Universe-Models

A Complete and Illustrated Technical Specification

Author: TSIGBE Comlan Alain

Version 2.0 - Enriched Document

September 21, 2025

Contents

Executive Summary

Overview

This document presents a complete technical specification of the **Universe-Models** framework, embodied by a **Cognitive Equivariant Agent (CEA)**. The objective is to create an autonomous agent capable of learning a generative, causal, and geometrically structured simulation of its environment from multimodal sensory data.

1 Introduction and Vision

1.1 Context and Motivation

The primary objective of a Universe-Model is to acquire, after training, an intrinsic and deep understanding of reality, and therefore of the universe. Unlike current learning systems, which require significant subsequent work to refine and improve the model, this framework aims for more autonomous comprehension.

The Universe-Models framework addresses a fundamental challenge in artificial intelligence: how can an agent build a deep and causal understanding of its world?

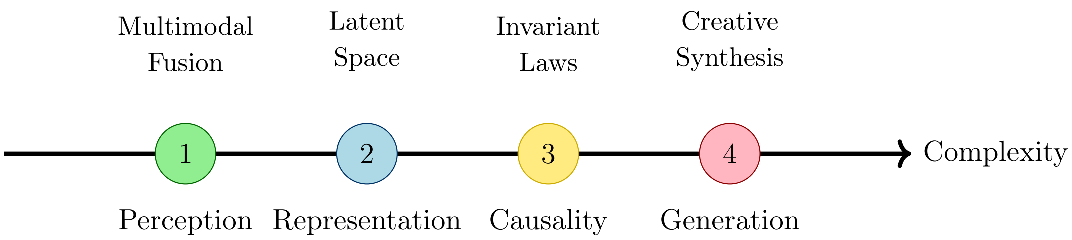

Figure 1: Cognitive processing pipeline of the agent

1.2 Fundamental Principles

I. **Robust Perception**: Coherent integration of noisy multimodal data.

II. **Active Uncertainty Management**: Explicit representation of doubt.

III. **Causal Disentanglement**: Separation of physical laws from the point of view.

IV. **Hierarchical Abstraction**: Multi-scale organization of concepts.

V. **Generative Expression**: Synthesis of coherent multimodal content.

2 Detailed Mathematical Architecture

2.1 Spaces and Notations

Fundamental Spaces

$$\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_M \quad \text{(Observation space)} \tag{1}$$
$$\mathcal{Z} = \mathbb{R}^d \quad \text{(Latent space of dimension } d\text{)} \tag{2}$$
$$\mathcal{A} = \{a_1, a_2, \ldots, a_K\} \quad \text{(Action space)} \tag{3}$$
$$\Theta = \{\phi, \theta, \eta, \psi, \omega, \boldsymbol{\Omega}_{\text{agent}}\} \quad \text{(Model parameters)} \tag{4}$$

2.2 Module 1: Robust Multimodal Perception

2.2.1 Perception Module Architecture

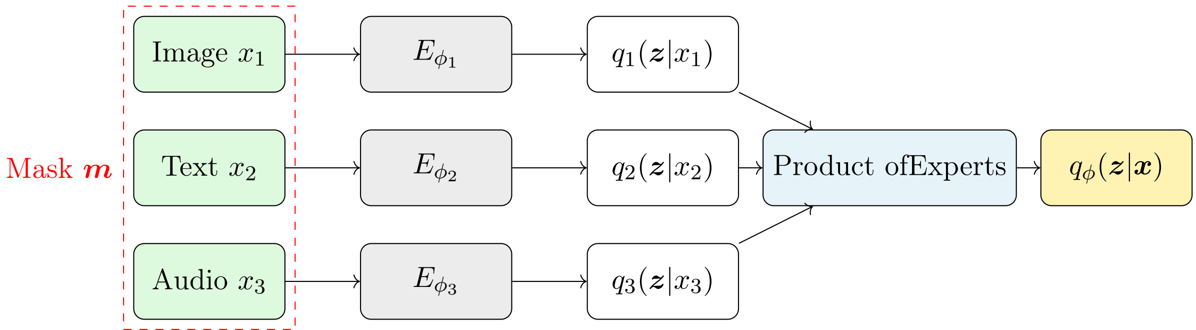

Figure 2: Architecture of the perception module with Product of Experts fusion

2.2.2 Product of Experts (PoE) Fusion - Mathematical Details

PoE fusion combines the beliefs from each modality in a probabilistic manner:

PoE Fusion Calculation

For Gaussian distributions $q_i(\boldsymbol{z}|x_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, the fused distribution is:

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}_{\text{vis}}) = \mathcal{N}(\boldsymbol{\mu}_\phi, \boldsymbol{\Sigma}_\phi) \tag{5}$$

where the precision (inverse covariance) is additive:

$$\boldsymbol{\Sigma}_\phi^{-1} = \sum_{i \in \text{vis}} w_i \boldsymbol{\Sigma}_i^{-1} + \lambda_{\text{reg}} \mathbf{I}_d \tag{6}$$

The fused mean is then:

$$\boldsymbol{\mu}_\phi = \boldsymbol{\Sigma}_\phi \left(\sum_{i \in \text{vis}} w_i \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i \right) \tag{7}$$

The weights w_i are calculated based on entropy:

$$w_i = \frac{\exp(-H[q_i])}{\sum_{j \in \text{vis}} \exp(-H[q_j])} \tag{8}$$

$$H[q_i] = \frac{1}{2} \log \left((2\pi e)^d \cdot \text{Det}(\boldsymbol{\Sigma}_i) \right) \tag{9}$$

2.2.3 Contrastive Alignment

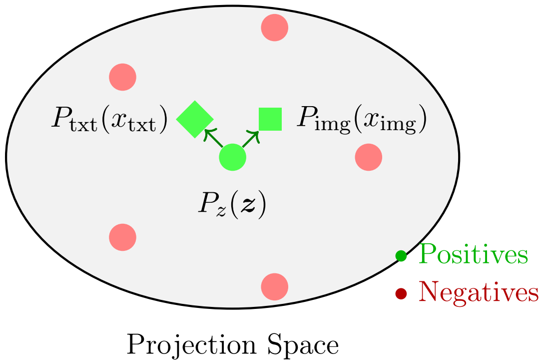

Figure 3: Contrastive alignment in the projection space

The InfoNCE contrastive loss function is written as:

$$\mathcal{L}_{\text{ctr}} = -\sum_{i=1}^{M} \mathbb{E}_{\mathcal{B}} \left[\log \frac{\exp(\text{sim}(P_z(\boldsymbol{z}), P_{m_i}(x_i))/\tau)}{\sum_{x_i' \in \mathcal{B}} \exp(\text{sim}(P_z(\boldsymbol{z}), P_{m_i}(x_i'))/\tau)} \right] \tag{10}$$

where $\text{sim}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^T \boldsymbol{b}}{||\boldsymbol{a}|| \cdot ||\boldsymbol{b}||}$ is the cosine similarity.

2.3 Module 2: Neural Geometry of the Latent Space

2.3.1 Geometric Structure and Uncertainty

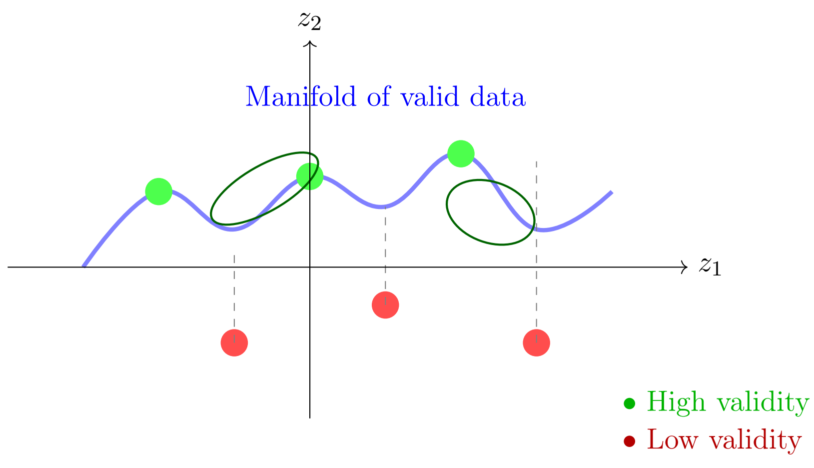

Figure 4: Manifold of valid data and covariance adaptation

2.3.2 Is-It-Valid (IIV) Estimator

The IIV estimator is a discriminator network that evaluates the probability that a latent point corresponds to real data:

IIV Estimator Architecture

$$D_\psi : \mathcal{Z} \to [0, 1] \tag{11}$$

$$p_{\text{valid}}(\boldsymbol{z}) = \sigma\left(\text{MLP}_\psi(\boldsymbol{z})\right) \tag{12}$$

where MLP_ψ is a multilayer perceptron and σ is the sigmoid function.

2.3.3 Adaptive Geometric Regularization

The geometric regularization loss sculpts the covariance matrix according to directional risk:

$$\mathcal{L}_{\text{geom}} = \sum_{k=1}^{d} w_k(\boldsymbol{\mu}_\phi) \cdot \text{Tr}\big(\mathbf{A}_k^{-1}\boldsymbol{\Sigma}_{\text{proj},k}\big) + \lambda_\kappa \cdot \max(0, \kappa(\boldsymbol{\Sigma}_\phi) - \kappa_{\max}) \tag{13}$$

where:

- $w_k(\boldsymbol{\mu}_\phi) = 1 - p_{\text{valid}}(\boldsymbol{\mu}_\phi + \epsilon \boldsymbol{u}_k)$ measures the risk in direction $\boldsymbol{u}_k$.
- $\mathbf{A}_k \in \text{SPD}(d)$ are learned matrices representing global uncertainty axes.
- $\kappa(\cdot)$ is the condition number of the matrix.

2.4 Module 3: Equivariant Causal Dynamics

2.4.1 Agent-Environment Separation

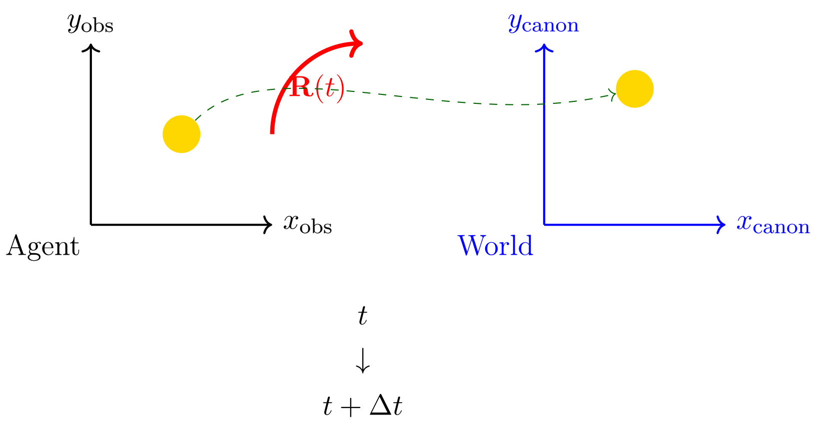

Figure 5: Transformation between observed (agent) and canonical (world) reference frames

The relationship between reference frames is expressed as:

Equivariant Transformation

$$\boldsymbol{z}_{\text{canon}}(t) = \mathbf{R}_{\text{agent}}^{-1}(t) \cdot \boldsymbol{z}_{\text{obs}}(t) \tag{14}$$

$$\mathbf{R}_{\text{agent}}(t) = \exp(t \cdot \boldsymbol{\Omega}_{\text{agent}}) \tag{15}$$

$$\boldsymbol{\Omega}_{\text{agent}} \in \mathfrak{so}(d) \quad \text{(skew-symmetric matrix)} \tag{16}$$

Dynamics in the canonical frame are simple:

$$\boldsymbol{z}_{\text{canon}}(t+1) = f_\eta(\boldsymbol{z}_{\text{canon}}(t), a_t) \tag{17}$$

In the observed frame, they become:

$$\boldsymbol{z}_{\text{obs}}(t+1) = \mathbf{R}_{\text{agent}}(t+1) \cdot f_\eta(\mathbf{R}_{\text{agent}}^{-1}(t) \cdot \boldsymbol{z}_{\text{obs}}(t), a_t) \tag{18}$$

2.4.2 Commutative Diagram

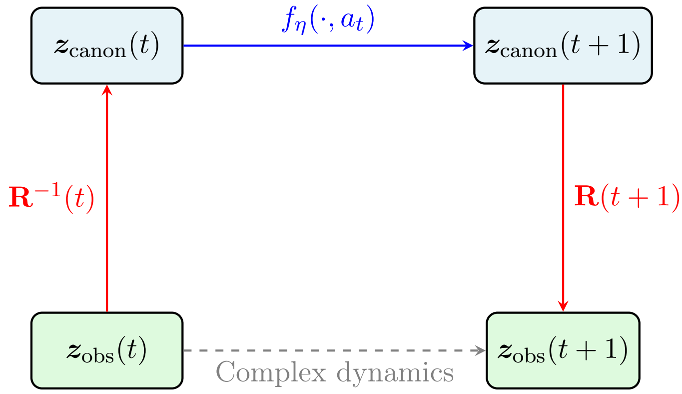

Figure 6: Commutative diagram of equivariant dynamics

2.5 Module 4: Hierarchical Conceptual Memory

2.5.1 Residual Vector Quantization (RVQ)

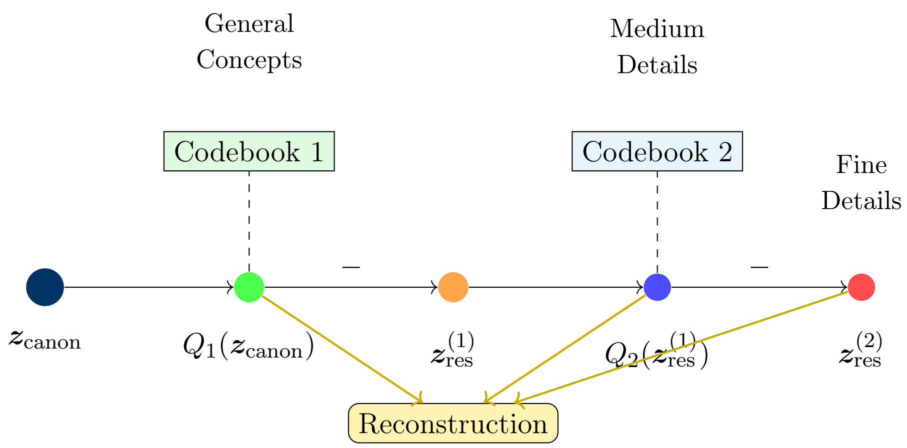

Figure 7: Hierarchical Residual Vector Quantization

2.5.2 Hierarchical Organization of Concepts

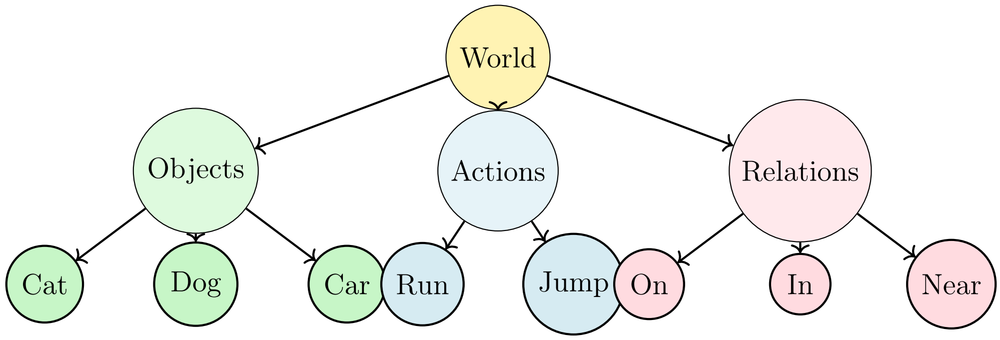

Figure 8: Conceptual hierarchy in memory

The abstraction loss combines the VQ-VAE objectives across all levels:

Abstraction Loss Function

$$\mathcal{L}_{\text{abstract}} = \sum_{l=1}^{L} \mathcal{L}_{\text{VQ}}^{(l)} \tag{19}$$

$$\mathcal{L}_{\text{VQ}}^{(l)} = \underbrace{\|\text{sg}\left[[]\, \boldsymbol{z}_{\text{res}}^{(l-1)}\right] - c_{l,k^*}\|_2^2}_{\text{Codebook update}} + \beta \underbrace{\|\boldsymbol{z}_{\text{res}}^{(l-1)} - \text{sg}\left[[]\, c_{l,k^*}\right]\|_2^2}_{\text{Commitment loss}} \tag{20}$$

where $\text{sg}\left[[]\,\cdot\right]$ is the stop-gradient operator and c_{l,k^*} is the nearest codebook vector.

2.6 Module 5: Generative Synthesis via Diffusion

2.6.1 Diffusion Process

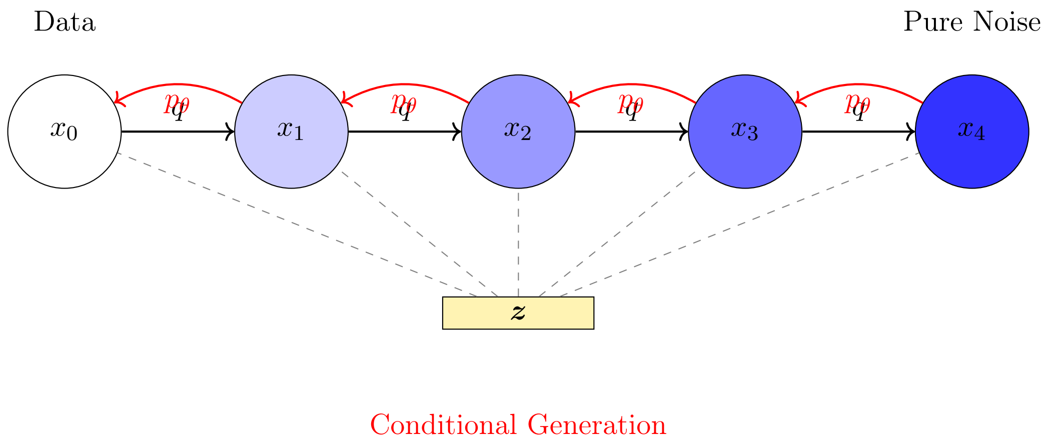

Figure 9: Conditional diffusion process

2.6.2 Denoising Network Architecture

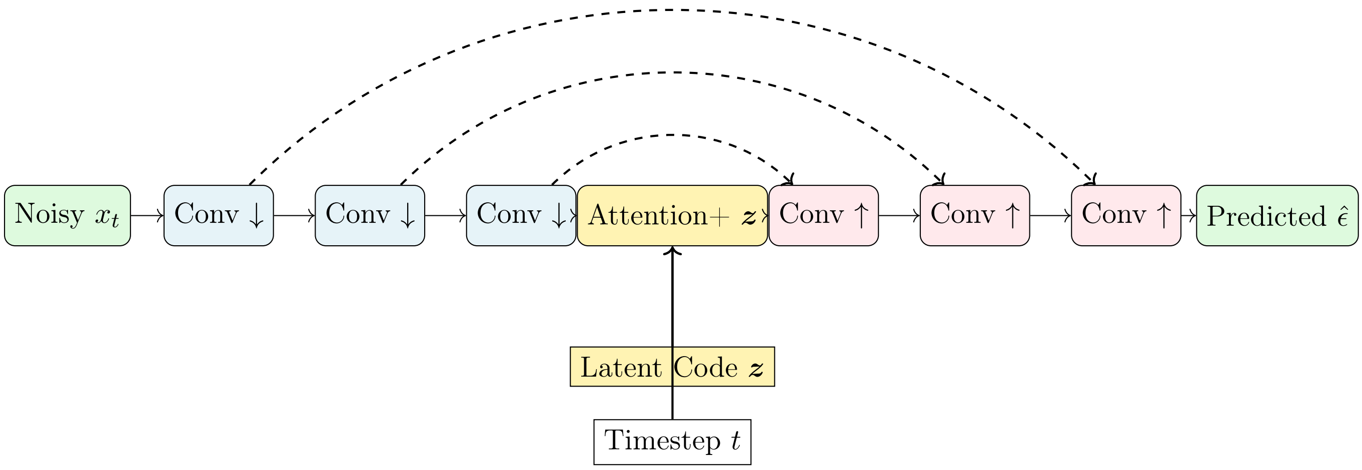

Figure 10: U-Net architecture for conditional diffusion

The loss function for diffusion is written as:

Simplified Diffusion Objective

$$\mathcal{L}_{\text{diff}} = \sum_{m=1}^{M} \mathcal{L}_{\text{diff}}^{(m)} \tag{21}$$

$$\mathcal{L}_{\text{diff}}^{(m)} = \mathbb{E}_{t \sim U[1,T], x_{m,0} \sim p_{\text{data}}, \boldsymbol{\epsilon} \sim \mathcal{N}(0,I)} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_m}(x_{m,t}, t, \boldsymbol{z})\|_2^2 \right] \tag{22}$$

where $x_{m,t} = \sqrt{\bar{\alpha}_t} x_{m,0} + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}$ is the noisy sample at time t.

3 Multi-Objective Optimization

3.1 Pareto Front Visualization

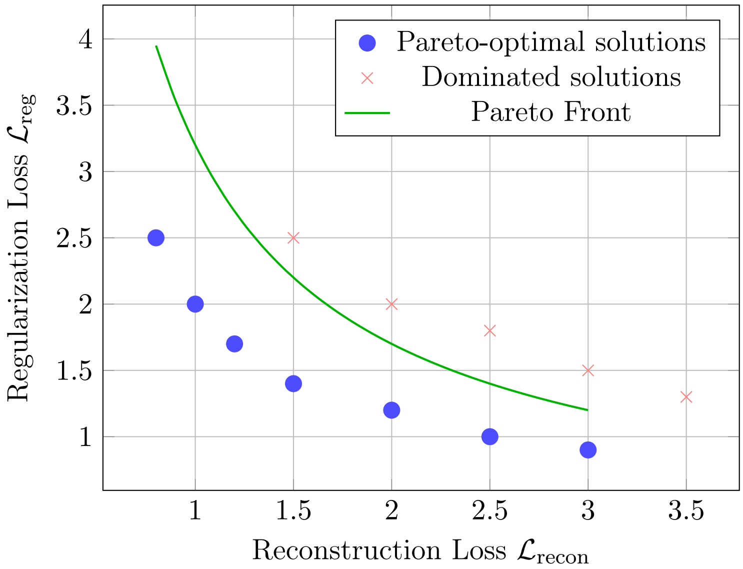

Figure 11: Pareto front for multi-objective optimization

3.2 Pareto-Optimal Gradient Descent Algorithm

Multi-Objective Optimization Problem

Given N objectives $\{\mathcal{L}_i\}_{i=1}^N$, we seek:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}\in\Delta^N} \left\| \sum_{i=1}^N \alpha_i \nabla_\Theta \mathcal{L}_i \right\|_2^2 \tag{23}$$

$$\text{where } \Delta^N = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^N : \sum_{i=1}^N \alpha_i = 1, \alpha_i \geq 0 \right\} \tag{24}$$

This quadratic problem can be solved efficiently using quadratic programming methods.

Algorithm 1 Pareto-Optimal Multi-Objective Optimization

Require: Losses $\{\mathcal{L}_i\}_{i=1}^N$, Parameters Θ, Learning rate γ

1: Compute gradients $\{\boldsymbol{g}_i = \nabla_\Theta \mathcal{L}_i\}_{i=1}^N$
2: Form the Gram matrix $\mathbf{G}_{ij} = \langle \boldsymbol{g}_i, \boldsymbol{g}_j \rangle$
3: Solve: $\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}\in\Delta^N} \boldsymbol{\alpha}^T \mathbf{G} \boldsymbol{\alpha}$
4: Compute combined gradient: $\boldsymbol{g}_{\text{combined}} = \sum_{i=1}^N \alpha_i^* \boldsymbol{g}_i$
5: Update: $\Theta \leftarrow \Theta - \gamma \cdot \boldsymbol{g}_{\text{combined}}$
6: Project onto constraints (e.g., $\boldsymbol{\Omega}_{\text{agent}} \in \mathfrak{so}(d)$)

4 Intrinsically Motivated Exploration

4.1 Components of the Intrinsic Reward

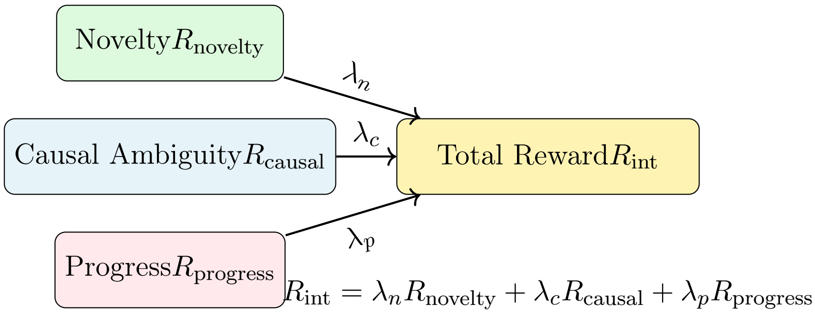

Figure 12: Architecture of the intrinsic reward

4.2 Exploration Strategies

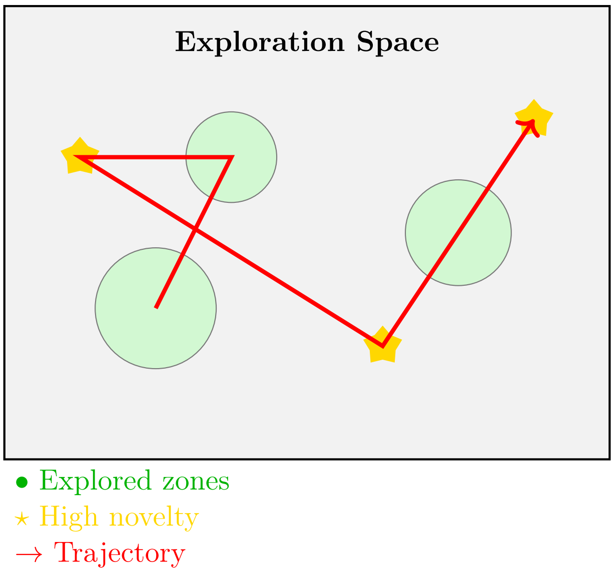

Figure 13: Curiosity-driven exploration strategy

5 Practical Implementation

5.1 Overall System Architecture

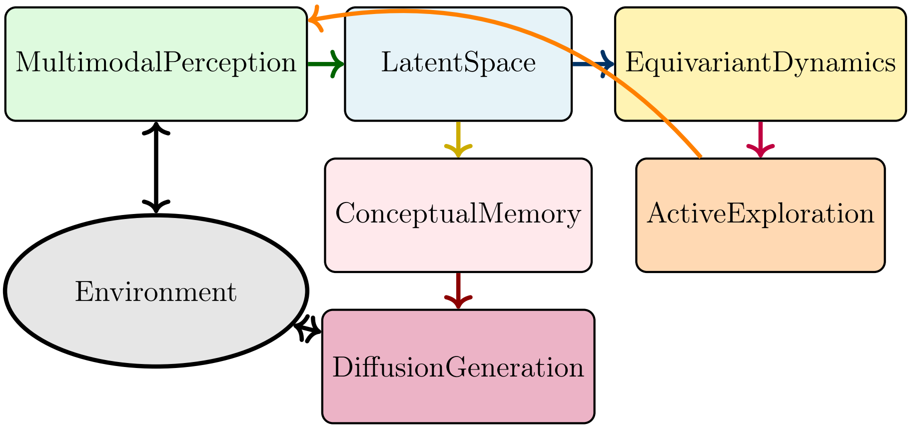

Figure 14: Overall architecture of the Universe-Models system

5.2 Main Pseudocode

```python
class UniverseModelsAgent:
    def __init__(self, config):
        # Initialize modules
        self.perception = MultiModalPerception(config.perception)
        self.latent_geometry = LatentNeuralGeometry(config.geometry)
        self.dynamics = EquivariantDynamics(config.dynamics)
        self.memory = HierarchicalMemory(config.memory)
        self.generator = DiffusionGenerator(config.generation)
        self.explorer = IntrinsicExplorer(config.exploration)

        # Multi-objective optimizer
        self.optimizer = ParetoOptimizer(self.parameters())

    def perceive(self, observations, mask=None):
        """Multimodal fusion with uncertainty management"""
        # Product of Experts fusion
        beliefs = []
        for modality_idx, obs in enumerate(observations):
            if mask is None or mask[modality_idx]:
                mu_i, sigma_i = self.perception.encoders[modality_idx](obs)
                weight = self.compute_entropy_weight(sigma_i)
                beliefs.append((mu_i, sigma_i, weight))

        # PoE Fusion
        z_mu, z_sigma = self.fuse_beliefs(beliefs)

        # Geometric regularization
        z_sigma = self.latent_geometry.regularize(z_mu, z_sigma)

        return z_mu, z_sigma

    def predict_dynamics(self, z_obs, action, t):
        """Equivariant dynamics prediction"""
        # Transform to canonical reference frame
        z_canon = self.dynamics.to_canonical(z_obs, t)

        # Simple dynamics in the canonical space
        z_canon_next = self.dynamics.forward(z_canon, action)
```

```python

        # Transform back to the observed reference frame
        z_obs_next = self.dynamics.from_canonical(z_canon_next, t+1)

        return z_obs_next

    def memorize(self, z_canon):
        """Hierarchical storage in memory"""
        codes = []
        residual = z_canon

        for level in range(self.memory.num_levels):
            code, indices = self.memory.quantize(residual, level)
            codes.append((code, indices))
            residual = residual - code

        return codes

    def generate(self, z, modality='image', num_steps=1000):
        """Conditional generation via diffusion"""
        # Initialize with pure noise
        x_t = torch.randn(self.generator.get_shape(modality))

        # Denoising process
        for t in reversed(range(num_steps)):
            # Predict noise
            noise_pred = self.generator.denoise(x_t, t, z, modality)

            # Update step
            x_t = self.generator.step(x_t, noise_pred, t)

        return x_t

    def explore(self, state):
        """Action selection based on curiosity"""
        # Compute intrinsic rewards
        novelty = self.explorer.compute_novelty(state)
        causal_ambiguity = self.explorer.compute_causal_ambiguity(state)
        learning_progress = self.explorer.compute_progress(state)

        # Weighted combination
        intrinsic_reward = (
            self.config.lambda_n * novelty +
            self.config.lambda_c * causal_ambiguity +
            self.config.lambda_p * learning_progress
        )

        # Action selection
        action = self.explorer.select_action(state, intrinsic_reward)
        return action

    def train_step(self, batch):
        """Multi-objective training step"""
        # Compute all losses
        losses = {
            'reconstruction': self.compute_reconstruction_loss(batch),
            'kl_divergence': self.compute_kl_loss(batch),
            'contrastive': self.compute_contrastive_loss(batch),
            'geometric': self.compute_geometric_loss(batch),
            'dynamics': self.compute_dynamics_loss(batch),
            'simplicity': self.compute_simplicity_loss(batch),
            'abstraction': self.compute_abstraction_loss(batch),
            'diffusion': self.compute_diffusion_loss(batch)
```

```
        }

        # Pareto-optimal optimization
        self.optimizer.step(losses)

        return losses
```

Listing 1: Main implementation of the Universe-Models framework

6 Evaluation Scenarios and Metrics

Evaluating such a complex cognitive agent requires a multi-faceted approach, combining controlled synthetic environments to validate specific capabilities with more realistic scenarios to test system integration and robustness.

6.1 Synthetic Test Environments

Simple virtual worlds allow for the isolation and quantification of each module's performance.

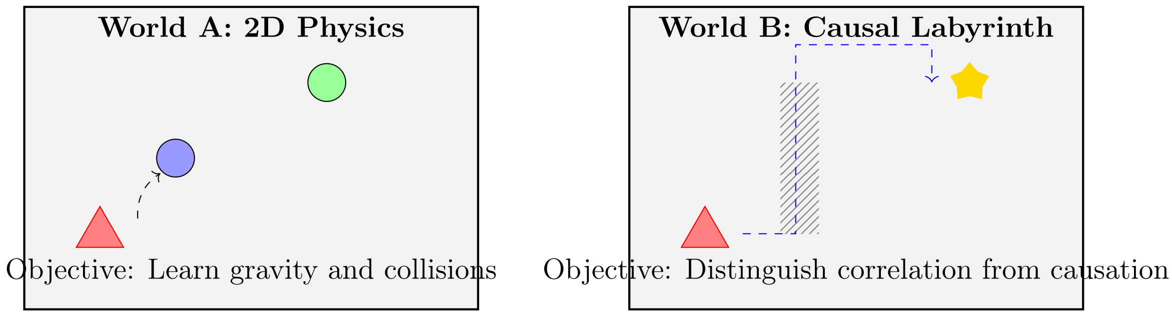

Figure 15: Examples of synthetic environments for evaluation

6.2 Real-World Scenarios

To assess performance under realistic conditions, we will use:

- **Robotic Navigation (Simulation)**: Using simulators like Habitat or Gibson to test the agent's ability to build a 3D model of its environment and navigate within it.
- **Video Understanding**: Using datasets like Kinetics or Something-Something to evaluate long-term dynamics prediction and comprehension of human actions.
- **Human-Agent Interaction**: Scenarios where the agent must interpret multimodal instructions (text, gestures) to accomplish a task.

6.3 Key Performance Indicators (KPIs)

Performance Indicators

The evaluation will be based on the following metrics, organized by module:

- **Perception**:
  - **Multimodal reconstruction error**: Quality of data generated in one modality from another (e.g., Image $\rightarrow$ Text).
  - **Contrastive alignment score**: Accuracy of cross-modal retrieval.
- **Latent Space**:
  - **Disentanglement score** (e.g., MIG, DCI): Measure of the separation of factors of variation.
  - **Uncertainty calibration** (Expected Calibration Error): Reliability of the covariance $\mathbf{\Sigma}_\phi$.
- **Causal Dynamics**:
  - **N-step prediction error**: L2 error between $\boldsymbol{z}_{\text{obs}}(t+N)_{\text{predicted}}$ and $\boldsymbol{z}_{\text{obs}}(t+N)_{\text{actual}}$.
  - **Simplicity score**: Norm of the environment's dynamics matrix, $\|f_\eta\|$, which should be minimized.
- **Generation**:
  - **Fréchet Inception Distance (FID)** for images.
  - **Mean Opinion Scores (MOS)** for audio and video.
- **Exploration**:
  - **State coverage**: Number of unique states visited in a given time.
  - **Learning speed**: Rate at which other metrics improve.

7 Potential Applications

The flexibility and depth of understanding of the Universe-Models framework pave the way for transformative applications in several domains.

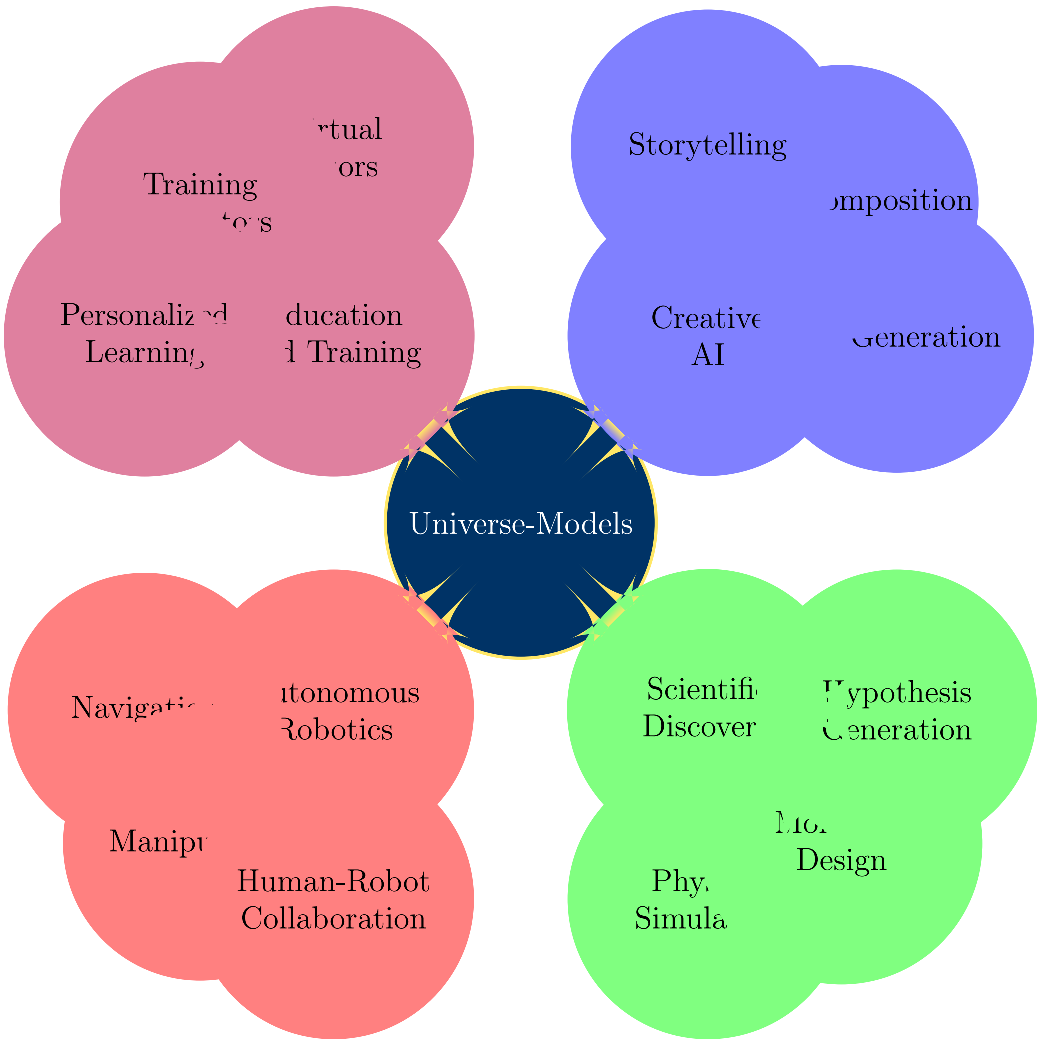

Figure 16: Potential applications of the Universe-Models framework

8 Conclusion and Perspectives

This document has presented a comprehensive technical specification for the Universe-Models framework, a cognitive architecture designed to learn a structured, causal understanding of the world from multimodal sensory data. The proposed system integrates several innovative components:

- A robust multimodal perception module that fuses information using a Product of Experts approach
- A geometrically structured latent space with adaptive uncertainty modeling
- Equivariant dynamics that separate agent-specific transformations from environment physics
- A hierarchical conceptual memory based on residual vector quantization
- A conditional diffusion process for multimodal generation
- A multi-objective optimization framework that balances competing objectives
- An intrinsically motivated exploration strategy driven by curiosity

8.1 Future Research Directions

Several promising research directions emerge from this work:

1. **Scaling Laws**: Investigating how performance scales with model size, data volume, and computational resources
2. **Transfer Learning**: Developing methods for efficient knowledge transfer between different environments and tasks
3. **Meta-Learning**: Enabling the system to rapidly adapt to new environments and tasks with minimal data
4. **Symbolic Reasoning**: Integrating symbolic reasoning capabilities with the subsymbolic representations learned by the system
5. **Social Intelligence**: Extending the framework to model and understand social interactions and theory of mind
6. **Ethical Considerations**: Developing methods for value alignment, transparency, and accountability in autonomous systems

The Universe-Models framework represents a step toward artificial general intelligence by integrating perception, reasoning, and action in a unified architecture. By learning a structured world model that captures the underlying causal structure of the environment, the system can not only interpret complex sensory data but also imagine, plan, and create in ways that were previously the domain of human intelligence.

Acknowledgments

The author would like to thank the open-source community for their contributions to machine learning and artificial intelligence research. Special thanks to the developers of PyTorch, TensorFlow, and other foundational tools that make this research possible.

References

[11pt, a4paper]article
[utf8]inputenc [T1]fontenc lmodern
[english]babel
geometry a4paper, left=2.5cm, right=2.5cm, top=2.5cm, bottom=2.5cm
[colorlinks=true, urlcolor=blue, linkcolor=black]hyperref
enumitem

Thematic Bibliography
Cosmology and Artificial Intelligence

A Synthesis of 2023-2024 Preprints

September 21, 2025

Theoretical Cosmology and Universe Models

1. **Boiza, C. G. (2024).** *Cosmological perturbations in a generalized axion-like field.* arXiv:2410.22467.
2. **Akarsu, Ö. (2024).** *CDM Tensions: Localising Missing Physics through Redshift and Scale.* arXiv:2402.04767.
3. **Shushi, T. (2024).** *The Universe as a Learning System.* arXiv:2402.14423.
4. **Saeed, J. (2024).** *Universal properties of the evolution in modified loop quantum cosmology.* arXiv:2406.06745.
5. **Muralidharan, V. (2024).** *Can Baby Universe Absorption Explain Dark Energy?* arXiv:2408.13306.
6. **Ignat'ev, Y. G. (2024).** *Cosmological models based on an asymmetric scalar doublet.* arXiv:2405.13607.
7. **Jean-Pierre, P. (2024).** *A bimetric cosmological model based on twin universes.* arXiv:2412.04644.
8. **Le Delliou, M., Del Popolo, A. (2024).** *An Anisotropic Model for the Universe.* arXiv:2410.06102.
9. **Wang, X., Zhang, Y.-L., Sasaki, M. (2024).** *Enhanced Curvature Perturbation and Primordial Black Hole Formation in Two-stage Inflation with a Break.* arXiv:2404.01846.
10. **Montandon, T., Hahn, O., Stahl, C. (2024).** *Simulating the Universe from the cosmological horizon to halo scales.* arXiv:2404.02783.

Artificial Intelligence and Astrophysics

11. **Audenaert, J., et al. (2024).** *The Multimodal Universe: Enabling Large-Scale Machine Learning with 100TB of Astronomical Scientific Data.* arXiv:2412.02527.
12. **Goh, L. W. K. (2024).** *Distinguishing Coupled Dark Energy Models using Neural Networks.* arXiv:2411.04058.
13. **Wang, Y. (2024).** *Can AI Understand Our Universe? Test of Fine-Tuning GPT with Astronomical Data.* arXiv:2404.10019.
14. **Dvorkin, C., et al. (2024).** *Neural Networks for Cosmological Model Selection and Parameter Estimation.* arXiv:2410.05209.
15. **Min, Z., et al. (2024).** *Deep Learning for Cosmological Parameter Inference from Dark Matter Halo Density Field.* arXiv:2404.09483.

16. **Murakami, K., et al. (2023).** *Non-linearity-free Prediction of the Growth-Rate using Convolutional Neural Networks*. arXiv:2305.12812.

17. **Rose, J. C., et al. (2024).** *Introducing the DREAMS Project: Dark Matter and Astrophysics with Machine Learning and Simulations*. arXiv:2405.00766.

18. **Pandey, S., et al. (2024).** *Creating Halos with Auto-Regressive Multi-stage Networks*. arXiv:2409.09124.

19. **Jagvaral, Y., et al. (2024).** *Geometric Deep Learning for Galaxy-Halo Connection*. arXiv:2409.18761.

20. **Luongo, O. (2024).** *Model Independent Cosmographic Constraints from DESI Data*. arXiv:2404.07070.

21. **Pal, S. (2023).** *ParamANN: A Neural Network to Estimate Cosmological Parameters*. arXiv:2309.15179.

22. **Chekanov, S. V. (2023).** *Estimation of the Chances to Find New Event Topologies at the LHC*. arXiv:2311.09012.

23. **Lu, Y. (2024).** *Machine Learning in Nuclear Physics: Applications and Challenges*. arXiv:2404.14948.

24. **Ragavendra, H. V., et al. (2024).** *Constraining Ultra Slow Roll Inflation using Cosmological Datasets*. arXiv:2404.00933.

25. **Gupta, K. R., Banerjee, A. (2024).** *Spatial Clustering of Gravitational Wave Sources with k-Nearest Neighbour Distributions*. arXiv:2404.01428.

26. **Khadka, N., et al. (2024).** *Breaking the Mass-Sheet Degeneracy in Strong Lensing Mass Modeling with Weak Lensing Observations*. arXiv:2404.01513.

27. **Libanore, S., Kovetz, E. D. (2024).** *Upcoming Searches for Decaying Dark Matter with ULTRASAT Ultraviolet Maps*. arXiv:2404.01556.

28. **Safdi, S., et al. (2024).** *A Semiblind Reconstruction of the History of Effective Number of Neutrinos Using CMB Data*. arXiv:2404.01124.

29. **Greig, B., et al. (2024).** *Blind QSO Reconstruction Challenge: Exploring Methods to Reconstruct the Ly Emission Line of QSOs*. arXiv:2404.01500.

30. **Rajesh Gupta, K., Banerjee, A. (2024).** *Spatial Clustering of Gravitational Wave Sources with k-Nearest Neighbour Distributions*. arXiv:2404.01428.