Learning to Plan in Latent Space with Joint-Embedding Predictive Architectures

Gabriele Gavino Pintus

Università degli Studi di Trieste
Research conducted at New York University

Supervisors: Prof. Luca Bortolussi & Prof. Alfredo Canziani

Master's Thesis Defense

Visual Navigation

Task: Navigate to a goal using only visual input

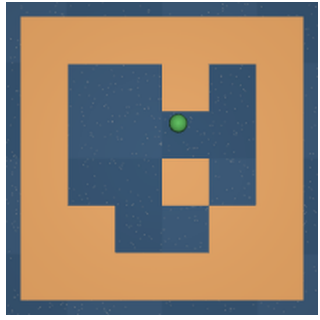
Given:

- Initial state $\boldsymbol{s}_0$
- Goal state $\boldsymbol{s}_g$
- 64×64 RGB observations

Not given:

- $\times$ x, y coordinates
- $\times$ Map of the maze
- $\times$ Reward signal

Key question: How can we plan without knowing where we are?



Learn a **world model** to predict and plan in latent space

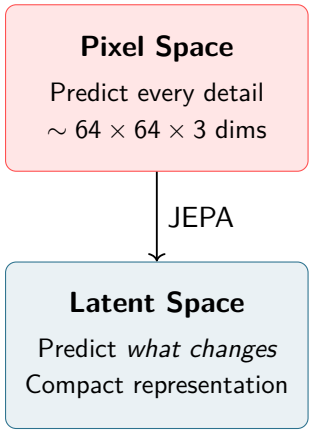
Learning World Models

World models enable agents to predict the consequences of actions before executing them.

The challenge

- High-dimensional observations (images)
- Redundant, noisy information
- Pixel prediction is expensive and wasteful

Key insight: Humans don't predict every pixel, we predict *what matters*.



We cannot predict the exact position of every leaf of the tree.
However we can foresee that it will bend toward the left.

Prediction in Representation Space

Joint-Embedding Predictive Architecture

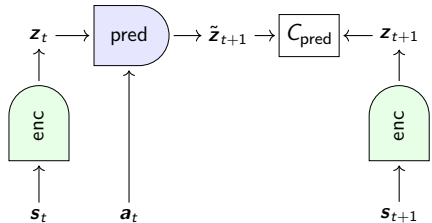
Instead of reconstructing pixels, predict in a *learned abstract space*:

- Encode observations → latent states
- Predict dynamics in latent space
- Ignore unpredictable details

Analogy: a tree in the wind

× Predict every leaf position

✓ Predict trunk bending direction



$$C_{\text{pred}}(\tilde{\boldsymbol{z}}_{t+1}, \boldsymbol{z}_{t+1}) = \|\tilde{\boldsymbol{z}}_{t+1} - \boldsymbol{z}_{t+1}\|^2$$

Static-Dynamic Decomposition via Learned Mask

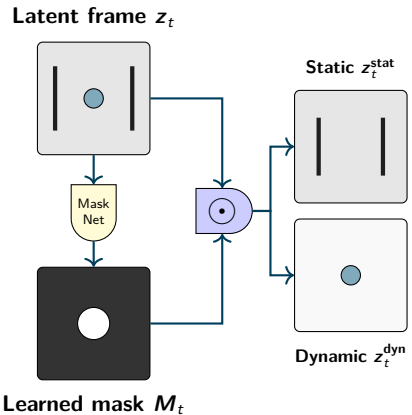
Problem In visual scenes, most pixels don't change.

Solution Learn a mask that separates

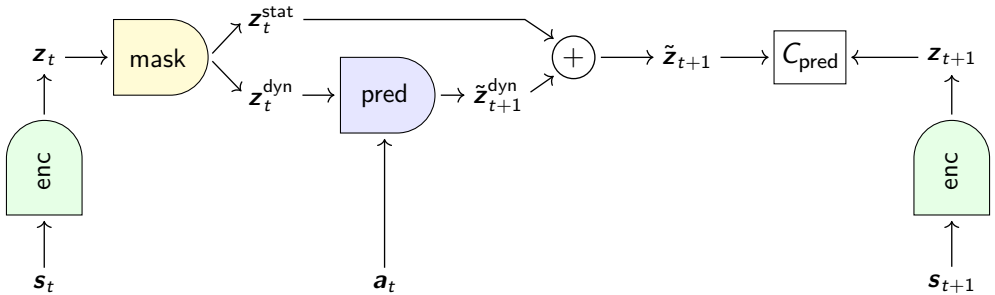
- Static walls, floor texture
- Dynamic agent, moving parts

Key equations

$$\mathbf{M}_t = \text{MaskNet}(\mathbf{z}_t)$$
$$\mathbf{z}_t^{\text{stat}} = \mathbf{z}_t \odot \mathbf{M}_t$$
$$\mathbf{z}_t^{\text{dyn}} = \mathbf{z}_t \odot (1 - \mathbf{M}_t)$$



JEPA overview



Encoder	CNN: $3 \times 64 \times 64 \mapsto 16 \times 26 \times 26$	40k
Predictor	CNN: $16 \times 26 \times 26 \mapsto 16 \times 26 \times 26$	50k
Mask extractor	CNN: $16 \times 26 \times 26 \mapsto 1 \times 26 \times 26$	5k

Dataset

Environment

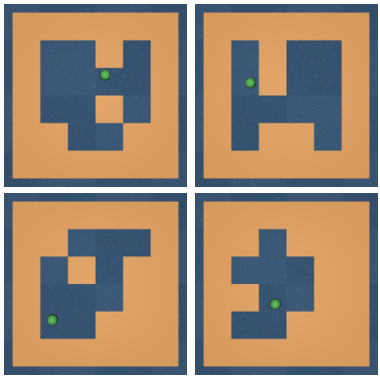
- 2D point-mass with momentum
- Continuous actions: forces in x, y

Observations

- RGB image ($64 \times 64 \times 3$)
- Velocity (v_x, v_y)
- No access to position!

Data Collection:

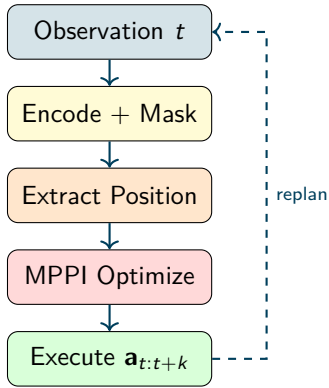
- 10 000 random trajectories ($H = 100$)
- 40 training + 40 test mazes



Planning in Latent Space: Model Predictive Control

MPPI Planning

- Sample $N = 512$ action trajectories
- Roll out in latent space ($H = 200$ steps)
- Extract position from mask
- Compute cost:
  distance to goal + action smoothness
- Aggregate samples
- Replan every k steps

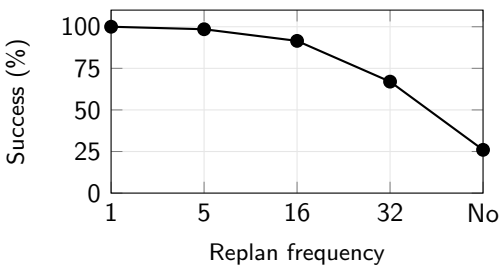


Experimental Results

Evaluation Setup

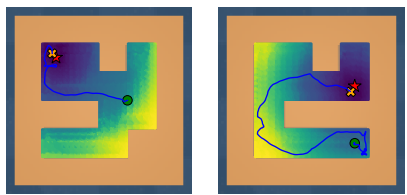
- 40 training + 40 test mazes
- 10 000 trajectories ($H = 100$)
- 200 evaluations per setting

Mi pa



The mask identifies dynamic regions

The distance field captures geometry



Conclusions

- Developed a **JEPA-based world model** for visual navigation in PointMaze
- Introduced a **learned mask mechanism** for static-dynamic decomposition
- Achieved **100% success rate** on 40 unseen mazes with frequent replanning

Limitations

- Euclidean cost ignores walls
- Non-differentiable position extraction

Future Directions

- Topology-aware costs
- Uncertainty quantification