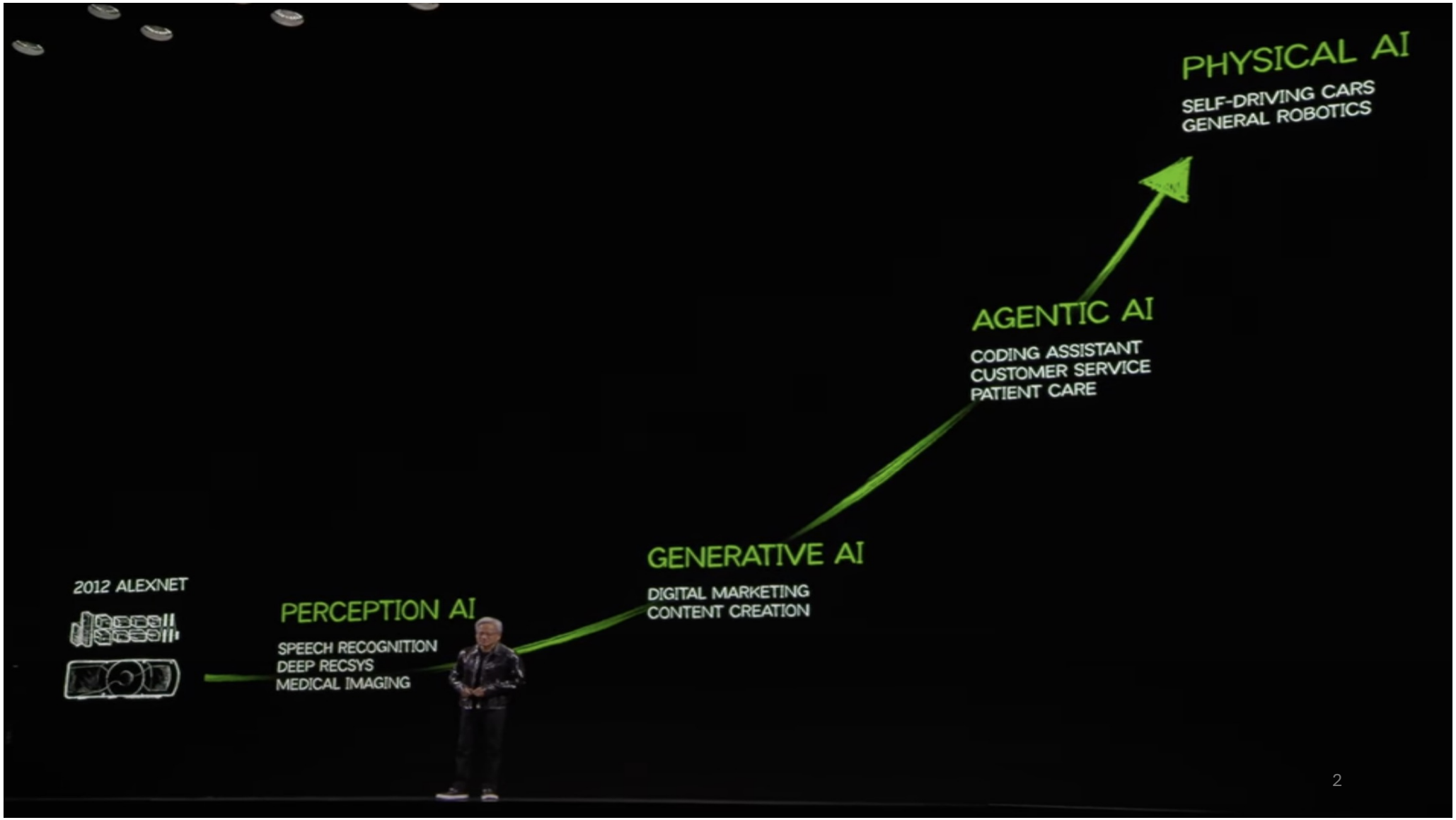
World Model

ziyu

2026-06-24


PHYSICAL AI
SELF-DRIVING CARS
GENERAL ROBOTICS
AGENTIC AI
CODING ASSISTANT
CUSTOMER SERVICE
PATIENT CARE
GENERATIVE AI
DIGITAL MARKETING
CONTENT CREATION
2012 ALEXNET
PERCEPTION AI
SPEECH RECOGNITION
DEEP RECSYS
MEDICAL IMAGING

Outline

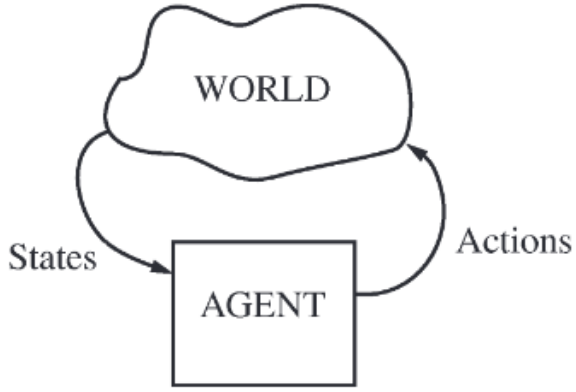
- 背景、问题设定
- World Model 的概念、能力
- 起点（world model 正式提出之前）
- Research line，经典工作
  - Dreamer
  - JEPA
- 总结与补充材料

MDP

- 智能体在每个时刻都能看到完整状态 s_t，然后做动作 a_t
- 两者结合，决定下一步世界怎么变，以及得到多少奖励。

$$s_{t+1} \sim P_E(s_{t+1} \mid s_t, a_t)$$

$$r_{t+1} \sim R_E(r_{t+1} \mid s_t, a_t, s_{t+1})$$



- MDP局限性：真实状态 s_t 不一定能被智能体直接看到。能看到的是 o_t（观测），来自真实状态 s_t $o_t \sim O_E(o_t \mid s_t)$

POMDP (partially observable ...)

- 环境：$\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, P_E, O_E, R_E, \gamma).$

$$s_{t+1} \sim P_E(s_{t+1} \mid s_t, a_t),$$
$$o_t \sim O_E(o_t \mid s_t),$$
$$r_{t+1} \sim R_E(r_{t+1} \mid s_t, a_t, s_{t+1}).$$

  - S：真实状态空间
  - A：动作空间
  - O：观测空间
  - P_E：真实环境动力学
  - O_E：观测模型
  - R_E：奖励模型
  - Gamma：折扣因子（衡量未来奖励的重要性，0~1，1 代表重视长期未来）

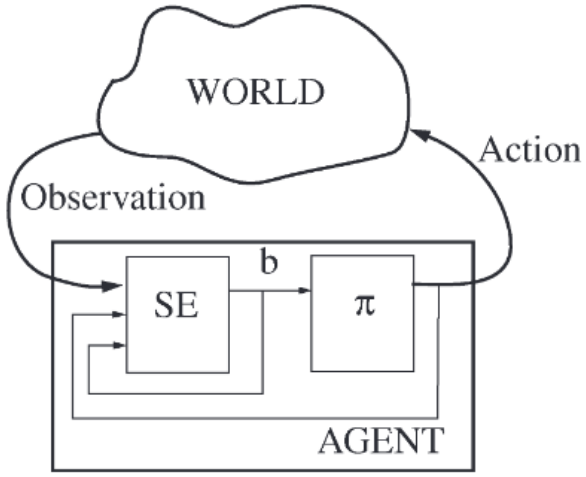
POMDP

$$\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, P_E, O_E, R_E, \gamma).$$

$$s_{t+1} \sim P_E(s_{t+1} \mid s_t, a_t),$$

$$o_t \sim O_E(o_t \mid s_t),$$

$$r_{t+1} \sim R_E(r_{t+1} \mid s_t, a_t, s_{t+1}).$$



WM 简单描述

$h_t = (o_0, a_0, r_1, o_1, \ldots, a_{t-1}, r_t, o_t)$ $\mathcal{M}_\theta$ $p_\theta(y_{t+1:t+H} \mid h_t, a_{t:t+H-1})$

智能体与环境交互产生的历史 → 世界模型 → 未来观测、潜在状态、奖励……（其他行为或任务相关变量）的分布

WM 简单描述

$h_t = (o_0, a_0, r_1, o_1, \ldots, a_{t-1}, r_t, o_t)$ $\mathcal{M}_\theta$ $p_\theta(y_{t+1:t+H} \mid h_t, a_{t:t+H-1})$

智能体与环境交互产生的历史 → 世界模型 → 未来观测、潜在状态、奖励……（其他行为或任务相关变量）的分布

WM 形式化描述

- 环境：POMDP（一般情况）
- 引入潜变量 z_t：$z_t \sim q_\theta(z_t \mid h_t).$

- 典型 WM 通常包含几部分：
  - State inference: $q_\theta(z_t \mid h_t)$
  - **Dynamics model**: $p_\theta(z_{t+1} \mid z_t, a_t)$
  - Observation model: $p_\theta(o_{t+1} \mid z_{t+1})$
  - Reward/cost model: $p_\theta(r_{t+1}, c_{t+1} \mid z_t, a_t, z_{t+1})$
  - Constraint model: $p_\theta(d_{t+1}, g_{t+1} \mid z_t, a_t, z_{t+1})$ done/termination, constraint

WM 形式化描述

$$p_\theta(o_{t+1:t+H}, r_{t+1:t+H}, z_{t+1:t+H} \mid h_t, a_{t:t+H-1})$$

WM 的关键能力

- 1. 表征世界状态（真实物理 vs 用于决策的统计量）
- 2. 预测世界（动力学、观测、奖励……），支持**planning**
- 3. 给出**分布**（probabilistic vs deterministic）
- 4. 反事实推理。
- 5. 支持不同的抽象层级和**尺度**（潜力）
  - E.g. pixels -> objects -> relations -> events -> plans

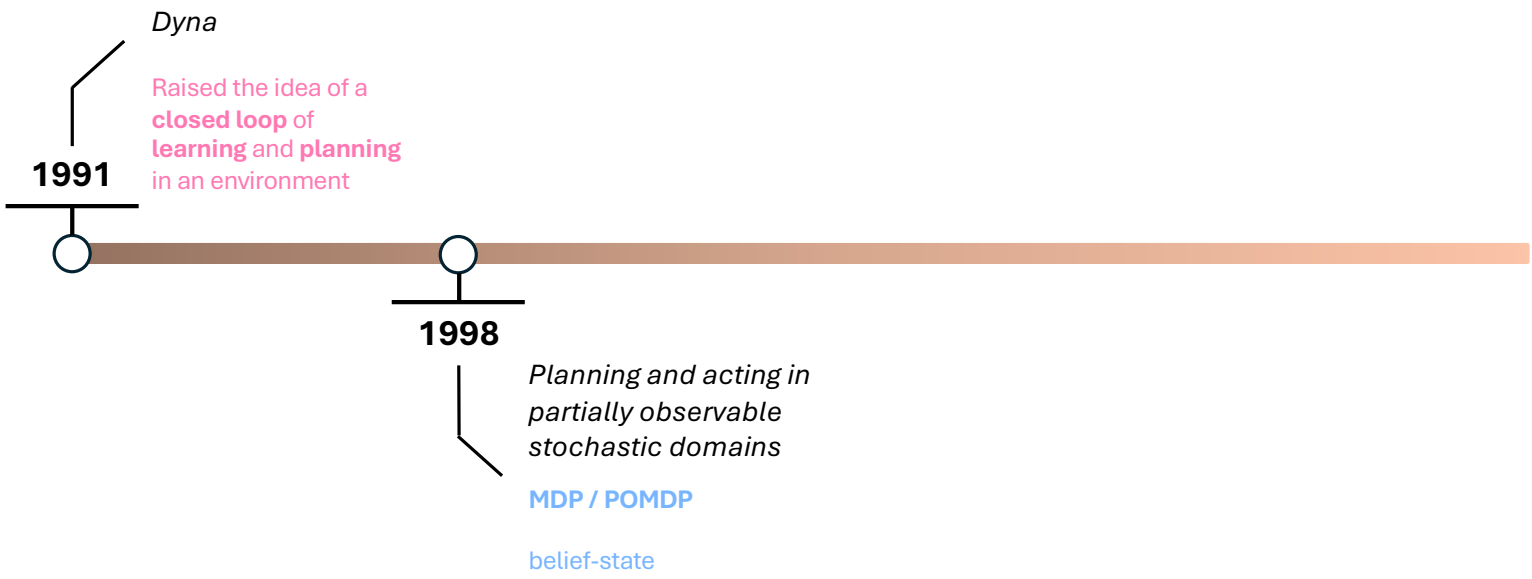
WM 定义总结

$$\mathcal{M}_\theta : (h_t, a_{t:t+H-1}) \mapsto p_\theta(Y_{t+1:t+H} \mid h_t, a_{t:t+H-1})$$

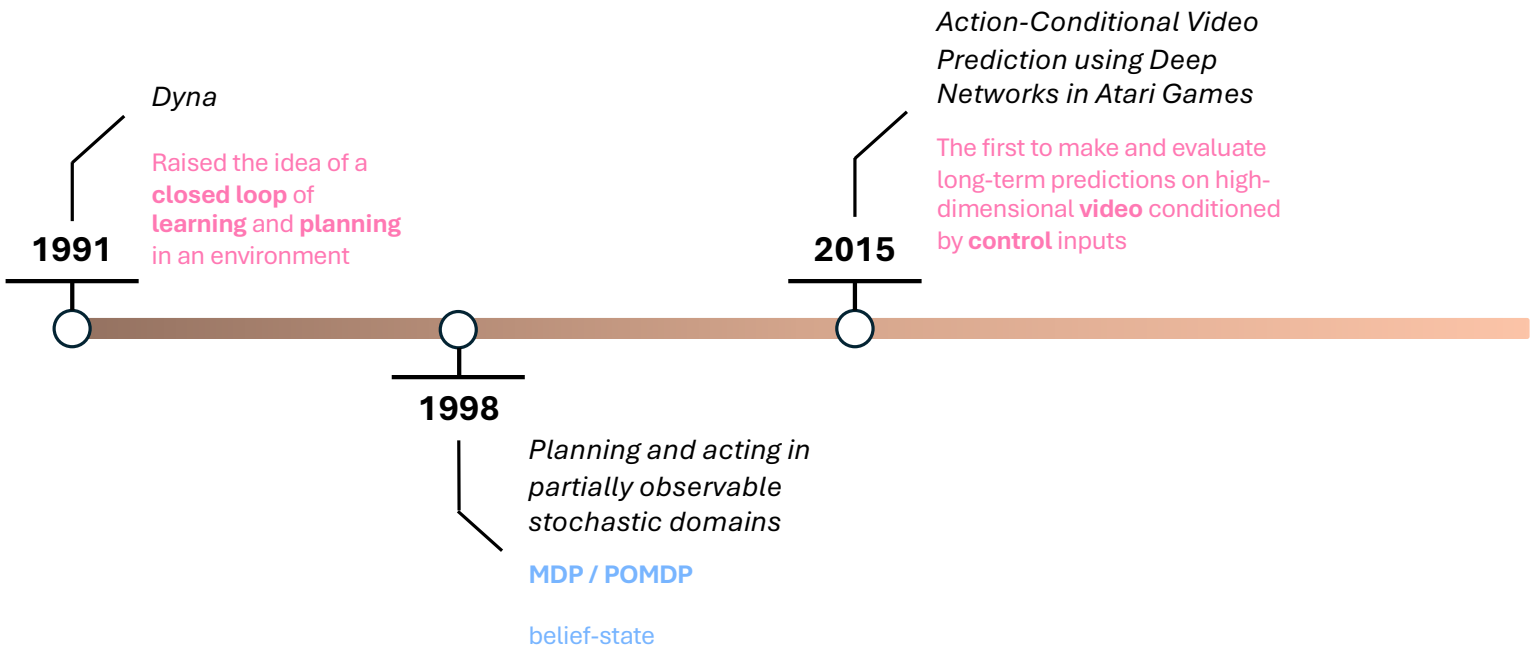
- World model 是智能体关于外部世界、自身身体、其他智能体以及行动后果的内部生成性/预测性模型。它从**历史经验**中推断当前**隐状态**，并在给定候选**行动或干预**的条件下，**预测**未来状态、观测、奖励、风险与约束的**分布**；
- 其目的不是完整复制世界，而是保留对预测、**推理**、**规划**和**控制**足够充分的结构。

Pre-World Model

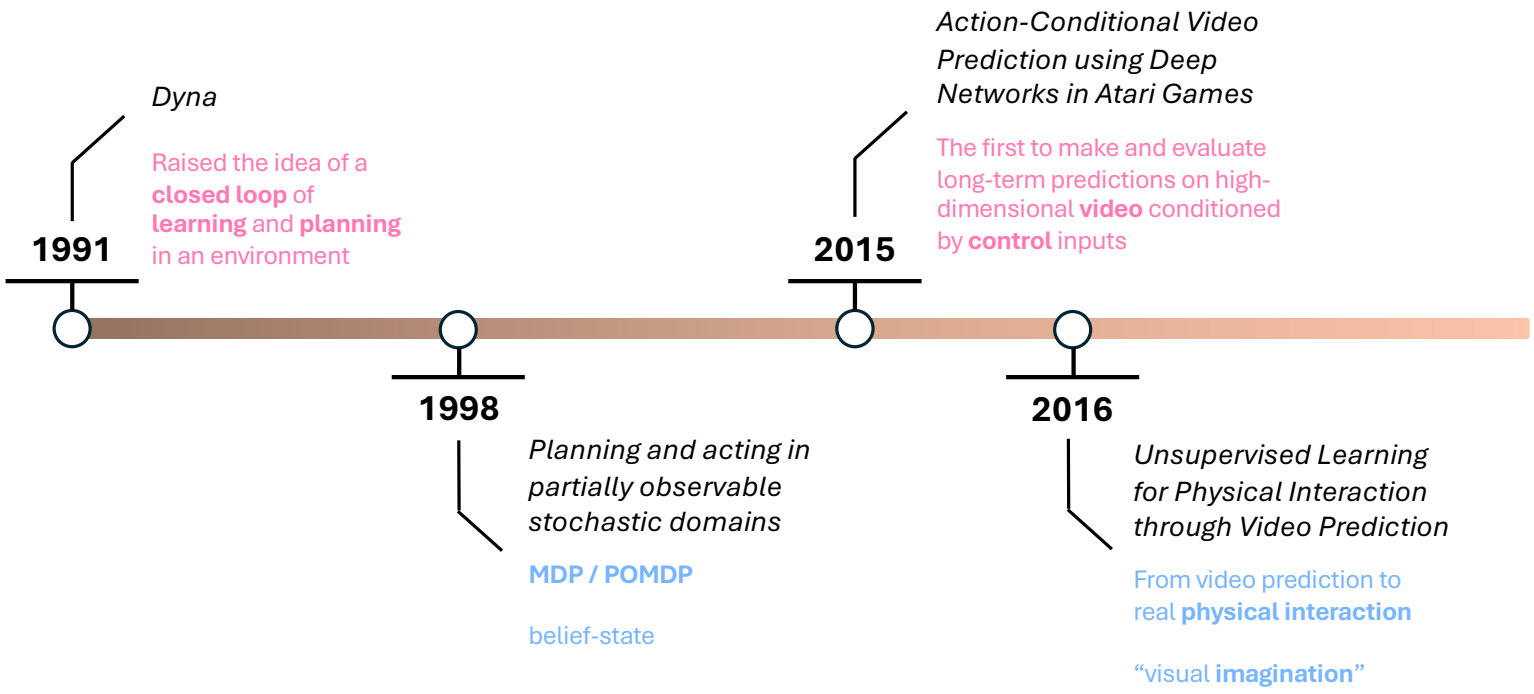
Pre-World Model



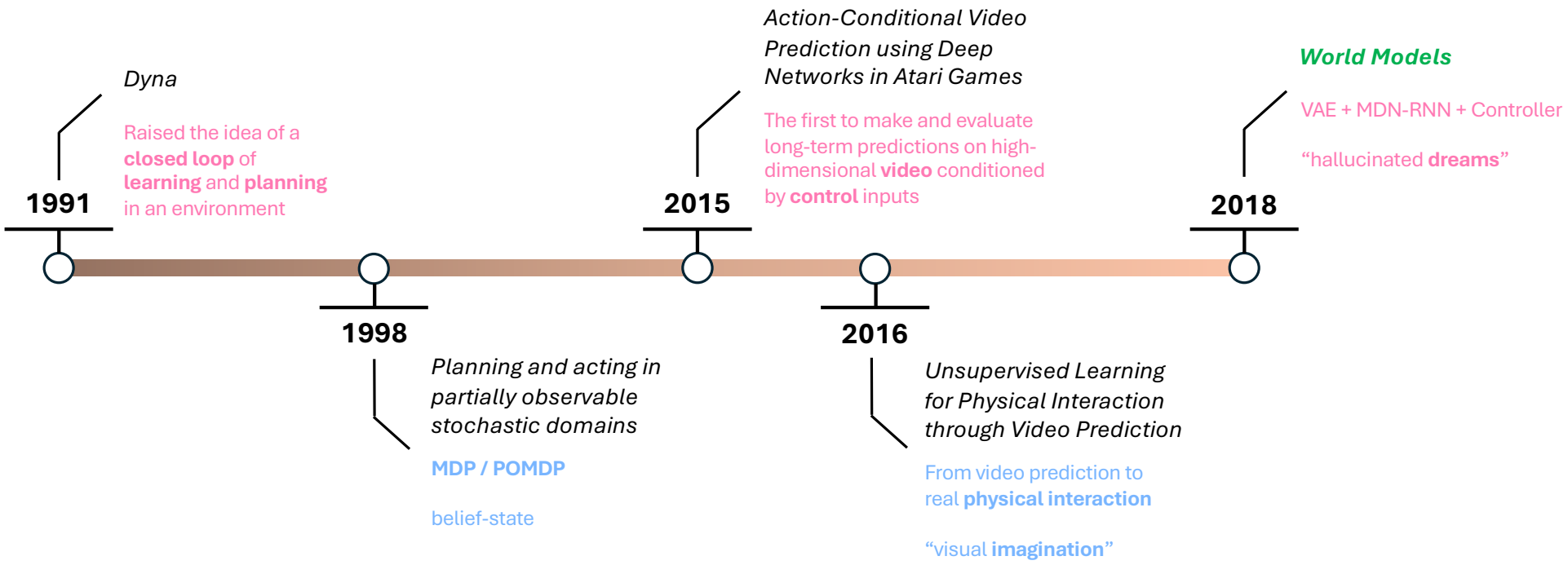
Pre-World Model



Pre-World Model

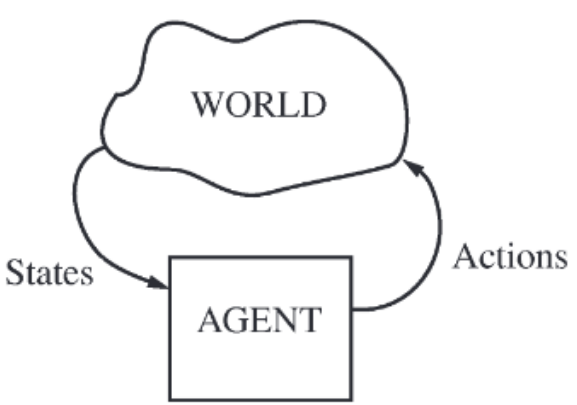


Pre-World Model

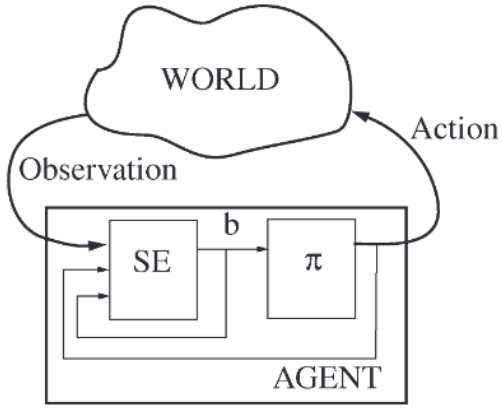


1991

1998

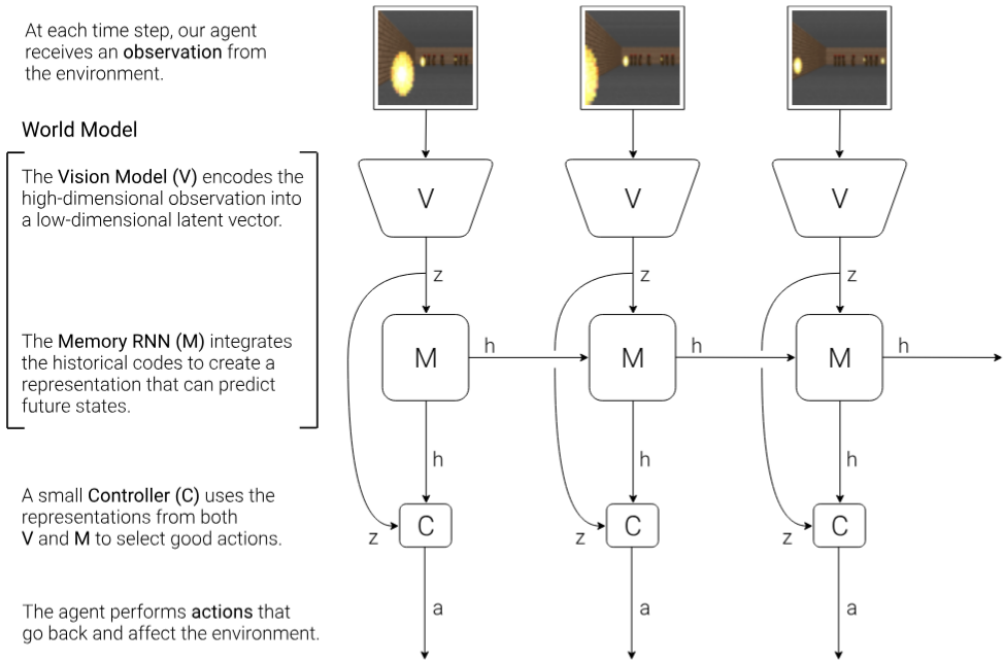


MDP



POMDP

2018



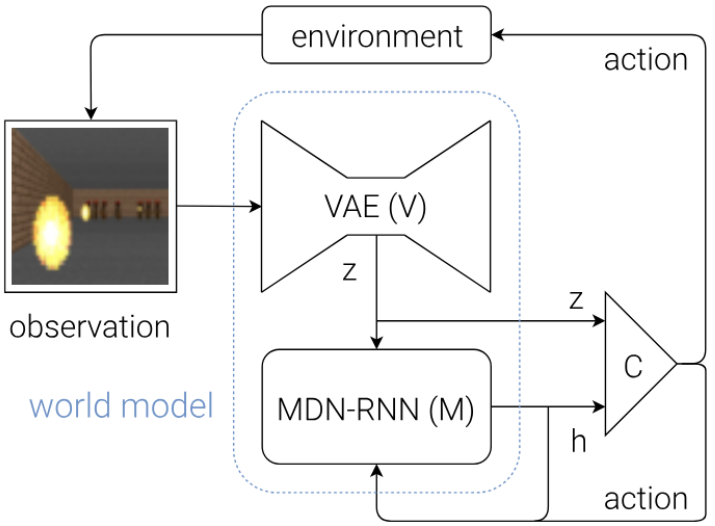
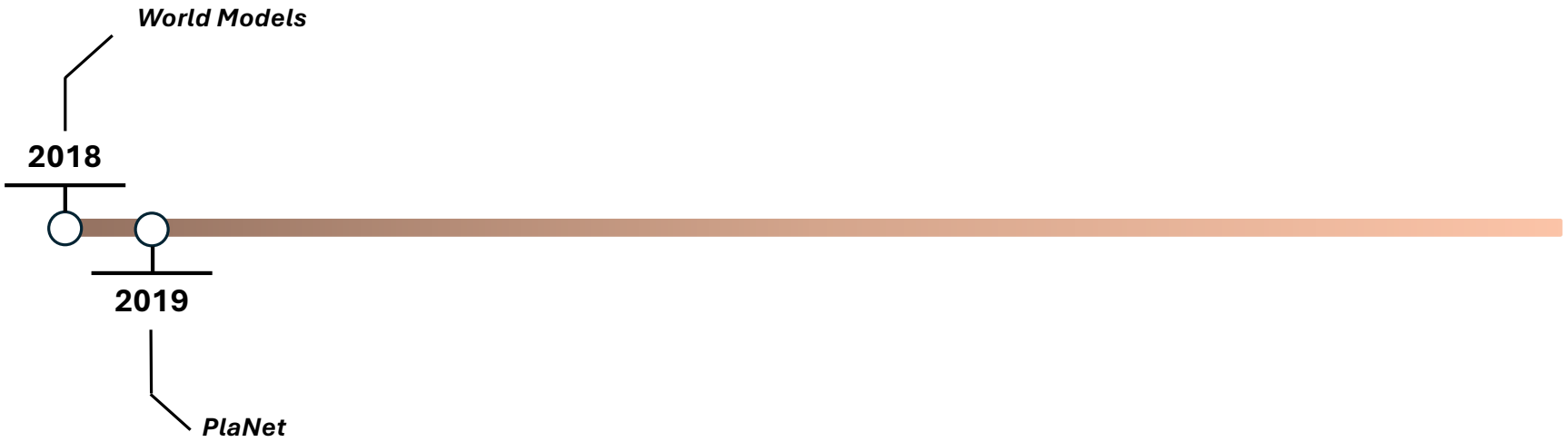


Figure 4. Our agent consists of three components that work closely together: **Vision (V)**, **Memory (M)**, and **Controller (C)**

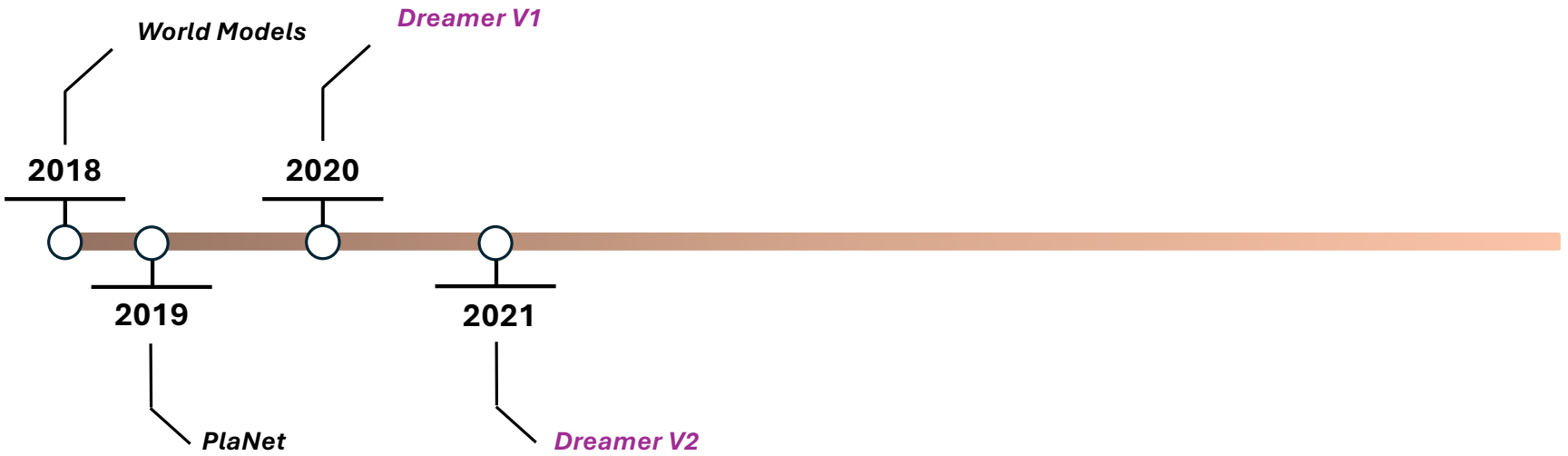
World Model

World Model

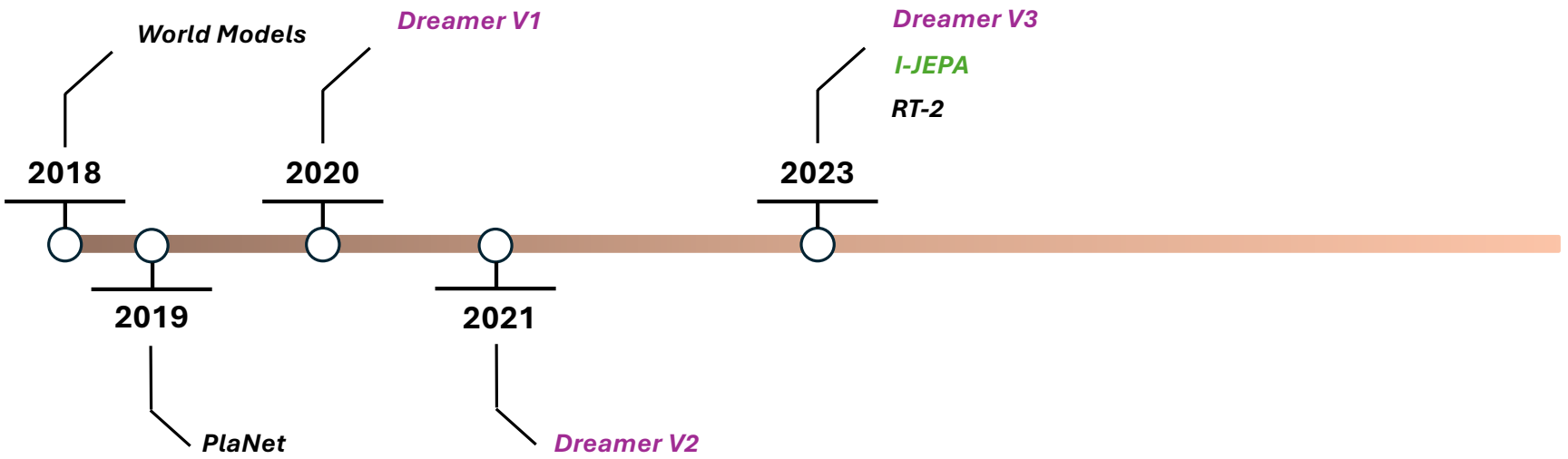


World Model

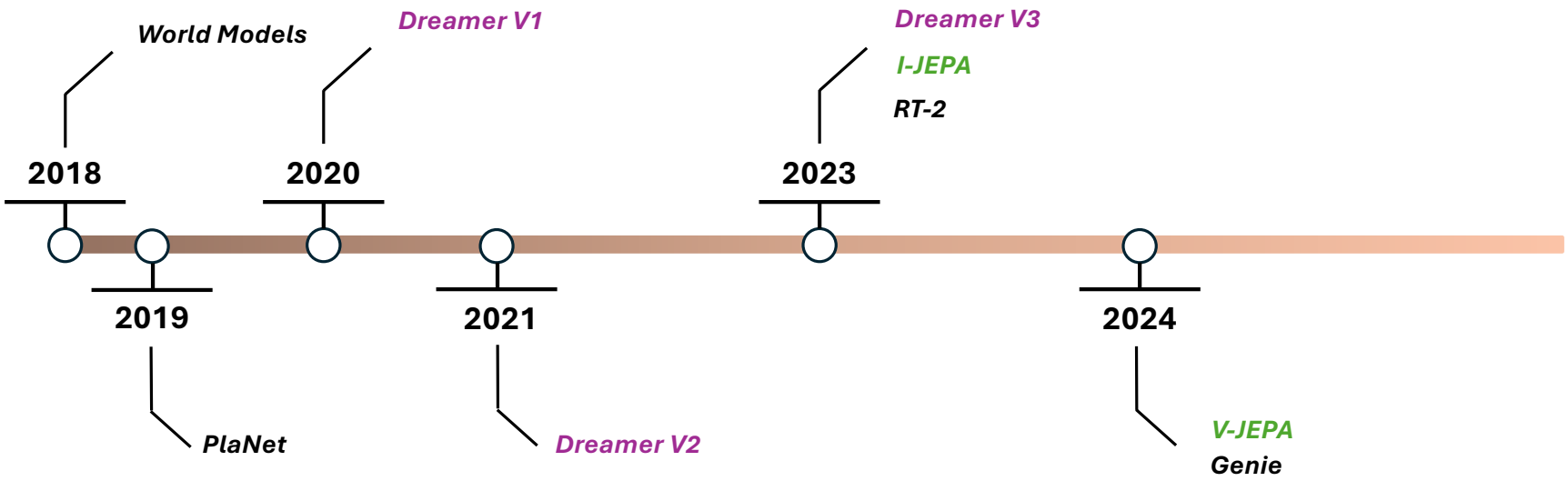
World Model


World Models
Dreamer V1
2018
2020
2019
2021
PlaNet
Dreamer V2

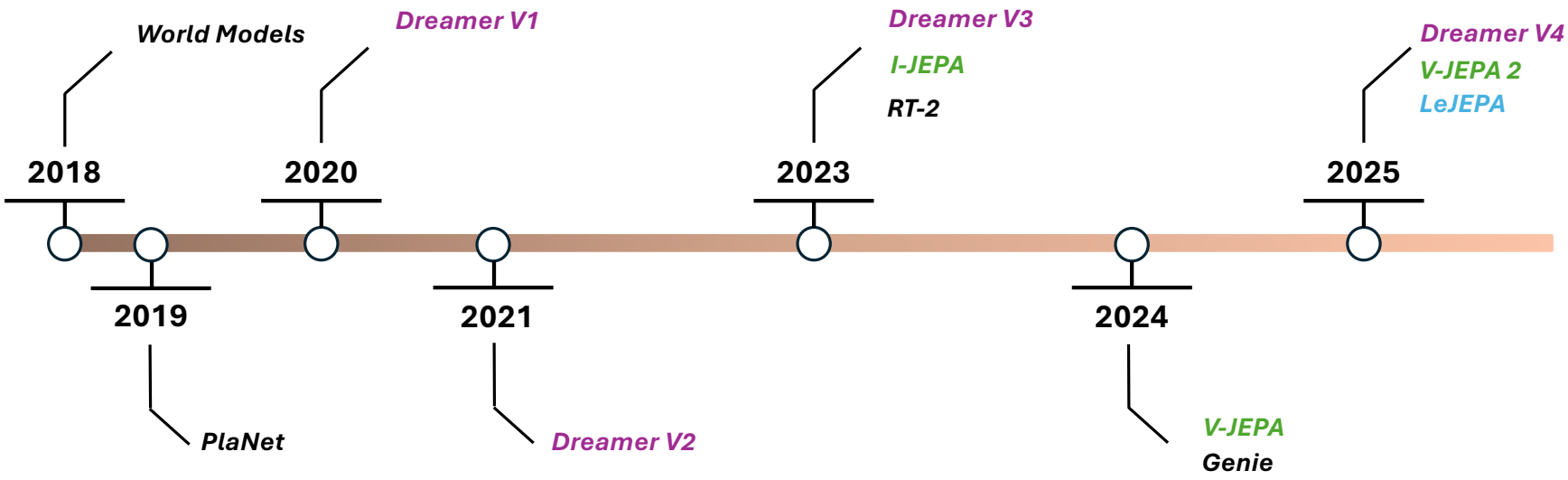
World Model


World Models
2018
PlaNet
2019
Dreamer V1
2020
Dreamer V2
2021
Dreamer V3
I-JEPA
RT-2
2023

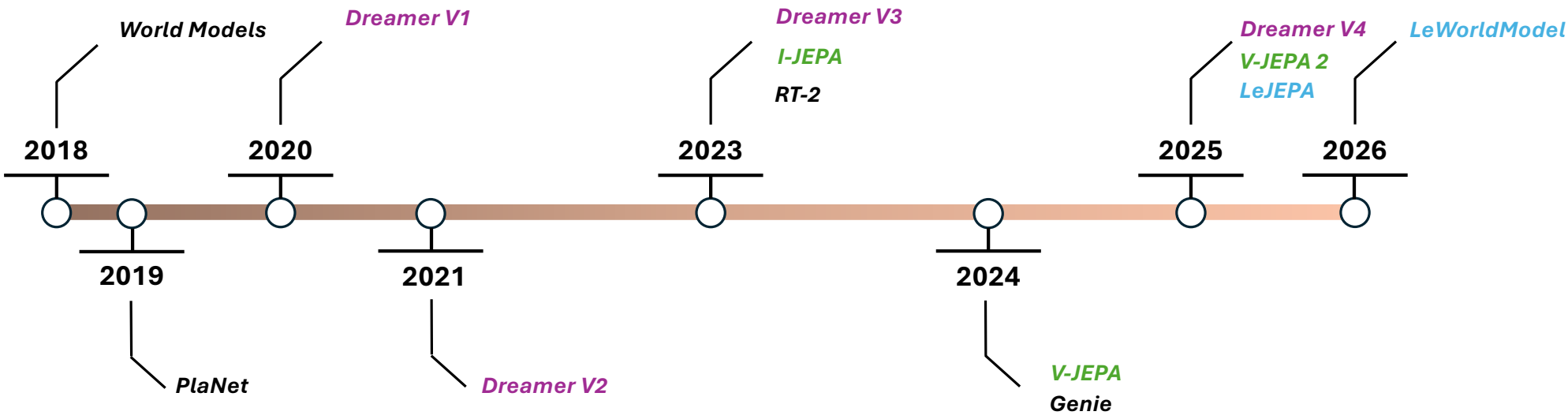
World Model


World Models
Dreamer V1
Dreamer V3
I-JEPA
RT-2
2018
2020
2023
2019
2021
2024
PlaNet
Dreamer V2
V-JEPA
Genie

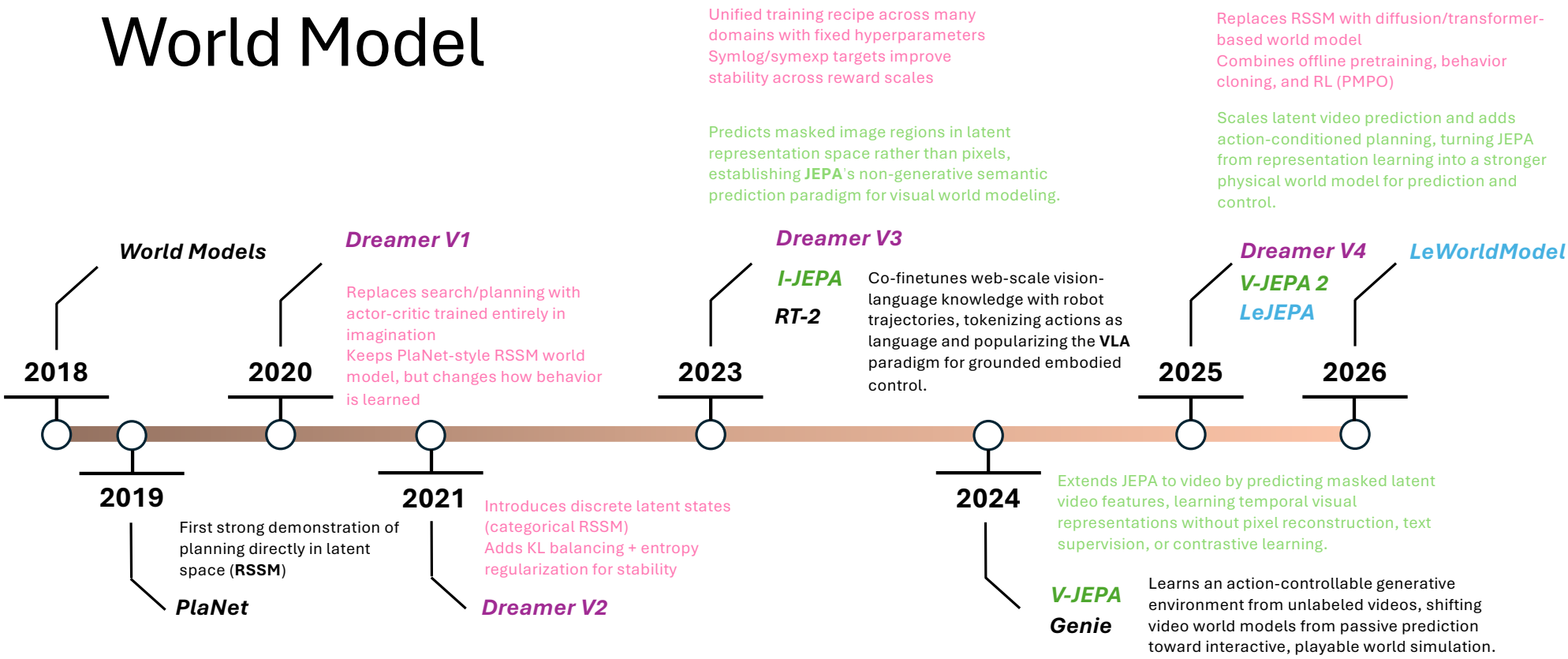
World Model


World Models
Dreamer V1
Dreamer V3
I-JEPA
RT-2
Dreamer V4
V-JEPA 2
LeJEPA
2018
2020
2023
2025
2019
2021
2024
PlaNet
Dreamer V2
V-JEPA
Genie

World Model


World Models
Dreamer V1
Dreamer V3
I-JEPA
RT-2
Dreamer V4
V-JEPA 2
LeJEPA
LeWorldModel
2018
2020
2023
2025
2026
2019
2021
2024
PlaNet
Dreamer V2
V-JEPA
Genie

World Model


Unified training recipe across many domains with fixed hyperparameters
Symlog/symexp targets improve stability across reward scales
Predicts masked image regions in latent representation space rather than pixels, establishing JEPA's non-generative semantic prediction paradigm for visual world modeling.
Replaces RSSM with diffusion/transformer-based world model
Combines offline pretraining, behavior cloning, and RL (PMPO)
Scales latent video prediction and adds action-conditioned planning, turning JEPA from representation learning into a stronger physical world model for prediction and control.
World Models
Dreamer V1
Dreamer V3
I-JEPA
RT-2
Dreamer V4
V-JEPA 2
LeJEPA
LeWorldModel
2018
2020
2023
2025
2026
Replaces search/planning with actor-critic trained entirely in imagination
Keeps PlaNet-style RSSM world model, but changes how behavior is learned
Co-finetunes web-scale vision-language knowledge with robot trajectories, tokenizing actions as language and popularizing the VLA paradigm for grounded embodied control.
2019
2021
2024
First strong demonstration of planning directly in latent space (RSSM)
PlaNet
Introduces discrete latent states (categorical RSSM)
Adds KL balancing + entropy regularization for stability
Dreamer V2
Extends JEPA to video by predicting masked latent video features, learning temporal visual representations without pixel reconstruction, text supervision, or contrastive learning.
V-JEPA
Genie
Learns an action-controllable generative environment from unlabeled videos, shifting video world models from passive prediction toward interactive, playable world simulation.

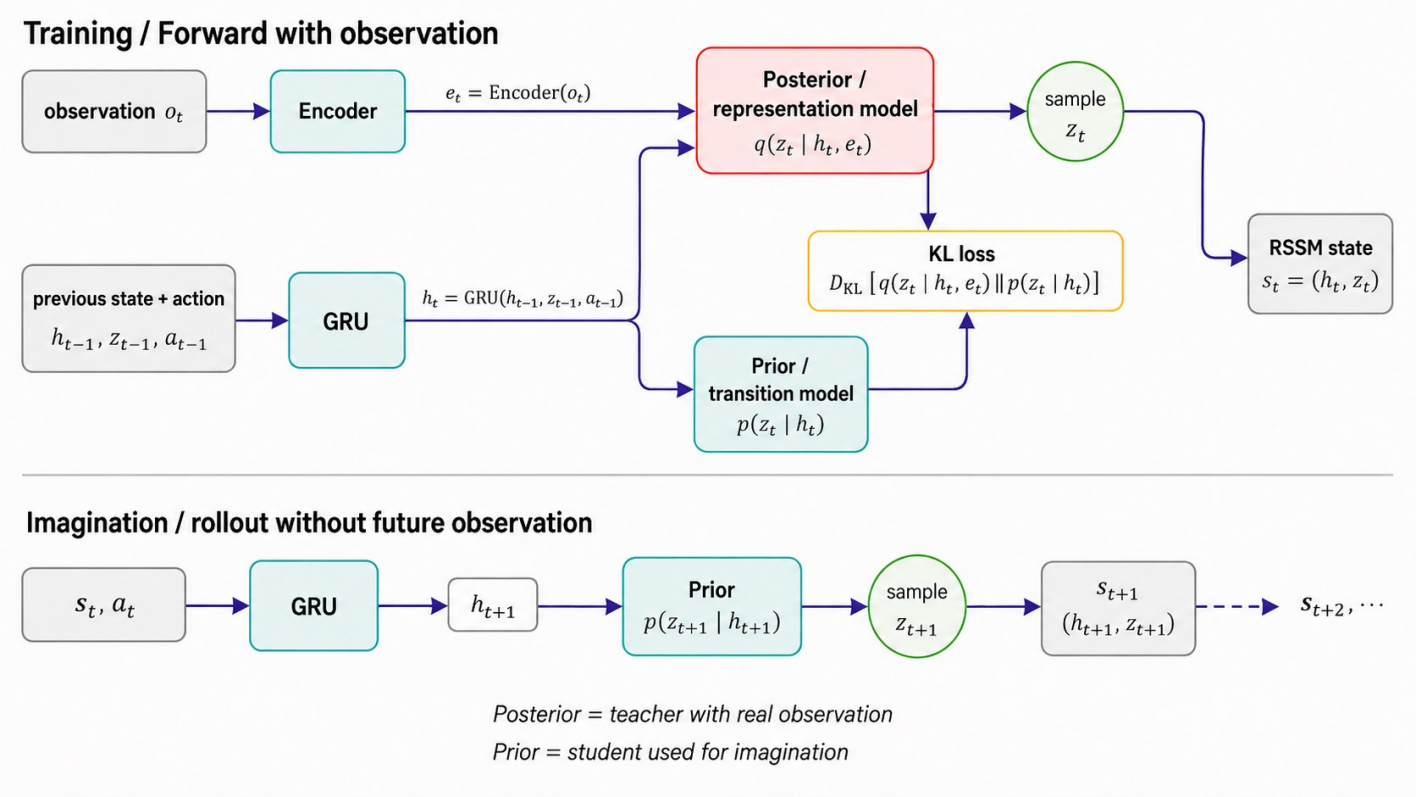
来源于 PlaNet

Recurrent State-Space Model (RSSM)

- 目标：从真实交互序列中学习一个latent world model：
  $(\boldsymbol{observation_t, action_t, reward_t, done_t})_{t=1}^{T}$

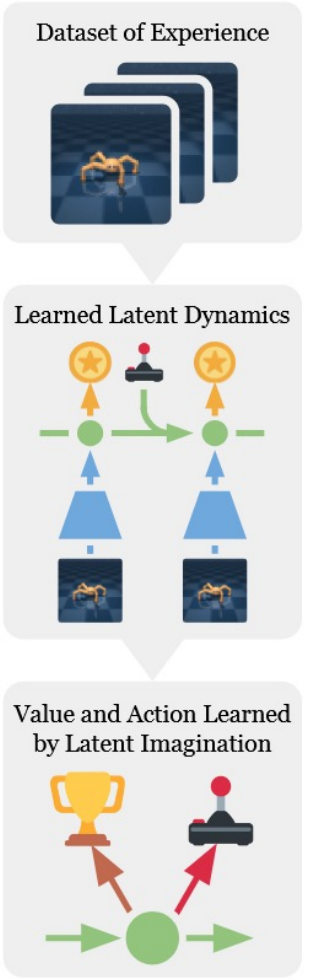
- 学到的是一个latent state：$\boldsymbol{s_t = (h_t, z_t)}$，其中$h_t$为deterministic history summary，z_t为uncertain latent state
  - *同时拥有长期记忆能力（历史信息）和不确定性建模能力（当前信息）

Recurrent State-Space Model (RSSM)

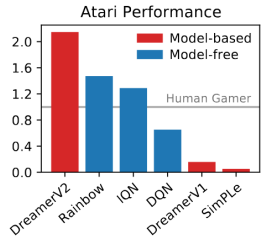


Dreamer V1

- 保留 PlaNet 工作中提出的 **RSSM**，但认为其 online planning (CEM) **控制策略**的效率较低，改成了 **actor-critic**。
  - 直接用 prior dynamics 在 latent space 生成 imagined trajectories，避免了需要在真实环境里或者用 CEM 进行反复搜索。
  - 在 imagined trajectories 上计算 imagined rewards，然后更新 policy (actor) 和 value function (critic)，所有梯度更新全部在 latent space 中进行。
- 训练部分在思想上是经典 actor-critic：
  - Actor / Action model: $q_\phi(a_\tau \mid s_\tau)$
    - 最大化value：$\max_\phi \mathbb{E}_{q_\theta, q_\phi}[\sum_{\tau=t}^{t+H} V_\lambda(s_\tau)]$;
  - Critic / Value model: $v_\psi(s_\tau)$
    - 学 value：$\min_\psi \mathbb{E}_{q_\theta, q_\phi}\left[\sum_{\tau=t}^{t+H} \frac{1}{2}|v_\psi(s_\tau) - V_\lambda(s_\tau)|^2\right]$，MSE loss，TD-style learning (target 是 imagined rewards + future value bootstrap)。
- 总结：Dreamer V1 = model-based actor-critic in latent imagination。
- 主要实验：DeepMind Control Suite 里面的 20 个视觉控制任务，基本都是连续动作。



Dreamer V2

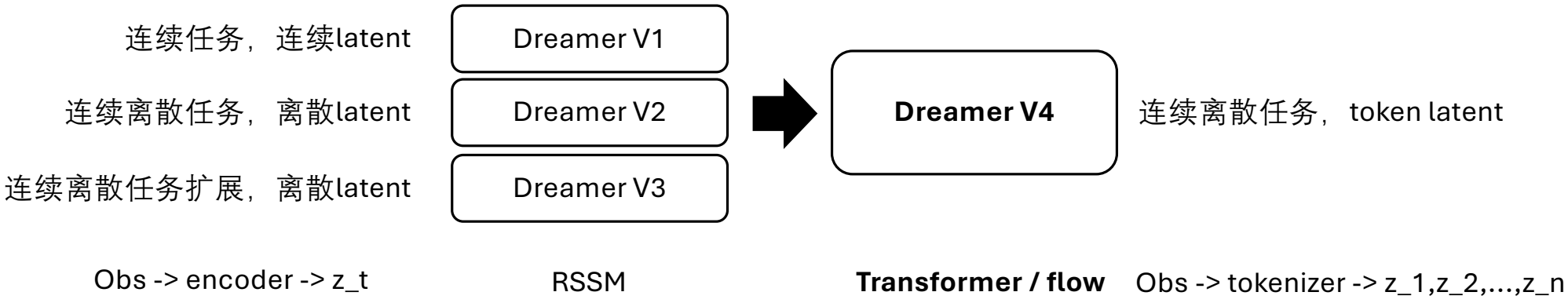


- 保留 Dreamer V1 的 actor-critic imagination 框架，但把训练机制修改，目的是能够打 Atari。
  - Atari benchmark 让研究者训练 AI agent 玩 Atari 2600 (一批很老的 2D 电视游戏)。很适合被形式化成从像素输入出发的离散 RL 问题。
  - Dreamer V2 表明 model-based agent 也能在这个 benchmark 上和传统 model-free 方法进行有力竞争。
- 具体改进：
  - 1. 建立在 PlaNet / Dreamer V1 的 world model 上，但把随机 gaussian latents 换成 **categorical variables**。
    - Insight：世界变化具有很强的离散结构。
  - 2. 用 **KL balancing** 改善了 RSSM prior 的学习，调整训练梯度的分配。
    - $\mathcal{L}_{\mathcal{KL}} = \alpha \,\mathrm{KL}\left(\mathrm{sg}\left(q_\phi(z_t \mid h_t, x_t)\right) \,\middle\|\, p_\phi(z_t \mid h_t)\right) + (1-\alpha)\,\mathrm{KL}\left(q_\phi(z_t \mid h_t, x_t) \,\middle\|\, \mathrm{sg}\left(p_\phi(z_t \mid h_t)\right)\right), \quad \alpha = 0.8$
  - 3. actor 训练改成适合离散动作（不是连续可微）的 REINFORCE-style actor learning。
  - 4. world model 加入 discount / continuation prediction，使 imagination 中的 value learning 更接近真实游戏过程，让模型知道一个 episode 是否结束。
  - 注意：actor / critic 训练过程中，world model 是 fixed 的，因此控制策略的学习不影响 latent representation。

Dreamer V3

- 主框架仍然不变，world model (RSSM) + actor + critic；具体任务从 Atari 扩展到 ProcGen, DMLab, Minecraft 等，都能 work。相比于前代工作的核心是跨 domain 任务的能力。最终在 8 类 domain，150+ 个任务上整体超过大量专精型模型，最有代表的结果是在 Minecraft 中从零学会挖钻石。
- 具体改进：
  - KL balancing → dynamic loss + representation loss + free bits
    - $\mathcal{L}_{dyn} = \max\left(1, \mathrm{KL}\left(\mathrm{sg}\left(q_\phi(z_t \mid h_t, x_t)\right) \,||\, p_\phi(z_t \mid h_t)\right)\right)$, $\mathcal{L}_{rep} = \max\left(1, \mathrm{KL}\left(q_\phi(z_t \mid h_t, x_t) \,||\, \mathrm{sg}\left(p_\phi(z_t \mid h_t)\right)\right)\right)$
    - 不同环境对 representation regularization 的需求不同，比如复杂 3D 环境中有很多控制无关细节，需要更强正则；静态背景游戏中单个像素可能很重要，需要弱正则。
  - Symlog / symexp transform，解决不同任务数值尺度差太多的问题。
    - 不再让网络直接预测原始尺度的巨大 reward / value，而是在压缩后的数值空间中学习，之后再变换回去。
  - Critic 不再预测单个标量，而是用 symexp twohot / categorical distribution 表示 return。
    - Return 不再被网络直接回归，而是让它做分类式预测（这个 return 的数值落在哪个 bin 的范围内？），梯度更加稳定。
  - ……
    - 很多其他工程性的改良，目的是跨 domain 的任务的学习表现更加稳定。
- 注意：
  - Ablation 显示，Dreamer V3 的表现仍然依赖 RSSM 的 reconstruction loss，和很多传统 RL 算法的性能依赖 reward / value prediction 不同。
  - Dreamer V1-V3 始终强调**生成/重建**的路线，和 JEPA 框架本质不同。

Dreamer 系列的范式转变

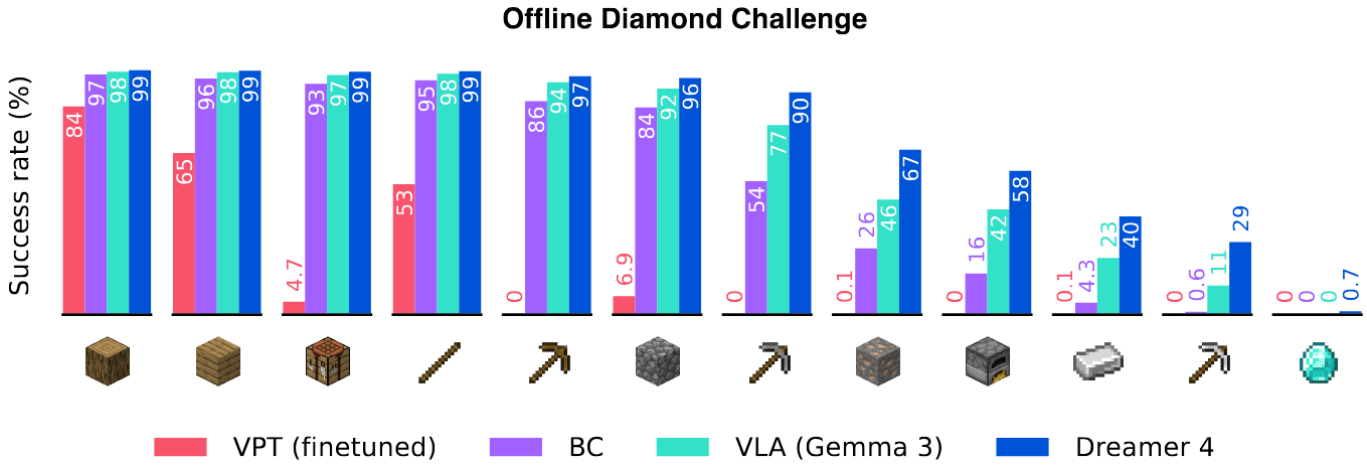


latent 组织方式完全变化。模型架构完全变化。Variational model 也变成 **diffusion-based** models。

“RSSM 对于很多简单任务仍然很好，但已经无法支撑这种高分辨率、大规模、多样化视频数据。” - Danijar Hafner

Dreamer V4

- 把 RSSM latent world model 改成 **scalable transformer / flow-based video world model**
- 目标： To train a strong world simulator with offline video data, so that an agent can be trained within this simulator



Phase 1

Dreamer V4
World model 部分

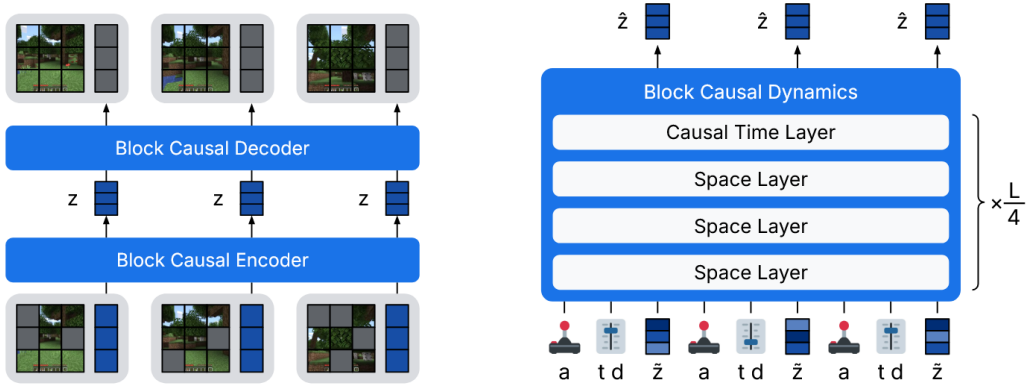
$$h_t = f(h_{t-1}, z_{t-1}, a_{t-1})$$

$$z_t \sim p(z_t \mid h_t)$$

$$\text{video} \xrightarrow{\text{tokenizer}} z_t$$

$$(z_{\leq t}, a_{\leq t}) \xrightarrow{\text{transformer dynamics}} z_{t+1}$$

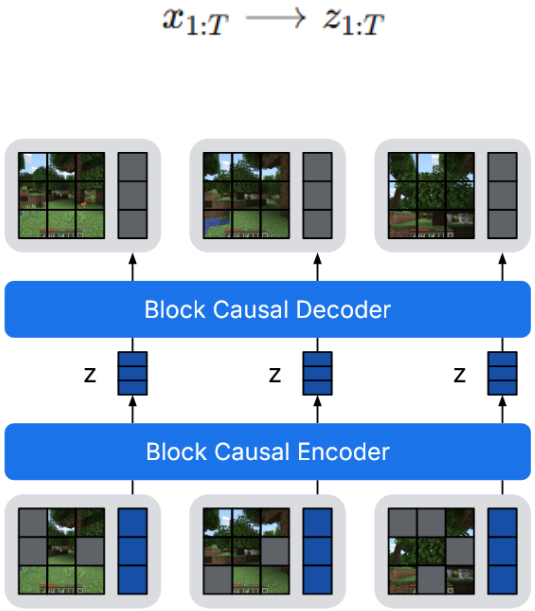
- 在 world model 部分，不再使用 RSSM，改为：
  - 1. **Causal tokenizer**：original video → continuous latent representation
  - 2. **Interactive dynamics model**：$p(future\ latent \mid past\ latents, action)$



(a) Causal Tokenizer (b) Interactive Dynamics

Causal Tokenizer

- **视频版** MAE tokenizer
  - 如果是 image tokenizer，每一帧是独立处理的：$z_t = E(x_t)$，没有时间关系。
  - 对于视频数据 $x_{1:T}$，先进行patchify，然后和 latent tokens 放在一起构成一个序列 $[X_1, L_1, X_2, L_2, \cdots, X_T, L_T]$，有两种 attention 来处理信息：
    - **空间 attention**：在同一帧内部，patch tokens 和 latent tokens 处理空间信息 X_t, L_t
    - **时间 attention**： latent tokens 还可以沿时间维度看，即 $L_t \leftarrow L_{<t}, X_{\leq t}$
  - 可理解成：每一帧内部做“空间压缩”；帧与帧之间做“因果时间信息传递”。
- 随机 drop 一些 patch，用 learned mask embedding 代替，来提升生成视频的空间一致性。
- 得到 L 后，过一层 linear projection 降维，并接 tanh activation 最终得到 z
- z 再经过 decoder 做重建，目标： $\mathcal{L}(\theta) = \mathcal{L}_{\text{MSE}}(\theta) + 0.2\mathcal{L}_{\text{LPIPS}}(\theta)$



(a) Causal Tokenizer

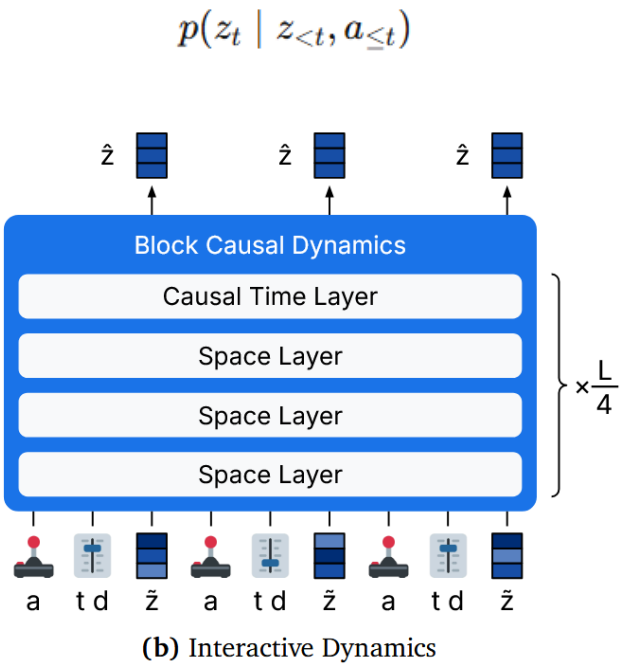
Interactive Dynamics Model

- Dynamics model 完全在上一步得到的 z 空间中学习。
- Shortcut flow matching model
- x-prediction（v-prediction 对一次性生成整个视频很有效，但长视频逐帧 rollout 时，高频误差容易累积）

- 核心：不同帧之间的关联
  - block-causal transformer 的 causal time attention + autoregressive rollout

$$\widehat{z_t^1} = f_\theta(z_{\leq t}, a_{\leq t}, \tau_{\leq t}, d_{\leq t})$$

  - 训练阶段，diffusion forcing 给不同视频帧不同程度的噪声 corruption，让模型学到每帧不同噪声水平的序列去噪能力。



(b) Interactive Dynamics

Agent Finetuning

- Causal tokenizer + interactive dynamics model 训练完成后，可以认为 Dreamer V4 world model pretraining 阶段完成。
- 这一阶段已经学到 $x_t \to z_t$ 和 $p(z_t|z_{<t}, a_{\leq t})$。但是，同一个视觉状态下，不同任务应该对应不同动作，所以模型需要读取 task condition，学习 task-conditioned policy 和 reward model。
- 下一阶段做法：在 dynamics model 中，插入 task / agent 相关 token，然后接 action head 和 policy head，让同一个模型能预测 $p(\text{action}, \text{reward} \mid \text{task})$

Agent Finetuning

- 这一阶段 model 的输入主要是 latent, action, **task / agent**, noisy representation, τ, d等
- Agent / task tokens 可以认为是理解世界状态，做决策和评估的 token（不是用来改变世界 dynamics 的）。
- **保护世界模型的因果关系**：单向 attention 设计。Agent tokens 可以看自身和其他所有模态信息，但其他任何信息不能看 agent tokens。因为未来的世界状态应该只能被 action 直接影响。

Agent Finetuning

- 保留预训练阶段的 video prediction 监督，增加两类监督：
  - Action （键盘 23 个 binary distribution、鼠标 121 类 categorical distribution）
    - $p_\theta(a_{t+n} \mid h_t)$
  - Reward（未来几步内，是否完成了某个事件？比如是否砍了树）
    - $\mathcal{L}(\theta) = -\sum_{n=0}^{L} \ln p_\theta(a_{t+n} \mid h_t) - \sum_{n=0}^{L} \ln p_\theta(r_{t+n} \mid h_t)$，这里设 $L = 8$
- Agent finetuning 阶段训练出来的模型，对于当前世界状态能够输出一串 $L = 8$ 的 action 序列和 reward 序列。
- 重点：学的是模仿数据行为的 policy（**behavioral cloning policy**），而不是直接优化哪个 action 最能完成任务或者最大化长期 reward。

Imagination RL

（类似 Dreamer V1-V3 的经典 model-based actor-critic RL）

- 1. World model 主体冻结。
- 2. 从真实数据中取一段 context。
- 3. 用当前 policy head 采样 action，用 world model 根据 action 生成下一步 z。
- 4. reward head 给 imagined state/action 标注 reward。
- 5. value head 估计长期价值。
- 6. 用 RL loss 更新 policy/value head。

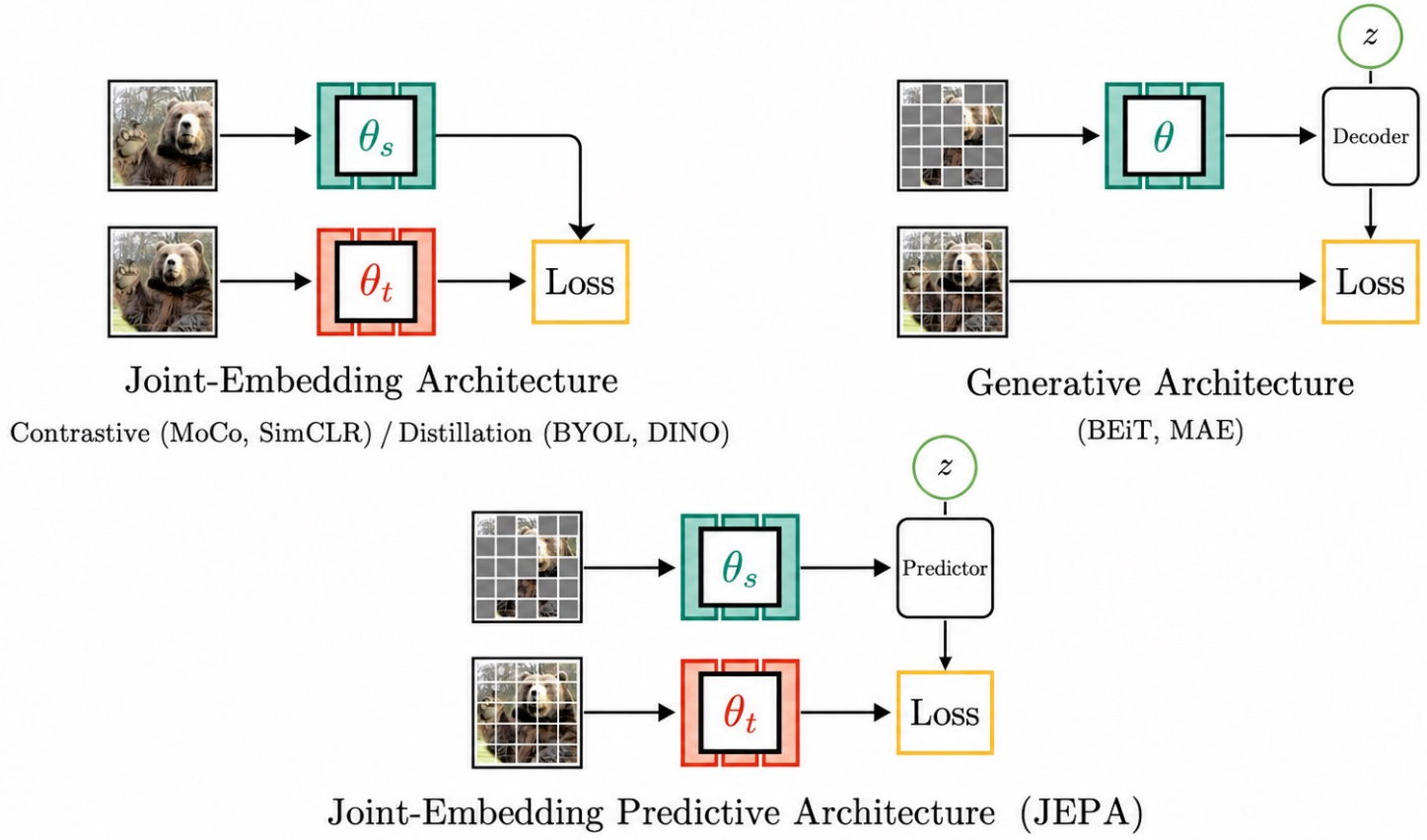
思考

- Action-conditioned world model 需要 agent，但 world model 需要agent吗？
- World 本身是独立于 agent 存在的？

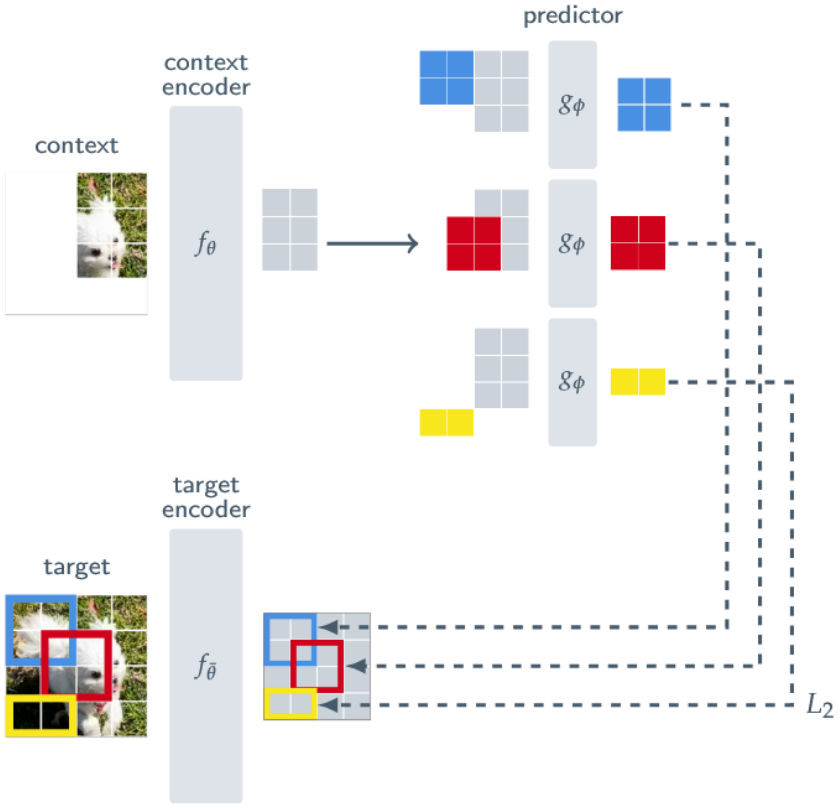
思考

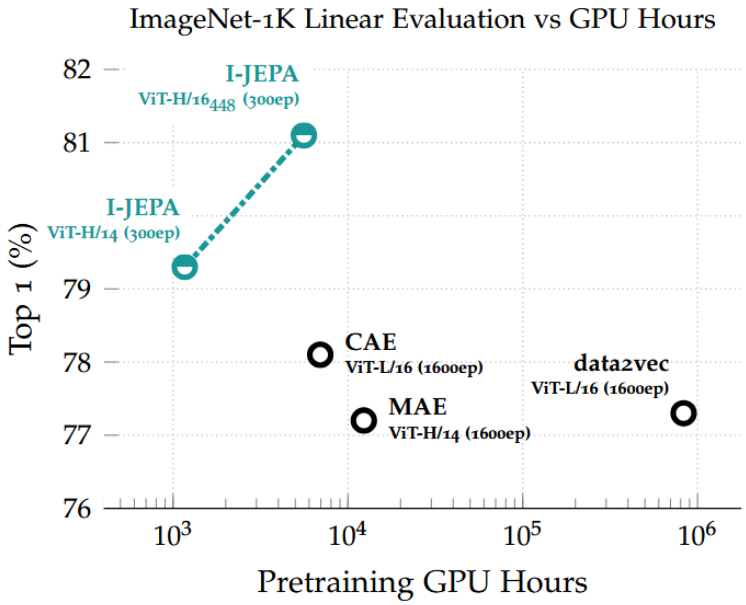
- Action-conditioned world model 需要 agent，但 world model 需要agent吗？
- World 本身是独立于 agent 存在的？
- 客观规律（如牛顿定律）不受主观意志、agent的observation、action的影响而存在。
- 但如果进入planning和control，就需要定义agent。

Joint Embedding Predictive Architectures (JEPA)

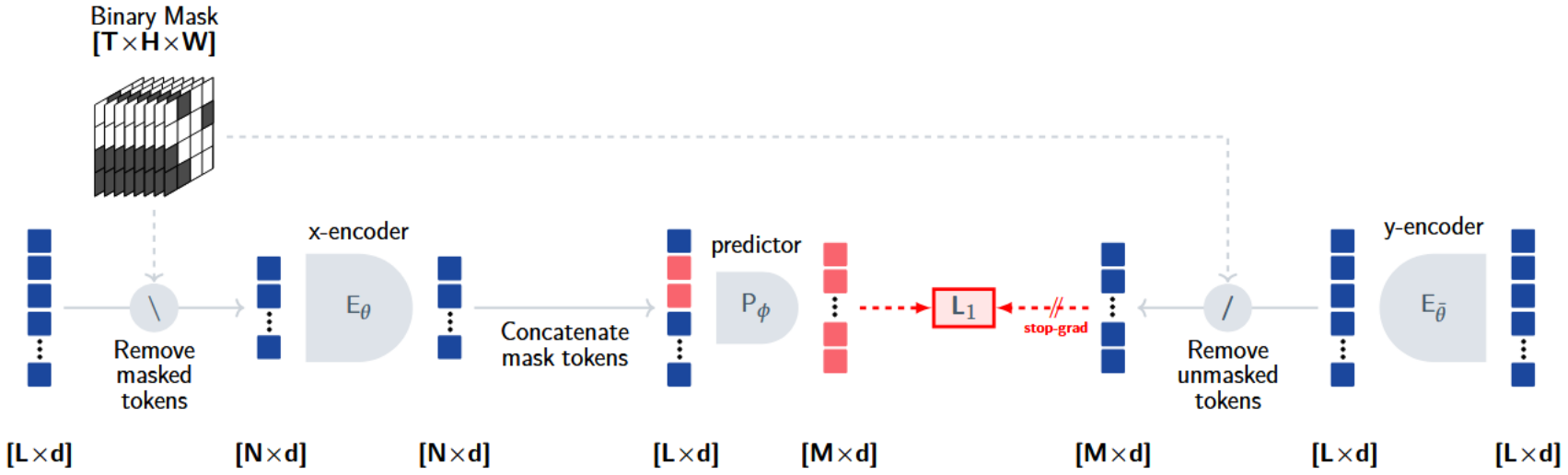


I-JEPA


predictor
context encoder
context
target encoder
target
f_θ
$f_{\bar{\theta}}$
g_ϕ
g_ϕ
g_ϕ
L_2

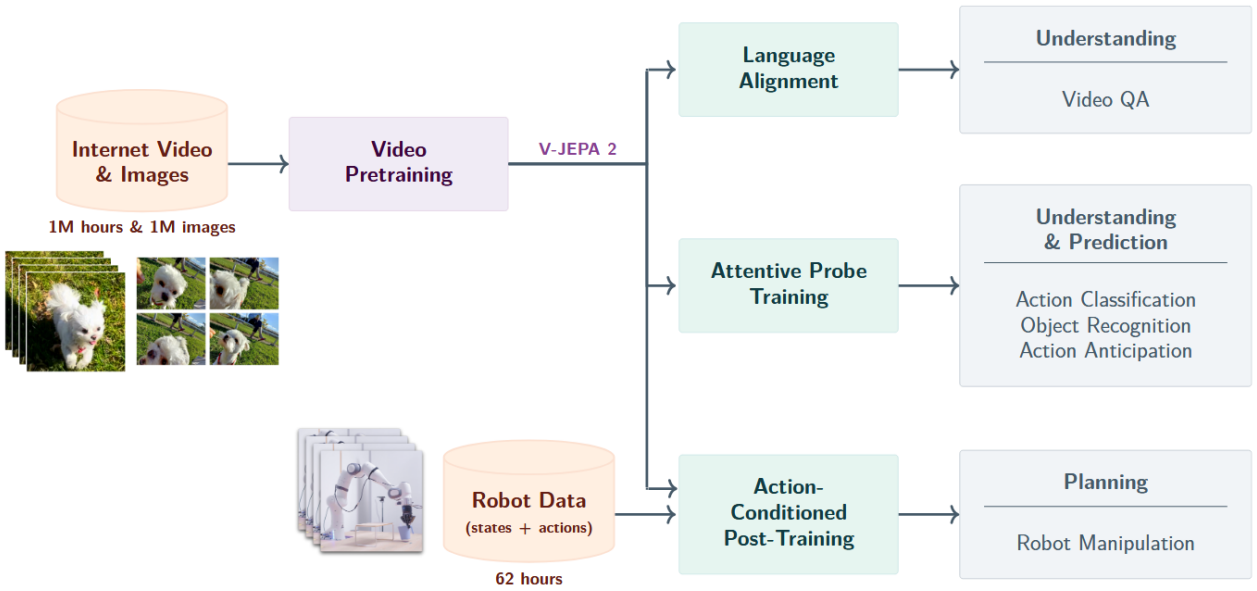

ImageNet-1K Linear Evaluation vs GPU Hours
Top 1 (%)
82
81
79
78
77
76
I-JEPA
ViT-H/16448 (300ep)
I-JEPA
ViT-H/14 (300ep)
CAE
ViT-L/16 (1600ep)
MAE
ViT-H/14 (1600ep)
data2vec
ViT-L/16 (1600ep)
10^3
10^4
10^5
10^6
Pretraining GPU Hours

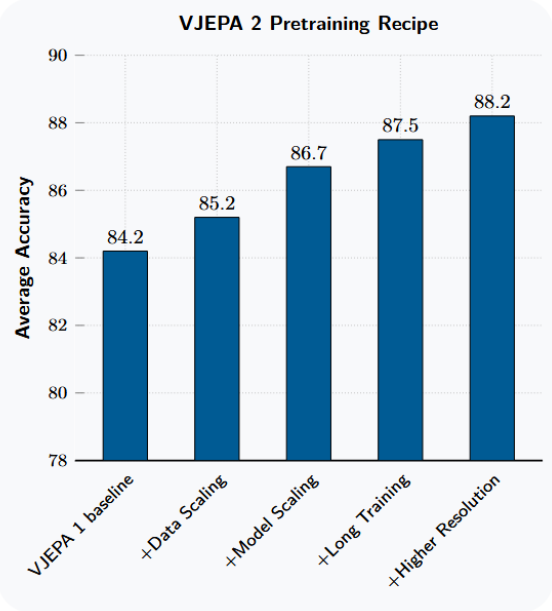
V-JEPA


Binary Mask
[T×H×W]
x-encoder
predictor
y-encoder
Remove masked tokens
Concatenate mask tokens
stop-grad
Remove unmasked tokens
[L×d]
[N×d]
[N×d]
[L×d]
[M×d]
[M×d]
[L×d]
[L×d]

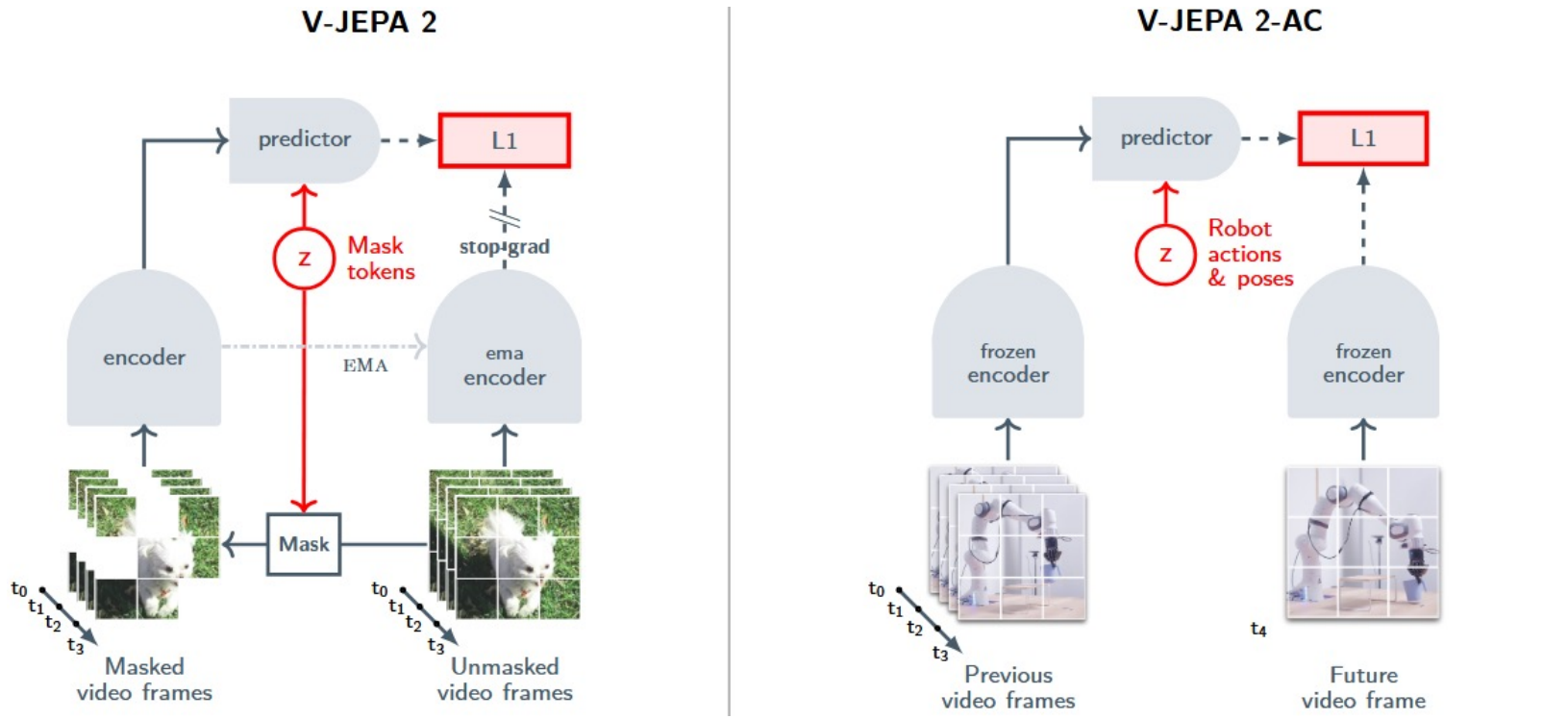
V-JEPA 2

Built upon V-JEPA, focusing on **scaling** and **specific scenarios**





V-JEPA 2-AC (world model)



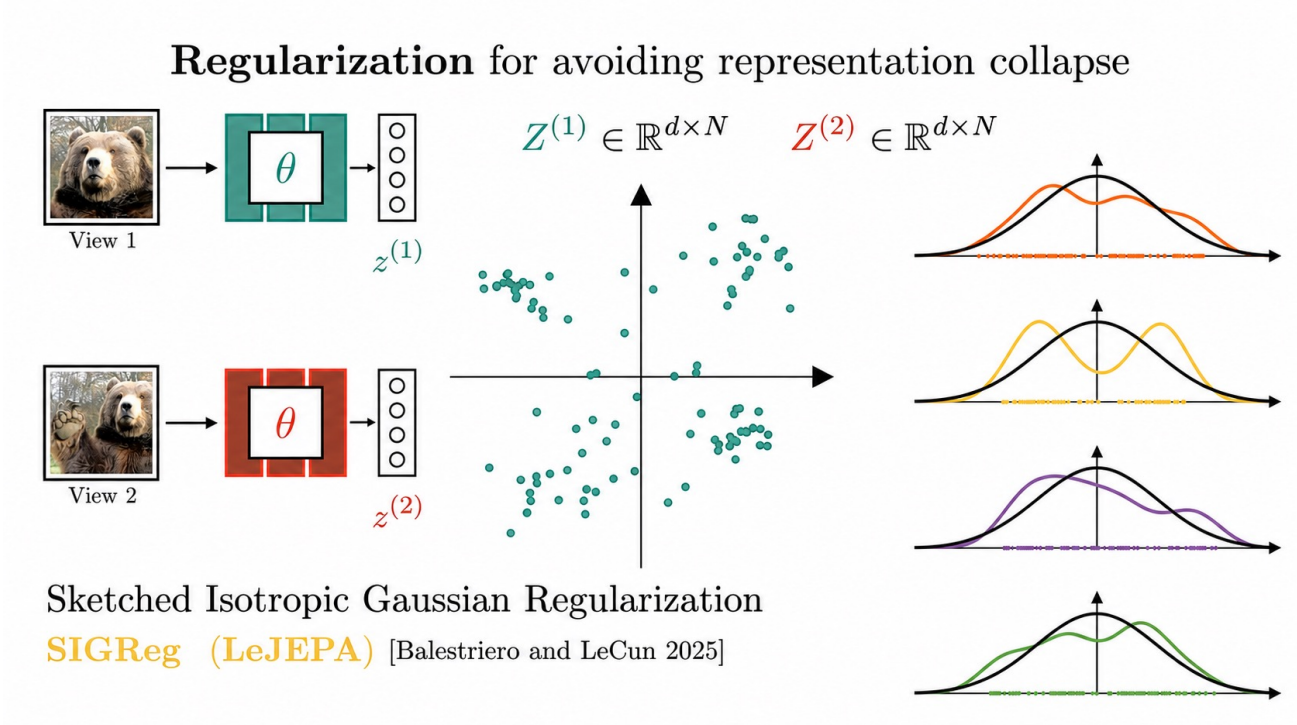
V-JEPA 2's Future work

Future work. There are several important avenues for future work to address limitations of V-JEPA 2. First, in this work we have focused on tasks requiring predictions up to roughly 16 seconds into the future. This enables planning for simpler manipulation tasks, like grasp and reach-with-object, from a single goal image. However, to extend this to longer-horizon tasks such as pick-and-place or even more complex tasks, without requiring sub-goals will require further innovations in modeling. Developing approaches for hierarchical models capable of making predictions across multiple spatial and temporal scales, at different levels of abstraction, is a promising direction.

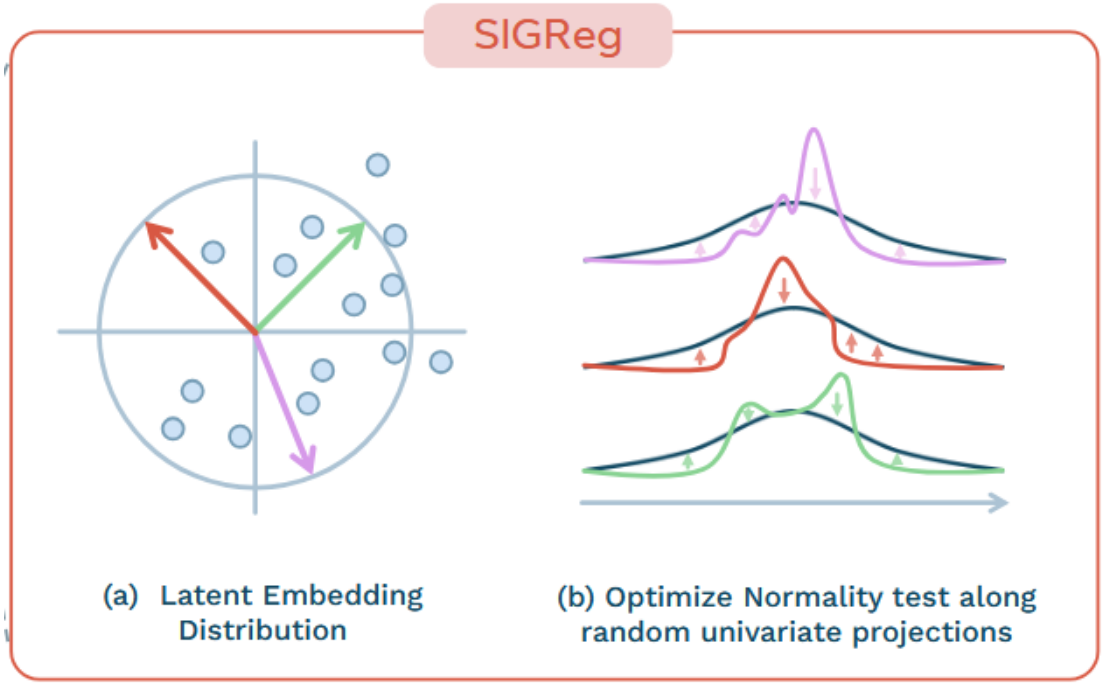
Second, as mentioned in Section 4, V-JEPA 2-AC currently relies upon tasks specified as image goals. Although this may be natural for some tasks, there are other situations where language-based goal specification may be preferable. Extending the V-JEPA 2-AC to accept language-based goals, e.g., by having a model that can embed language-based goals into the V-JEPA 2-AC representation space, is another important direction for future work. The results described in Section 7, aligning V-JEPA 2 with a language model, may serve as a starting point.

Finally, in this work we scaled V-JEPA 2 models up to a modest 1B parameters. The results in Section 2 demonstrated consistent performance improvements while scaling to this level. Previous work has investigated scaling vision encoders to as large as 20B parameters (Zhai et al., 2022; Carreira et al., 2024). Additional work is needed in this direction to develop scalable pre-training recipes that lead to sustained performance improvements with scale.

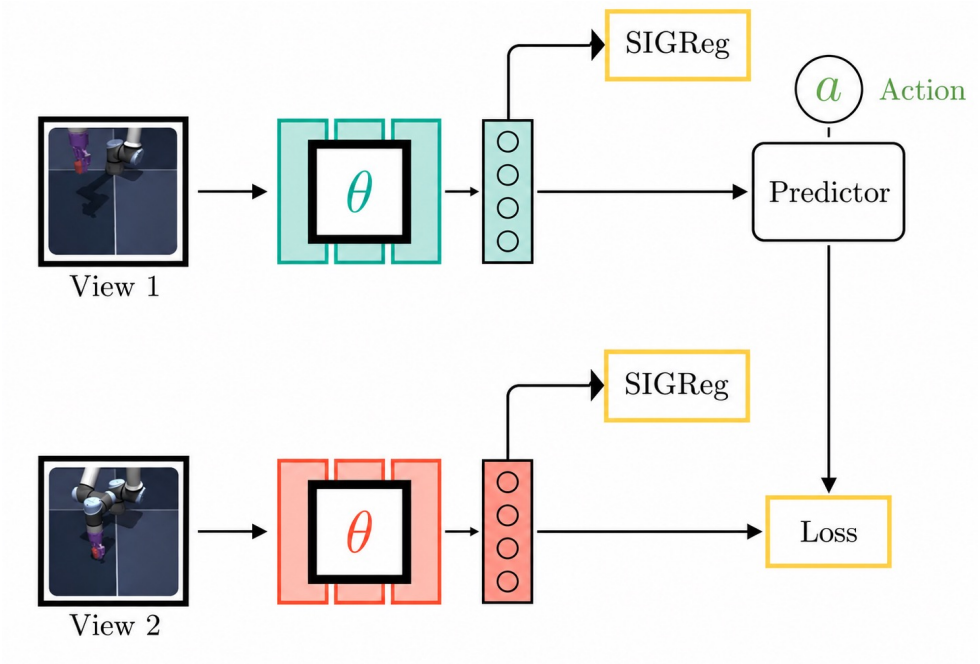
LeJEPA


Regularization for avoiding representation collapse
View 1
View 2
θ
θ
$z^{(1)}$
$z^{(2)}$
$Z^{(1)} \in \mathbb{R}^{d \times N}$
$Z^{(2)} \in \mathbb{R}^{d \times N}$
Sketched Isotropic Gaussian Regularization
SIGReg (LeJEPA) [Balestriero and LeCun 2025]

LeJEPA


SIGReg
(a) Latent Embedding Distribution
(b) Optimize Normality test along random univariate projections

LeWorldModel



补充材料

- World Model & VLA 模型综述：
  - https://song2yu.github.io/world-model-vla/
- **Robotics & World Models Reading Club 07 — Los Altos**
  - https://luma.com/srhe0vuo
- Danijar Hafner (Dreamer 系列作者) 主页
  - https://danijar.com/
- Other JEPA resources
  - LeCun talk at Harvard： https://www.youtube.com/watch?v=yUmDRxV0krg
  - A Path Towards Autonomous Machine Intelligence：
    - https://openreview.net/pdf?id=BZ5a1r-kVsf
    - https://awxlong.github.io/assets/pdf/autonomous-ml-lecunnpdf.pdf
    - https://zhuanlan.zhihu.com/p/535157931
    - https://www.youtube.com/watch?v=OKkEdTchsiE
  - Deep Dive into Yann LeCun's JEPA： https://rohitbandaru.github.io/blog/JEPA-Deep-Dive/
  - **How AI Learned to Teach Itself [JEPA] (By Jia-Bin Huang)**：
    https://www.youtube.com/watch?v=gVEr2cnDE_8&t=19s