CIS 6280 · FALL 2026

WORLD MODELS

LECTURE 7

Generative Models I · Latent Variables and Adversarial Learning

Jiatao Gu

University of Pennsylvania · September 15, 2026

Course Logistics

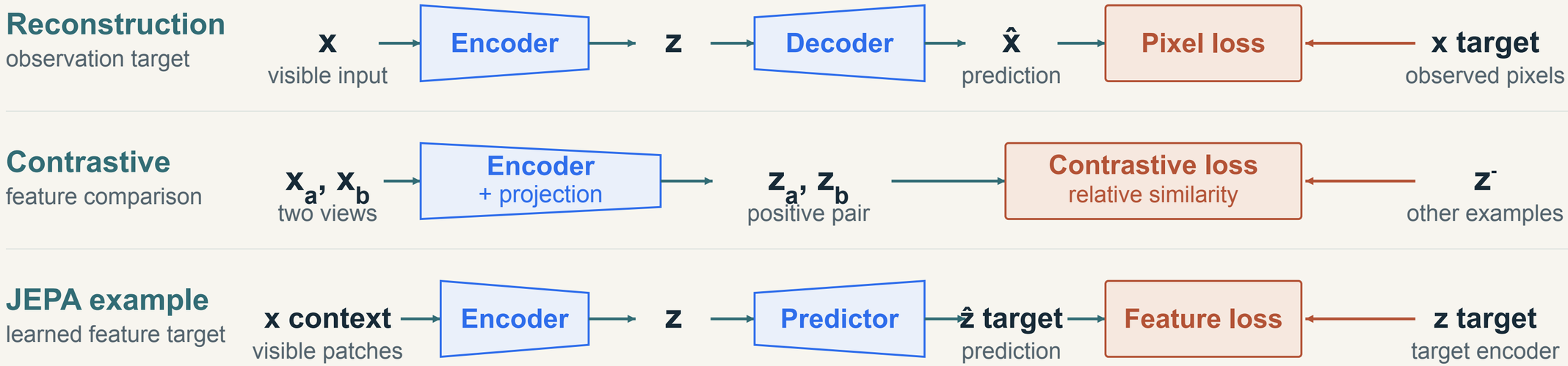
When	Announcement
Assignment 1	Released Sep 10 · due **Sep 24**. Details and submission on Canvas.
Today · Sep 15	**Tentative:** virtual via Zoom. Link and confirmation on Canvas.
Thu · Sep 17	**Tentatively canceled.** Check Canvas for confirmation.
Next lecture · Sep 22	**Tentative:** virtual via Zoom. L8: autoregressive models.

Recap + Yann video: 20 min. VAE and its SSM connection: 43 min. GAN: 5 min. World Models: 15 min. + 2 min buffer.

Bring last week's question forward: what information does the learning signal preserve?

Last Week: Learning Signals

Follow the arrows: what is encoded, predicted, and compared?



Two target spaces: observations and learned features. Contrastive learning also compares learned features.

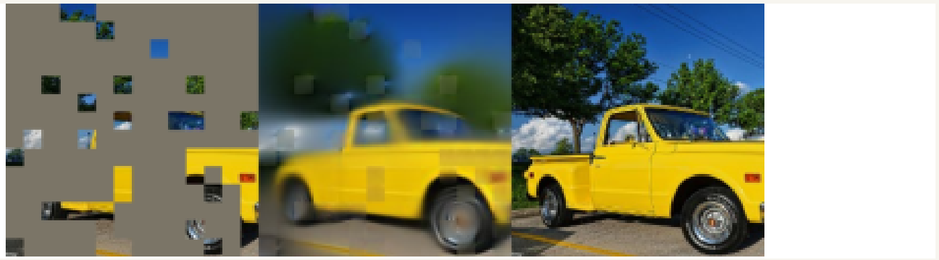
> **Does lower pretraining loss mean a better representation?**
>
> Only after we specify what information the downstream task needs.

The target and the loss shape what the representation is rewarded for preserving.

Reconstruction

Example: MAE

Target pixels come from the image; they are hidden from the encoder.



Visible patches → reconstruction → original

$$\mathcal{L}_{\text{rec}} = \frac{1}{|M|} \sum_{k \in M} \|\hat{x}_k - x_k\|^2$$

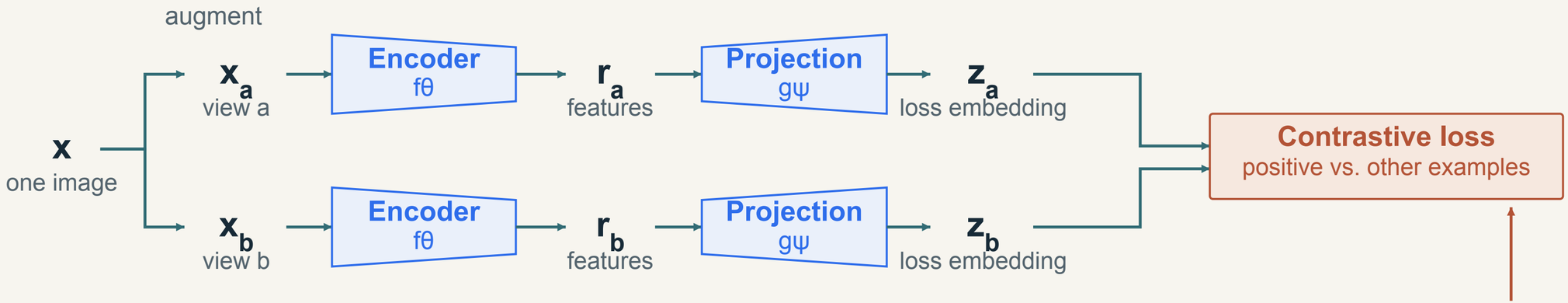
Learning pressure: recover missing visual information.

Task mismatch: a tiny traffic light may matter more than a large texture region.

> **Is color a nuisance?**
>
> For some shape tasks, yes. For traffic lights, color can determine the action.

Reconstruction can learn useful features. Pixel importance and task importance may differ.

Contrastive Learning



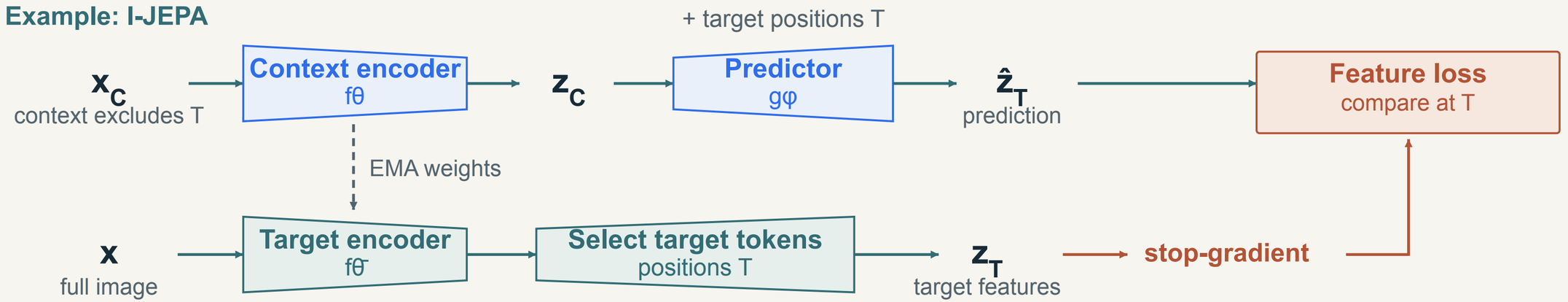
$$\ell_i = -\log \frac{\exp(\mathrm{sim}(z_i, z_i^+)/\tau)}{\sum_{j \neq i} \exp(\mathrm{sim}(z_i, z_j)/\tau)}$$

The positive is included in the denominator. The downstream readout often uses r, before projection.

> **Should two crops agree if one crop removes the object?**
>
> The positive-pair choice is an assumption about which information should survive.

Augmentations choose what to match; negatives choose what to distinguish.

Learned Targets: Two Training Recipes



Training recipe	Examples	What accompanies agreement?
Teacher-based	DINO / BYOL / I-JEPA	Slow targets + recipe-specific asymmetry
Distribution regularization	VICReg / LeJEPA	Feature variation / distribution constraints

LeJEPA uses view matching + SIGReg, without a teacher. The graph above is the I-JEPA example.

> **Does a learned target automatically become semantic?**
>
> No. Its useful information still needs a downstream test.

Separate the prediction task, the source of the target, and the mechanism controlling collapse.

Which Representation Is Better?

Better for a specified task, with a specified readout and budget.

Required information	Useful evaluation	What that result does not establish
Object category	Frozen encoder + matched classification probe	Motion or action-conditioned dynamics
Position and orientation	Matched spatial readout on held-out scenes	Unobserved velocity or hidden forces
Future motion	Predict held-out trajectories from history	Reliable long-horizon control
Action consequences	Rollout tests and closed-loop task success	Reliability outside the tested setting

> **Can method A win classification while method B is better for control?**
>
> Yes. They can retain different information, and the readout can access it differently.

Compare the same task, data, encoder budget and readout. Then inspect the failures.

Prediction and Uncertainty

With a fixed target representation, squared error favors the conditional mean.

$$g^*(c) = \mathbb{E}[Y \mid c]$$

A constructed example

Two equally likely target values:

$$Y = -1 \quad \text{or} \quad Y = +1$$

$$g^*(c) = 0$$

Zero minimizes MSE, even when zero is not a possible target.

Two target spaces

Pixels: averaging alternatives can blur a prediction.

Features: averaging still occurs if the features distinguish those alternatives.

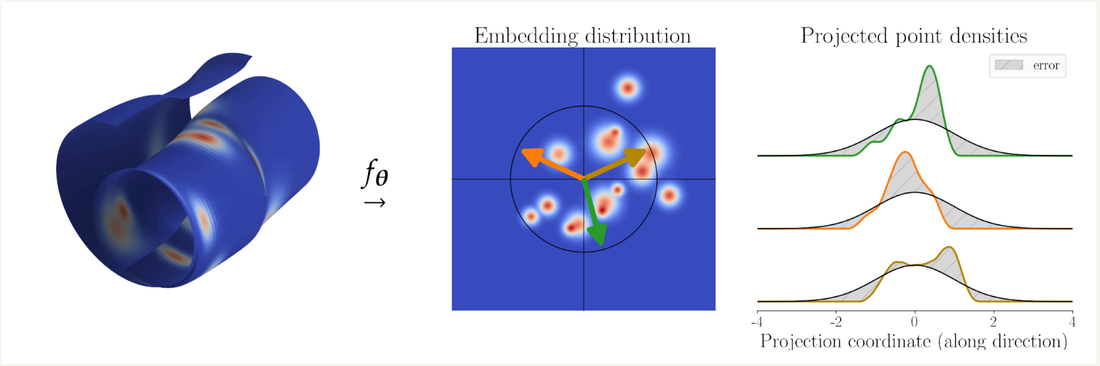
A feature map can also remove an unpredictable detail. We must decide whether that detail matters.

Must every predictor sample all possible outcomes?

Only if the task needs those alternatives. A point prediction can be useful when relevant uncertainty is small.

Target space and uncertainty modeling are separate choices. This motivates today's generative models.

SIGReg: Across Images, Along Directions



LeJEPA Figure 2: projections and their distributions

$$Z \in \mathbb{R}^{B \times d}, \quad A \in \mathbb{R}^{d \times K}$$

$$U = ZA \in \mathbb{R}^{B \times K}$$

One column of U:
B values from B different images.

A contains K sampled unit directions. Compare each column's distribution with a standard Gaussian.

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\text{match}} + \lambda R_{\text{SIGReg}}$$

Do we assign a random Gaussian target vector to each image?

No. Matching relates views of one image. SIGReg constrains a distribution across images.

Views should agree within an image. Features should remain distributed across images.

SIGReg: Distribution and Information

For one unit direction, let $u_b = a^\top z_b$. At frequency ω:

$$C = \frac{1}{B}\sum_b \cos(\omega u_b), \quad S = \frac{1}{B}\sum_b \sin(\omega u_b)$$

$$\left[C - e^{-\omega^2/2}\right]^2 + S^2$$

One frequency's discrepancy from the Gaussian reference.

What the constraint addresses	What still needs evidence
Identical features incur a distribution penalty	Optimization always avoids collapse
Sampled projections approach the reference	An exact high-dimensional distribution match
A nonconstant, spread-out embedding	Task-relevant position, motion or action effects

> **Suppose z records only an irrelevant variable, already distributed as N(0, I).**
>
> The Gaussian test cannot decide that the variable is irrelevant. The task and evaluation must do that.

A distribution constraint discourages degeneracy. It does not identify task-relevant information.

Representations for World Models

A representation defines what the predictor can use and what it predicts.

Component	Interface	Question to test
Observation encoding	$z_t = f(o_t)$	What is recoverable from this observation?
State inference	$p(s_t \mid o_{\leq t}, a_{<t})$	What does the observation history tell us about the hidden state?
Predictive dynamics	$p(s_{t+1} \mid s_t, a_t)$	How do candidate actions change the future?

Here s_t is the hidden state from L3. The posterior describes uncertainty about it; an observation feature z_t is not automatically the state.

> **A perfect reconstruction of one frame: enough to predict the next frame?**
>
> Not if two hidden velocities produce the same image. History or a belief over hidden state may be needed.

Representation learning can support a world model. Predictive dynamics must also be specified and tested.

World Models and JEPA

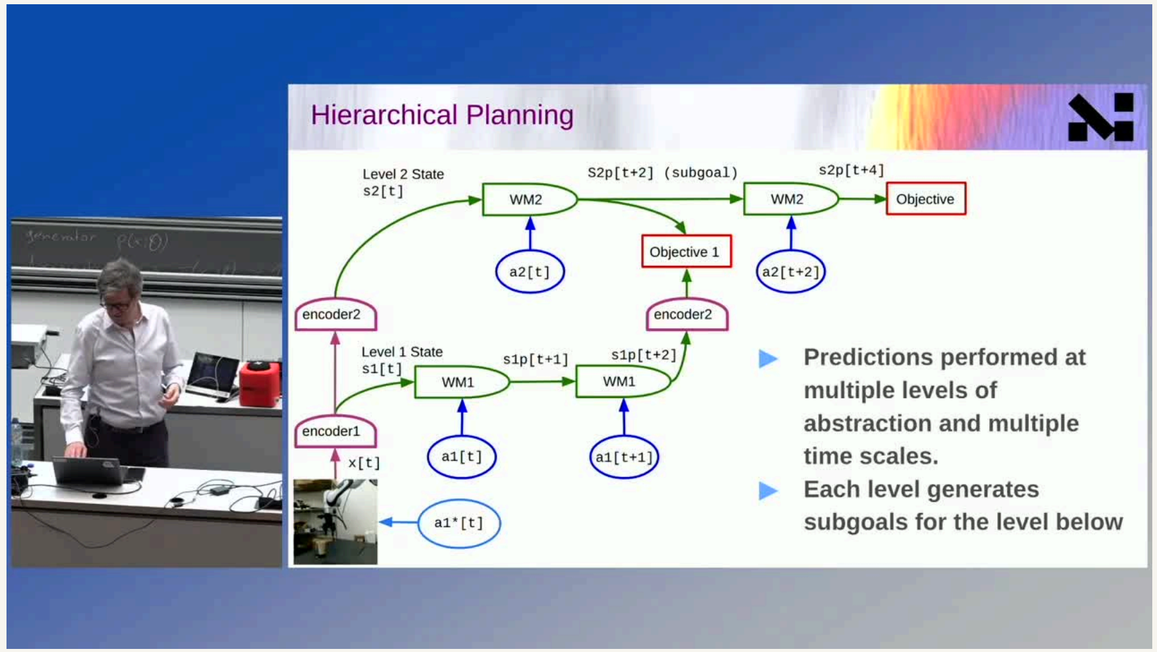
Example	What it supplies	World-model question
I-JEPA	Masked image feature prediction	What adds temporal evolution?
V-JEPA 2 pretraining	Predictive video feature learning	What is visible at forecasting time?
V-JEPA 2-AC	Action-conditioned latent dynamics	Do imagined actions support control?
Ha & Schmidhuber (2018)	VAE + MDN-RNN dynamics	Can the generative model support behavior?

JEPA is an approach to learning and predicting representations.
World modeling asks what a system can predict about an environment.

> **Does “predicting a target” always mean “predicting a future”?**
>
> No. Masked completion can use context unavailable at the forecast time. Check the information boundary.

Today: how latent-variable and adversarial models learn distributions over possible observations.

Yann LeCun on Generative Models



Yann LeCun · 21:06–25:08 · 4 min 02 sec

While watching: which details should a predictive representation keep or ignore?

Keep this question in mind: what does a reconstruction objective ask z to retain?

From Features to Hidden Causes

L5 / L6 connection · We learned features by choosing what an encoder should predict.

Starting point	Direction	Question
L5 / L6: representations	$x \xrightarrow{f} z$	What information survives the encoder?
L3 / L4: state-space models	$s_t \longrightarrow o_t$	How does a hidden state generate an observation?

Can we learn a model of observations and use inference in that model to obtain a representation?

> **Where should we begin?**
>
> With the familiar state → observation relation, using just one observation first.

First specify how observations are generated. Then ask how to infer their hidden causes.

Learning the Observation Model

L3 / L4 recall · For a fixed action sequence:

$$p(s_{0:T}, o_{0:T} \mid a_{0:T-1}) = p(s_0) \prod_{t=0}^{T} p(o_t \mid s_t) \prod_{t=0}^{T-1} p(s_{t+1} \mid s_t, a_t)$$

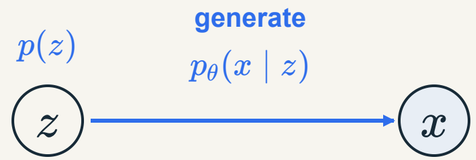
Keep only the initial observation: T = 0. Rename s_0 as z, and o_0 as x.

$$p(s_0, o_0) = p(s_0)p(o_0 \mid s_0) \quad \longrightarrow \quad p_\theta(z, x) = p(z)p_\theta(x \mid z)$$

Familiar object	Learned image model
Initial-state prior	Choose a latent prior $p(z)$.
Observation model	Use a neural network to parameterize $p_\theta(x \mid z)$.

z is an observation representation. This factorization alone does not make it a predictive state.

Latent-Variable Generative Models



z: latent · x: observation

$$p_\theta(x, z) = p(z)\, p_\theta(x \mid z)$$

1. Draw a latent

$$z \sim p(z) = \mathcal{N}(0, I)$$

2. Draw an observation

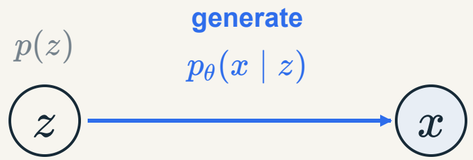
$$x \sim p_\theta(x \mid z)$$

$$p_\theta(x) = \int p(z)\, p_\theta(x \mid z)\, dz$$

The decoder turns a simple latent distribution into a flexible distribution over observations.

Latent variables let many different explanations generate different observations.

Decoder Likelihoods



z: latent · x: observation

L5 / L6 connection · A squared reconstruction loss corresponds to a particular likelihood.

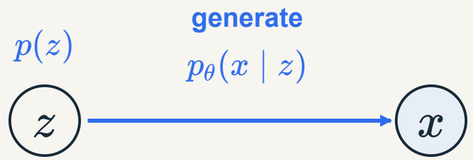
$$p_\theta(x \mid z) = \mathcal{N}\left(x; \mu_\theta(z), \sigma_x^2 I\right)$$

$$-\log p_\theta(x \mid z) = \frac{\|x - \mu_\theta(z)\|^2}{2\sigma_x^2} + C$$

Observation model	Reconstruction term
Fixed-variance Gaussian	Scaled squared error; C is constant in θ and z.
Bernoulli, for binary observations	Binary cross-entropy.
A richer conditional distribution	A different likelihood and a different modeling assumption.

Reconstruction is a modeling choice about errors, noise, and unresolved variation.

Maximum Likelihood



z: latent · x: observation

$$\max_\theta \ \mathbb{E}_{x \sim p_{\text{data}}} \log p_\theta(x)$$

$$\log p_\theta(x) = \log \int p(z) p_\theta(x \mid z)\, dz$$

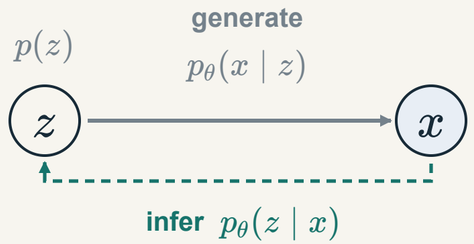
Sampling is easy: draw z, then decode.

Scoring an observed x can be hard: integrate over all latent explanations.

> **Could we just average likelihoods from prior samples?**
>
> In principle yes. In high dimensions, most prior samples may explain the observed x poorly.

An easy sampler does not automatically give an easy likelihood.

Posterior Inference



$$p_\theta(z \mid x) = \frac{p(z)p_\theta(x \mid z)}{p_\theta(x)}$$

The posterior asks which latents could explain this particular observation.

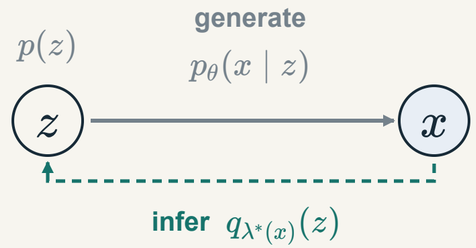
L3 / L4 connection · Gaussian conditioning solved this inference problem exactly.

Linear Gaussian observation model	Neural observation model
Gaussian conditioning had a closed form.	The posterior is generally not available in closed form.
A distribution represented uncertainty about state.	We still want a distribution over plausible latents.

Next question: can we find a tractable distribution that approximates this posterior?

The inference question is familiar; the neural observation model makes the computation harder.

Variational Inference



Turn posterior inference into an optimization problem over distributions.

1. Choose a tractable family. For example, Gaussians with adjustable mean and variance.

$$q_\lambda(z) = \mathcal{N}\left(z; \mu, \mathrm{diag}(e^\ell)\right), \qquad \lambda = (\mu, \ell)$$

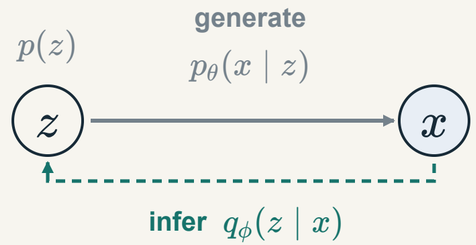
2. Fit its parameters to the posterior. Hold the model θ and observation x fixed.

$$\lambda^*(x) = \arg\min_\lambda D_{\mathrm{KL}}(q_\lambda(z) \parallel p_\theta(z \mid x))$$

We can optimize a separate λ for every x. A neural network is not required.

Variational inference = choose a distribution family, then optimize the approximation.

Amortized Inference: An Encoder



Optimize separately for each input	Share an inference network
$x \longrightarrow \lambda^*(x)$	$x \xrightarrow{\text{encoder}_\phi} \lambda_\phi(x)$
Run an optimization loop for every x.	Predict the parameters in one forward pass.

$$\lambda_\phi(x) = (\mu_\phi(x), \ell_\phi(x)), \qquad q_\phi(z \mid x) := q_{\lambda_\phi(x)}(z)$$

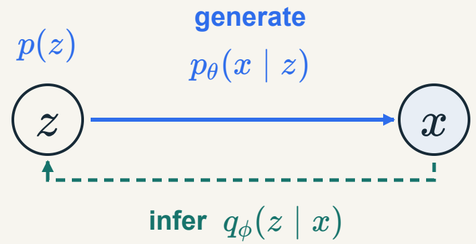
$$\min_\phi \ \mathbb{E}_x D_{\mathrm{KL}}(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x))$$

Training this shared encoder with a variational objective is **amortized variational inference**.

Next: rewrite this objective using the joint density we can evaluate.

The encoder shares the work of approximate inference across observations.

ELBO: An Exact Identity



$$
\begin{aligned}
D_{\mathrm{KL}}(q_\phi(z \mid x) \| p_\theta(z \mid x)) &= \mathbb{E}_q[\log q_\phi(z \mid x) - \log p_\theta(z \mid x)] \\
&= \mathbb{E}_q[\log q_\phi(z \mid x) - \log p_\theta(x, z)] + \log p_\theta(x)
\end{aligned}
$$

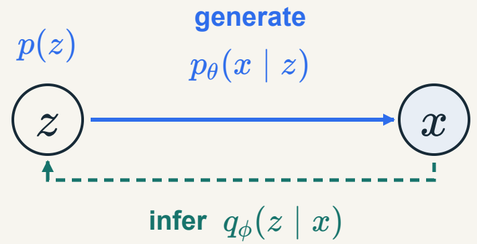
$$
\mathcal{L}_{\mathrm{ELBO}}(x) := \mathbb{E}_q[\log p_\theta(x, z) - \log q_\phi(z \mid x)]
$$

$$
\log p_\theta(x) = \mathcal{L}_{\mathrm{ELBO}}(x) + D_{\mathrm{KL}}(q_\phi(z \mid x) \| p_\theta(z \mid x))
$$

The final KL is nonnegative. Therefore ELBO ≤ log likelihood.

The posterior approximation determines how tightly the ELBO bounds the likelihood.

ELBO: Reconstruction and Prior



$$\mathcal{L}_{\mathrm{ELBO}}(x) = \mathbb{E}_q[\log p(z) + \log p_\theta(x \mid z) - \log q_\phi(z \mid x)]$$

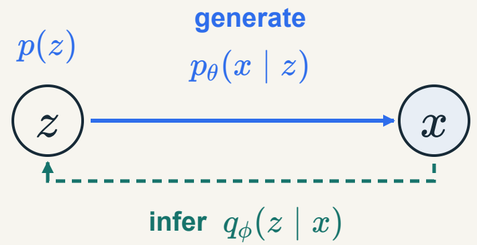
$$\mathcal{L}_{\mathrm{ELBO}}(x) = \underbrace{\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x \mid z)}_{\text{reconstruction likelihood}} - \underbrace{D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))}_{\text{prior KL}}$$

L5 / L6 connection · The reconstruction term now sits inside a latent-variable likelihood objective.

Maximize the ELBO over both the inference network ϕ and the decoder θ.

Fit observations while making inferred latents compatible with the generative prior.

ELBO: Tightness



Quantity	What it compares
Training penalty	$D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))$ · posterior approximation vs prior
Gap to log likelihood	$D_{\mathrm{KL}}(q_\phi(z \mid x) \| p_\theta(z \mid x))$ · approximation vs true posterior

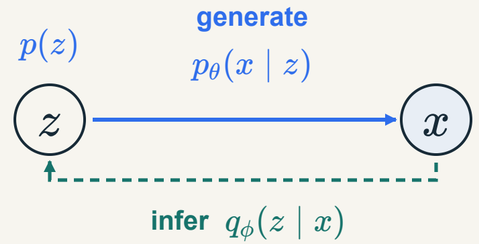
$$q_\phi(z \mid x) = p_\theta(z \mid x) \quad \Longrightarrow \quad \mathcal{L}_{\mathrm{ELBO}}(x) = \log p_\theta(x)$$

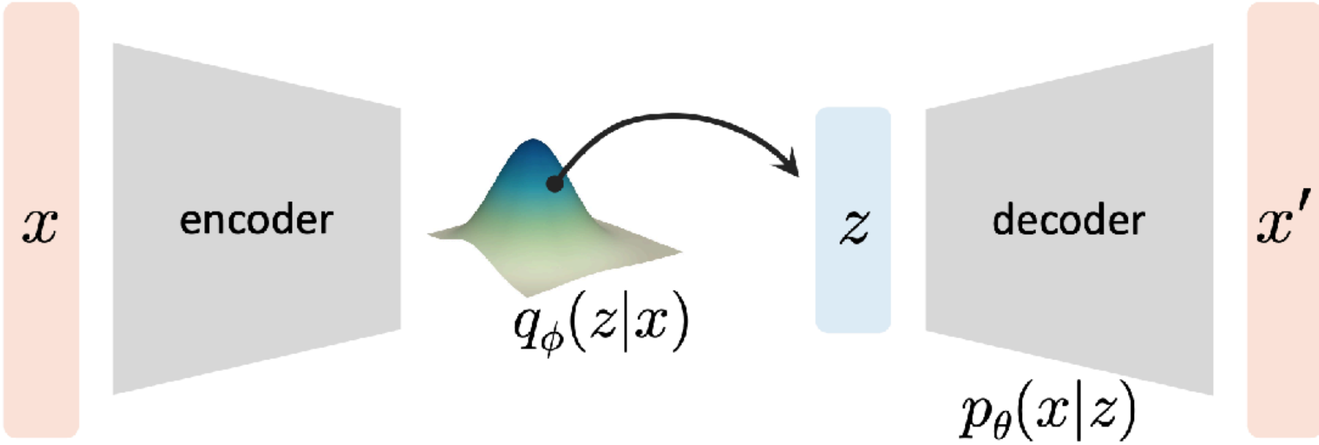
Does a tight bound require the training prior KL to be zero?

No. The true posterior can differ substantially from the prior.

A tight ELBO means accurate inference for the current generative model.

The Variational Autoencoder





Infer a latent distribution; sample a code; predict an observation distribution.

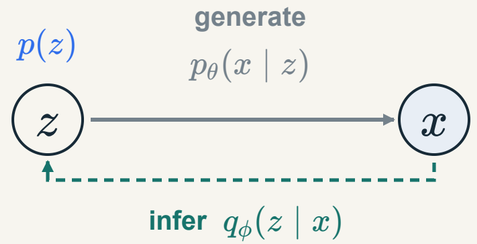
Encoder ϕ
Infer a distribution over latents from x.

Decoder θ
Assign likelihood to x given a sampled z.

$$\min_{\theta,\phi} \; -\mathcal{L}_{\text{ELBO}}(x)$$

An autoencoder architecture trained as approximate maximum likelihood.

Gaussian Posteriors



$$q_\phi(z \mid x) = \mathcal{N}(\mu, \mathrm{diag}(e^{\ell})), \qquad p(z) = \mathcal{N}(0, I)$$

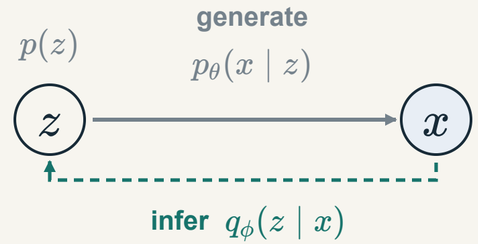
The encoder outputs μ and log variance ℓ. All vectors have d latent coordinates.

$$D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z)) = \frac{1}{2} \sum_{j=1}^{d} \left(\mu_j^2 + e^{\ell_j} - 1 - \ell_j \right)$$

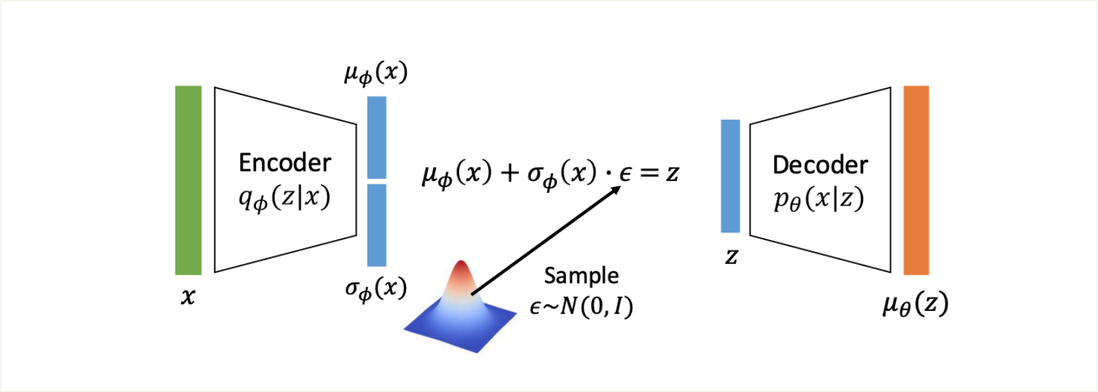
> **What does the KL prefer when considered alone?**
>
> μ = 0 and ℓ = 0, so every conditional posterior becomes N(0, I).

The reconstruction term must reward information strongly enough to retain it.

Reparameterization



Why? The reconstruction loss must update the encoder through a sampled z.



Separate the random noise from the learned parameters.

Problem

A detached draw from qϕ gives no gradient path back to μϕ and ℓϕ.

Solution

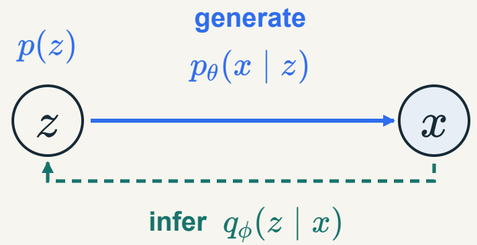
Draw parameter-free noise ε. Compute z as a differentiable function of the encoder outputs.

$$\epsilon \sim \mathcal{N}(0, I), \qquad z = \mu_\phi(x) + \exp(\ell_\phi(x)/2) \odot \epsilon$$

$$\text{Backprop:} \quad \text{reconstruction loss} \longrightarrow z \longrightarrow (\mu_\phi, \ell_\phi) \longrightarrow \phi$$

Same sampling distribution; now reconstruction can train the encoder by backpropagation.

A VAE Training Step

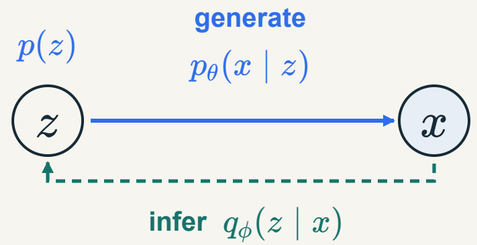



```
# x: [B, ...]; encoder outputs: [B, d]
mu, logvar = encoder(x)
eps = randn_like(mu)
z = mu + exp(0.5 * logvar) * eps

nll = -decoder_distribution(z).log_prob(x).sum_over_event_dims()
kl  = 0.5 * (mu**2 + exp(logvar) - 1 - logvar).sum(-1)
loss = (nll + kl).mean()

optimizer.zero_grad()
loss.backward()
optimizer.step()
```

One latent sample per example gives a Monte Carlo estimate of the reconstruction expectation.

Sum likelihood terms over the observation; average the combined objective over examples.

Reconstruction and Generation



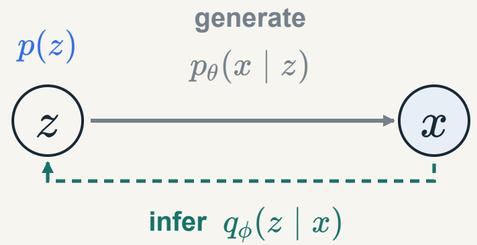
Operation	Latent source	What it tests
Reconstruct an observed x	$z \sim q_\phi(z \mid x)$	Can a code inferred from x explain x?
Generate a new x	$z \sim p(z)$	Do prior samples decode into plausible observations?

$$\text{Generation:} \qquad z \sim p(z), \quad x \sim p_\theta(x \mid z)$$

> **Does excellent reconstruction guarantee excellent prior samples?**
>
> No. Inferred latents may occupy regions that the prior visits differently.

Evaluate prior generation separately from reconstruction.

SIGReg and VAE Regularization



L5 / L6 connection · A shared idea: solve a prediction task while organizing the latent distribution.

$$\text{task loss} + \text{latent distribution regularization}$$

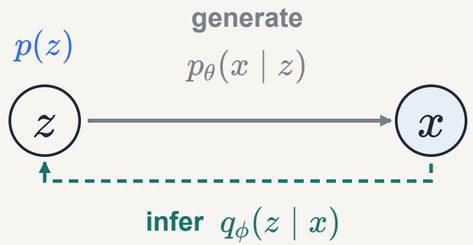
	SIGReg in representation learning	Prior KL in a VAE
Task signal	Predict / agree across related views	Reconstruct the observed input
Latents	One feature vector $z = f_\phi(x)$ per view	A distribution $q_\phi(z \mid x)$ per input
Gaussian matching	Across inputs: the distribution of feature vectors	For each input: its conditional distribution vs $p(z) = \mathcal{N}(0, I)$

The connection is close: VAE regularization includes aggregate distribution matching.

Its per-input KL also penalizes information carried about x. We can separate these two effects.

Both organize latent distributions; the VAE prior KL adds a cost for input information.

What the VAE KL Adds



$$q_\phi(z) := \mathbb{E}_{x \sim p_{\text{data}}} q_\phi(z \mid x)$$

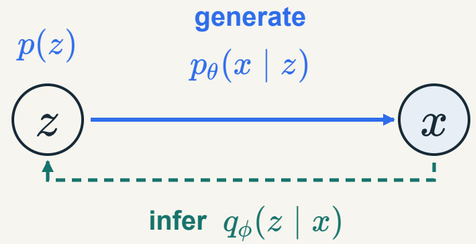
$$\mathbb{E}_x D_{\text{KL}}(q_\phi(z \mid x) \| p(z)) = \underbrace{I_q(X; Z)}_{\text{input information}} + \underbrace{D_{\text{KL}}(q_\phi(z) \| p(z))}_{\text{aggregate matching}}$$

Same Gaussian aggregate, different information: a one-dimensional example.

	No input information	Input-dependent codes
$q_\phi(z \mid x)$	$\mathcal{N}(0, 1)$ for every x	$\mathcal{N}(\mu(x), \frac{1}{4})$; $\mu(X) \sim \mathcal{N}(0, \frac{3}{4})$
Aggregate $q_\phi(z)$	$\mathcal{N}(0, 1)$	$\mathcal{N}(0, 1)$
Average VAE KL	0	$\log 2$ nats of input information

Gaussian aggregate matching can coexist with informative codes; the VAE KL also charges for that information.

Posterior Collapse



$$q_\phi(z \mid x) \approx p(z), \qquad I_q(X; Z) \approx 0$$

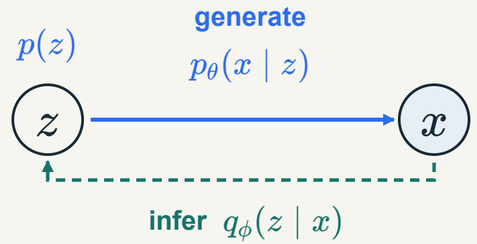
The encoder carries little input-specific information. The decoder may learn to ignore z.

$$p_\theta(x \mid z) = p_\theta(x) \quad \Longrightarrow \quad p_\theta(z \mid x) = p(z)$$

> **Can z still have nonzero variance during posterior collapse?**
>
> Yes. z can remain Gaussian noise while carrying no information about x.

Posterior collapse is loss of input information, not necessarily a constant latent vector.

Rate and Distortion



$$D = \mathbb{E}_{x,q_\phi(z|x)}[-\log p_\theta(x \mid z)], \qquad R = \mathbb{E}_x D_{\mathrm{KL}}(q_\phi(z \mid x) \| p(z))$$

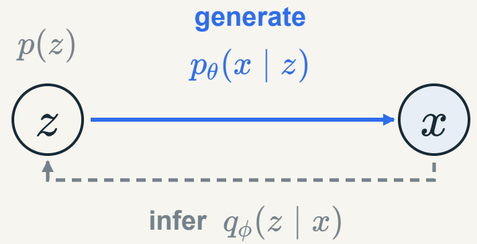
$$\min_{\theta,\phi} \; D + \beta R$$

Choice	Pressure on the solution
Higher β	A larger penalty for information and prior mismatch.
Lower β	More emphasis on reconstructing observations.
β = 1	The standard negative ELBO for the specified likelihood.

L5 / L6 connection · Information useful for reconstruction may differ from information useful for a task.

Choosing a bottleneck does not identify the right semantics automatically.

Decoder Means and Samples



L5 / L6 connection · A squared-error point predictor returns a conditional mean.

$$\mu^*(z) = \mathbb{E}[X \mid Z = z] \qquad \text{for a fixed-variance Gaussian decoder}$$

Unresolved alternatives

If the same z leaves several appearances possible, the decoder mean averages them.

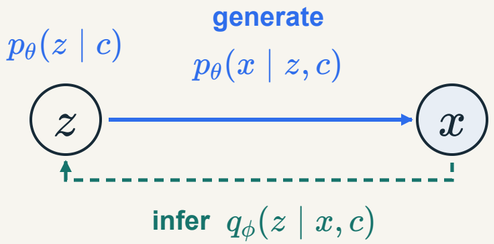
Latent and decoder choices

Latents can distinguish alternatives. A richer decoder can represent residual uncertainty.

> **Is blurry output an unavoidable property of every VAE?**
>
> No. It depends on the information in z, the likelihood family, architecture, and optimization.

Distinguish a conditional mean, a conditional sample, and a sample from the full model.

Conditional Variational Autoencoders



Let c be information available when making a prediction. Let x be the target.

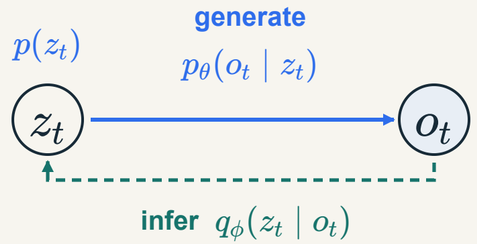
$$p_\theta(x, z \mid c) = p_\theta(z \mid c)\, p_\theta(x \mid z, c)$$

$$\mathcal{L}(x, c) = \mathbb{E}_{q_\phi(z|x,c)} \log p_\theta(x \mid z, c) - D_{\mathrm{KL}}(q_\phi(z \mid x, c) \| p_\theta(z \mid c))$$

During training	At prediction time
Infer z using both c and the observed target x.	Sample z using c; the future target x is unavailable.
Example: c = history + candidate action.	Generate possible next observations conditioned on c.

Training may infer from the target; prediction must respect what is actually known.

Latents for Prediction



L5 / L6 connection · A good representation must preserve information relevant to the intended prediction.

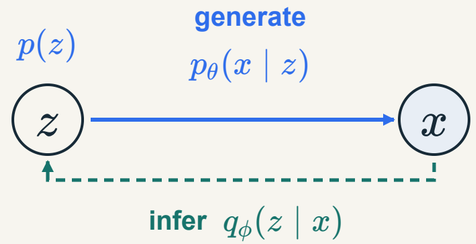
$$z_t \sim q_\phi(z_t \mid o_t), \qquad h_t = F_\psi(z_{<t}, a_{<t})$$

Driving scene	Reconstruct this image	Predict / control
Roadside texture	May account for many pixels.	May have little task relevance.
Traffic-light color	A small region of the image.	Can determine whether to stop.
Vehicle velocity	Not determined by a single position-only frame.	History can reveal motion.

z_t encodes the current observation; h_t summarizes earlier observations and actions. Neither is automatically a sufficient state.

Static compression becomes a world-model component when connected to temporal prediction.

VAE: The Four Objects



Object	Role
$p(z)$	Generate a latent before seeing an observation.
$p_\theta(x \mid z)$	Specify an observation likelihood.
$p_\theta(z \mid x)$	The true posterior induced by the generative model.
$q_\phi(z \mid x)$	A tractable learned approximation to that posterior.

> **Which network is needed to generate a new, unconditional observation?**
>
> The decoder and a prior sampler. The encoder is used for inference from observed data.

Next: learn a sampler through discrimination instead of an explicit likelihood.

Generative Adversarial Learning

$$z \sim p(z), \qquad x = G_\theta(z)$$

Learn a sample distribution through a real-versus-generated classification task.

Discriminator D	Generator G
Learn to distinguish data samples from generated samples.	Change generated samples so D is less able to distinguish them.

$$\min_G \max_D \; \mathbb{E}_{x \sim p_{\text{data}}} \log D(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D(G(z)))$$

Alternate D and G updates. The basic formulation supplies a sampler, but no posterior encoder q(z | x).

Adversarial discrimination is another way to learn a distribution from data.

Contrastive Learning and GANs

L5 / L6 connection · Both use discrimination. What is the label, and what is the final product?

	Contrastive representation learning	GAN training
Discrimination task	Identify matching views among alternatives	Identify real versus generated samples
Trainable object of interest	An encoder that exposes useful features	A generator that changes the sample distribution
Why alternatives matter	They shape invariance and separation	They provide a moving distribution to match
Typical evaluation	Frozen features + specified downstream task	Sample distribution, fidelity, and coverage

> **Is the GAN discriminator automatically a good general-purpose encoder?**
>
> It may learn useful features, but the adversarial objective does not establish that by itself.

A discrimination loss is a mechanism; its labels and training loop determine its purpose.

Mode Collapse

A generator can produce convincing samples from too few parts of the data distribution.

Constructed example	Red outcomes	Blue outcomes
Data distribution	50%	50%
Collapsed generator	100%	0%

Every generated red outcome might look valid. Half the possible outcomes are still missing.

> **Would checking only a few attractive samples reveal the problem?**
>
> Not reliably. We also need coverage and conditional diversity checks.

Sample quality and distribution coverage are different evaluation axes.

Learned Environment Models

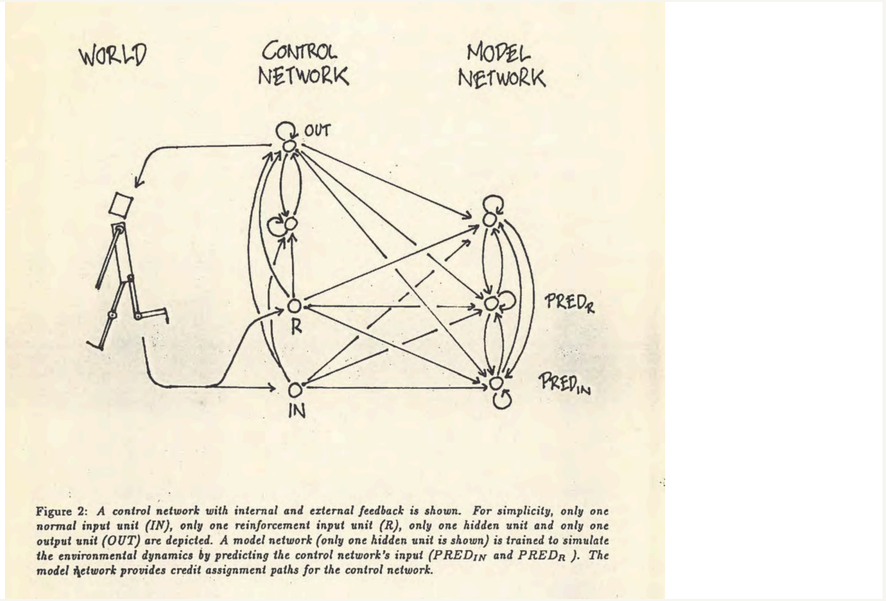


Figure 2: *A control network with internal and external feedback is shown. For simplicity, only one normal input unit (IN), only one reinforcement input unit (R), only one hidden unit and only one output unit (OUT) are depicted. A model network (only one hidden unit is shown) is trained to simulate the environmental dynamics by predicting the control network's input ($PRED_{IN}$ and $PRED_R$). The model network provides credit assignment paths for the control network.*

Schmidhuber (1990), Figure 2 · callback to L2

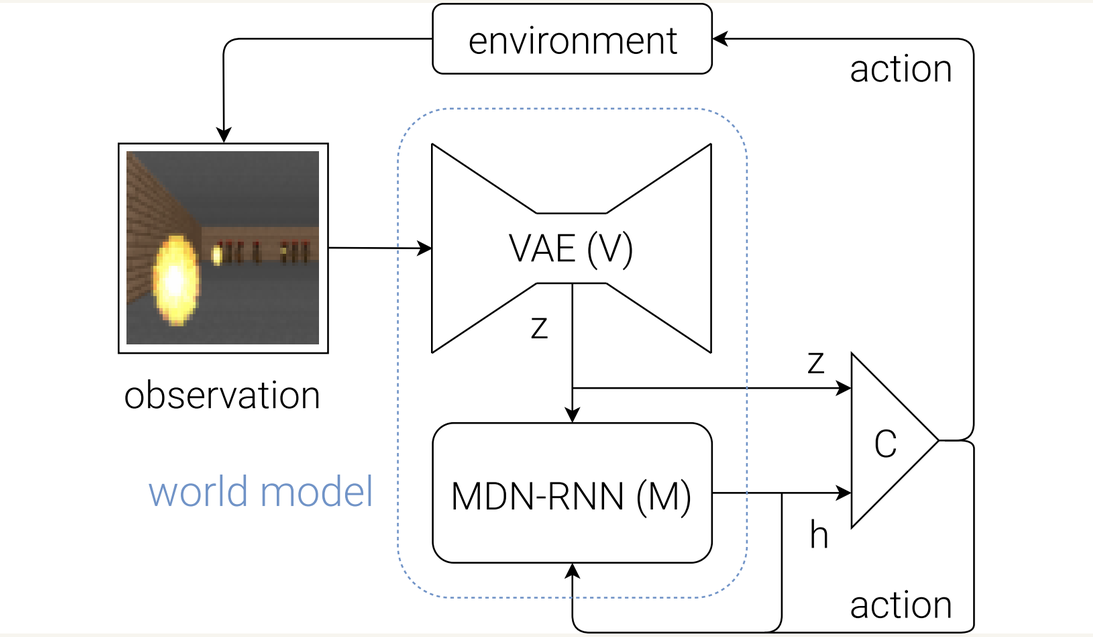
Model network
Predict future sensory and reinforcement inputs.

Control network
Use the learned model to improve behavior.

The idea already connects memory, prediction, and action.

World modeling is a functional goal with a history beyond any one representation objective.

World Models (2018)



Original system schematic · Ha & Schmidhuber (2018)

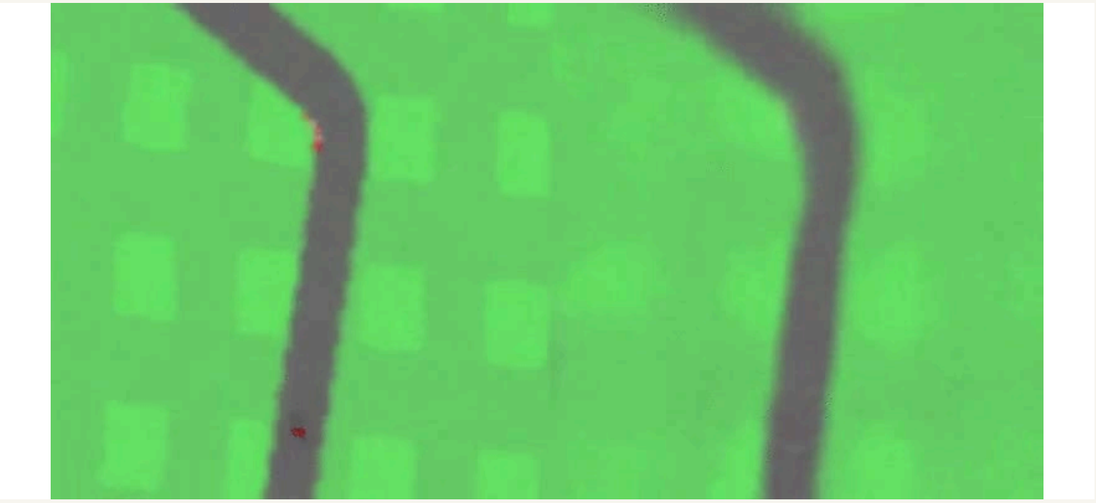
V · Vision
VAE compresses observations.

M · Memory
MDN-RNN predicts distributions over future latents.

C · Controller
A small policy acts from the learned features.

The VAE is one component of the system—not the whole world model.

Vision: A VAE for Observations



CarRacing · observation and VAE reconstruction

$$z_t \sim q_\phi(z_t \mid o_t)$$

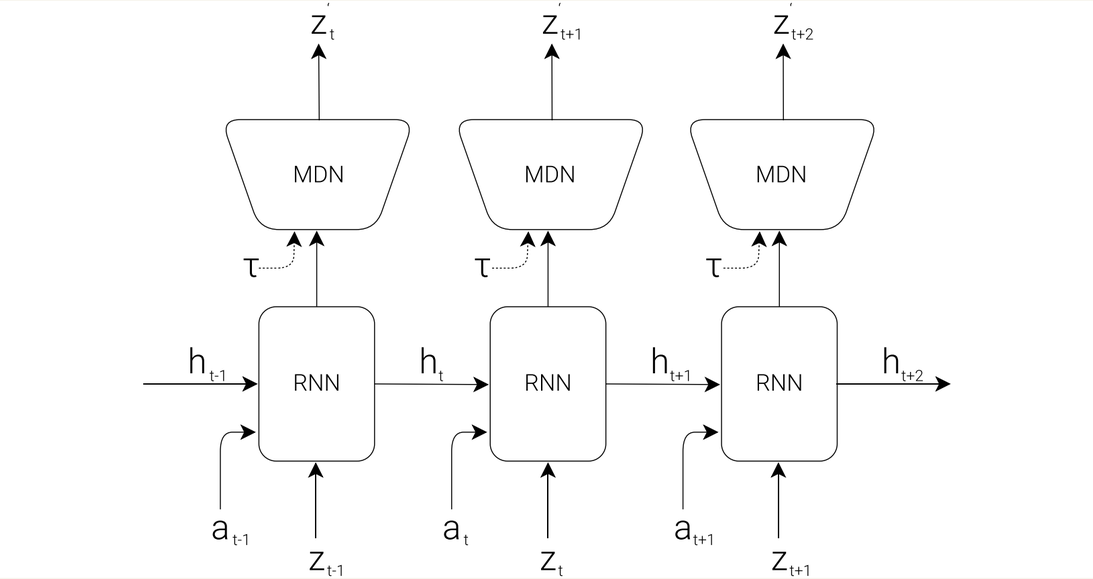
$$p_\theta(o_t \mid z_t)$$

Learn from collected observations using a reconstruction term and a prior KL.

This objective does not directly ask which details affect future reward.

Reconstruction creates a compact interface; predictive usefulness remains an empirical question.

Memory: An MDN-RNN



Original recurrent prediction diagram

$$h_{t+1} = f_\psi(h_t, z_t, a_t)$$

$$p_\psi(z_{t+1,j} \mid z_t, a_t, h_t) = \sum_{k=1}^{K} \pi_{t,j,k}\, \mathcal{N}(z_{t+1,j}; \mu_{t,j,k}, \sigma^2_{t,j,k})$$

For each latent coordinate j, predict mixture weights, means, and variances.

The released implementation models coordinates independently given the recurrent context.

Memory summarizes history; the mixture describes possible next latents.

Learning Latent Dynamics

Stage	Training data	Objective
1 · Vision	Observations collected from the environment	Train the VAE.
2 · Memory	Encoded observations + recorded actions	Train next-latent likelihood; freeze V.
3 · Control	Interaction using V and M features	Optimize task return; freeze V and M.

$$\mathcal{L}_M = -\sum_t \log p_\psi(z_{t+1} \mid z_t, a_t, h_t)$$

Training histories contain observed latents; imagined histories will contain model samples.

A one-step likelihood objective must ultimately support repeated prediction.

Observation and Imagination

With a real observation	Inside the learned model
$z_t \sim q_\phi(z_t \mid o_t)$	$\tilde{z}_{t+1} \sim p_\psi(\cdot \mid \tilde{z}_t, a_t, h_t)$
Infer a latent from a new sensor input.	Sample a next latent from the transition model.
New observations can correct the history.	No new observation corrects an erroneous rollout.

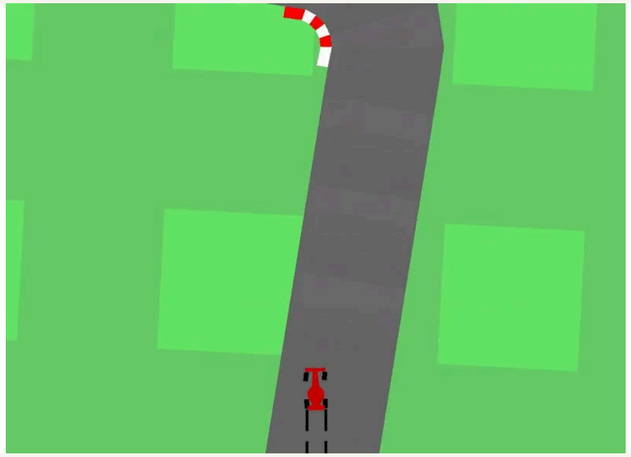
$$a_t = C_\eta(\tilde{z}_t, h_t), \qquad h_{t+1} = f_\psi(h_t, \tilde{z}_t, a_t)$$

A decoder can visualize imagined latents; the controller need not consume decoded pixels.

Imagination replaces incoming observations with model-generated possibilities.

Control with Latent Features

CarRacing · controller with z only



CarRacing · controller with z and recurrent memory

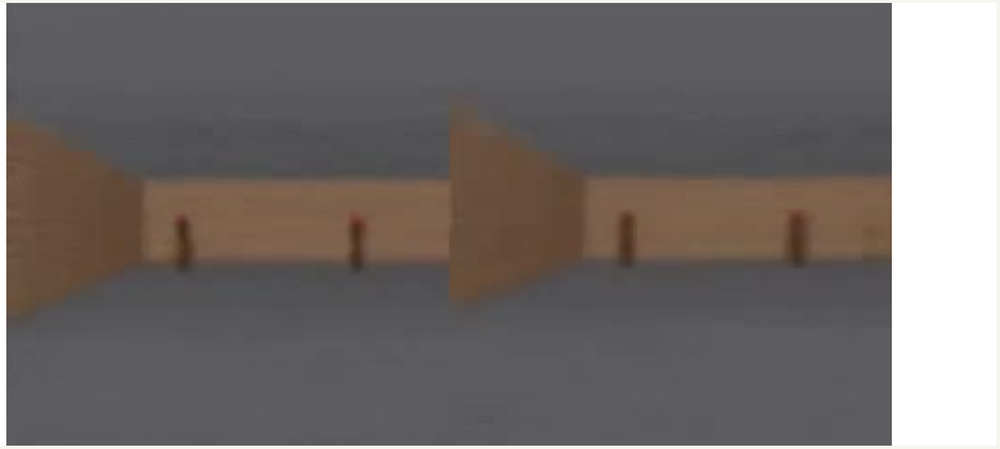
$$a_t = C_\eta(z_t, h_t)$$

CarRacing: the controller was optimized in the real environment using learned features.

The paper’s controller used CMA-ES. This experiment is not online model-predictive control.

Memory can help control even when the policy does not plan by explicit online rollouts.

Learning Inside a Dream



VizDoom observations and their VAE reconstructions

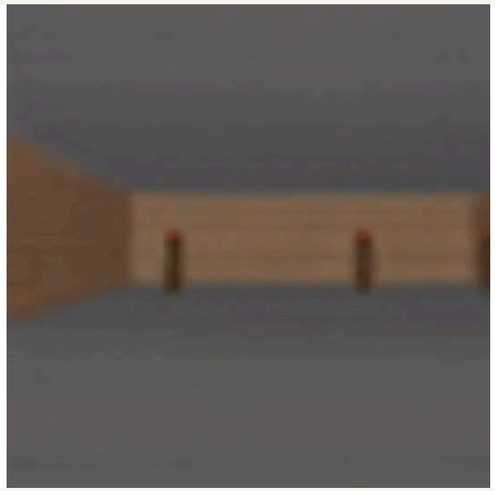
Controller transferred to the real VizDoom environment

VizDoom experiment: train the controller in a learned latent simulator, then evaluate transfer.

M predicts next latents and episode termination. Survival supplies the task objective.

The stronger test is whether behavior learned inside the model works in the environment.

Model Exploitation



World Models · behavior that exploits an imperfect learned environment

A controller is optimized against the model it experiences.

It may discover actions where that model is wrong.

Good performance in an imagined environment can therefore fail to transfer.

The paper varied sampling temperature to study robustness. More noise is not a general correction for model bias.

Evaluate the model where the resulting policy takes it—not only on the original dataset.

Representations, Dynamics, and Utility

Layer	Question	Evidence
Representation	What survives the encoder?	Readouts under stated task and budget.
Dynamics	What follows this history and action?	Conditional prediction and multi-step rollouts.
Utility	Can these predictions support behavior?	Planning or control in the environment.

JEPA-style features, VAE latents, or other representations can participate in a predictive system.

> **Can we identify a world model from the encoder's training loss alone?**
>
> No. Specify temporal and action conditioning, what the system predicts, and how those predictions are used.

World models are defined by predictive capability and use; JEPA is one route to learning them.

Three Collapse Problems

Context	What collapses?	A diagnostic question
Noncontrastive representation learning	Different inputs receive indistinguishable features.	Do features vary and preserve useful information?
VAE posterior collapse	The latent carries little information about its input.	Does q(z | x) depend on x? Does the decoder use z?
GAN mode collapse	Generated samples cover too little of the target distribution.	Which valid outcomes are missing?

> **Is checking latent variance enough to rule out all three?**
>
> No. Variance, input information, and output coverage are different properties.

Name the random variable, the distribution, and the information that was lost.

Next: Autoregressive Prediction

For a fixed candidate action sequence:

$$p(o_{1:T} \mid a_{1:T-1}) = p(o_1) \prod_{t=1}^{T-1} p(o_{t+1} \mid o_{\le t}, a_{\le t})$$

Today	Next steps
Representations preserve selected information.	L8 · Autoregressive prediction and rollout.
VAE and GAN learn distributions.	L11 · RSSM: recurrent state, prior, and inference.
World Models links prediction to control.	Evaluate long-horizon behavior.

Exit question: What does your model know at time t, and what can it predict under an action?

A plausible next sample is only the beginning of a useful rollout.