# LANGUAGE-REASONING JEPA: SEARCHING IMAGINED FUTURES OVER LANGUAGE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A language model reasons by generating text about the future and reads its next action from the same likelihood that generated the text, and it has no separate mechanism for imagining what an action would do. Action-conditioned joint-embedding predictive architectures (JEPAs) provide such a mechanism in robot planning, and we ask whether the same construction supports reasoning over language. We read a sentence as a partial observation of a decision process and a derivation step as an action, and its observed consequence is what a JEPA is trained to predict. Transition prediction fixes the dynamics of the latent space but leaves its geometry free. We therefore add a goal-conditioned energy head, trained only by ranking outcomes that the environment has executed, and plan by receding-horizon search over imagined latent trajectories. We call the resulting model Language-Reasoning JEPA (LR-JEPA) and evaluate it in two synthetic environments rendered entirely in language, where every policy selects its next step from the same menu of admissible actions and we measure overshoot, the number of executed actions relative to the shortest solution. On chain-structured deduction, language models trained on the same trajectories are optimal inside the trained lengths and reach overshoot 1.69 to 1.91 beyond them, where a random policy reaches 2.01. The same ranking head on the frozen language model reaches 1.03 at search depth one and falls to the random level by depth four, since a language model cannot roll its state forward, whereas LR-JEPA improves from 1.31 to 1.09 over the same depths. On dependency graphs the search degrades with depth, which we trace to the drift of imagined trajectories.

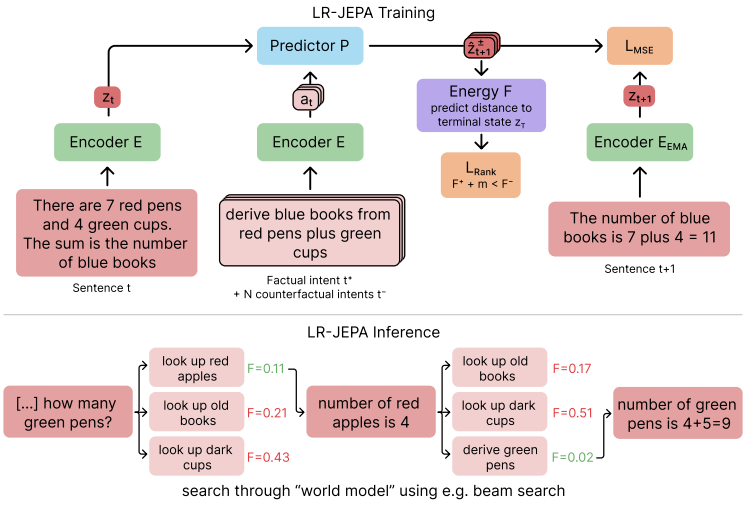


Figure 1: Training (top). The encoder $E_\theta$ maps the history to a state $z_t$, the predictor $P_\varphi$ imagines successors of the factual $(+)$ and $N$ counterfactual $(-)$ intents, and the energy head $F_\psi$ learns to rank the factual endpoint below the counterfactual ones. Inference (bottom) is an energy-guided beam over imagined latent trajectories, committing to the best one's first action.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

# 1 INTRODUCTION

When a language model acts as an agent, its next action is the sampled or most probable continuation, and reinforcement-learning post-training strengthens exactly this readout (OpenAI, 2024; Guo et al., 2025). Such a model reasons by generating text about the future and then reads its decision from the same likelihood that generated the text. The search this induces is disorderly, since sampled chains of thought revisit earlier states and contradict themselves (Wang et al., 2023b; Chen et al., 2025; Yue et al., 2025). The architecture contains no mechanism that imagines what an action would do before the model commits to it.

Such a mechanism exists in robot planning, where action-conditioned joint-embedding predictive architectures (JEPAs) act as world models, and a planner scores their imagined latent futures against a goal and commits to the first action of the best imagined trajectory (LeCun, 2022; Assran et al., 2025; Zhou et al., 2025; Maes et al., 2026). We transfer this construction to language by reading it as the observation channel of a partially observed decision process, in which a sentence is a partial observation, a derivation step is an action, and the sentence that the step produces is its observed effect. The reasoning state is the set of interaction histories that share the same future under every action sequence, and an action-conditioned JEPA is trained to predict the consequence of an action on that state.

Even an accurate predictor does not yet give the planner what it needs, because transition prediction fixes the dynamics of the latent space but leaves its geometry free, so a re-embedding of the latents can reverse which imagined successor lies closer to a goal (Proposition 1). We therefore attach a goal-conditioned energy head and train it only by ranking outcomes that the environment has executed, because a correct local order is all that shortest-path planning requires (Proposition 2). The resulting model, Language-Reasoning JEPA (LR-JEPA), plans with a receding-horizon beam search over imagined latent trajectories, so that search depth is test-time computation alone.

We study LR-JEPA in two synthetic environments rendered entirely in language, a chain-structured deduction task and a DAG-structured arithmetic task. In both the state is observed only through sentences, an action is a short sentence, and its effect is another sentence, and the menu of admissible actions is the same for every policy, so that the only difficulty is which action to take. During training the environments also execute alternative actions, which the ranking supervision requires, and we regard this executor as the assumption that does not yet transfer to natural language. Under this interface we compare LR-JEPA with language models trained on the same text, with looped and sentence-level language models, with likelihood-based lookahead, and with the same ranking head attached to a frozen language model.

We make three contributions:

1. We formulate reasoning over language as a partially observed decision process, with the quotient of histories by equality of futures under every action sequence as its state, and we show that an exact predictor leaves goal-distance orderings unidentified while a correct local order suffices for shortest-path planning.
2. We introduce LR-JEPA, to our knowledge the first action-conditioned JEPA that trains and searches over language, together with an energy head that supplies the missing order. The head is trained by ranking executed counterfactual consequences over hindsight-relabelled goals and all horizons, and it serves as the objective of the planner.
3. We evaluate LR-JEPA against language models from 4.8M to 200M parameters, looped and sentence-level language models, likelihood lookahead, temporal-difference and latent-distance energies, and the same ranking head on a frozen language model, all under one interface, and we separate the contributions of the ranking head, the predictive state and the search depth.

# 2 RELATED WORK

**Planning with learned world models.** Model-based planning learns a simulator of the environment and plans inside it, and latent world models (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020), the value-equivalent model of MuZero (Schrittwieser et al., 2020) and the learned values of TD-MPC2 (Hansen et al., 2024) share one template, in which the planner imagines candidate futures, scores them against a goal, commits to the first action and replans. Joint-embedding predictive architectures apply this template in representation space. A JEPA computes its prediction loss in latent space

(LeCun et al., 2006; LeCun, 2022), as I-JEPA and V-JEPA do for images and video (Assran et al., 2023; Bardes et al., 2024), and conditioning the predictor on an action turns it into a latent world model. V-JEPA 2 plans manipulation zero-shot by scoring imagined futures against a goal image (Assran et al., 2025), DINO-WM plans over frozen visual features (Zhou et al., 2025), and LeWorldModel trains such a model end to end from pixels (Maes et al., 2026). These planners rest on two properties of their observations that language does not offer. An image of the scene contains the whole world the planner acts in, so the state is fully observed, and one image corresponds to one world state, so a distance between latents is a distance between states. A sentence shows one fragment of the state of a derivation, and one state has many wordings. We therefore define the state as a quotient of histories (Section 3) and replace the latent-distance readout by a learned order, which is necessary because the distance readout plans at the random level in our setting (Section 6).

**Goal direction on JEPA latents.** A world model imagines futures, but the planner must also know which imagined future lies closer to the goal. Visual JEPA planners read this from the latent distance to a goal embedding. Recent work learns a value on the latents instead, where TD-JEPA learns latent-predictive representations with temporal-difference targets (Bagatella et al., 2026), and value learning and latent-action planning on JEPA representations follow the same recipe (Destrade et al., 2025; Zhang et al., 2026). We compare against temporal-difference training of our own head under identical architecture, data and planner. Our head learns an order and no value, in a pairwise form that comes from preference learning (Bradley & Terry, 1952; Christiano et al., 2017), and its goals are relabelled to states the trajectory actually reached, as in hindsight experience replay (Andrychowicz et al., 2017) and goal-conditioned supervised and contrastive learning (Ghosh et al., 2021; Eysenbach et al., 2022). It differs from both in what it compares, imagined endpoints of alternatives the environment has executed, and in what it is used for, as the objective of a beam over imagined trajectories. Constraining the score to be a quasimetric (Wang et al., 2023a) is a natural alternative parametrization of such a head, which we ablate.

**Reasoning and test-time search with language models.** An autoregressive language model (Brown et al., 2020) acts by taking its most likely continuation, and test-time search improves on this by rolling the model forward as its own textual world model or by scoring partial derivations with a verifier (Yao et al., 2023; Hao et al., 2023; Lightman et al., 2024; Wan et al., 2024; Lample et al., 2022; Zelikman et al., 2024; Lehnert et al., 2024; Gandhi et al., 2024). Looped transformers share one block across depth and thereby spend test-time computation without generating text (Dehghani et al., 2019; Giannou et al., 2023; Geiping et al., 2025). Latent reasoning moves part of the computation out of token space, as Coconut iterates thought vectors (Hao et al., 2025), large concept models autoregress over sentence embeddings (LCM team et al., 2024), LLM-JEPA adds embedding prediction to language-model finetuning (Huang et al., 2026), and JEPA reasoners pair a latent reasoner with a decoder (Liu et al., 2025). None of these couples the latent state to an action-conditioned dynamics model that is searched at test time. We include likelihood lookahead, looped models and a sentence-level model as baselines, the last being the closest likelihood-trained analogue of our state abstraction. Choice-based text-game agents (He et al., 2016; Hausknecht et al., 2020; Yao et al., 2020) score a menu of textual actions against a text state, as all our policies do, and LR-JEPA inserts an imagined latent rollout between the state and the score.

**Length generalization as the test.** Next-token models trained on short instances fail on longer ones across arithmetic, composition and planning (Anil et al., 2022; Dziri et al., 2023; Zhou et al., 2024; Bachmann & Nagarajan, 2024; Valmeekam et al., 2023). We adopt this protocol as our test and train every model on short problems and evaluate it on longer ones, so that success requires a procedure that extends beyond the trained lengths. Our second environment follows the arithmetic generator of Ye et al. (2025). Its dependency graphs let us vary the structure of the necessary computation while holding everything else fixed.

## 3 Language as a Partially Observed Decision Process

We model reasoning over text as a partially observed decision process $\mathcal{M}$ with semantic state $s_t$, semantic action $u_t$, language observation $x_t$, and language action $a_t$ grounded by a map $\alpha(a_t) = u_t$. The agent conditions on the interaction history

$$h_t = (x_0, a_0, x_1, \ldots, a_{t-1}, x_t).$$

Following predictive state representations (Littman et al., 2001; Singh et al., 2004), let $Y$ collect future task-relevant observations and terminal outcomes, $C$ future costs, and $A$ future feasible-action sets, all up to the episode horizon $\tau$. Histories $h, h'$ are *predictively equivalent* when every policy over actions admissible from both induces the same future law,

$$h \sim_{\text{pred}} h' \iff \mathcal{L}_\pi(Y_{t+1:\tau}, C_{t:\tau}, A_{t:\tau} \mid h) = \mathcal{L}_\pi(Y_{t+1:\tau}, C_{t:\tau}, A_{t:\tau} \mid h') \quad \forall \pi, \tag{1}$$

and the reasoning state is the equivalence class $q(h) = [h]_{\sim_{\text{pred}}}$. Histories with identical futures under every action sequence meet in the quotient whatever their wording, and the quotient retains everything planning needs, since predictively equivalent histories admit the same policies and share every value (Proposition 3, Appendix A).

The supervision that identifies this quotient is the observed consequence of an action, which is exactly what an action-conditioned JEPA is trained to predict. Prediction alone does not recover the quotient, since a constant encoder predicts its collapsed targets perfectly, and Section 4 shows that prediction also underdetermines the geometry that planning needs. Our experiments instantiate $\mathcal{M}$ with deterministic finite-horizon processes in which executing $a$ from $h$ appends one consequence, the successor $T(h, a)$. We leave stochastic environments, where a squared-error predictor recovers only the conditional mean of the successors, to future work.

## 4 Language-Reasoning JEPA

LR-JEPA consists of three jointly trained parts, (1) a predictive backbone that learns the latent dynamics of derivation, (2) an energy head that gives the latent space the order that planning needs, and (3) a receding-horizon search loop (Figure 1). We take actions to be *intent phrases*, short sentences drawn from a finite per-problem catalogue. In FSA the phrase is a rule sentence of the problem, and in iGSM it names the quantity to compute. The environment executes the grounding $\alpha$ of a phrase and appends the consequence. We call every position of a demonstrated trajectory an *anchor*. At each anchor the generator also executes a sample of the other catalogue entries, records their consequences, and continues some of these branches for several steps. These executed alternatives are the *counterfactuals*. They supply negatives for the energy head and transitions for the predictor during training, and at test time no model sees them.

### 4.1 Latent dynamics of derivation

A causal Transformer encoder $E_\theta$ maps the token stream of a history $h$ to hidden states, the reasoning state $z_h$ is the last one, and candidate phrases are encoded in context by the same backbone,

$$z_h = E_\theta(h), \qquad z_a = E_\theta(a \mid h), \qquad \widehat{z}_{h,a} = P_\varphi\big(z_h,\, z_a\big), \tag{2}$$

where the causal Transformer predictor $P_\varphi$ produces the imagined successor and $\theta$, $\varphi$, $\psi$ name the encoder, predictor and head parameters. Following visual JEPAs (Assran et al., 2023; 2025), an exponential moving average (EMA) copy $\bar{E}$ supplies targets held fixed by the stop-gradient $\mathrm{sg}[\cdot]$, and the backbone minimizes a smooth-$\ell_1$ consequence-prediction loss

$$\mathcal{L}_{\text{pred}} = \mathbb{E}_{(h,a)}\Big[\, \ell_1^{\text{sm}}\big(\widehat{z}_{h,a} - \mathrm{sg}\big[\bar{E}(T(h,a))\big]\big)\Big], \tag{3}$$

taken over executed and counterfactual actions, so the predictor learns the consequences of the whole local menu, and all representations are trained from scratch.

### 4.2 Predictive ambiguity

Nothing in Equation 3 constrains latent distances, so even an exact predictor does not determine which imagined state lies closer to a goal.

**Proposition 1** (Predictive ambiguity)**.** *Suppose $P(E(h), z_a) = E(T(h,a))$ on every realized transition, let $Z \subset \mathbb{R}^{d_z}$ be the realized latent states, and let $b : Z \to \mathbb{R}^{d_z}$ be any injection. Then $\widetilde{E} = b \circ E$ and $\widetilde{P}(b(z), z_a) = b(P(z, z_a))$ reproduce every realized transition, and a suitable $b$ reverses any strict goal-distance comparison among distinct realized states, so goal-distance orderings are not identified by transition prediction. (Proof in Appendix A.)*

The statement is an existence claim over the unconstrained function class, and Section 6 measures whether architectural bias shrinks the ambiguity in practice. The goal-distance readout of visual JEPA

planners (Assran et al., 2025; Maes et al., 2026) therefore has no guarantee here, and in our implementation a value function trained by temporal differences (Schrittwieser et al., 2020; Hansen et al., 2024) reaches 1.76 at best at depth eight (Section 6), so we instead learn the comparison directly as an order.

### 4.3 Energy head and ranking supervision

An MLP head $F_\psi(z_h, z', z_g, k)$ scores a possibly imagined state $z'$, reached from anchor $z_h$, against a goal representation $z_g$ at horizon $k$, an energy in the free-energy convention (LeCun et al., 2006; LeCun, 2022), lower is better. Every ranking signal the head receives is a Bradley–Terry-style pairwise constraint (Bradley & Terry, 1952; Christiano et al., 2017) over outcomes the environment actually executed,

$$\mathcal{L}_{\mathrm{rank}}(\mathcal{C}) = \mathbb{E}_{(+,-)\sim\mathcal{C}} \log\Big(1 + \exp\big(\tfrac{1}{\beta}\,(F^{+} - F^{-} + m)\big)\Big), \qquad F^{\pm} = F_\psi\big(z_h, \widehat{z}^{\pm}, z_g, k\big), \quad (4)$$

with margin $m$ and temperature $\beta$. The head scores imagined endpoints $\widehat{z}^{\pm}$ that the predictor produces, and the label of a pair records which continuation the demonstration took, so the supervision consists of executed outcomes although the scored objects are imagined. Three constraint sets instantiate the loss. In *hindsight long-horizon ranking*, written $\mathcal{C}_{\mathrm{hind}}$, we relabel the goal to an observed future state after hindsight replay (Andrychowicz et al., 2017) and require the observed continuation's endpoint to undercut the counterfactual endpoints, for every anchor and every horizon $k$ up to the remaining length. In *feasibility ranking*, written $\mathcal{C}_{\mathrm{feas}}$, the endpoints of feasible actions must undercut infeasible ones, which governs shallow search, and removing $\mathcal{C}_{\mathrm{feas}}$ raises overshoot at depth one from 1.31 to 1.63. In *goal-swap ranking*, written $\mathcal{C}_{\mathrm{swap}}$, an endpoint must score better against its own goal than against another problem's. We train with all three, and Section 6 ablates each in turn. During training the goal input $z_g$ encodes the relabelled future state, and at test time it encodes the problem context, in which the prompt states the goal. The latest relabels of a trajectory lie at its terminal state, which is the state the context describes, and feeding the encoded terminal state itself as the test-time goal changes no result at any depth (Table 4).

Conversely, a correct local order is all that shortest-path planning requires, in the following sense. Let $D^\star(h)$ be the shortest remaining action count in a deterministic unit-cost problem, finite at $h_0$, and call $a$ optimal at $h$ if $D^\star(T(h,a)) = D^\star(h) - 1$.

**Proposition 2** (Local ordering suffices)**.** *If a learned score ranks, at every history reachable from $h_0$ by optimal actions, every optimal action strictly above every suboptimal one, then greedy selection reaches the goal in exactly $D^\star(h_0)$ actions, and a score that differs from such a score by less than $\gamma/2$, where $\gamma > 0$ is its smallest gap, inherits the guarantee. (Proof in Appendix A.)*

Equation 4 orders only sampled counterfactual pairs, so it need not deliver this premise everywhere, and search over imagined rollouts inherits no guarantee, and the proposition covers only greedy selection, so the beam search is a heuristic extension, so Section 6 measures where depth helps and where it fails. We also regularize the latents with VICReg (Bardes et al., 2022), compare it against the SIGReg objective of LeJEPA (Balestriero & LeCun, 2025), and add a horizon-ranking and a counterfactual reconstruction term, with every weight listed in Appendix B. Removing any one of these terms changes overshoot at depth eight by at most 0.02 (Table 5).

### 4.4 Receding-horizon planning

Inference is energy-guided beam search over imagined latent trajectories with a receding horizon (Algorithm 1, Appendix B). From root state $z$ and menu $M$ the predictor rolls each beam forward under continuations $p = (a_1, \ldots, a_\ell)$ drawn from $M$, so that a candidate beyond the first step may be infeasible at the imagined state and the feasibility ranking of Section 4.3 has to penalize it, and beams are kept by mean prefix energy

$$s(p) = \frac{1}{\ell}\sum_{i=1}^{\ell} F_\psi\big(z, \widehat{z}_i, z_g, i\big), \qquad \widehat{z}_i = P_\varphi\big(\widehat{z}_{i-1}, z_{a_i}\big), \;\; \widehat{z}_0 = z. \quad (5)$$

Only the first action of the best depth-$d$ trajectory is executed, the environment returns the actual consequence, and planning repeats, so depth is pure test-time computation and no parameter depends on it.

**FSA deduction** c0 is south. c0 is olive. If c0 is south and c0 is olive, then c1 is juniper. · · · Prove that c8 is juniper. *If c0 is south and c0 is olive, then c1 is juniper.* c1 is juniper. *If c1 is juniper and c0 is south, then c1 is amber-side.* c1 is amber-side. · · · c8 is juniper.

**iGSM** the number of red pens is 7. the number of blue books equals the number of red pens plus the number of green cups. the number of green cups is 4. how many blue books are there? *look up the number of red pens.* so the number of red pens is 7. · · · *derive blue books from red pens plus green cups.* so the number of blue books is 7 plus 4 = 11.

Figure 2: Truncated example episodes. Blue marks environment text (prompt and consequences), *orange italics* the intent phrases (actions), and green the terminal sentence state that ends the episode. In FSA the prompt's rule sentences double as the action catalogue.

## 5 Experimental Setup

We compare LR-JEPA with language models trained on the same text in a chain-structured and a DAG-structured environment. Appendix B completes every architectural and training detail.

**Environments.** Each environment is a deterministic generator that presents a problem in text, consisting of prompt sentences, a catalogue of intent phrases that serve as the admissible actions, and a goal statement, and executing a phrase appends its consequence to the history. The episode ends once the model derives the *terminal sentence state*, the sentence that establishes the goal, which the prompt names without spelling it out (Figure 2). Problems are sampled fresh at every step, and both model families consume byte-identical streams of oracle-demonstrated shortest solutions, so exact imitation is optimal on the trained lengths.

*FSA deduction* is a layered rule system that simulates a finite-state automaton. The actions are the rule sentences of the prompt, executing one appends its conclusion, and the shortest proof through $L$ layers chains $2L-1$ applications, so the necessary computation is a chain. Training uses 6 to 10 layers and the out-of-distribution (OOD) band 11 to 16 layers, with catalogues of 40 to 80 rules. *iGSM* poses arithmetic dependency problems after Ye et al. (2025). A problem states some quantities directly, defines others through arithmetic on earlier ones with values modulo 23, and asks for one of them. Each action computes one currently computable quantity, the necessary computations form a DAG, and any topological order solves the problem. Training uses 3 to 9 necessary steps and evaluation 10 to 16.

The two environments differ in how much of an action's consequence the action itself reveals. An FSA rule states its own conclusion, so imagining one step is easy, and we keep it easy on purpose, because the benchmark scores which action a policy selects, given that every policy faces the same menu and is measured by how directly it reaches the goal. A random policy that imagines every consequence perfectly still runs at $2.01$, and two baselines in Section 6 confirm that easy prediction confers no planning ability. In iGSM the consequence of an action is a computed value that the action phrase does not contain, so the same selection problem is posed where one-step imagination is no longer free.

**Models and baselines.** We train a standard GPT-style decoder-only language model by next-token cross-entropy on the stream of executed trajectories, at 5M, 20M, 50M, 90M and 200M parameters with rotary position embeddings. As a policy it scores each candidate by its length-normalized conditional log-likelihood and commits to the best one. We also train two looped language models at the reference width, the weight-shared stack of Dehghani et al. (2019) and the input-injected recurrence of Geiping et al. (2025), which share one block across depth and spend a chosen number of inference iterations. We further train a sentence-level language model in the manner of LCM team et al. (2024), the closest likelihood-trained analogue of the state abstraction in LR-JEPA, which pools every sentence into one latent and autoregresses over those latents, in a plain and in a looped variant (Appendix B). We train LR-JEPA itself at 10M, 20M, 50M and 90M parameters. Both families see the same executed trajectories, while the losses of LR-JEPA additionally consume the counterfactuals of Section 4, which next-token prediction cannot use.

Two further baselines separate the contribution of the ranking head from that of the predictive state. The *frozen-LM head* baseline attaches the identical head to the reference language model, frozen after its own training, and trains it on the same ranking pairs. At depth one the successor is the hidden state after a candidate phrase. At depth $d$ it is the hidden state after $d$ menu phrases appended to the history without their outcomes, read under the beam of LR-JEPA and scored by the same head against the same history and prompt states, which is the only way a language model can imagine a continuation. The *TD*-LR-JEPA baseline compares ranking with the value-based training of JEPA planners (Bagatella et al., 2026; Destrade et al., 2025). It keeps the architecture, data and planner of Section 4 and trains the same head as a negated value function by temporal-difference bootstrapping

Table 1: FSA deduction, overshoot (1.00 optimal, OOD 2.01 random, 1.96 first-feasible), mean ± std, $n$=100 except the frozen-LM head row ($n$=20) and the TD row at depth 8 ($n$=10). LR-JEPA uses search depth 4, the frozen-LM head row uses LR-JEPA's head, ranking pairs and beam on the frozen reference language model, and the TD row (depth 1 in distribution, depth 8 out) changes only the energy-head supervision.

| Model | ID (6–10) | OOD (11–16) |
|---|---|---|
| LM 20M | $1.00 \pm 0.00$ | $1.88 \pm 0.04$ |
| LM 50M | $1.00 \pm 0.00$ | $1.80 \pm 0.04$ |
| LM 90M | $1.00 \pm 0.00$ | $1.91 \pm 0.04$ |
| Looped (shared / injected) | $1.00 \pm 0.00$ | $1.89 \pm 0.08$ / $1.89 \pm 0.04$ |
| Sentence LM (plain / looped) | $1.00 \pm 0.00$ | $1.74 \pm 0.03$ / $1.69 \pm 0.05$ |
| Ranking head on frozen LM (depth 1 / 8) | $1.00 \pm 0.00$ / $1.97 \pm 0.07$ | $1.03 \pm 0.04$ / $1.96 \pm 0.06$ |
| LR-JEPA 10M | $1.13 \pm 0.03$ | $\mathbf{1.09 \pm 0.04}$ |
| LR-JEPA 20M | $1.13 \pm 0.03$ | $1.17 \pm 0.12$ |
| LR-JEPA 50M | $1.16 \pm 0.03$ | $\mathbf{1.11 \pm 0.03}$ |
| LR-JEPA 90M | $1.40 \pm 0.31$ | $1.45 \pm 0.34$ |
| TD (SARSA / expectile) | $1.95 \pm 0.05$ / $2.02 \pm 0.02$ | $1.93 \pm 0.06$ / $2.02 \pm 0.02$ |

against a unit cost per step, in a SARSA variant and in the expectile variant of implicit Q-learning (Kostrikov et al., 2022). Two reference policies calibrate the scale of every result, a *random* policy, which samples uniformly among the feasible actions, and a *first-feasible* policy, which always takes the first entry of the environment's enumeration and thereby exposes any ordering bias in the menu.

**Evaluation.** Every policy faces the same interface, a shuffled menu of feasible actions from which it commits to one entry, and the deeper imagined steps of LR-JEPA draw from this root menu. Throughout, *depth* denotes the search depth of the planner, the number of imagined steps, and problem size is always reported separately. Under a budget of four times the necessary length every menu policy eventually succeeds, including the random one, so accuracy is uninformative, and what separates the policies is how *directly* they reach the goal. We therefore measure *overshoot*, the number of executed actions divided by the necessary count on solved episodes, for which 1.0 is optimal and 2.01 is the random level on FSA. The menu factors out the proposal of actions, which we leave to future work (Section 8). Headline results use 100 episodes and three seeds, and we report the mean and standard deviation over seeds.

## 6 Results

**Chain-structured deduction.** Table 1 compares LR-JEPA with the language model on FSA deduction. Inside the trained length distribution the language model is exactly optimal at every scale that we trained to its optimum. Beyond it, the language model collapses to 1.69 to 1.91 at every scale and recipe, against first-feasible at 1.96 and random at 2.01. The collapse survives a grid of 18 model widths and layer counts, hundredfold learning-rate sweeps and a 200M model at $1.77 \pm 0.03$. LR-JEPA at depth 4 reaches 1.09 at 10M and 1.11 at 50M. The 90M model varies widely at our training budget, at $1.45 \pm 0.34$. Inside the trained lengths the likelihood readout remains the better policy, at 1.00 against 1.13 to 1.16, because one wasted action costs a larger fraction of a shorter problem. Latent search pays off beyond the trained lengths, where at budgets of 1.25 to 1.5 times the shortest length LR-JEPA reaches a mean success of at least 90 percent against 5 to 16 percent for the 50M language model (Appendix C).

**Search with a frozen language model.** The frozen-LM head row of Table 1 holds the head, its ranking pairs and the beam fixed and varies only the representation underneath. At depth one the head on the frozen language model reaches 1.03 beyond the training lengths. One-step selection on a chain therefore rests on the ranking supervision alone and needs neither the encoder nor the predictor of LR-JEPA. As soon as the beam looks ahead the two systems part ways, since a language model has no simulator and its imagined continuation is only its hidden state after reading candidate phrases appended to its context, and under the beam of LR-JEPA its overshoot rises to 1.72 at depth two and to 1.96 at depth eight, the random level. Supplying the environment as an oracle simulator, by appending the true consequence of every imagined phrase, does not restore search, as the same head then reaches 1.50, 1.95 and 1.97 at depths two, four and eight. A head that is additionally trained across horizons on the same executed rollouts reaches 1.39, 1.58 and $1.66 \pm 0.36$ (Table 4). Over

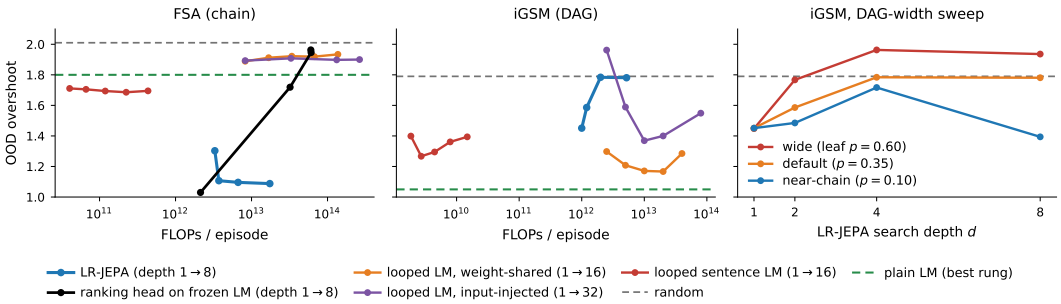


Figure 3: Test-time computation. Left and center, OOD overshoot against inference FLOPs per episode (log axis, accounting in Appendix B). Right, identical iGSM checkpoints under a sweep of the generator's DAG width, where depth harms planning on wide DAGs and helps on near-chains. Dashed lines mark the random policy in grey and the plain language model at its best iteration count in green.

the same beam, LR-JEPA improves from 1.31 at depth one to 1.11 at depth two and 1.09 at depths four and eight. A language model trained on serialized counterfactual branches reaches 2.02, and likelihood lookahead in the manner of RAP and tree of thoughts (Hao et al., 2023; Yao et al., 2023) reaches 1.84 and 1.79. The predictive state is therefore what enables search, and on chains search accounts for the difference between 1.31 and 1.09.

Figure 3 places every family on a common axis of estimated inference FLOPs per episode. On FSA, LR-JEPA improves monotonically with depth from 1.31 to 1.09, while the looped language model spends up to $1.4 \times 10^{14}$ FLOPs without leaving the 1.89 to 1.93 range. The ranking head on the frozen language model moves the other way, from 1.03 at depth one to the random level by depth four.

**Training signal for the energy head.** Next, we vary how the energy is obtained, holding the architecture, the data and the planner fixed. The TD row of Table 1 trains the head by temporal-difference bootstrapping, and both forms, SARSA and the expectile form of implicit Q-learning, plan at the random level. A ten-point hyperparameter sweep leaves them there, because the value either diverges or collapses to a constant (Appendix C). Replacing $F_\psi$ by a latent distance to the context encoding, the readout of visual JEPA planners that Proposition 1 calls unidentified, gives 1.63 and 1.89 at depths one and eight, and a distance to the encoding of the true solved state still reaches the random level from depth four (Table 4). The component ablations of Table 5 in Appendix C vary one part of the 10M model at a time. Three components are load-bearing at depth eight, where the reference reaches 1.09, since removing the hindsight ranking gives 1.72, replacing the MLP head by a quasimetric gives 2.02, and replacing the causal predictor by an MLP gives 1.56. Feasibility ranking governs shallow search, and every other component can be removed within one standard deviation. Depth-monotone profiles depend on the packed training configuration, in which every anchor of a trajectory shares one batch. Scale is non-monotone, with 20M trailing 10M and 90M varying widely, so the demonstrated regime is 10M to 50M under one configuration (Appendix C).

**Transfer to dependency graphs.** We transfer the FSA training configurations to iGSM unchanged, at the 10M class for LR-JEPA and the reference scale for the language model (Appendix C). The language model plans near-optimally in both regimes, at $1.05 \pm 0.05$ and $1.05 \pm 0.03$. The depth search of LR-JEPA *degrades* from $1.45 \pm 0.01$ at depth one to $1.78 \pm 0.21$ at depth eight, and the random policy reaches 1.79. Loop iterations improve the language model on iGSM (Figure 3), while the curve of LR-JEPA rises with depth. Section 7 traces this to the structure of the dependency graph.

## 7 Analysis

**Depth inversion on dependency graphs.** The graph of necessary computation is a chain on FSA and a directed acyclic graph on iGSM, which admits any topological order. Learning-rate changes in either direction and a four-fold longer budget leave the iGSM profile anti-monotone (Appendix C). Sweeping graph width through the generator's leaf probability on identical checkpoints isolates the mechanism (Figure 3). On wide graphs, at leaf probability 0.60, depth search degrades from $1.45 \pm 0.02$ at depth one to $1.94 \pm 0.08$ at depth eight. On near-chains, at leaf probability 0.10, the

effect reverses and depth eight reaches $1.39 \pm 0.01$, below depth one. At fixed weights, the harm that depth causes is therefore monotone in graph width.

We attribute the inversion to how far an imagined trajectory drifts relative to the movement of a single action. On a directed acyclic graph every minimal element of the remaining order is already a correct next action, so the useful decision is one step away and selection at depth one needs no lookahead, which is why depth one stays above the random level, at $1.45$ against $1.79$. Deeper search must compare states several actions ahead, where the per-step error of the predictor is of the order of the movement that one action produces, so by depth $d$ the drift dominates the imagined endpoint. On a chain the beam needs only the gold path, so the same predictor helps there. Width sharpens the effect, because the more actions are simultaneously correct, the smaller the energy differences the beam must resolve against a fixed amount of drift. This account explains the depth trend, while the gap to the language model at depth one, $1.45$ against $1.05$, shows that the transferred head is also weaker on this domain.

**Linear probes of the frozen LM.** Ridge probes decode summary statistics of the FSA symbolic state from the language model almost exactly ($R^2 \geq 0.94$ in both bands), and a linear probe on its out-of-band representations identifies a gold action at top-1 accuracy $0.976$, while the same model plans at $1.9$ through its likelihood readout, so the information sufficient for near-optimal action is linearly present and the failure lies in the policy readout. Conversely, the latent of LR-JEPA decodes these counts worse, at $R^2 \approx 0.6$, while it plans near-optimally, so linear decodability of the symbolic state and planning competence come apart in both directions.

## 8 Limitations and Future Work

The environments are synthetic and deterministic, actions are intent phrases from a finite catalogue that an oracle executor grounds, and goals are given as states. The environment executes the counterfactual actions that the ranking supervision requires, which is the one assumption that does not yet transfer to natural language, and symbolic information is confined to evaluation and to training labels, and no policy receives it. Open vocabulary, ambiguity of reference and the proposal of actions are outside the interface, which supplies every policy with a menu of admissible actions.

Two questions lead from here to a general JEPA language model, the first being what an action is in natural language. Hierarchical JEPA (LeCun, 2022) matches the structure of language well, since token decisions form the lowest level, which plans towards sentence states, which in turn plan towards paragraph states. Hierarchical planning raises its own difficulties, among them off-manifold states under free test-time optimization, and we regard it as the natural next step. The second question is where counterfactuals come from once no executor is available, and on-policy rollouts of the model itself can replace the executor during training, and an energy objective that does not depend on contrastive pairs would remove the requirement altogether. The depth inversion on dependency graphs (Section 7) marks a third, more specific target, a predictor with drift below the movement of one action on wide graphs.

## 9 Conclusion

We introduced Language-Reasoning JEPA, an action-conditioned JEPA that plans over language by searching imagined latent trajectories, and an energy head, trained by ranking executed outcomes, that supplies the order which transition prediction leaves free. Under one interface and against language models from 4.8M to 200M parameters, looped and sentence-level models, likelihood lookahead and value-based alternatives, ranking supervision alone selects a next step at least as well on a frozen language model as on the JEPA, while the predictive state is what turns test-time computation into better plans. We regard this as a step towards language world models that reason natively in latent space, by imagining the consequences of what they might say before committing to it.

## AI Use Statement

In this work we used generative AI tools for writing and debugging the data generators, the training and evaluation scripts and the analysis code, for tabulating results, and for editing the text of the

manuscript under the direction of the authors. We also used them to check the bibliography, and every entry was subsequently verified against its DOI by the authors. We have not used generative AI tools for generating research ideas, and no model-written text or code enters the training data. All AI-assisted code was reviewed and tested by the authors, every number in the paper was checked against the evaluation outputs, and all text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

## Reproducibility Statement

Appendix B specifies every architecture, objective weight, generator parameter, and evaluation protocol, Appendix C reports every result in detail, and Appendix E lists the software environment and the configurations used for training and evaluation.

## References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 2017.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38546–38556, 2022. doi: 10.52202/068431-2793.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction. In *Proceedings of the International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2296–2318, 2024.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *International Conference on Learning Representations*, 2026.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.1093/biomet/39.3-4.324.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? On the overthinking of long reasoning models. In *Proceedings of the International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9487–9499, 2025.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Matthieu Destrade, Oumayma Bounou, Quentin Le Lidec, Jean Ponce, and Yann LeCun. Value-guided action planning with JEPA world models. *arXiv preprint arXiv:2601.00844*, 2025. World Modeling Workshop.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems*, volume 36, pp. 70293–70332, 2023. doi: 10.52202/075280-3081.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35603–35620, 2022. doi: 10.52202/068431-2580.

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. Stream of search (SoS): Learning to search in language. In *Conference on Language Modeling*, 2024.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems*, volume 38, pp. 46242–46293, 2025. doi: 10.52202/085713-1380.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11398–11442, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, 2023. doi: 10.18653/v1/2023.emnlp-main.507.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. Interactive fiction games: A colossal adventure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7903–7910, 2020. doi: 10.1609/aaai.v34i05.6297.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1621–1630, 2016. doi: 10.18653/v1/P16-1153.

Hai Huang, Yann LeCun, and Randall Balestriero. LLM-JEPA: Large language models meet joint embedding predictive architectures. In *International Conference on Learning Representations*, 2026.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

Guillaume Lample, Timothée Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. HyperTree proof search for neural theorem proving. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26337–26349, 2022. doi: 10.52202/068431-1910.

LCM team, Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, et al. Large concept models: Language modeling in a sentence representation space. *arXiv preprint arXiv:2412.08821*, 2024.

Yann LeCun. A path towards autonomous machine intelligence. *OpenReview preprint*, 2022.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In Gökhan Bakır, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan (eds.), *Predicting Structured Data*, pp. 191–246. MIT Press, 2006.

Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qinqing Zheng, Paul McVay, Michael Rabbat, and Yuandong Tian. Beyond A*: Better planning with transformers via search dynamics bootstrapping. In *Conference on Language Modeling*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1555–1561, 2001.

Bingyang Kelvin Liu, Ziyu Patrick Chen, and David P. Woodruff. JEPA-reasoner: Decoupling latent reasoning from token generation. *arXiv preprint arXiv:2512.19171*, 2025.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

OpenAI. Learning to reason with LLMs. OpenAI blog, `https://openai.com/index/learning-to-reason-with-llms/`, September 2024.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Conference on Uncertainty in Artificial Intelligence*, pp. 512–519, 2004. doi: 10.5555/1036843.1036905.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, pp. 38975–38987, 2023. doi: 10.52202/075280-1693.

Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. AlphaZero-like tree-search can guide large language model decoding and training. In *Proceedings of the International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 49890–49920, 2024.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proceedings of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36411–36430, 2023a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023b.

Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. Keep CALM and explore: Language models for action generation in text-based games. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 8736–8754, 2020. doi: 10.18653/v1/2020.emnlp-main.704.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822, 2023. doi: 10.52202/075280-0517.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. In *International Conference on Learning Representations*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, pp. 64304–64339, 2025. doi: 10.52202/085713-1933.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. In *Conference on Language Modeling*, 2024.

Zhenghao Zhang, Yuanxiang Wang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Tianyu Zong, Hongzhu Yi, Guoqing Chao, Xingchen Chen, Tiankun Yang, Chenxi Bao, Tao Yu, Jingjing Zhou, and Jungang Xu. Delta-JEPA: Learning action-sensitive world models via latent difference decoding. *arXiv preprint arXiv:2606.31232*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135, 2025.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? A study in length generalization. In *International Conference on Learning Representations*, 2024.

# A PROOFS

**Setting.** Throughout, $\mathcal{M}$ is the partially observed decision process of Section 3. A policy $\pi$ maps histories to distributions over actions, and $\pi$ is *admissible* at $h$ if it assigns positive probability only to feasible actions along every trajectory it induces from $h$. For a horizon $\tau \leq \infty$, write $W_{h,\pi} = (Y_{t+1:\tau}, C_{t:\tau}, A_{t:\tau})$ for the random future observation–cost–feasibility sequence induced by following $\pi$ from $h$, taking values in the canonical sequence space with its product $\sigma$-algebra.

**Proposition 3** (Control sufficiency)**.** *Assume the per-step cost and the termination event are measurable functions of $(Y, C, A)$. If $h \sim_{\mathrm{pred}} h'$, then the admissible policy sets at $h$ and $h'$ coincide, $V^\pi(h) = V^\pi(h')$ for every admissible policy $\pi$, and therefore $V^\star(h) = V^\star(h')$.*

*Proof of Proposition 3.* First, the admissible policy sets coincide. The feasibility processes $A_{t:\tau}$ have the same law under every policy by Equation 1. In particular the feasible set at the root, $A_t$, is almost surely equal under $h$ and $h'$, and recursively the reachable feasibility structure agrees, so a policy is admissible at $h$ iff it is admissible at $h'$.

Fix an admissible $\pi$. By assumption the return $G = \sum_{k \geq t} \mathbf{1}\{\text{not terminated before } k\}\, c_k$ is a measurable function $g(W_{h,\pi})$ of the future sequence, since the per-step cost and the termination event are measurable functions of $(Y, C, A)$. Equation 1 states $W_{h,\pi} \overset{d}{=} W_{h',\pi}$, and measurable functions of equal-in-distribution random elements are equal in distribution, so $g(W_{h,\pi}) \overset{d}{=} g(W_{h',\pi})$, and taking expectations,

$$V^\pi(h) = \mathbb{E}[g(W_{h,\pi})] = \mathbb{E}[g(W_{h',\pi})] = V^\pi(h').$$

(In the finite-horizon deterministic environments of this paper $G$ is bounded, so the expectations exist, and in general one assumes integrability.) Because the two admissible sets coincide and $V^\pi$ agrees on them pointwise,

$$V^\star(h) = \sup_\pi V^\pi(h) = \sup_\pi V^\pi(h') = V^\star(h'). \qquad \square$$

*Proof of Proposition 1.* For the first claim, let $z = E(h)$ be realized and $a$ feasible at $h$. Then

$$\widetilde{P}(\widetilde{E}(h), z_a) = \widetilde{P}(b(E(h)), z_a) = b(P(E(h), z_a)) = b(E(T(h, a))) = \widetilde{E}(T(h, a)),$$

so every realized transition is reproduced exactly. $\widetilde{P}$ is well defined on the realized pairs because $b$ is injective on $Z$.

For the second claim, let $z_+, z_-, z_g \in Z$ be distinct with $\|z_+ - z_g\| < \|z_- - z_g\|$. Define $b$ on $Z$ by exchanging $z_+$ and $z_-$ and fixing every other point, so that $b(z_+) = z_-$, $b(z_-) = z_+$, and $b(z) = z$ otherwise. This $b$ is a bijection of $Z$ onto itself, hence an injection into $\mathbb{R}^{d_z}$, and

$$\|b(z_+) - b(z_g)\| = \|z_- - z_g\| > \|z_+ - z_g\| = \|b(z_-) - b(z_g)\|,$$

reversing the goal-distance order while, by the first claim, leaving every realized transition intact. No transition-prediction loss evaluated on realized data distinguishes $(E, P)$ from $(\widetilde{E}, \widetilde{P})$, so goal-distance orderings are not identified by transition prediction. $\square$

*Proof of Proposition 2.* Let $\sigma$ be the learned score and consider greedy selection $a_t \in \arg\max_a \sigma(h_t, a)$. We prove by induction on $d = D^\star(h)$ that from any history $h$ reachable from $h_0$ by optimal actions, greedy selection reaches the goal in exactly $d$ actions.

In the base case $d = 0$ the history is terminal and zero actions are used. For the inductive step let $D^\star(h) = d > 0$. Since $D^\star(h_0)$ is finite and the problem is deterministic with unit costs, the set of optimal actions at $h$ is nonempty. By hypothesis every optimal action at $h$ scores strictly above every suboptimal one, so the greedy argmax is an optimal action $a$, and $D^\star(T(h, a)) = d - 1$. The successor $T(h, a)$ is again reachable from $h_0$ by optimal actions, so the ranking hypothesis applies to it, and by the inductive hypothesis greedy selection reaches the goal from $T(h, a)$ in exactly $d - 1$ further actions, for a total of $d$.

For the robustness claim, let $\sigma^\star$ be a score with minimal gap $\gamma > 0$ between every optimal and every suboptimal action at every history covered by the hypothesis, and let $\widehat{\sigma}$ satisfy $|\widehat{\sigma}(h, a) - \sigma^\star(h, a)| < \gamma/2$ for all such $(h, a)$. For optimal $a^+$ and suboptimal $a^-$ at the same history,

$$\widehat{\sigma}(h, a^+) > \sigma^\star(h, a^+) - \tfrac{\gamma}{2} \geq \sigma^\star(h, a^-) + \gamma - \tfrac{\gamma}{2} > \widehat{\sigma}(h, a^-),$$

**Algorithm 1** Energy-guided latent planning (one environment step)

$z \leftarrow E(h)$; $M \leftarrow$ menu of feasible actions at $h$; $B \leftarrow \{(z, [\,], 0)\}$
**for** $\ell = 1$ to $d$ **do**
  $B' \leftarrow \emptyset$
  **for all** $(z', p, s) \in B$, candidate $a \in M$ not in $p$ **do**
    $\widehat{z} \leftarrow P(z', z_a)$; $s' \leftarrow s + F_\psi(z, \widehat{z}, z_g, \ell)$
    add $(\widehat{z},\ p{+}[a],\ s')$ to $B'$
  **end for**
  $B \leftarrow$ best elements of $B'$ by mean prefix energy $s'/\ell$, keeping 4 continuations per root candidate
**end for**
execute first action of $\arg\min_{(\cdot,p,s)\in B} s$

so $\widehat{\sigma}$ satisfies the strict-ranking hypothesis and the first part applies. □

## B PROTOCOL DETAILS

**Architectures.** Table 2 lists every model in the paper. Nominal names round the measured trainable-parameter counts; the EMA teacher doubles LR-JEPA's stored parameters but is not trained. The looped LMs use the reference width with one weight-shared block, and the input-injected recipe trains with randomized iteration counts (mean 8, max 32) and truncated backpropagation (Geiping et al., 2025). TD-LR-JEPA and the iGSM transfer runs use the 10M-class LR-JEPA architecture, and the transfer LM cells use the reference LM.

Table 2: Model architectures. Parameter counts are measured trainable parameters where recorded at training time, otherwise nominal. Nominal size names round these counts, so sizes are approximately parameter-matched across families and not exactly (e.g. LM 90M ≈ 92M against LR-JEPA 90M = 111M trainable).

| Model | width | layers | trainable params |
|---|---|---|---|
| LM reference | 256 | 6 | 4.8M |
| LM 20M | 384 | 8 | 15.9M |
| LM 50M | 512 | 16 | ≈50M |
| LM 90M | 768 | 13 | ≈92M |
| LM 200M | 1024 | 16 | ≈201M |
| Looped LM (both recipes) | 256 | 1 shared block | ≈1M/loop |
| Sentence LM | 256 | enc 2 / state 4 / dec 2 | 7.3M |
| Looped sentence LM | 256 | enc 2 / 1 shared state block / dec 2 | 5.7M |
| LR-JEPA 10M | 128 (packed recipe) | enc 4 / pred 2 | 7.7M |
| LR-JEPA 20M | 384 | enc 5 / pred 3 | 19.8M |
| LR-JEPA 50M | 512 | enc 11 / pred 5 | 59.7M |
| LR-JEPA 90M | 768 | enc 9 / pred 4 | 111.1M |

**Training.** All models train from scratch on freshly sampled problems, with one pass and no repeats. LR-JEPA runs train for a fixed wall-clock budget (2,200 s for 10M-class runs, 1–2 h for larger sizes) with AdamW, learning rate $10^{-4}$ for JEPA runs and $3 \times 10^{-4}$ for LMs unless a sweep is reported, in bf16, and the release pins the resulting optimizer-step counts per run. The EMA teacher follows a momentum schedule from $0.99$ to $0.999$, and dropout is zero everywhere. The tokenizer is whitespace word-level with only pad/unk specials, and sentence boundaries survive as period tokens. LM runs use the identical executed-trajectory stream with cross-entropy on solution tokens and train under matched wall-clock budgets (1,900 s at the reference scale, while larger sizes share the budget of the corresponding LR-JEPA size). The *packed* recipe groups all anchors of a trajectory into consecutive batch rows (batch 38 with 19-way anchor grouping), so every gold-chain position receives counterfactual and rollout supervision. This recipe at width 128 is the 10M model.

**Objectives.** The ranking losses are pairwise logistic (Bradley–Terry) objectives on energy differences with margin $0.5$ and temperature $0.3$. The reference weights are latent consequence prediction

(smooth-$\ell_1$, weight 1.0, executed and counterfactual), hindsight long-horizon ranking 1.0 (horizons up to the full remaining length), feasibility ranking 4.0, goal-swap ranking 1.0, horizon ranking 1.0 (horizons $\{1, 2, 4, 8\}$), VICReg 1.0 (variance target 1.0, covariance 0.04), auxiliary next-phrase LM head 1.0, action decoder 1.0, counterfactual state reconstruction 1.0. As for counterfactual volume, every gold-chain position of a trajectory serves as an anchor (19 per trajectory in the packed recipe), and each anchor has 26 candidate actions (2 feasible, 16 infeasible, 8 redundant) with up to 8 counterfactual rollouts, so LR-JEPA consumes roughly an order of magnitude more executed transitions than the LM's single-path stream. Exact per-run counts are recorded in the release.

**FSA generator.** FSA deduction problems simulate a finite-state automaton as layered implication chains over attribute facts. Each layer introduces rules with branching factor 4 and two side facts per step, catalogues hold 40–80 rules, and the feasible menu at a step contains the not-yet-applied rules with all antecedents derived. Standard deviations of $\pm 0.00$ in result tables denote agreement to the reported precision.

**Planner and encoders.** The beam is root-balanced. Every root-menu candidate is expanded (capped at 64), each surviving beam keeps its 4 best energy-guided continuations per level, and trajectories are aggregated by mean prefix energy. These beam parameters were fixed a priori, not tuned. At inference the energy anchor is the root state at every beam level with the level $\ell$ as horizon input, a pattern training covers by placing anchors at all trajectory positions with horizons up to the full remaining length. Both families use rotary position embeddings. The hindsight goals used to train the energy head are EMA encodings of achieved states, while the test-time goal input is the context encoding in which the prompt states the goal. The goal-swap loss is one explicit pressure tying the two, but its removal is within one standard deviation (Table 5), and the hindsight goals themselves reconcile the two, since their latest relabels approach the terminal state that the context describes.

**Evaluation.** Closed-loop episodes run to a $4\times$ budget over the necessary length, evaluation problems are drawn from held-out seeds, and candidate menus are shuffled. Overshoot is computed over solved episodes. Free-generation (answer-tag) episodes decode without interruption up to a token budget matching the $4\times$ action budget, split into sentences, and replay symbolically. Strict success requires every printed sentence to be a catalogue rule or the true consequence of the previous rule and the goal to be derived.

**Sentence LM.** A two-layer encoder pools every sentence of the stream (prompt sentence, intent phrase, outcome) into one latent, a four-layer causal state model runs over the sequence of sentence latents, and a two-layer decoder cross-attending to the current state latent is trained by next-token cross-entropy on the next intent phrase. As a policy it scores each candidate phrase by length-normalized decoder log-likelihood from the current state latent, exactly as the token LM does from its token context; the looped variant shares one state block across depth as the weight-shared looped LM does and is evaluated at 1–16 iterations.

**FLOP accounting.** LR-JEPA evaluation records backbone token positions and predictor/energy forward counts per episode, and FLOPs are $2N$ per parameter-position with per-module parameter counts. Looped-LM FLOPs multiply the single-pass block cost by the iteration count, with token positions computed from the scoring protocol and measured episode lengths. The two accountings share the backbone convention, and predictor and energy-head costs are exact for LR-JEPA and absent for the LM. Sentence-LM FLOPs count the chunk encoder over the tokens of the two sentences added per step, the state model over every sentence latent present at that step (multiplied by the iteration count when looped), and the decoder over every menu candidate's tokens, with measured per-module parameter counts (`figures/make_sentence_lm_flops.py`).

## C Complete results

**iGSM robustness.** Lowering the LR-JEPA learning rate to $3 \times 10^{-5}$ gives 1.67–1.97 across depths, raising it to $3 \times 10^{-4}$ gives 1.48–1.80, and quadrupling the training time leaves the depth profile anti-monotone ($1.58 \rightarrow 1.80$), so the depth inversion is not specific to the transferred hyperparameters or budget.

Table 3: Transfer of both recipes to iGSM (overshoot), $n$=10 episodes. LR-JEPA uses depth 1, looped LMs report their best iteration count and were evaluated only out of distribution, and the random policy reaches $1.79 \pm 0.00$ OOD.

| Model | ID | OOD |
|---|---|---|
| LM | $1.05 \pm 0.05$ | $\mathbf{1.05 \pm 0.03}$ |
| Looped, weight-shared | – | $1.17 \pm 0.04$ |
| Looped, input-injected | – | $1.37 \pm 0.43$ |
| LR-JEPA | $1.55 \pm 0.03$ | $1.45 \pm 0.01$ |

**Evaluation episode counts.** FSA finals use $n$=100 at depths 1–4 (both bands) and 50M ID depth 8, while 50M OOD depth 8 uses $n$=25 and 10M OOD depth 8 uses $n$=10 (deep OOD evaluations exceed 8 h walltimes at $n$=100). iGSM runs use $n$=10.

**FSA depth grids (OOD 11–16, overshoot).**

| cell | d1 | d2 | d4 | d8 |
|---|---|---|---|---|
| LR-JEPA 10M (packed×128) | 1.31 | 1.11 | 1.09 | 1.09 |
| LR-JEPA 20M | 1.56 | 1.37 | 1.17 | 1.15 |
| LR-JEPA 50M | 1.18 | 1.17 | 1.11 | 1.11 |
| LR-JEPA 90M (lr $10^{-4}$) | 1.21 | 1.31 | 1.42 | 1.52 |
| LR-JEPA 90M (lr $3\times10^{-5}$) | 1.38 | 1.41 | 1.45 | 1.43 |
| packed, $d_{\text{model}}$=256 | 1.55 | 1.35 | 1.20 | 1.19 |
| width 512 | 1.23 | 1.23 | 1.22 | 1.21 |

**Rotary depth grids (OOD 11–16, overshoot, $n$=10).** The rotary-embedding runs of Appendix B give $1.33/1.28/1.20/1.19$ at depths $1/2/4/8$ for packed $d_{\text{model}}$=256, $1.33/1.33/1.35/1.36$ at 20M, and $1.25/1.23/1.19/1.19$ for the 10M recipe at a four-fold longer budget. The depth ordering matches the learned-embedding rows above: the packed and long-budget recipes improve with depth while 20M is flat to anti-monotone. At the default learning rate the profile is depth-anti-monotone, $3 \times 10^{-5}$ recovers a flat profile on average, and four-fold longer training at the default rate does not recover it. The main-text LR-JEPA number is the 10M run, and the main-text 90M row reports learning rate $3 \times 10^{-5}$.

**200M LM.** A 200M LM reaches $1.00$ in distribution and $1.77 \pm 0.03$ on depths 11–16, reproducing the collapse at forty times the reference scale.

**Learning-rate gates.** Token-LM OOD overshoot under lr $\in \{3\times10^{-5}, 10^{-4}, 10^{-3}, 3\times10^{-3}\}$ is $1.95/1.95/1.95/1.91$ with ID $1.00$ throughout, and at 90M it is $1.94/1.80$ for lr $10^{-4}/10^{-3}$. The LR-JEPA 10M recipe under lr $\in \{3\times10^{-5}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}\}$ gives OOD depth-4/8 values of $1.19/1.17$, $1.09/1.08$, $1.33$ (depth 4 only), and $1.22/1.23$, and the reported $10^{-4}$ is within $0.01$ of best at depth.

**Budget curves (FSA OOD, depth 4, $n$=100).** At budget $1.1\times$ the necessary length, LR-JEPA 50M succeeds on $0.61$ of episodes and 10M on $0.74$, while the 50M token LM is at $0.00$ there and $0.05$–$0.16$ at $1.25$–$1.5\times$. Both families saturate to $1.0$, near $1.25$–$1.5\times$ for LR-JEPA and near $2\times$ for the LM.

**Baselines and ablations (Table 4).** The ranking head on the frozen reference LM is a two-layer MLP on the concatenation of the LM's state before the candidate, its state after reading the candidate phrase, and the prompt state, trained for 3,000 steps with the counterfactual and feasibility pairs of Section 4 on the same anchors ($n$=20 episodes). At depth $d > 1$ the same head scores sequences of candidate phrases appended to the history, drawn from the root menu without repetition and appended without their outcome sentences, so that the successor input is the LM's state after reading all $d$ phrases while the other two inputs remain the history state and the prompt state. The beam is the one of LR-JEPA (every root expanded, deeper candidates from the root menu, branch 4, at most 64 expansions, mean-prefix aggregation), and the on-policy gold-pick rate falls from $0.93$–$1.00$ at depth 1 to $0.50$–$0.64$ at depth 2 and $0.47$–$0.51$ at depths 4 and 8, i.e. chance among the two or three

Table 4: Baselines and ablations on the central comparison (FSA, overshoot, mean ± std; OOD is depths 11–16; $n$ in Appendix C). The ranking-head and probe rows put the LM's own representation under LR-JEPA's readout; the serialized and search-matched rows give the LM LR-JEPA's data and test-time search; the ruler row removes the learned energy from LR-JEPA.

| Variant | Depth | ID (6–10) | OOD (11–16) |
|---|---|---|---|
| Ranking head on frozen LM | 1 | $1.00 \pm 0.00$ | $1.03 \pm 0.04$ |
| | 2 | $1.78 \pm 0.21$ | $1.72 \pm 0.12$ |
| | 4 | $1.97 \pm 0.07$ | $1.95 \pm 0.07$ |
| | 8 | $1.97 \pm 0.07$ | $1.96 \pm 0.06$ |
| Same, true consequence appended after every imagined step | 1 | $1.00 \pm 0.00$ | $1.03 \pm 0.04$ |
| | 2 | $1.38 \pm 0.28$ | $1.50 \pm 0.19$ |
| | 4 | $1.89 \pm 0.11$ | $1.95 \pm 0.04$ |
| | 8 | $1.90 \pm 0.10$ | $1.97 \pm 0.02$ |
| Same, head also trained across horizons | 1 | $1.00 \pm 0.00$ | $1.06 \pm 0.06$ |
| | 2 | $1.22 \pm 0.19$ | $1.39 \pm 0.28$ |
| | 4 | $1.41 \pm 0.34$ | $1.58 \pm 0.37$ |
| | 8 | $1.47 \pm 0.32$ | $1.66 \pm 0.36$ |
| Gold-action probe on frozen LM as policy | 1 | $1.00 \pm 0.00$ | $1.01 \pm 0.01$ |
| LM trained on serialized counterfactuals | 1 | $1.00 \pm 0.00$ | $2.02 \pm 0.02$ |
| Search-matched LM | 2 | $1.00 \pm 0.00$ | $1.84 \pm 0.06$ |
| | 4 | $1.00 \pm 0.00$ | $1.79 \pm 0.09$ |
| LR-JEPA with layer-normalized $L_1$ distance instead of $F_\psi$ | 1 | $1.42 \pm 0.30$ | $1.63 \pm 0.34$ |
| | 8 | $1.58 \pm 0.34$ | $1.89 \pm 0.15$ |
| LR-JEPA with cosine ruler instead of $F_\psi$ | 1 | $1.39 \pm 0.26$ | $1.53 \pm 0.33$ |
| | 8 | $1.55 \pm 0.34$ | $1.82 \pm 0.17$ |
| LR-JEPA with hindsight goal input | 1 | $1.22 \pm 0.05$ | $1.15 \pm 0.04$ |
| | 8 | $1.14 \pm 0.03$ | $1.08 \pm 0.01$ |

candidates that advance the chain. Giving it the true outcome sentence as well (candidate-privileged) reaches $1.00$ in band and $1.03$ out of band at depth 1, and the same head under the beam with the true consequence appended after every imagined phrase, so that the environment acts as an oracle simulator, still falls to $1.50$, $1.95$ and $1.97$ at depths 2, 4 and 8 out of band ($n$=20). The horizon-matched variant trains the same head additionally on the hindsight long-horizon pairs of Section 4.3, the observed $h$-step continuation against four random feasible $h$-step rollouts from the anchor with $h$ uniform up to the remaining length and every outcome sentence replayed by the environment, and it receives the horizon as an input, which the beam sets to the level. Its out-of-band overshoot at depths $1/2/4/8$ is $1.06/1.39/1.58/1.66$ ($\pm 0.06/0.19/0.37/0.36$), so matched supervision together with an oracle executor slows but does not prevent the degradation. The probe policy trains a gold-action MLP probe on the same frozen features inside the training band and picks the argmax at every step ($n$=10); its out-of-band gold-pick rate is $0.99$–$1.00$ against $0.53$–$0.58$ for the likelihood readout. The search-matched LM appends each candidate rule and its conclusion, rescores by likelihood, and commits to the best root at lookahead 2 and 4 ($n$=10, branch 4, at most 64 expansions). The layer-normalized $L_1$ readout replaces $F_\psi$ by the distance between the imagined endpoint and the context encoding in the same beam ($\ell_1$ $1.63 \pm 0.34$ at depth 1 and $1.89 \pm 0.15$ at depth 8 out of band, $n$=10); the cosine ruler gives $1.53 \pm 0.33$ and $1.82 \pm 0.17$, so both rulers degrade with depth. The hindsight-goal row feeds the EMA encoding of the achieved terminal state as $z_g$ and equals the context-goal result at every depth, which leaves open how much the head reads from $z_g$ as opposed to the anchor state that encodes the same context, a question we have not tested directly ($1.15/1.10/1.08/1.08$ against $1.15/1.10/1.08/1.08$ at depths $1/2/4/8$ for the rotary 10M LR-JEPA). Counterfactual counts of 2, 8, and 26 per anchor (depth 4) give $1.10/1.08/1.09$ out of band and $1.17/1.15/1.15$ in band.

**Sentence LM.** The sentence LM (7.3M, $n$=100) is exactly optimal in distribution and reaches $1.74 \pm 0.03$ on depths 11–16; the looped sentence LM (5.7M) is at $1.00$ in distribution at every iteration count and at $1.71/1.71/1.69/1.69/1.70$ out of distribution for $1/2/4/8/16$ iterations (best $1.69 \pm 0.05$ at 8), so extra iterations move it by less than $0.03$.

**Probes.** Ridge $R^2$ for remaining steps, feasible count, and fact count, with probes trained in distribution, is $0.99/1.00/1.00$ on the token LM's ID holdout and $0.39/0.71/0.69$ OOD, and probes trained inside the OOD band reach $0.94/0.97/0.97$. LR-JEPA 10M sits at $\approx 0.56$–$0.64$ in band in both regimes, the random-init encoder at $\approx 0.40$–$0.55$ in band, and MLP probes fall below ridge in every OOD configuration. Gold-action identification with an MLP on action-in-context features reaches AUC $1.0$ for all three encoders including random-init, and the LM's OOD top-1 is $0.976$. On iGSM, steps-to-go $R^2$ is $\leq 0.35$ for every model.

Table 5: Component ablations on FSA, out-of-distribution overshoot at depths 1 and 8, means, $n$=10 per configuration, change against the reference recipe (1.31 at $d$=1, 1.09 at $d$=8) in parentheses. The reference's standard deviation is 0.27 at $d$=1 and 0.03 at $d$=8, so the conclusions rest on the $d$=8 column.

| ablation | $d$=1 | $d$=8 | ablation | $d$=1 | $d$=8 |
|---|---|---|---|---|---|
| hindsight ranking | 1.66 (+.35) | 1.72 (+.63) | goal-swap ranking | 1.30 (−.01) | 1.09 (±.00) |
| MLP energy → quasimetric | 1.95 (+.64) | 2.02 (+.93) | next-phrase LM head | 1.16 (−.15) | 1.09 (±.00) |
| causal → MLP predictor | 1.64 (+.33) | 1.56 (+.47) | action decoder | 1.26 (−.05) | 1.09 (±.00) |
| feasibility ranking | 1.63 (+.32) | 1.13 (+.04) | counterfactual state | 1.27 (−.04) | 1.09 (±.00) |
| horizon ranking | 1.19 (−.12) | 1.07 (−.02) | VICReg | 1.10 (−.21) | 1.08 (−.01) |
| add advantage regression | 1.28 (−.03) | 1.10 (+.01) | VICReg → SIGReg | 1.22 (−.09) | 1.13 (+.04) |

**Recipe-dependence of depth-monotonicity.** Depth-monotone OOD profiles require the packed narrow recipe, while the wide-batch default ($1.57/1.57/1.65/1.66$) and the longest-rollout variant are anti-monotone and fraction/width variants are flat. Full per-run tables ship with the release.

**Temporal-difference sweep (FSA OOD, overshoot, $n$=10).** Ten runs vary only the value-training hyperparameters of the TD row, holding architecture, data and planner fixed: SARSA-style bootstrapping at learning rates $3\times10^{-5}$, $10^{-4}$, $3\times10^{-4}$ and $10^{-3}$ and at discounts 0.90 and 0.999, and the expectile form at $\tau \in \{0.7, 0.9, 0.99\}$ plus $\tau$=0.9 at $3\times10^{-5}$. Eight of ten give 2.00–2.04 at every depth, one gives 1.97–2.04, and the best, SARSA at $10^{-3}$, reaches $1.86/1.77/1.76/1.76$ at depths $1/2/4/8$; the ranking recipe is at 1.09. The profiles are depth-invariant to three decimals in eight runs, which is the signature of an energy that does not order states, and the training traces show two ways of arriving there: the residual grows by one to two orders of magnitude in seven runs (to 12.5 at $10^{-4}$ and 21.7 at discount 0.999, from $\approx 1.0$), while the expectile at $\tau$=0.99 instead collapses to a near-constant value with residual 0.014. The head's counterfactual-depth ranking accuracy, chance 0.5, is 0.08–0.66 across the sweep against 0.87–0.93 for the ranking-trained reference.

**Oracle-distance readout (FSA OOD, overshoot, $n$=10).** Ranking the menu by latent distance from the imagined endpoint to the encoding of the *true solved state*, an oracle goal that removes the predictor's error from the readout while keeping the learned ruler, gives $1.62 \pm 0.42$, $1.79 \pm 0.16$, $2.01 \pm 0.05$ and $2.01 \pm 0.02$ at depths $1/2/4/8$ against a random floor of 2.02. It is at chance from depth 4 on, and the spread closes as it gets there. This is consistent with Proposition 1: a correct goal does not rescue a distance readout.

## D Qualitative traces

Verbatim closed-loop episodes for LR-JEPA (depth 4) and the token LM on identical FSA problems, in and out of distribution, are provided with the code release (`traces_pw128_id_ood.txt`). A representative OOD LR-JEPA episode executes two feasible off-path rules, then locks onto the gold chain and follows it to the goal without further deviation. Representative LM free-generation transcripts produce the correct first rule and then continue with well-formed rules that do not exist in the problem's catalogue.

## E Reproducibility

All environments are deterministic generators seeded per problem index, and every reported result names its generator band and evaluation episode count. The release contains the training and evaluation entry points, launch scripts for every run in the paper, the frozen code snapshot hash, per-run result files including per-episode records, and model checkpoints for the headline runs. The complete result grid is enumerated in the release manifest.