Mimir: A Hierarchical Context Architecture for Grounded Perception

Your Name

October 5, 2025

Abstract

This document specifies a complete, self-improving system architecture designed to learn a grounded model of language and other modalities. The architecture leverages a symmetric Teacher-Student bootstrapping paradigm for all sensory streams. Initially, powerful pre-trained models act as "Teachers" to train dedicated Proteus instances on the geometry of primitive perceptual embeddings. These Proteus models then provide the ground truth for training streamlined "Student" models, each using a modality-appropriate local context-forming mechanism (e.g., an EGNN for vision). The outputs of these unimodal experts are then used to train a higher-level ‘Proteus-M‘ instance on the geometry of co-occurrence, which discovers the structure of a Shared Multimodal Latent Space (SMLS). The fuzzy membership over this shared space provides a unified context vector for higher-level reasoning engines, grounding abstract concepts in perception.

1 Introduction

A primary challenge in artificial intelligence is grounding abstract concepts in raw sensory data. This paper presents a hierarchical context architecture designed to solve this problem by progressively building richer context through a sequence of distinct learning stages. Our design philosophy is built on four core principles:

1. **Symmetric Bootstrapping:** We kickstart learning by distilling knowledge from large pre-trained Teacher models into a common engine, Proteus.
2. **Modality-Appropriate Local Context:** The system uses different, specialized mechanisms for initial context formation, respecting the native structure of different data types (e.g., spatial for images, sequential for text).
3. **A Unified Multimodal Manifold:** The system discovers the structure of a shared multimodal space by training a dedicated ‘Proteus-M‘ instance on the concatenated outputs of the unimodal experts.

4. **Recursive Self-Improvement:** The system can use its own generated outputs as a new ”ground truth” to train more refined versions of itself.

This paper will detail this multi-stage pipeline, from the initial unimodal student training to the final discovery of the shared multimodal latent space.

2 Unimodal Student Training

The first stage of the pipeline trains independent, efficient ”Student” models that learn to reproduce the concepts discovered by large Teacher models, but using only local context. This acts as a variable-to-fixed-size data processing layer where computational cost is proportional to input fidelity.

2.1 The Visual Student (‘Student-V‘)

The visual student is an E(2)-Equivariant Graph Network (EGNN) that operates on a k-NN graph of image patches. Its objective is not only compression but to learn a **disentangled latent** where content/identity is separated from pose/scale/illumination.

Concretely, the EGNN produces a variable-sized set of per-patch latents $\{(z_{inv,i}, z_{eq,i})\}$. Training uses a **multi-part objective** that induces a *learned* partition of each latent into an *invariant* feature subspace z_{inv} and a complementary *equivariant/covariant* subspace z_{eq}:

- **Reconstruction:** a decoder reconstructs the input from the full latent z, preserving information.
- **Invariance:** an invariance criterion (e.g., VICReg/Barlow Twins or InfoNCE on augmented pairs) is applied so that $z_{inv}(x) \approx z_{inv}(T\,x)$ for SE(2)+scale transforms T, yielding a stable feature subspace suitable for Proteus.
- **Equivariance:** a complementary loss encourages $z_{eq}(T\,x) \approx R(T)\,z_{eq}(x)$ for a simple representation R of the transform (e.g., rotation in a low-dimensional head), concentrating transformation factors outside z_{inv}.

Per-patch manifold and densification:

1. The set of invariant patch vectors $\{z_{inv,i}\}$ is passed through a pre-trained, modality-specific Proteus instance (e.g., ‘Proteus-V‘). This yields sparse, variable-dimension membership vectors $\{\mu_{V,i}\}$.
2. The sparse set $\{\mu_{V,i}\}$ is passed through a lightweight Elastic Autoencoder (a ”primitive codec”) to produce dense embeddings $\{\nu_{V,i}\}$.
3. The set $\{(\nu_{V,i}, z_{eq,i})\}$ is then summarized by an Attention-Based Aggregator into a single, fixed-size, disentangled signature for the image: $(\Phi_{V,inv}, \Phi_{V,eq})$.

Masking at multiple levels: The same mask-based partitioning is applied *wherever a representation is produced.* Concretely: (i) at the **patch level**, each latent z_i is split into $(z_{inv,i}, z_{eq,i})$ by a mask head m^{patch}; (ii) at the **aggregator level**, the output signature Φ_V is also split into $(\Phi_{V,inv},\, \Phi_{V,eq})$ by an independent mask head m^{agg}. Each level receives its own invariance/equivariance objectives appropriate to its scope (local SE(2)+scale for patches; global transforms/state for the image-level signature), plus the binary bias, capacity target, and cross-covariance regularizers to preserve a clean separation.

This multi-part objective is not a collection of independent heuristics, but a system of competing pressures designed to be the minimal set of constraints for learning a useful, disentangled representation. The **Invariance Loss** acts as the primary differentiable proxy for the stationarity that Proteus requires, forcing the model to ignore nuisance variables. However, this pressure alone would lead to a trivial solution (e.g., $z_{inv} = 0$), a problem known as representation collapse. The **Reconstruction Loss** provides the necessary counter-pressure, ensuring that the latent vector retains enough information to be useful. Finally, the **Equivariance Loss** provides a structured destination for the nuisance information that the invariance loss necessarily discards, organizing it in a predictable way for downstream temporal models. The mask regularization terms then enforce a clean separation between these two subspaces. This principled trade-off is the core of the student's learning process.

Unimodal concept manifold and densification. Across many inputs, the fixed-size invariant signatures $\{\Phi_{V,inv}\}$ are used to train a second Proteus instance (e.g., a holistic 'Proteus-Image') that learns the manifold of complete visual concepts. At inference, its sparse output (e.g., $\Psi_{V,inv}$) is passed through an Elastic Autoencoder (a "holistic codec") to yield a final dense, stable, fixed-size unimodal vector $C_{V,inv}$. The pair $(C_{V,inv}, \Phi_{V,eq})$ is the final per-image output forwarded to higher levels.

Latent partition and objective (generic form). For any representation vector $r \in \mathbb{R}^D$ (patch-level z_i or aggregated signature Φ_V), a learned mask $m(r) \in [0,1]^D$ produces

$$r_{inv} = m(r) \odot r, \qquad r_{eq} = (\mathbf{1} - m(r)) \odot r.$$

Patch-level objectives use local SE(2)+scale augmentations; aggregate-level objectives use global transforms/state. We optimize the weighted objective

$$\mathcal{L} = \lambda_{\text{rec}}\, \mathcal{L}_{\text{rec}} + \lambda_{\text{inv}}\, \mathcal{L}_{\text{inv}} + \lambda_{\text{eq}}\, \mathcal{L}_{\text{eq}} + \lambda_{\text{bin}}\, \mathcal{L}_{\text{bin}} + \lambda_{\text{size}}\, \mathcal{L}_{\text{size}} + \lambda_{\text{cov}}\, \mathcal{L}_{\text{cov}}.$$

A typical instantiation is identical to the per-patch formulation, replacing z with the appropriate r and and using the transform representation $R(T)$ consistent with that level (e.g., image-wise rotation for Φ_V).

2.2 The Sequential Students (‘Student-T‘, ‘Student-A‘)

The students for temporal data like text and audio use a gated recurrent architecture. The core model is a multi-rate EMA bank (a ”mixture of decays”) that allows the model to capture context over multiple time scales simultaneously. As with the visual student, an Attention-Based Aggregator is used to convert the variable-length sequence of context-aware outputs into a single, fixed-size signature vector for downstream processing.

3 Multimodal Context Bootstrapping

This crucial stage teaches the system how different modalities relate to each other by learning the geometry of their co-occurrence.

3.1 Training ‘Proteus-M‘

A new, high-level Proteus instance, ‘Proteus-M‘, is trained on a dataset of concatenated signature vectors produced by the unimodal students. For each co-occurring event (e.g., an image-text pair), a single vector is created:

$$v_{\text{multimodal}} = [v_{\text{frame\_sig}}, v_{\text{text\_sig}}, v_{\text{audio\_sig}}, \dots]$$

‘Proteus-M‘ discovers the natural clusters in this space, which represent abstract, multimodal concepts. For example, it learns a single ”dog” concept that is activated by images of dogs, the word ”dog”, and the sound of barking.

3.2 The Shared Multimodal Latent Space (SMLS)

The result of this process is a trained ‘Proteus-M‘ model that defines a Shared Multimodal Latent Space. The fuzzy membership vector over this space, `v_Context_Unified`, serves as the final, unified context for any given moment in time, providing a rich, abstract representation to the higher-level cognitive core. This model is also bidirectional, enabling concept-driven generative synthesis.

4 The Bootstrapping Philosophy

A foundational principle of this architecture is the use of a Teacher-Student bootstrapping paradigm to kickstart the learning process. We posit that this is not merely a practical shortcut, but reflects a deeper principle about efficient knowledge acquisition.

We can think of the powerful, pre-trained ”Teacher” models (e.g., large language or vision transformers) as an analogue to evolution. They are the product of an immense, ”brute-force” optimization process over vast datasets, resulting in a strong, general-purpose understanding of the world. An agent learning ”from scratch” would be prohibitively expensive and slow, akin to a single organism attempting to recapitulate millions of years of evolution.

Instead, our approach leverages the distilled knowledge of these ”evolutionary” teachers. The initial Proteus instances learn the essential geometry of the world from the teachers' outputs. The streamlined Student models then learn to reproduce this essential geometry in a much more efficient, self-supervised manner. This positions the agent to begin its own lifecycle of learning from a highly advanced starting point, allowing it to focus its limited computational resources on adapting and refining its knowledge, rather than discovering every foundational concept from first principles.

5 Conclusion

We have specified a complete, staged, and bootstrapping architecture for a grounded perceptual system. By separating the system into clear stages—unimodal expertise, shared context learning, and high-order temporal analysis—it resolves complex dependencies in a principled way. By using Proteus at multiple levels of abstraction, the system creates a truly unified and scalable foundation for the final Mimir engine to discover deep, abstract concepts from any combination of sensory inputs. This architecture provides a robust and extensible framework for building perceptually grounded AI systems that can learn, adapt, and improve over time.

References