RL as Internal Regularization Preventing JEPA Representation Collapse

Jiacan Yu
Johns Hopkins University
jyu197@jh.edu

Siyi Chen
Johns Hopkins University
schen357@jhu.edu

Abstract

We investigate the representation collapse phenomenon in Joint Embedding Predictive Architectures (JEPA). We study a setting where a JEPA encoder is integrated into a reinforcement learning (RL) pipeline as part of a policy network. Our theoretical analysis demonstrates that under such a setup, a partially collapsed encoder cannot be a global optimum when trained jointly with an RL objective. This suggests that the RL objective can act as an effective mechanism to prevent encoder collapse. We hypothesize that, rather than being a failure mode, representation collapse may indicate an inherent tendency toward simplicity in the learned representation space. While the simplicity of the resulting representations needs more experimental study, our work provides theoretical support for this possibility and motivates future investigation.

1 Introduction

PLDM (Planning with Latent Dynamics Model) (Sobal et al., 2025), or DINO-WM (Zhou et al., 2025), is a way to generate actions from a dynamics model, also known as a world model. It adopts a Joint Embedding Predictive Architecture (JEPA), as shown by the square in Fig.1. During training, there is an observation s_t of the environment at every time step t. The observation is usually an image of the environment. The encoder E_ϕ turns it into the encoding vector z_t. The dynamics model W_ψ takes the encoding vector and an action a_t and predicts the resulting vector $\hat{z}_{t+1}$. The loss is a measure of the difference between $\hat{z}_{t+1}$ and z_{t+1} obtained by encoding the observation s_{t+1} after actually performing a_t in the environment. During test time, we provide the initial observation as s_t, and the desired goal observation as s_{t+1}. They are encoded by E_ϕ, which gives z_t and z_{t+1}. We do a search in the action space of W_ψ (e.g. using gradient descent) to find the action that, once performed on z_t, will lead to a vector that is closest to z_{t+1}.

A problem with JEPA is that if we do not fix the encoder during the training time, the encoder will degenerate, that is, it will encode every observation to the same vector, which will successfully minimize the loss to 0, but makes the world model useless. This is also known as representation collapse. One way to fix this is to add regularization terms that encourage the increase in variance of the encoding vectors in a batch (Bardes et al., 2022). Another way is to add an additional projection layer to the encoder at $t+1$, and to set the rest of its parameters to the exponential moving average of the encoder at t (Grill et al., 2020). A comprehensive list of current methods to prevent representation collapse in the JEPA architecture is given in Section 2.2 of (Drozdov et al., 2024).

We consider JEPA in an RL framework, as shown in Fig.1; then a solution of representation collapse appears naturally: the RL training objective can work as a regularizer to prevent encoding collapse. This work provides theoretical support. We first prove that, under reasonable assumptions, a fully collapsed encoder is not a global optimum. Then we generalize the proof to show that a partially collapsed encoder also cannot be a global optimum.

We hypothesize that the model will learn a simple representation space of the environment. The RL objective will extract the features necessary to achieve high reward, and the encoder's

$$
\begin{tikzcd}
a_t \sim P_\theta\big(a_t \mid E_\phi(s_t)\big) & & V_\theta\big(E_\phi(s_t)\big) \\
& z_t \arrow[ul, "\pi_\theta"] \arrow[ur, "V_\theta(\cdot)"] \arrow[r, "W_\psi(\,\cdot\,{,}a_t)"] & z_{t+1} \\
& s_t \arrow[u, "E_\phi"] \arrow[r, "a_t"] & s_{t+1} \arrow[u, "E_\phi"]
\end{tikzcd}
$$

Figure 1: The square represents JEPA. A policy network and a value network are put on top of JEPA to train the model in an RL environment, similar to (Kenneweg et al., 2025).

tendency to be degenerate will keep the feature space minimal. Our future work will examine the simplicity of the representation space.

2 Related Work

Kenneweg et al. (2025) examine the feasibility of combining the JEPA architecture with RL. They conducted an experiment in the Cart Pole environment, showing empirically that receiving information about the gradient of the RL loss is better than applying external regularization. Our contribution is complementary. We focus on why RL gradients prevent representation collapse in principle, without relying on any additional regularization. We supply a proof showing that a collapsed encoder cannot be globally optimal once a policy loss is present. We show that the mechanism is task-agnostic as long as the RL environment satisfies our assumptions.

3 Method

Suppose we have an RL environment where the model will need to perform multiple actions until the end of an episode. We will add a policy and value network on top of the JEPA architecture. The policy $\pi_\theta(a_t \mid E_\phi(s_t))$ will predict actions based on the encoding vectors produced by the JEPA architecture. The value network $V(E_\phi(s_t))$ will also be based on the encoding vectors. We will collect the trajectory until the end of the episode $t = T$. Note that we don't use the world model to search for the action.

During training time, we can use some policy gradient method, like A2C, plus a loss term for JEPA to train π_θ, W_ψ, and E_ϕ together. The model will perform N rollouts as described above to get $\{\tau_i\}_{i=1\ldots N}$, where $\tau = (s_0, a_0, s_1, a_1, ..., s_T, a_T)$. A reward will be assigned to each of them. For policy gradient, we will maximize:

$$
\begin{aligned}
\mathcal{J}(\theta, \phi, \psi) = \mathbb{E}_{\{\tau_i\}_{i=1\ldots N}} \Bigg[& \sum_{t=0}^{T-1} \log \pi_\theta(a_t \mid E_\phi(s_t)) \cdot A_t \\
& - \lambda \sum_{t=0}^{T-1} \left\| W_\psi(E_\phi(s_t), a_t) - E_\phi(s_{t+1}) \right\|^2 \Bigg] \qquad (1)
\end{aligned}
$$

where the first term is the standard policy gradient objective, the second term is the loss for the JEPA architecture, A_t is the advantage in policy gradient, and λ is a hyperparameter controlling the weight of the two training objectives. The whole objective is differentiable, so we may train it end-to-end. Especially, the gradient of the policy gradient objective will flow back to the encoder.

This framework can be considered as a normal policy gradient, where the encoder is just the first few layers of the policy and the value network. The difference is that an extra JEPA loss is applied to these layers to help them capture the dynamics of the environment.

3.1 Intuition about Preventing Collapse

For simplicity, let us suppose that the necessary information about the environment is fully observable given the observation at one time step, so now the policy and value only depend on the current time step. Suppose E_ϕ is already degenerate, so it encodes two different observations s_a and s_b to the same vector z_d, then the policy will predict the same probability distribution over actions $P(a_t) = \pi_\theta(a_t|z_d)$. But the optimal action distributions for s_a and s_b are likely to differ. Since the RL objective is to update the policy and the encoder to match these optimal distributions, the encoder can be updated to encode s_a and s_b to different encoding vectors, so that the policy can predict two different probability distributions to approach the two optimal distributions at the same time.

4 Fully Collapsed Encoder

Proposition 1 (Fully collapsed encoder is sub-optimal)**.** *Let*

$$\mathcal{J}(\theta,\phi,\psi) \;=\; \underbrace{\mathcal{J}_{\mathrm{PG}}(\theta,\phi)}_{-\textit{ expected return}} \;+\; \underbrace{\mathcal{J}_{\mathrm{JEPA}}(\phi,\psi)}_{\textit{world-model loss}}$$

be the loss to be minimized. Assume

1. ***Collapsed-policy Improvability.*** *For* every *state-independent policy π_c there exist a state s_Δ and an action a_Δ with $Q^{\pi_c}(s_\Delta, a_\Delta) > V^{\pi_c}(s_\Delta)$ i.e. any constant action distribution is non-greedy somewhere.*
2. ***Lipschitz continuous world model.*** *For every action, $\|W_\psi(z_1,a) - W_\psi(z_2,a)\| \le L\|z_1 - z_2\| \quad \forall z_1, z_2$.*

Then a fully collapsed encoder $E_\phi(s) \equiv z_d$ cannot minimize $\mathcal{J}$ globally.

4.1 Set-up and notation

- Encoder $E_\phi : \mathcal{S} \to \mathbb{R}^d$.
- Policy head is a **single linear** layer: $\ell_\theta(z) = Wz + b$, $\pi_\theta(a\,|\,z) = \mathrm{softmax}_a(\ell_\theta(z))$.
- Under the collapsed encoder each state shares latent z_d and therefore the same action distribution $\pi_c(a) := \pi_\theta(a\,|\,z_d)$.

4.2 A one-dimensional "private channel" for state s_Δ

Introducing the encoder uncollapse. We want to show that lower loss exists when the encoder is uncollapsed. Set

$$E_{\phi'}(s) = \begin{cases} z_d + \varepsilon u & s = s_\Delta, \\ z_d & \text{otherwise,} \end{cases} \qquad 0 < \varepsilon \ll 1.$$

Only the encoding vector of state s_Δ moves by distance ε to the direction of u, which is a special direction that keeps the old logits intact.

1. **The constraints on u.**
$$Wu = 0 \quad \text{and} \quad u^\top z_d = 0.$$
2. **Why such a u exists.** Let
$$\mathcal{N} := \ker W = \{v \in \mathbb{R}^d \mid Wv = 0\}, \qquad \mathcal{H} := z_d^\perp = \{v \in \mathbb{R}^d \mid v^\top z_d = 0\}.$$

- $\mathcal{N}$ is a vector sub-space with $\dim \mathcal{N} = d - \mathrm{rank}(W)$.
- $\mathcal{H}$ is a hyper-plane, hence $\dim \mathcal{H} = d - 1$.

By the dimension formula for intersections of sub-spaces,

$$\dim(\mathcal{N} \cap \mathcal{H}) \;\geq\; \dim \mathcal{N} + \dim \mathcal{H} - d \;=\; (d - \mathrm{rank}\, W) + (d-1) - d \;=\; d - \mathrm{rank}\, W - 1.$$

Therefore the intersection $\mathcal{N} \cap \mathcal{H}$ is non-trivial (contains a non-zero vector) whenever

$$d - \mathrm{rank}(W) \;\geq 2.$$

$\mathrm{rank}(W)$ is at most the number of the actions $|\mathcal{A}|$, but the dimension of the encoding space $d \gg |\mathcal{A}|$, so the above condition is likely to hold.

Designing a weight tweak that targets only s_Δ. We add a correction to the policy weight matrix

$$\delta W := k\, e_{a_\Delta} u^\top, \qquad k > 0 \text{ (free parameter)}$$

where e_{a_Δ} is the one-hot row vector whose a_Δ-th entry is 1.

1. *All unchanged states* $(z = z_d)$.

$$\begin{aligned}(W + \delta W) z_d &= W z_d \;+\; k\, e_{a_\Delta} u^\top z_d \\ &= W z_d \;+\; k\, e_{a_\Delta} \underbrace{(u^\top z_d)}_{=0} \\ &= W z_d \;=\; \ell_\theta(z_d).\end{aligned}$$

   Hence the logits of every state that still maps to z_d remain *exactly* what they were before the tweak.

2. *The moved state* $(z = z_d + \varepsilon u)$.

$$\begin{aligned}(W + \delta W)(z_d + \varepsilon u) &= W z_d + \varepsilon W u \;+\; k\, e_{a_\Delta} u^\top z_d \;+\; k\, \varepsilon e_{a_\Delta} u^\top u \\ &= W z_d + \varepsilon \underbrace{W u}_{=0} + k\, e_{a_\Delta} \underbrace{(u^\top z_d)}_{=0} + k \varepsilon e_{a_\Delta} \underbrace{(u^\top u)}_{=1} \\ &= \ell_\theta(z_d) \;+\; k\varepsilon\, e_{a_\Delta}.\end{aligned}$$

   Thus *only* the logit of action a_Δ in state s_Δ gains an increment

$$\boxed{\Delta \ell = k\, \varepsilon.}$$

4.3 How one extra logit lowers the policy loss

We compute the first–order change in the policy loss.

Soft-max Jacobian. For logits ℓ and probabilities $\pi = \mathrm{softmax}(\ell)$,

$$\frac{\partial \pi_i}{\partial \ell_j} \;=\; \pi_i \big(\delta_{ij} - \pi_j\big).$$

First-order probability shifts. Let $p_i := \pi_c(i)$ be the original probabilities, and set the logit increment on a_Δ to $\Delta\ell := k\varepsilon$ Then

$$\delta \pi_{a_\Delta} = p_{a_\Delta}(1 - p_{a_\Delta})\, \Delta \ell =: \eta \qquad \delta \pi_{j \neq a_\Delta} = -p_j\, p_{a_\Delta}\, \Delta \ell = -\frac{p_j}{1 - p_{a_\Delta}}\, \eta. \tag{2}$$

Change in the state value of s_Δ. Write $Q_i := Q^{\pi_c}(s_\Delta, i)$ and $V^{\pi_c}(s_\Delta) = \sum_i p_i Q_i$. Then

$$\begin{aligned}
\delta V(s_\Delta) &= \sum_i \delta\pi_i \, Q_i \\
&= \eta Q_{a_\Delta} - \frac{\eta}{1 - p_{a_\Delta}} \sum_{j \neq a_\Delta} p_j Q_j \\
&= \eta Q_{a_\Delta} - \frac{\eta}{1 - p_{a_\Delta}} \big[V^{\pi_c} - p_{a_\Delta} Q_{a_\Delta} \big] \\
&= \frac{\eta}{1 - p_{a_\Delta}} \big[Q_{a_\Delta} - V^{\pi_c} \big] \\
&= \frac{\eta}{1 - p_{a_\Delta}} A^{\pi_c}(s_\Delta, a_\Delta).
\end{aligned}$$

Contribution to the global policy objective. Because the loss we minimize is $\mathcal{J}_{\text{PG}} = -\mathbb{E}_s[V^\pi(s)]$, an increase in V reduces the loss. Only s_Δ contributes to first order, so

$$\begin{aligned}
\Delta\big(\mathcal{J}_{\text{PG}}\big) &= -d_\pi(s_\Delta)\, \delta V(s_\Delta) \\
&= -d_\pi(s_\Delta)\, \frac{\eta}{1 - p_{a_\Delta}}\, A^{\pi_c}(s_\Delta, a_\Delta) + O(\varepsilon^2).
\end{aligned}$$

Insert $\eta = p_{a_\Delta}(1 - p_{a_\Delta})\, k\varepsilon$ (from 2) to obtain the linear improvement

$$\boxed{\Delta\mathcal{J}_{\text{PG}} = -k_1\, k\, \varepsilon + O(\varepsilon^2)}, \qquad k_1 := d_\pi(s_\Delta)\, p_{a_\Delta}\, A^{\pi_c}(s_\Delta, a_\Delta) \;>\; 0. \tag{3}$$

Here the minus sign reflects that $\mathcal{J}_{\text{PG}}$ decreases (linear in $k\varepsilon$), while the prefactor k_1 is strictly positive because the advantage is positive by assumption 1.

4.4 How the JEPA loss changes

Let e be the prediction error of the world model before the encoder is uncollapsed, and e' be the prediction error of the world model after the encoder is uncollapsed on the affected state s_Δ.

$$e := \big\| W_\psi(z_d, a) - z_d \big\|, \quad e' := \big\| W_\psi(z_\Delta, a) - z_d \big\|.$$

$$\begin{aligned}
e'_t &= \| W_\psi(z_\Delta, a) - z_d \| \\
&= \| W_\psi(z_\Delta, a) - W_\psi(z_d, a) + W_\psi(z_d, a) - z_d \| \\
&\leq \| W_\psi(z_\Delta, a) - W_\psi(z_d, a) \| + e_t \\
&\leq L\varepsilon + e_t.
\end{aligned}$$

Assumption 2 gives the last inequality. Squaring and expanding:

$$e'^2 \;\leq\; L^2\varepsilon^2 + 2L\varepsilon\, e + e^2 \;\;\Longrightarrow\;\; e'^2 - e^2 \;\leq\; L^2\varepsilon^2 + 2L\varepsilon\, e. \tag{4}$$

Let $E_{\max} := \max e$ (finite during training) and let m be the number of time steps whose state equals s_Δ (necessarily $m \leq H$, the horizon). Summing (4) over those steps gives the JEPA loss increment:

$$\boxed{\Delta\mathcal{J}_{\text{JEPA}} \;\leq\; C\,\varepsilon + C'\varepsilon^2,} \qquad C := 2mL\, E_{\max},\; C' := mL^2. \tag{5}$$

4.5 Choosing k so the policy win beats the JEPA cost.

Combine 5 with 3:

$$\Delta\mathcal{J} = (-k_1 k + C)\,\varepsilon + C'\varepsilon^2 + O(\varepsilon^2).$$

Select $k > \dfrac{2C}{k_1}$ and then choose ε so small that the quadratic remainder is negligible. With this choice $\Delta\mathcal{J} < 0$. ■

5 Partially Collapsed Encoder

Theorem 1 considers full representation collapse, where the encoder maps every state to a single latent vector z_d. That result remains too weak for practice: an encoder may collapse only a subset of states. To complete the picture, we show that even this partial collapse cannot be globally optimal.

5.1 Modifications to Previous Proof

We need one additional assumption to show that a partially collapsed encoder is not optimal. We will make modifications to how the weight tweak is designed. The rest of the proof remain the same.

Proposition 2 (Two-state collapse is sub-optimal)**.** *Let $\mathcal{J} = \mathcal{J}_{\mathrm{PG}} + \mathcal{J}_{\mathrm{JEPA}}$ be the total loss. Suppose*

1. ***Non-greedy action.*** *For the two collapsed states, here exists an action a_Δ with $A^{\pi_c}(s_a, a_\Delta) > 0$.*
2. ***Lipschitz world model.*** *For all $a \in \mathcal{A}$, $\|W_\psi(z_1, a) - W_\psi(z_2, a)\| \le L\|z_1 - z_2\| \quad \forall z_1, z_2 \in \mathbb{R}^d$.*
3. ***Two-dimensional latent slack.*** *The encoder outputs do not exhaust latent space:*

$$\dim \mathcal{S} \;\le\; d - \mathcal{A} - 2.$$

Then any encoder that maps s_a and s_b to the same code z_d cannot be a global minimizer of $\mathcal{J}$.

5.2 Set-up

Assume the encoder collapses two distinct states $s_a \neq s_b$ into the same encoding vector:

$$E_\phi(s_a) = E_\phi(s_b) = z_d,$$

while the embeddings of all other states may or may not equal z_d.

Set the uncollapsed encoder

$$E_{\phi'}(s) = \begin{cases} z_d + \varepsilon u & s = s_a, \\ E_\phi(s) & \text{otherwise,} \end{cases} \qquad 0 < \varepsilon \ll 1.$$

Let

$$P := \big[\, z_d,\ z(s_1),\ z(s_2), \ldots \big] \in \mathbb{R}^{d \times K}$$

collect all the encoding vectors except $E_\phi(s_a)$. Here $z(s_i) = E_\phi(s_i)$ for each remaining state s_i. Define $\mathcal{S} := \operatorname{span}(P)$ and $\mathcal{N} := \mathcal{S}^\perp$. By Assumption 3, $\mathcal{N}$ is non-trivial.

5.3 Designing a weight tweak that isolates only s_a

Pick a direction u. Consider the sub-space

$$\mathcal{U} \;:=\; \ker W \;\cap\; z_d^{\perp}.$$

Because W is a single linear layer with $\operatorname{rank}(W) \le |\mathcal{A}|$, its kernel has dimension $\dim\ker W \ge d - |\mathcal{A}|$. Intersecting with the hyper-plane $z_d^{\perp}$ reduces dimension by at most one, giving

$$\dim\mathcal{U} \;\ge\; d - |\mathcal{A}| - 1.$$

Assumption 3 meanwhile bounds the span: $\dim\mathcal{S} \le d - |\mathcal{A}| - 2$. Hence

$$\dim\mathcal{U} \;>\; \dim\mathcal{S} \quad\Longrightarrow\quad \mathcal{U}\setminus\mathcal{S} \text{ is not trivial.}$$

We may therefore choose a *unit* vector

$$u \in \mathcal{U}\setminus\mathcal{S}, \quad \text{i.e. } Wu = 0,\; u^{\top} z_d = 0,\; u \notin \mathcal{S}.$$

Property $u \notin \mathcal{S}$ is crucial later: it guarantees that we can find a companion vector $w \in \mathcal{N}$ with $w^{\top} u \neq 0$.

Find a compensator v. Because $\dim\mathcal{N} \ge 2 + |\mathcal{A}|$, pick $w \in \mathcal{N}$ such that $w^{\top} u \neq 0$ and set

$$v := -u + w.$$

Then

$$(u^{\top} + v^{\top})P = w^{\top} P = 0, \qquad 1 + v^{\top} u = w^{\top} u \neq 0.$$

Rank-1 correction. Define

$$\boxed{\delta W := k\, e_{a_\Delta}\big(u^{\top} + v^{\top}\big), \qquad k > 0 \text{ free parameter}}$$

5.4 Effect on logits

For all other states $s \neq s_a$, which are encoded to $E_\phi(s) = z$, we have:

$$(W + \delta W)z = Wz + k\, e_{a_\Delta}(u^{\top} + v^{\top})z = Wz,$$

because every such z is a column of P.

For the moved state s_a, its new embedding is $z_d + \varepsilon u$ with $0 < \varepsilon \ll 1$:

$$\begin{aligned}(W + \delta W)(z_d + \varepsilon u) &= Wz_d + \varepsilon W u + k\, e_{a_\Delta}(u^{\top} + v^{\top})z_d + k\varepsilon e_{a_\Delta}(u^{\top} + v^{\top})u \\ &= Wz_d + k\varepsilon e_{a_\Delta}\big(1 + v^{\top} u\big),\end{aligned}$$

using $Wu = 0$ and $(u^{\top} + v^{\top})z_d = 0$. Because $1 + v^{\top} u = w^{\top} u \neq 0$, the logit of action a_Δ increases by

$$\boxed{\Delta\ell = k\varepsilon\, w^{\top} u \neq 0}.$$

5.5 Finishing the proof

1. *Policy-gradient gain.* Only s_a changes to first order, so $\Delta\mathcal{J}_{\mathrm{PG}} = -k_1' k\varepsilon + O(\varepsilon^2)$ with $k_1' = d_\pi(s_a)\, p_{a_\Delta}\, A^{\pi_c}(s_a, a_\Delta) w^{\top} u > 0$ by Assumption 1.
2. *JEPA cost bound.* The world-model error grows at most $C\varepsilon + C'\varepsilon^2$ (only steps with $s_t = s_a$ are affected), exactly as in the full-collapse proof.
3. *Choose parameters.* Pick $k > 2C/k_1'$ and sufficiently small ε; then $\Delta\mathcal{J} = (-k_1' k + C)\varepsilon + O(\varepsilon^2) < 0$.

Hence the encoder that merges $\{s_a, s_b\}$ is *not* globally optimal, completing the proof of Proposition 2. ■

6 Conclusion

We have found new parameters for the policy head that strictly minimize the loss, so a partially collapsed encoder cannot be globally optimal. Since such parameters exist when the policy head is just a single linear layer, we can expect that they also exist for more complex policy network architectures.

Our assumptions are realistic. Assumption 1 requires that the two collapsed states s_a and s_b need to be handled differently by the policy, that is, they are two meaningfully different states in the RL task. Assumption 2 should be satisfied in practice, since most neural networks are Lipschitz continuous. Assumption 3 should be satisfied in most cases, since a latent space is usually high-dimensional, and encoding vectors usually do not span the entire space. Therefore, our analysis supports the feasibility of preventing representation collapse with RL objective.

7 Acknowledgment

The authors would like to express their deepest gratitude to Prof. Lenhart Schubert for his insightful discussions and invaluable editorial guidance throughout this work. The authors are also grateful to Mingrui Liu for many stimulating conversations. Their support and feedback were essential.

References

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022. URL https://arxiv.org/abs/2105.04906.

Katrina Drozdov, Ravid Shwartz-Ziv, and Yann LeCun. Video representation learning with joint-embedding predictive architectures, 2024. URL https://arxiv.org/abs/2412.10925.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL https://arxiv.org/abs/2006.07733.

Tristan Kenneweg, Philip Kenneweg, and Barbara Hammer. Jepa for rl: Investigating joint-embedding predictive architectures for reinforcement learning, 2025. URL https://arxiv.org/abs/2504.16591.

Vlad Sobal, Wancong Zhang, Kynghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models, 2025. URL https://arxiv.org/abs/2502.14819.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning, 2025. URL https://arxiv.org/abs/2411.04983.