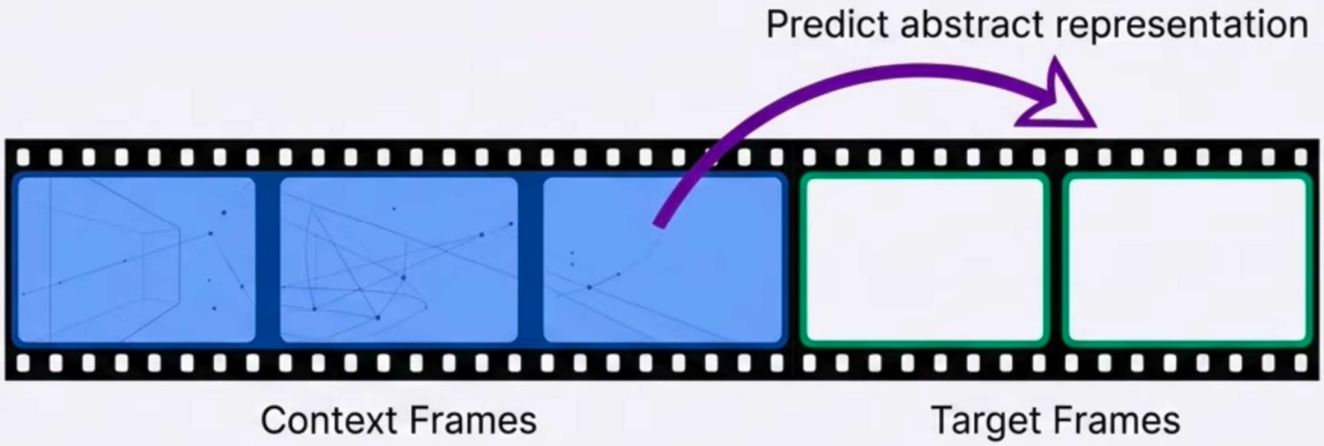
I-JEPA significantly outperforms MAE in linear probing on ImageNet, indicating more semantic representations.

Top-1 Accuracy:
I-JEPA ViT-H/14: **79.3%** vs.
MAE ViT-H/14: 77.2%

No Augmentations Needed

Achieves this performance without any hand-crafted view augmentations, unlike invariance-based methods.

Extending Prediction into the Fourth Dimension: V-JEPA

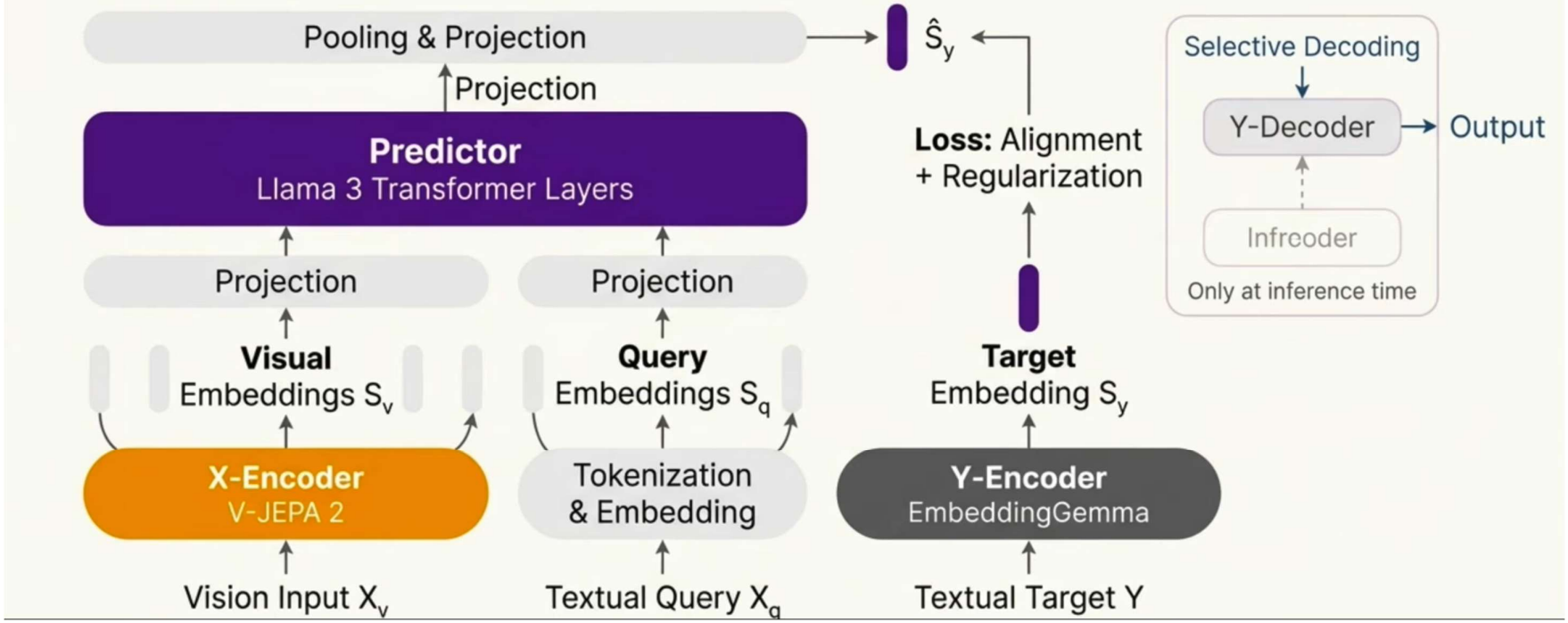


The JEPA principle is not limited to static images. V-JEPA learns powerful video representations by predicting the future in an abstract, semantic space.

While a powerful standalone model, V-JEPA's key evolution is serving as the visual backbone—the **X-Encoder**—for a more ambitious architecture.

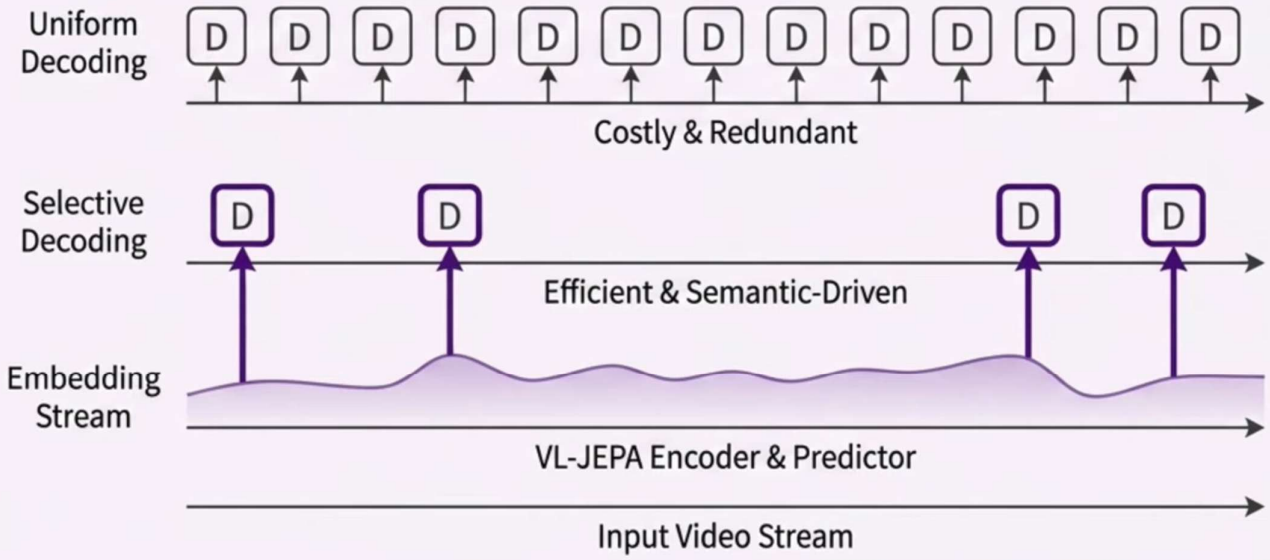
VL-JEPA: A Unified Architecture for Vision and Language

To create a non-generative model for general vision-language tasks. It takes a visual input and a textual query to predict the continuous embedding of the target text.



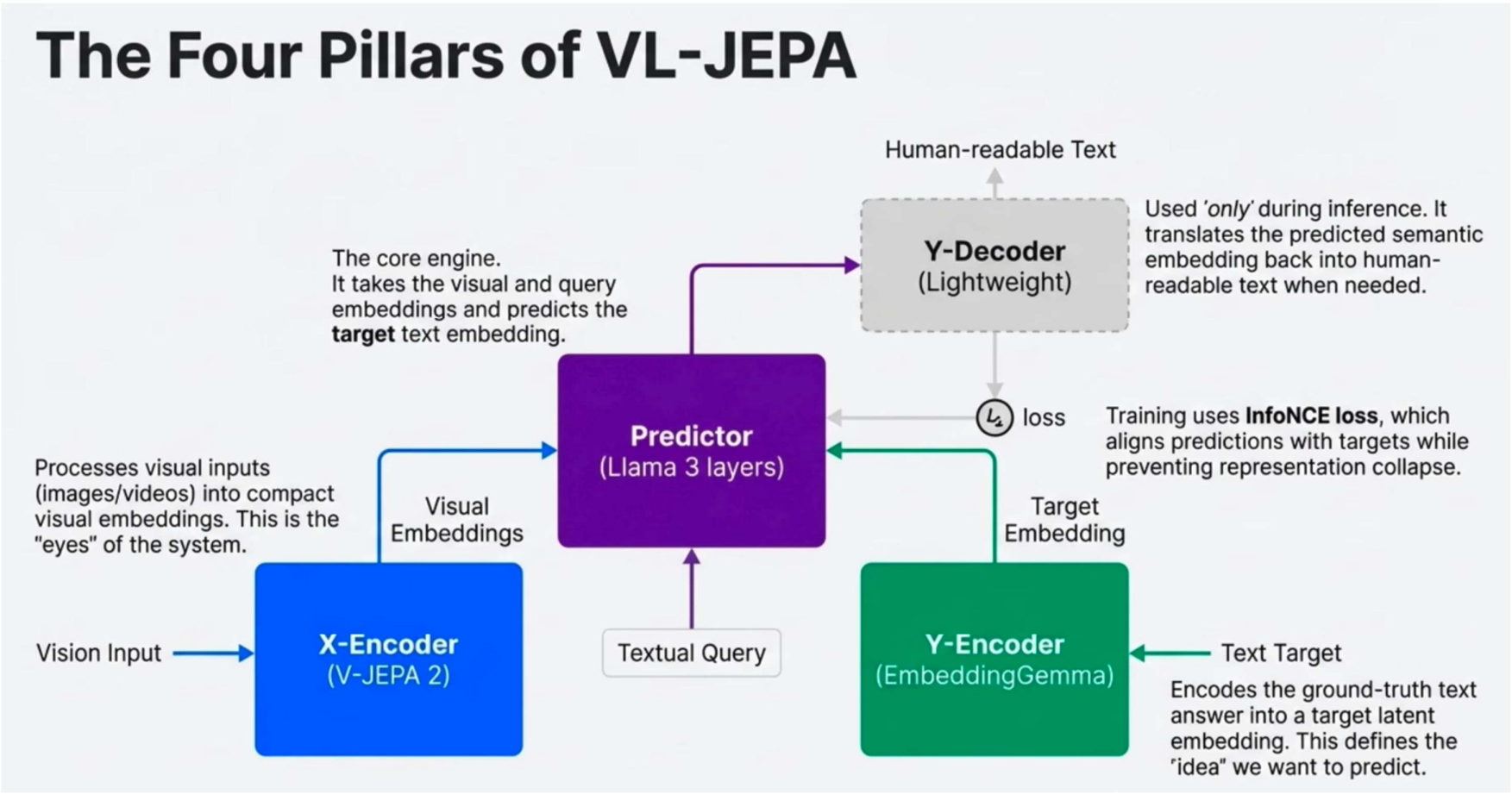
The Efficiency Breakthrough: Real-Time Performance with Selective Decoding

Standard VLMs must generate text token-by-token—an expensive, continuous process unsuitable for real-time video where updates are needed only when something changes.



This strategy enables always-on semantic monitoring while reducing the number of decoding operations by **~2.85x** with similar performance compared to uniform decoding.

The Four Pillars of VL-JEPA

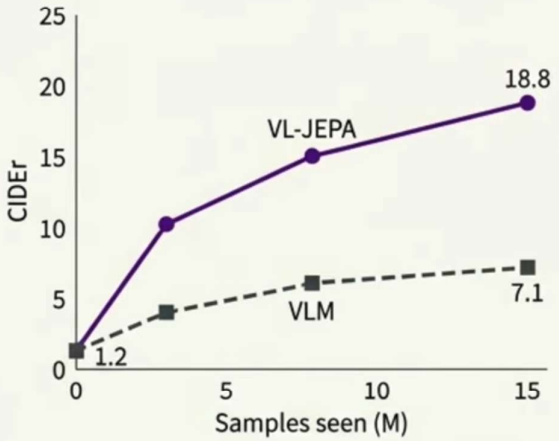


Embedding vs. Token Prediction: A Controlled Comparison

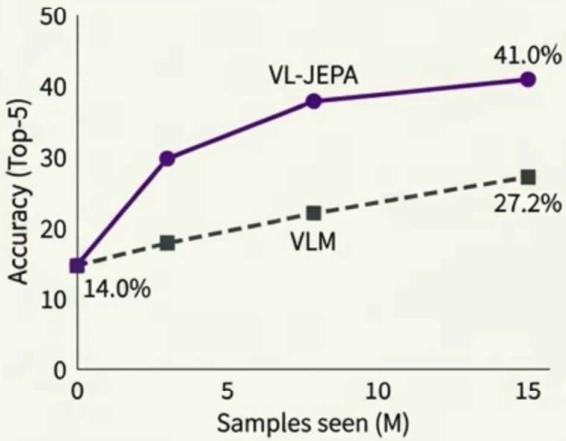
Experiment Setup

A strictly aligned comparison: same vision encoder, training data, and iterations. The only difference is the objective: embedding prediction (VL-JEPA) vs. next-token prediction (VLM).

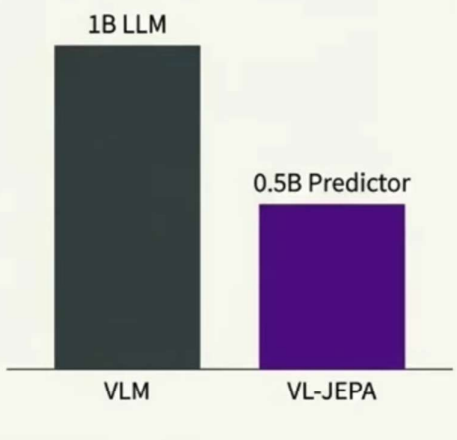
Zero-shot Video Captioning



Zero-shot Video Classification



Trainable Parameters



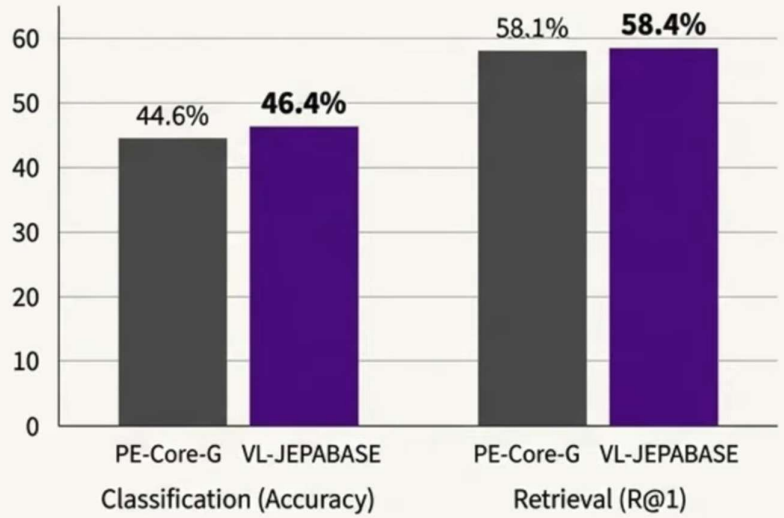
"This controlled comparison highlights the benefit of predicting embeddings rather than tokens, showing both higher sample efficiency and stronger absolute performance."

A Generalist Model for Classification, Retrieval, and VQA

A single, unified VL-JEPA architecture excels across a range of tasks without modification.

Zero-Shot Classification & Retrieval

Average Performance (8 Datasets)



Visual Question Answering

VQA Performance vs. Established VLMs

Benchmark	InstructBLIP / Qwen-VL	VL-JEPASFT (1.6B)
GQA (compositional reasoning)	49.5 / 59.3	**60.8**
TallyQA (complex counting)	68.0	**67.4**
POPE (object hallucination)	79.0	**84.2**

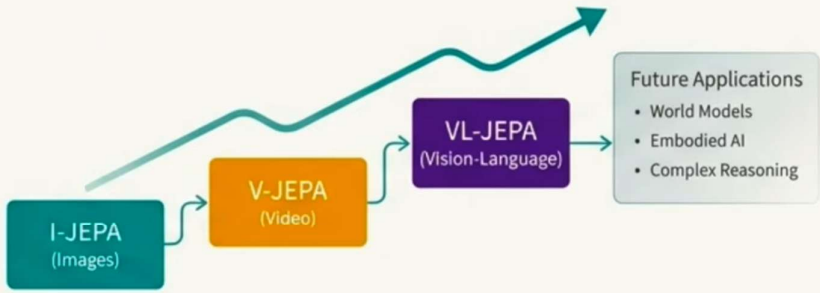
The JEPA Family: An At-a-Glance Comparison

Feature	I-JEPA	V-JEPA	VL-JEPA
Primary Domain	Static Images	Video	Vision + Language
Input	Image Context Block	Video Context Clip	Visual Input + Text Query
Prediction Target	Representation of Image Target Block	Representation of Video Target Clip	Continuous Embedding of Text Target
Architecture Core	ViT Encoder + Lightweight Predictor	ViT Encoder (e.g., V-JEPA 2)	Modular: V-JEPA 2 (X-Enc), Llama 3 (Pred), EmbeddingGemma (Y-Enc)
Loss Function	L2 Distance	Embedding Space Loss	InfoNCE (Contrastive)
Key Innovation	Learning semantics without augmentations	Extending prediction to the time domain	Separating semantic prediction from text generation
Efficiency Gain	Faster convergence than pixel models	Foundation for efficient video encoding	Selective Decoding (~2.85x fewer ops)

The JEPA Trajectory: More Abstract, More Efficient, More Capable

The Journey Recapped

- It began with **I-JEPA**, proving that predicting abstract representations in images could learn powerful semantic features more efficiently than reconstructing pixels.
- It evolved into **VL-JEPA**, a multimodal system that separates semantic understanding from language generation, unlocking new capabilities like selective decoding.



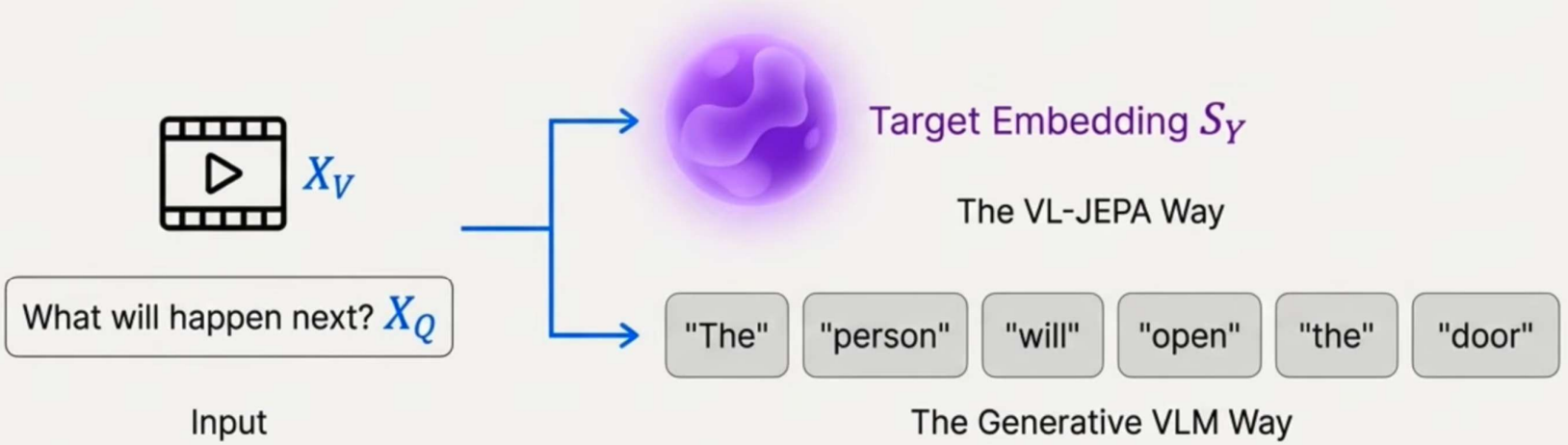
The Core Principle

The fundamental innovation is moving the prediction task from the data space (pixels, tokens) to an abstract, semantic embedding space. This unlocks greater efficiency, better semantic learning, and novel functionalities.

The Culmination: VL-JEPA Unites Vision and Language

VL-JEPA expands the predictive architecture to its logical conclusion: given a visual context and a text query, it predicts the *semantic meaning* of the answer.

This is the **first non-generative** model for general-domain vision-language tasks. It separates the act of *semantic prediction* from the act of *text generation*.



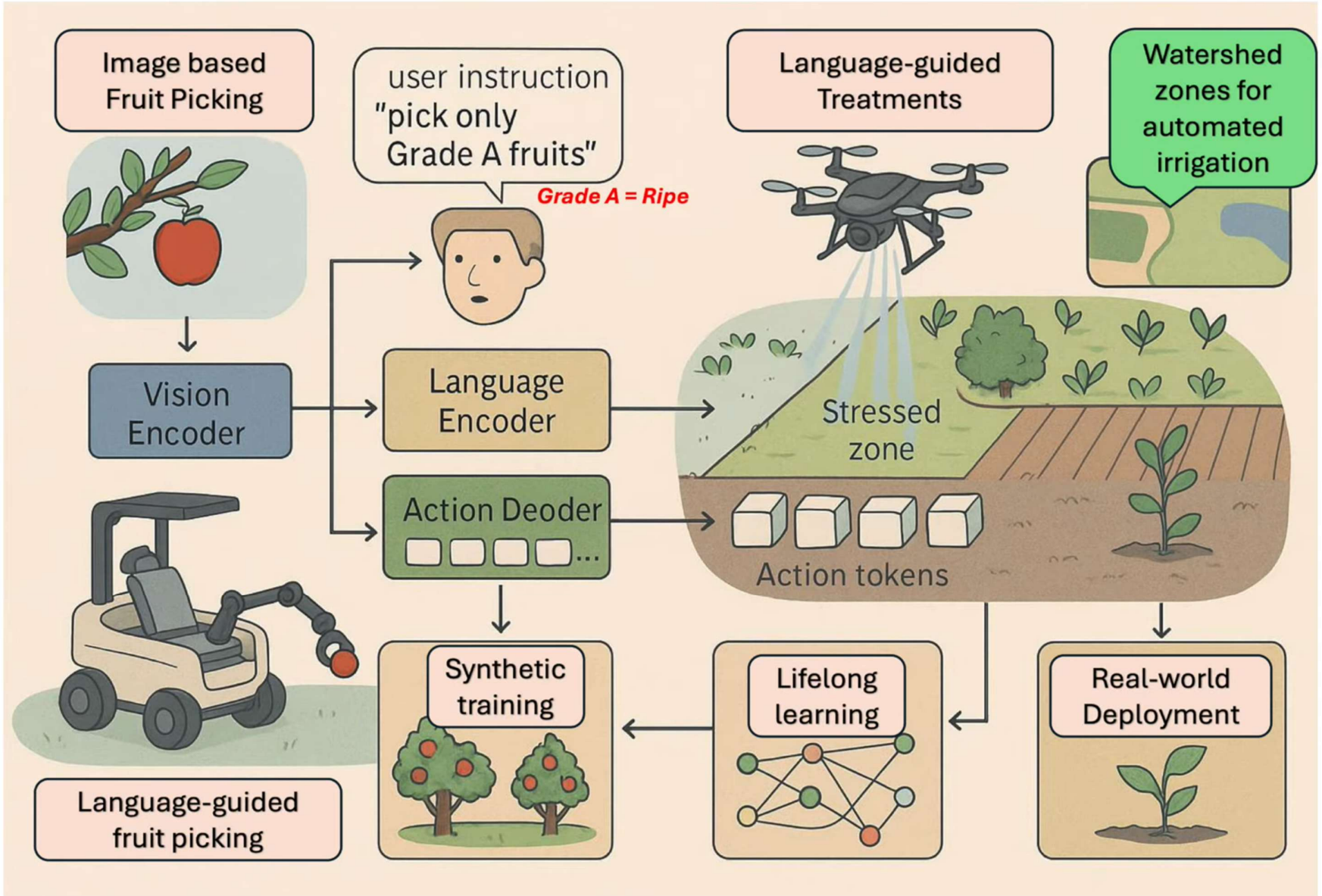


Image based Fruit Picking
user instruction "pick only Grade A fruits"
Grade A = Ripe
Language-guided Treatments
Watershed zones for automated irrigation
Vision Encoder
Language Encoder
Action Deoder
Stressed zone
Action tokens
Synthetic training
Lifelong learning
Real-world Deployment
Language-guided fruit picking

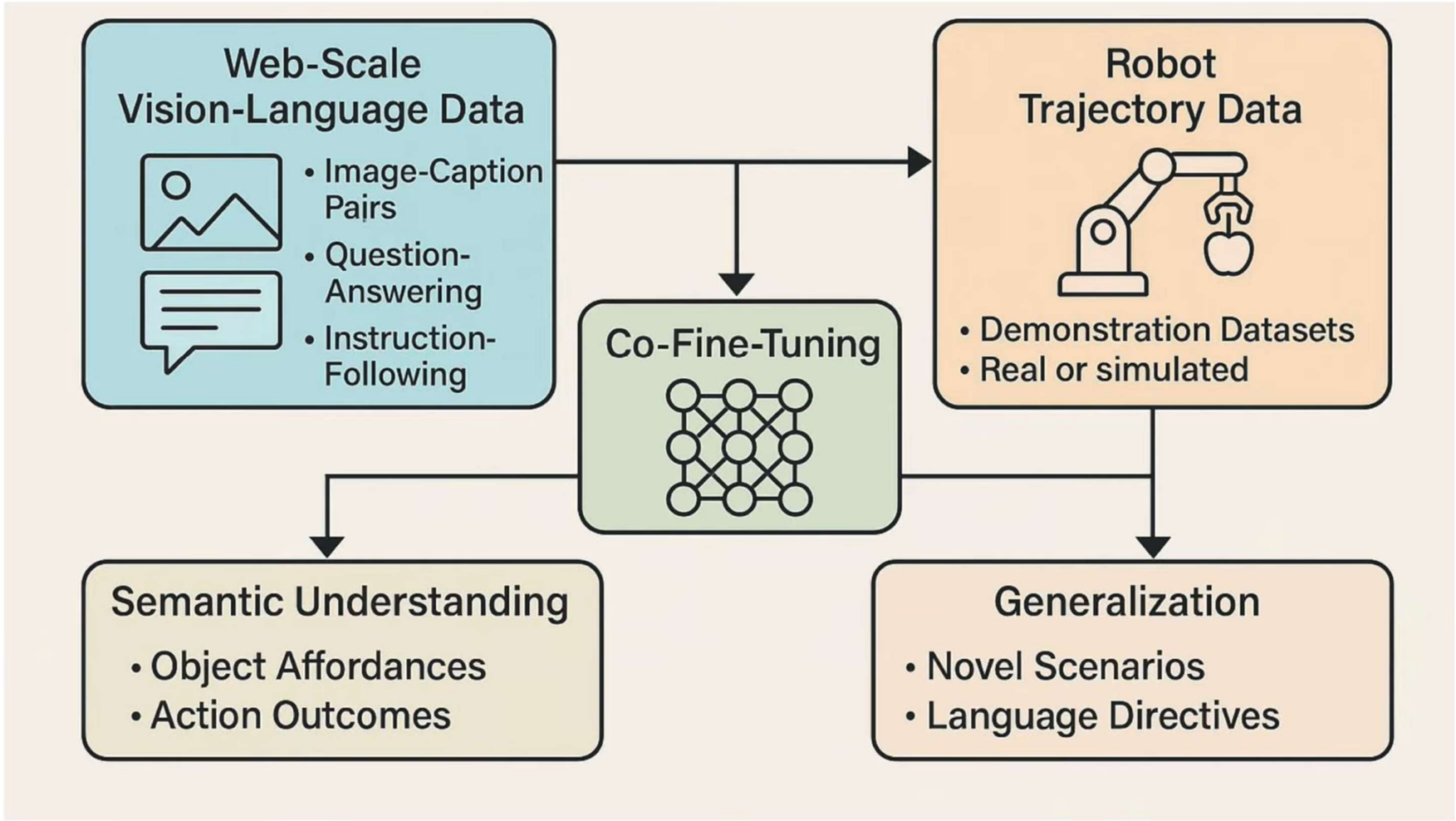
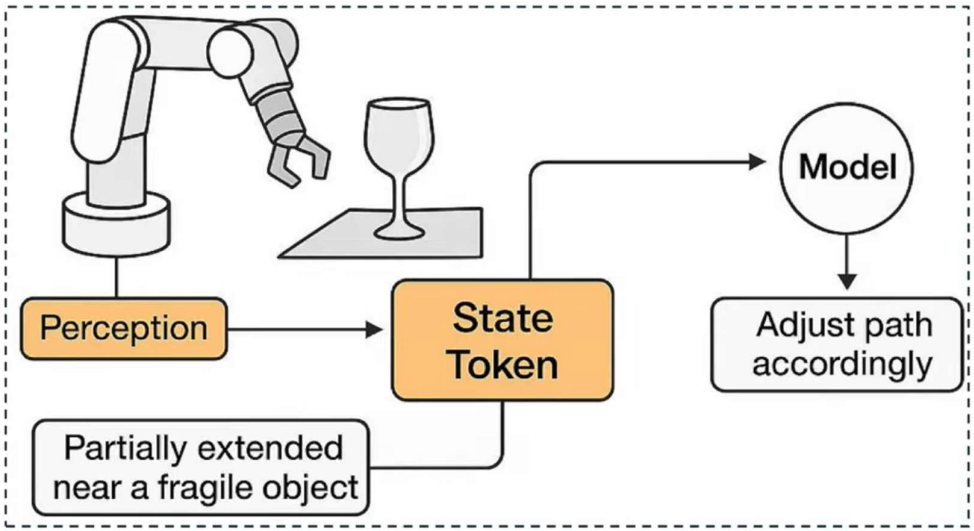
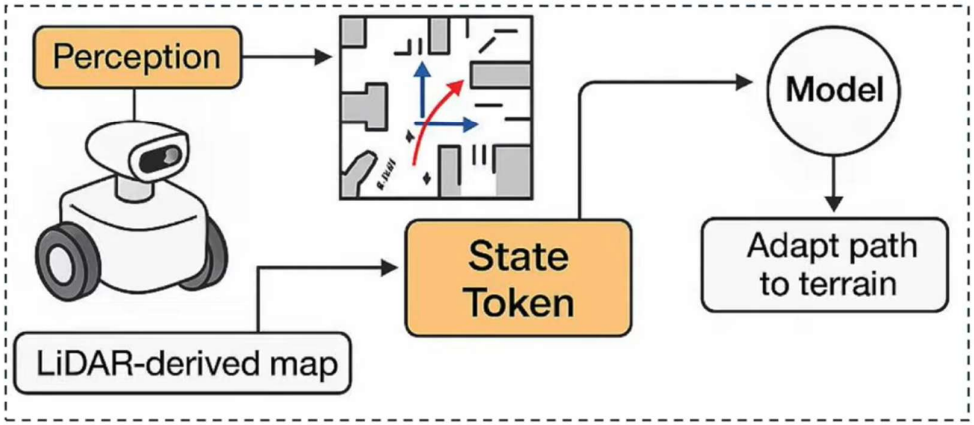


Figure 10: **Learning Paradigms: Data Sources and Training Strategies for VLAs.**


Model
Perception
State Token
Adjust path accordingly
Partially extended near a fragile object

(a)


Perception
Model
State Token
Adapt path to terrain
LiDAR-derived map

(b)

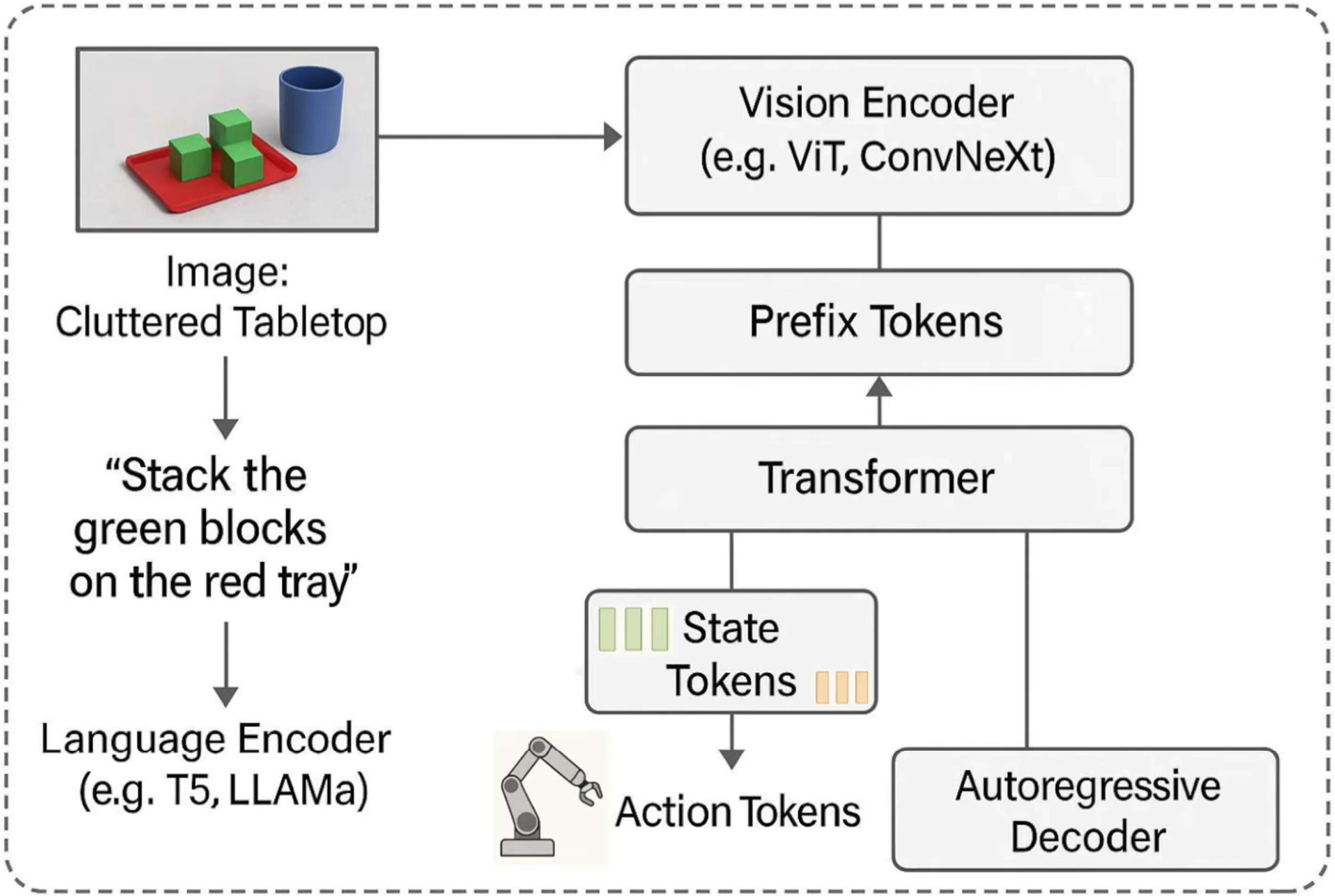


Image:
Cluttered Tabletop
"Stack the green blocks on the red tray"
Language Encoder (e.g. T5, LLAMa)
Vision Encoder (e.g. ViT, ConvNeXt)
Prefix Tokens
Transformer
State Tokens
Action Tokens
Autoregressive Decoder

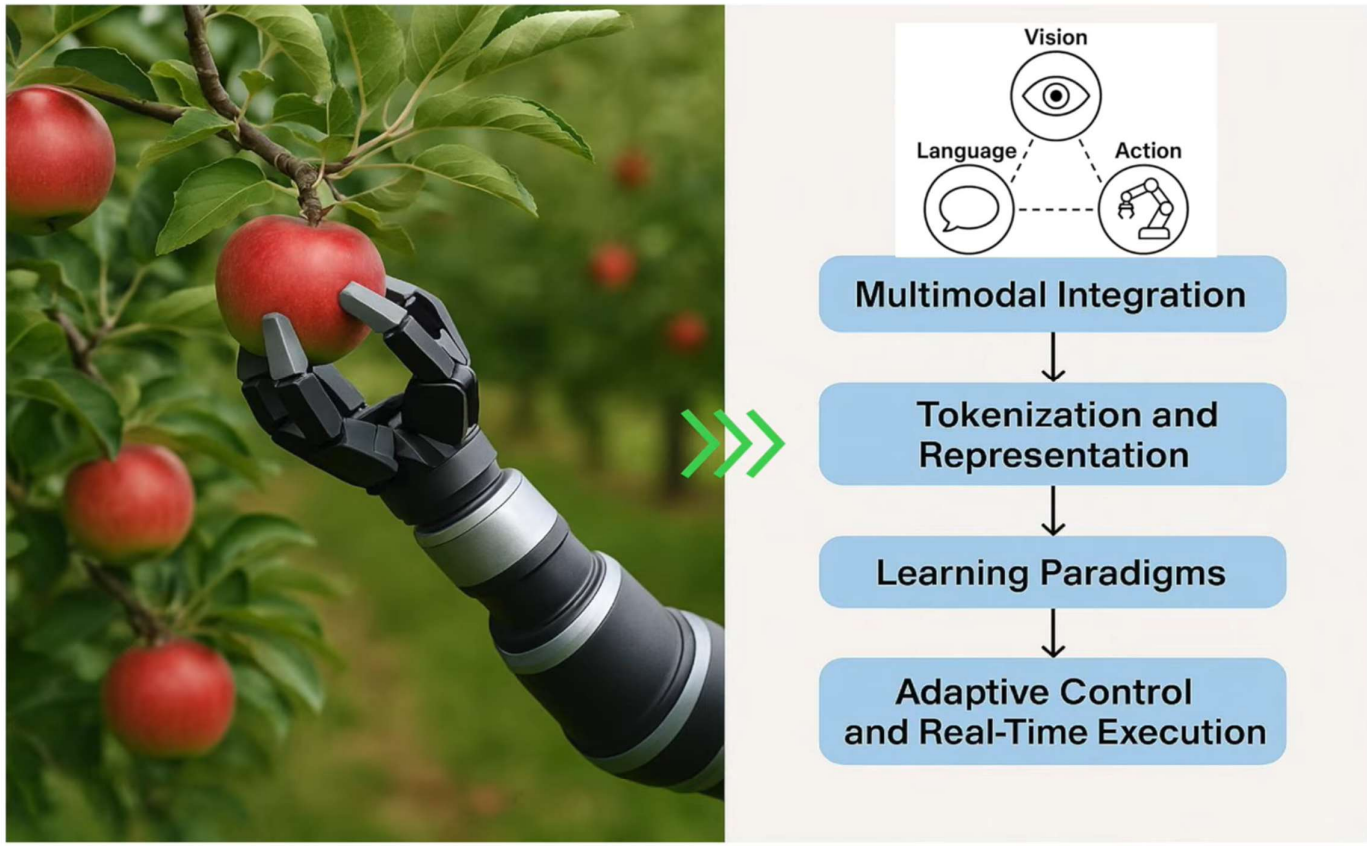


Figure 5: **Foundational Concept of VLA Models (in an Apple-Picking Scenario)** This illustration depicts a robotic arm autonomously picking a ripe apple in an orchard, guided by a VLA model. On the right, a flowchart outlines the four key stages of VLA models: Multimodal Integration, Tokenization and Representation, Learning Paradigms, and Adaptive Control and Real-Time Execution.

The JEPA Lineage: A Side-by-Side Comparison

Feature	I-JEPA	V-JEPA	VL-JEPA
Primary Domain	Static Images	Video	Vision + Language
Input	Image Context Block	Video Context	Visual Input + Text Query
Prediction Target	Image Block **Representation**	Video **Representation**	Text Target **Embedding**
Core Goal	Semantic Image Representation	Video Representation Learning	Multimodal Understanding
Loss Function	L2 Distance	(Embedding Space)	InfoNCE (Contrastive)
Key Efficiency	No manual augmentations	Efficient video encoding	Selective Decoding (2.85x gain)

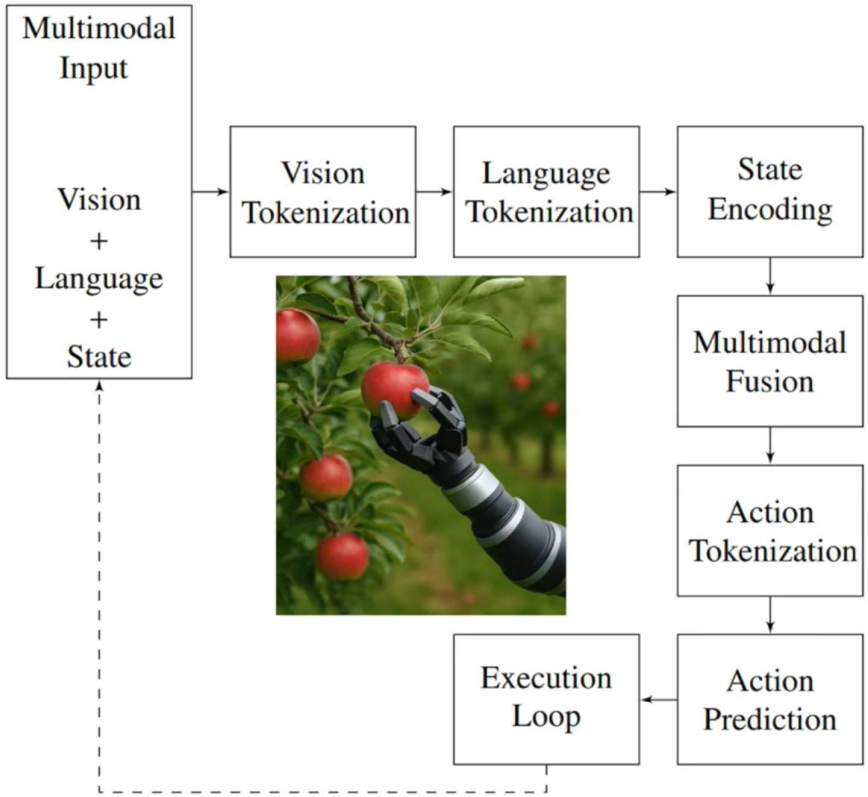
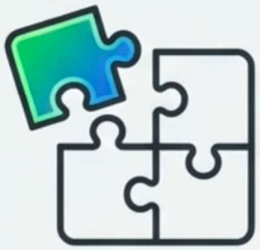


Figure 9: **Illustrating the process of how VLAs Encode the World.** VLAs encode the world by converting vision, language, and sensor inputs into tokens, fusing them through cross-attention, predicting action sequences via transformers, and executing tasks with real-time feedback-enabling robots to interpret scenes, follow instructions, and adapt actions dynamically.

From Seeing to Understanding

I-JEPA



Learns the parts of the world by imagining what's hidden.

V-JEPA

Learns the flow of the world by predicting what happens next.

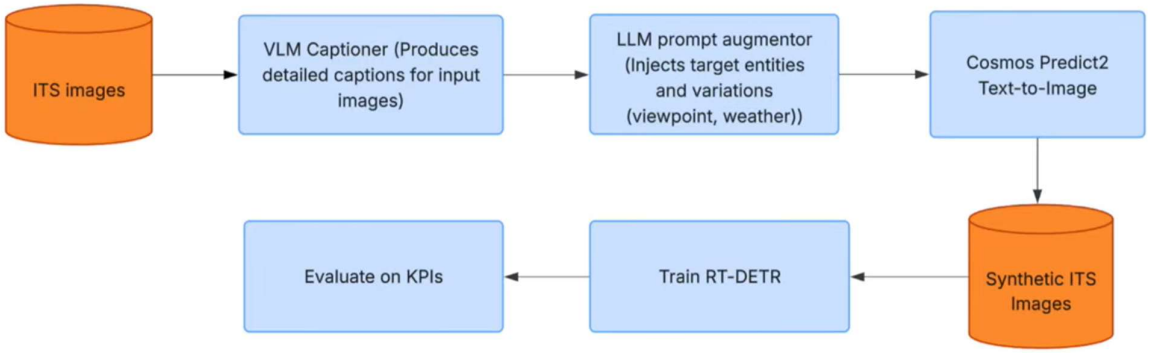
VL-JEPA

Learns the *meaning* of the world by predicting the idea behind events.

The **JEPA** paradigm marks a **fundamental shift**: from models that learn by **reconstructing** the world's surface, to models that learn by **predicting its abstract, semantic core**. This is **not just a more efficient way** to learn; it's a step towards more **general** and **robust** machine intelligence.

Cosmos Predict 2 Pipeline Components

Architecture



Component explanations

- VLM Captioner: Produces faithful, detailed captions from example ITS images to seed generation.
- LLM Prompt Augmenter: Injects target entities and variations (viewpoint, weather) under strict realism rules.
- Cosmos Predict 2 Text-to-Image: Generates high-quality ITS images aligned with prompts.
- Train RT-DETR: Fine-tune the detector .
- Evaluate on KPIs: Measure improvements across ACDC, SUTD, DAWN (e.g., AP50 per class and weather).

Cosmos Curator - A GPU-accelerated video curation pipeline built on Ray. Supports multi-model analysis, content filtering, annotation, and deduplication for both inference and training data preparation.

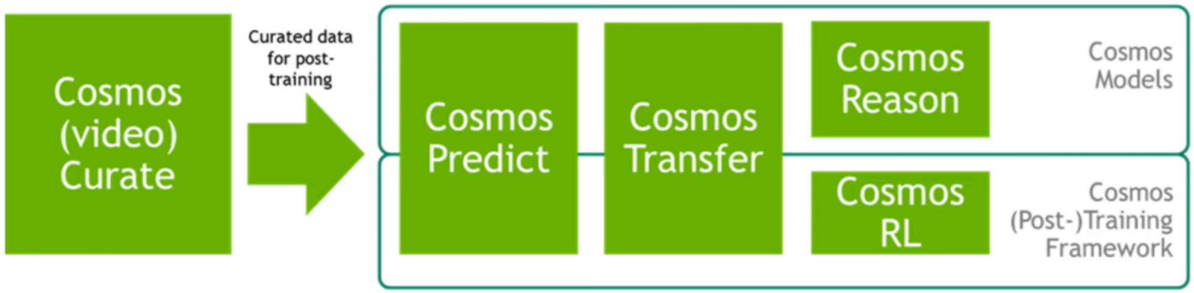
Cosmos Predict - A diffusion transformer for future state prediction. Provides text-to-image and video-to-world generation capabilities, with specialized variants for robotics and simulation. Supports custom training for domain-specific prediction tasks.

Cosmos Transfer - A multi-control video generation system with ControlNet and MultiControlNet conditioning (including depth, segmentation, LiDAR, and HDMap). Includes 4K upscaling capabilities and supports training for custom control modalities and domain adaptation.

Cosmos Reason - A 7B vision-language model for physically grounded reasoning. Handles spatial/temporal understanding and chain-of-thought tasks, with fine-tuning support for embodied AI applications and domain-specific reasoning.

Cosmos RL - A distributed training framework supporting both supervised fine-tuning (SFT) and reinforcement learning approaches. Features elastic policy rollout, FP8/FP4 precision support, and optimization for large-scale VLM and LLM training.

All models include pre-trained checkpoints and support custom training for domain-specific adaptation. The diagram below illustrates component interactions across inference and training workflows.



The Future is Semantic: Latent-Space Prediction as a Path to More Efficient AI

Summary of Key Advantages

1. **Simplifies the Learning Target**: Focuses on meaning by abstracting away linguistic noise.
2. **Improves Efficiency**: Delivers stronger performance with fewer parameters and lower training costs.
3. **Enables Real-Time Inference**: Drastically reduces latency via non-autoregressive prediction and selective decoding.
4. **Provides Architectural Unity**: A single model handles generation, retrieval, and classification.

The Path Forward

VL-JEPA demonstrates a powerful alternative to standard token-generative models. This approach is a foundation for future work on multi-modal latent space reasoning, like visual chain-of-thought.

By predicting in a shared, abstract representation space, we move closer to more intelligent, versatile, and practical AI systems.