A Blueprint for Language-Native World Models

Niel Ok
Stanford University
nielok@stanford.edu

April 2025

Abstract

We propose a theoretical framework for language-native world modeling, where natural language descriptions of environments are encoded into structured latent vectors composed of interpretable components, each corresponding to an aspect of world state (e.g., object position, agent action, environmental affordance). Instead of predicting the next token, our system autoregressively forecasts the evolution of semantic latent states under hypothetical action sequences. We extend Shannon's notion of entropy minimization beyond text generation and define reasoning as compression over semantically meaningful world trajectories.

The system comprises three key modules: (1) a *semantic encoder* that maps text to latent world states; (2) a *latent dynamics model* that predicts how the latent world evolves over time under specified actions; and (3) a *goal verifier* that scores simulated outcomes for goal alignment. Each latent state can optionally be projected into a learned space optimized for simulation and then decoded back into structured form for interpretation.

Unlike conventional prompting-based approaches, we separate semantic interpretation, latent prediction, and evaluation. Simulation is performed entirely in latent space, not token space, allowing for modular, interpretable, and efficient reasoning. This blueprint recasts language modeling as predictive compression over latent semantic configurations and enables agents that autoregressively simulate not text but meaningfully structured futures.

1 Background and Related Work

1.1 Shannon Entropy and Predictive Compression

Claude Shannon's foundational work on information theory [8] introduced entropy as a measure of uncertainty and a limit on lossless compression. In modern machine learning, this underpins next-token prediction, where models minimize cross-entropy loss to approximate the source distribution. Transformer-based language models such as GPT [5, 1] optimize this objective to generate coherent textual continuations. However, their focus remains on syntactic fluency rather than semantic world understanding.

1.2 World Models in Reinforcement Learning

Model-based reinforcement learning (MBRL) explores the use of internal models that simulate environment dynamics for planning. Systems like PlaNet [3], Dreamer [2], and MuZero [7] learn latent representations of physical environments and predict transitions under action sequences. These world models operate over sensorimotor inputs, often grounded in visual or proprioceptive data, and focus on predicting observable outcomes for downstream decision-making.

1.3 Language Models as Semantic Simulators

Recent research explores the potential of language models to simulate abstract processes using natural language as both input and output. Works on chain-of-thought prompting [9], tool integration [6], and generative agents [4] demonstrate LLMs' ability to model reasoning steps, simulate social interactions, and carry out tasks. However, these approaches remain constrained to surface-level token manipulation. They lack a principled notion of latent world states or trajectory modeling beyond token-level transitions.

While transformer-based language models model sequences through token recurrence and attention, their simulation capacity is ultimately tethered to surface-level syntax. They excel at syntactic fluency but conflate grammatical continuity with causal or semantic coherence. Because they operate entirely in token space, they are limited in their ability to explicitly represent structured world dynamics or track evolving latent variables over time. This restricts their utility in tasks requiring consistent reasoning, long-horizon planning, or counterfactual imagination.

1.4 Our Contribution: Language-Native World Modeling

We introduce a novel framework that shifts the focus of language models from surface-level token prediction to structured semantic reasoning. In our approach, natural language descriptions of environments and goals are encoded into structured latent vectors, where each dimension corresponds to an interpretable semantic variable (e.g., object status, agent intent, environmental affordance). These latent representations are then evolved through time using a history-aware dynamics model and evaluated via a goal verifier.

Unlike prior approaches that generate token continuations or simulate environments through sensorimotor embeddings, our framework operates entirely in latent semantic space. This enables interpretable planning, compositional simulation, and language-grounded reasoning without reliance on symbolic logic or grounded sensory inputs.

Our key contributions are:

- A unified architecture that decomposes interpretation, simulation, and evaluation into modular components: a semantic encoder, a latent dynamics model, and a verifier.
- A hybrid latent space design that retains symbolic interpretability while allowing optimized projection for efficient simulation.
- A formalization of reasoning as entropy minimization over future latent trajectories, extending Shannon entropy from token sequences to structured world-state evolution.
- A new paradigm of language-native planning—grounded solely in natural language but decoupled from autoregressive token generation.

This framework bridges information theory, model-based reasoning, and semantic grounding to reimagine language models not as generators of text, but as engines of structured, compressible meaning.

2 Language-Native World Modeling Framework

We propose a framework that shifts from next-token prediction to structured world-state prediction, enabling language models to simulate and reason about environments described in natural language. Our system interprets natural language observations and goals, translates them into latent representations, and predicts how the world might evolve under hypothetical actions.

2.1 Problem Setup

The input to our system is a *time-indexed sequence* of natural language observations and optional actions, describing how an environment has evolved over time. At each timestep t, we assume access to:

- A natural language description of the current environment E_t
- (Optionally) past actions $a_{t-K:t-1}$
- A desired goal state G, also described in language

We encode these into structured latent representations:

$$z_t = f_{\text{enc}}(E_t), \quad g = f_{\text{enc}}(G)$$

Here, z_t represents the structured latent world state at time t, and g encodes the goal configuration. Crucially, the system does not treat E_t in isolation—rather, it models the environment's evolution as a *sequence* of latent states $z_{t-K:t}$ generated from a sequence of textual observations $E_{t-K:t}$.

Actions a_t are defined as vectors that specify intended modifications to the world (e.g., "pick up the cup"). However, not all latent dimensions are mutable. We define an affordance function:

$$a_t \in \phi(z_t)$$

which returns the set of valid action vectors conditioned on the current world state z_t. This enforces semantic plausibility by filtering out illogical or impossible changes. It might not be necessary for a well-trained latent dynamics model, but it provides a valuable inductive bias that constrains the action space, improves sample efficiency during planning, and enhances interpretability by explicitly modeling which transitions are feasible given the current semantic configuration.

To simulate how the world evolves, we apply a *history-aware latent dynamics model*:

$$z_{t+1} = f_{\text{dyn}}(z_{t-K:t}, a_{t-K:t})$$

The latent dynamics model operates autoregressively: at each step, it uses the previously predicted latent state z_{t+i} (rather than ground-truth latents) to generate the next z_{t+i+1}. This enables open-loop simulation over arbitrary horizons and avoids exposure bias that would arise from teacher-forced rollouts. Unlike traditional world models that assume Markovian dynamics, our model conditions on a sliding window of past latent states and actions—i.e., $z_{t+i-K:t+i}, a_{t+i-K:t+i}$—to model the next state. This design captures long-range temporal dependencies such as narrative structure, memory, and causal chaining in language. While autoregressive in structure, the model functions over structured latent space rather than token sequences, enabling recursive generation of coherent semantic trajectories that reflect evolving world dynamics.

The system rolls forward through time, generating a *trajectory* of future latent states:

$$\{z_{t+1}, z_{t+2}, \ldots, z_{t+H}\}$$

To evaluate the outcome, a verifier $V(z_{t+H}, g)$ scores how well the final predicted state aligns with the goal. Planning involves selecting the action sequence that maximizes this alignment:

$$a^* = \arg\max_{a_{t:t+H}} V(z_{t+H}, g)$$

Example. **History:** t–2: "The robot enters the kitchen." $t-1$: "It scans the counter for objects." t: "A cup lies on the floor."

Goal: "The cup is placed back on the counter."

Each sentence is encoded into a structured latent, forming the input sequence $z_{t-2:t}$. An action like "pick up the cup" is sampled within $\phi(z_t)$, and the model simulates future states $z_{t+1:t+H}$, selecting actions that bring the final state closer to g.

2.2 System Architecture

Our framework operates over sequences of latent world states derived from sequences of language. It consists of three key modules:

1. **Semantic Encoder.** A large language model encodes each time-indexed environment description E_t and goal G into structured latent vectors. Each dimension in z_t corresponds to a semantically meaningful world variable (e.g., agent location, object status, available actions), preserving temporal consistency and interpretability.
2. **Latent Dynamics Model.** A predictive model simulates how the environment evolves, using a window of past latent states and actions to forecast the next latent state:

$$z_{t+1} = f_{\text{dyn}}(z_{t-K:t}, a_{t-K:t})$$

The latent dynamics model operates autoregressively: at each step, it uses the previously predicted latent state z_{t+i} (rather than ground-truth latents) to generate the next z_{t+i+1}. This enables open-loop simulation over arbitrary horizons and avoids exposure bias that would arise from teacher-forced rollouts. Unlike traditional world models that assume Markovian dynamics, our model conditions on a sliding window of past latent states and actions—i.e., $z_{t+i-K:t+i}, a_{t+i-K:t+i}$—to model the next state. This design captures long-range temporal dependencies such as narrative structure, memory, and causal chaining in language. While autoregressive in structure, the model functions over structured latent space rather than token sequences, enabling recursive generation of coherent semantic trajectories that reflect evolving world dynamics.
3. **Verifier.** A downstream model scores alignment between the final predicted world state and the goal representation, enabling trajectory evaluation and action planning.

This design treats the entire modeling task as a *sequence-to-sequence prediction problem* in latent semantic space, rather than in token space.

2.3 Hybrid Latent Representations: From Structure to Optimization

To balance semantic interpretability with simulation efficiency, we introduce a two-stage encoding pipeline: a structured semantic latent is first projected into a compact, optimized space, and then discretized via a learned vocabulary of symbolic latent tokens.

Formally, let $\{z_{t-K}, \ldots, z_t\}$ be a time-indexed sequence of structured latent vectors derived from natural language. Each z_t is projected into an intermediate continuous space via:

$$z_t^{\text{cont}} = f_{\text{proj}}(z_t)$$

To enable discrete reasoning and efficient planning, we define a finite vocabulary $\mathcal{Z} = \{e_1, e_2, \ldots, e_N\} \subset \mathbb{R}^d$ of learnable latent codes, or *codebook entries*, where each e_i represents a symbolic configuration of the world.

We then assign each projected latent to its nearest symbolic token in the codebook:

$$z'_t = \text{nearest\_code}(z_t^{\text{cont}}) = \arg\min_{e_i \in \mathcal{Z}} \|z_t^{\text{cont}} - e_i\|^2$$

The latent dynamics model operates over sequences of these discrete, symbolic latents:

$$z'_{t+1} = f_{\text{dyn}}(z'_{t-K:t}, a_{t-K:t})$$

To support human-level interpretability, supervision, and debugging, we define a structured decoder that maps symbolic latents back into interpretable structured form:

$$\tilde{z}_t = f_{\text{struct-dec}}(z'_t)$$

This round-trip mechanism—structured latent to optimized vector to symbolic token and back—preserves the semantic content of natural language while supporting efficient latent simulation and discrete entropy computation. By leveraging a learned latent vocabulary, we enable the use of token-like semantics within continuous neural systems, unifying the strengths of symbolic reasoning with differentiable modeling.

3 Information-Theoretic Foundations

Our framework is grounded in the principle that intelligent behavior emerges from the ability to compress and predict structured sequences of world representations. Drawing from Shannon's insight—that entropy bounds the minimal number of bits required to describe a signal—we extend this concept from sequences of linguistic tokens to trajectories of latent semantic states, each encoding interpretable aspects of an evolving environment.

3.1 From Token Entropy to Latent Trajectory Entropy

Traditional language models quantify uncertainty over the next token x_{t+1} given a sequence history $x_{1:t}$ using conditional entropy:

$$H(X_{t+1} \mid x_{1:t}) = -\sum_{x \in \mathcal{V}} P(x \mid x_{1:t}) \log P(x \mid x_{1:t})$$

Here, $\mathcal{V}$ is the vocabulary of possible tokens. This formulation captures uncertainty in syntactic continuation, but remains agnostic to the underlying semantic or causal structure that may drive the evolution of the described environment.

3.2 Latent Representation of Sequential Context

We instead operate over structured semantic states derived from sequences of natural language descriptions. Let:

- $h_{t-K:t}$: a sequence of historical language inputs describing the environment and agent behavior over the last K steps,
- $z_{t-K:t} = f_{\text{enc}}(h_{t-K:t})$: a corresponding sequence of structured latent vectors summarizing the semantic world state at each timestep.

Each latent vector $z_i \in z_{t-K:t}$ encodes interpretable semantic variables such as object properties, agent positions, and affordance structure. The evolution of the world is modeled as a transition through this latent space:

$$z_{t+1} = f_{\text{dyn}}(z_{t-K:t}, a_{t-K:t})$$

3.3 Entropy over Latent Trajectories

To generalize the notion of entropy to structured world modeling, we define the uncertainty over the next latent state z_{t+1}—or more generally, over a full future trajectory $\{z_{t+1}, \ldots, z_{t+H}\}$—conditioned on the observed history:

$$\begin{aligned} &H(Z_{t+1} \mid z_{t-K:t}, a_{t-K:t}) \\ = -\sum_{z \in \mathcal{Z}} \quad &P(z_{t+1} = z \mid z_{t-K:t}, a_{t-K:t}) \\ &\cdot \log P(z_{t+1} = z \mid z_{t-K:t}, a_{t-K:t}) \end{aligned}$$

Here, $\mathcal{Z}$ denotes the space of valid structured latent world states. This formulation captures the semantic uncertainty over how the world may evolve given past context and agent actions.

This entropy formulation implicitly assumes an autoregressive rollout procedure, where each predicted state z_{t+i+1} depends on previously generated latents $z_{t+i-K:t+i}$ and actions. This open-loop structure reflects the model's belief distribution over future semantic trajectories, rather than sampling from ground-truth sequences. As such, entropy over latent trajectories captures uncertainty not just in state transitions, but in the model's capacity to simulate plausible futures recursively.

3.4 Reasoning as Predictive Compression

Under this framework, reasoning is reframed as the process of minimizing semantic entropy across time: learning a dynamics model that compresses and predicts structured trajectories of latent world states. The lower the entropy over future states, the more structured and coherent the agent's internal model of the world is.

The latent dynamics model f_{dyn} thus serves as a compression engine: it transforms a sequence of past states and actions into a predictive representation of what comes next. Successful modeling across a planning horizon implies the system has internalized causal, temporally extended patterns—allowing it to simulate hypothetical futures and evaluate their alignment with goals.

In contrast to token-level prediction, our formulation operates on interpretable, semantically grounded representations. This aligns reasoning with information-theoretic principles: intelligent behavior emerges from minimizing uncertainty over structured world-state trajectories, not over strings.

4 Planning in Latent Space

Our framework enables goal-directed reasoning by simulating the evolution of semantic world states entirely within a structured latent space. Unlike traditional language models that operate in token space, we treat latent representations—derived from sequences of language inputs—as the substrate for prediction, planning, and evaluation. This design allows efficient and interpretable simulation without decoding intermediate states into natural language.

4.1 Structured Latent Trajectory Modeling

Given a history of natural language environment descriptions $h_{t-K:t}$, we encode them into a sequence of structured latent states:

$$z_{t-K:t} = f_{\text{enc}}(h_{t-K:t}), \quad g = f_{\text{enc}}(G)$$

Here, g encodes the desired goal state. Starting from this sequence $z_{t-K:t}$, and given a corresponding history of past actions $a_{t-K:t}$, the latent dynamics model simulates the next world state:

$$z_{t+1} = f_{\text{dyn}}(z_{t-K:t}, a_{t-K:t})$$

By unrolling this transition iteratively over a planning horizon H, we generate a full trajectory of predicted future world states:

$$\{z_{t+1}, z_{t+2}, \ldots, z_{t+H}\}$$

Each predicted latent state reflects a semantically coherent configuration of the world, grounded in the temporal logic and affordances of prior states. Simulation occurs entirely in latent space, avoiding the inefficiencies of autoregressive language generation while preserving interpretability through structured representations.

Decoding via f_{dec} is optional and used for explanation, debugging, or alignment with human supervision. The core planning mechanism operates entirely on compressed, structured trajectories.

4.2 Goal Evaluation and Trajectory Selection

To determine whether a simulated trajectory leads toward the intended goal G, we evaluate the final predicted state z_{t+H} against the goal representation g using a verifier function $V(z_{t+H}, g)$. This function can take various forms:

- A learned model trained to regress task completion scores
- A similarity metric (e.g., cosine similarity) between z_{t+H} and g
- A contrastive model that distinguishes goal-aligned outcomes from distractors

The planning objective becomes a trajectory optimization problem:

$$a^* = \arg\max_{a_{t:t+H}} V(z_{t+H}, g)$$

where a^* is the sequence of actions that leads to the predicted state most consistent with the goal. This evaluation guides action selection within the latent space simulation loop.

4.3 Planning Loop

Our structured planning system proceeds as follows:

1. Encode the recent history $h_{t-K:t}$ into a latent sequence $z_{t-K:t}$
2. Generate or sample candidate action sequences $a_{t:t+H}$ constrained by $\phi(z_t)$
3. Roll out future latent states using the autoregressive dynamics model f_{dyn}, where each predicted state z_{t+i} is generated based on previously predicted latents $z_{t+i-K:t+i}$ and actions
4. Score the predicted final state z_{t+H} using the verifier $V(z_{t+H}, g)$
5. Select the best-scoring trajectory for execution, further rollout, or decoding

Because the latent dynamics model operates over a symbolic vocabulary of learned tokens, planning proceeds over discrete trajectories in latent space, enabling entropy-based evaluation and efficient candidate enumeration. This architecture enables efficient, interpretable planning that is fully grounded in natural language input yet decoupled from token-level generation. It eliminates the need for symbolic logic engines or grounded physics simulators by leveraging the structured semantics of language-derived latent space. By treating world modeling as sequence-to-sequence prediction in meaning space, our approach supports long-horizon, goal-directed reasoning under uncertainty.

5 Discussion

This paper proposes a theoretical blueprint for *language-native world modeling*—an architectural paradigm in which reasoning, planning, and simulation are carried out not over sequences of tokens, but over sequences of structured latent states derived from natural language. We reframe language models not as string generators but as predictive compressors operating over evolving trajectories of semantic meaning.

We intentionally do not present empirical results. As an independent researcher, the resources necessary to implement, train, and evaluate this framework are beyond the current scope. However, this work is designed to serve as a conceptual foundation—akin to early proposals in model-based reinforcement learning and probabilistic programming—that invites empirical exploration from the research community.

Several key design principles emerge from this framework:

- **Modular Latent Reasoning.** Our system decomposes the reasoning process into interpretable modules: a semantic encoder that processes sequences of language into structured latents, a latent dynamics model that simulates transitions across time, and a verifier that evaluates alignment with goals. This structure supports transparent, compositional reasoning distinct from monolithic prompting.
- **Structured-to-Optimized Latents.** Semantic latent states retain interpretable variables grounded in language. These can be projected into optimized latent embeddings for efficient simulation and then decoded back into structured form, enabling both flexibility and traceability.
- **Reasoning as Predictive Compression.** Intelligence is operationalized as entropy minimization over future latent trajectories. Rather than generating probable token sequences, the model reduces uncertainty over how the world evolves—selecting action sequences that yield semantically coherent, goal-directed futures.

While each component—semantic encoding via LLMs, sequence modeling via learned dynamics, and vector-space goal evaluation—builds on established techniques, their integration into a unified, language-grounded trajectory simulator is novel. Crucially, our framework aligns with both the strengths of language models and the structure of sequential decision-making, offering a pathway toward interpretable agents capable of simulating semantically rich environments.

Future empirical work could investigate several questions: Can structured-to-learned latent projections improve generalization across tasks? Does entropy over latent state trajectories correlate with cognitive complexity or planning difficulty? Do latent-space rollouts outperform token-space decoders in robustness and alignment?

Although our framework diverges from traditional autoregressive decoding in token space, it retains an autoregressive structure in latent space: the dynamics model recursively generates each future state from previously predicted latents and actions. This preserves temporal continuity and supports open-ended semantic simulation across planning horizons.

This paper is ultimately a provocation and an invitation: to reimagine language models not merely as predictors of strings, but as engines of structured simulation. To model not what is said, but what is meant. To build agents that act in latent worlds and reason over semantic sequences—compressing the unfolding of experience not word by word, but thought by thought.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901, 2020.

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

[3] 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 36th International Conference on Machine Learning (ICML)*, volume 97, pages 2555–2565. PMLR, 2019.

[4] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 1–22. ACM, 2023.

[5] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[6] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

[7] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[8] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824–24837, 2022.