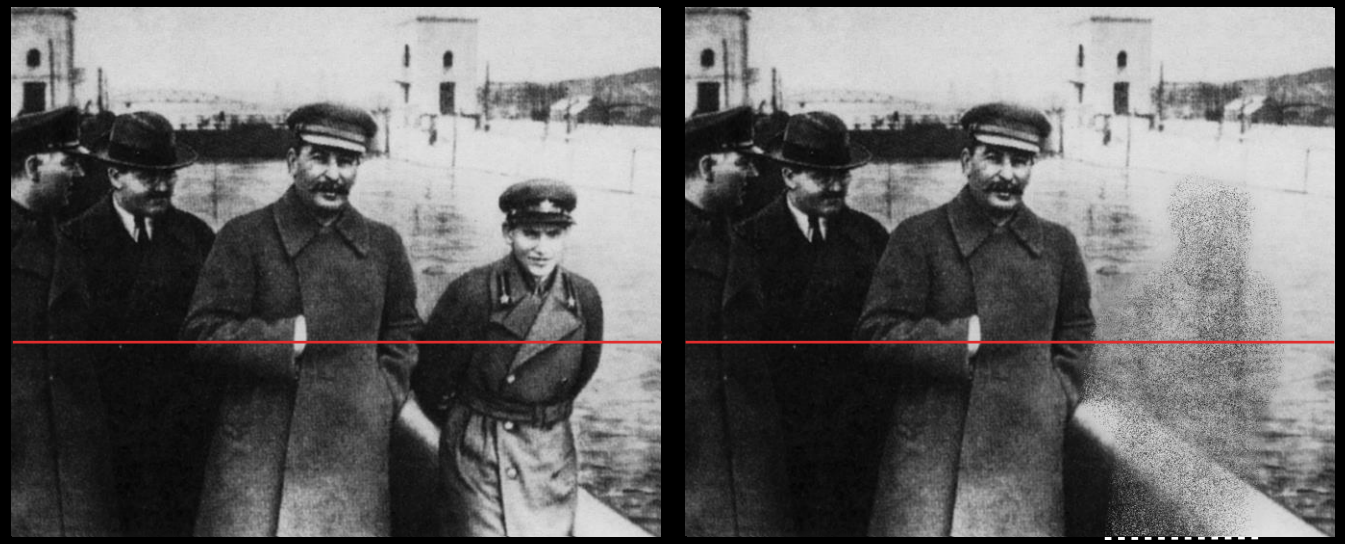


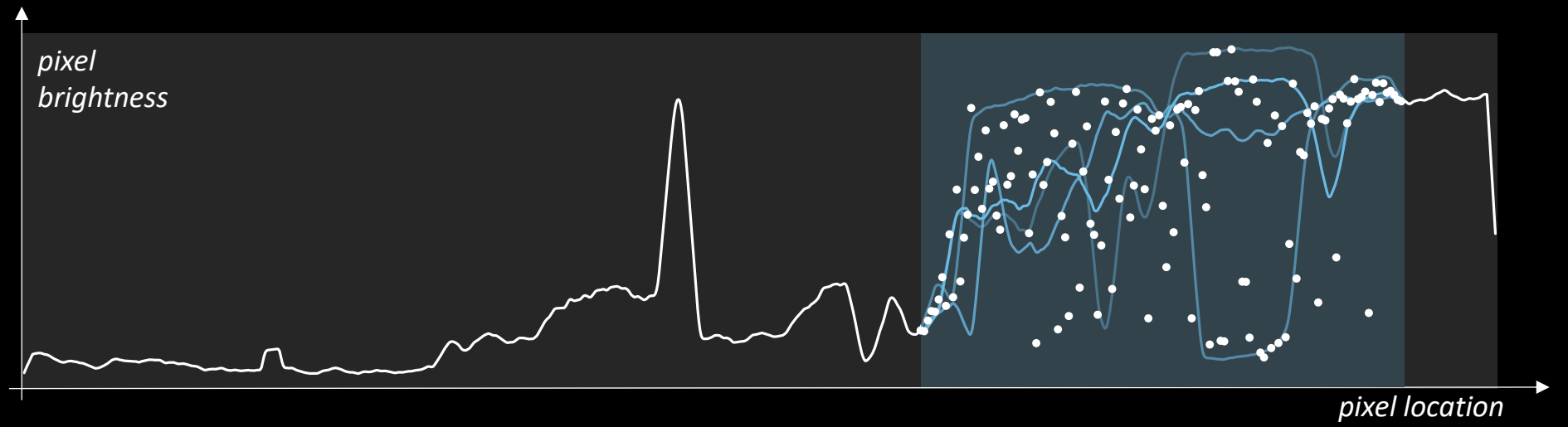
Stalin with Nikolai Yezhov, April 1937

Give a formula for how the gap might be filled, using probability notation.

draw from $(Y \mid Y_x = y_x \text{ for all } x \in \mathcal{K})$

$\Pr_Y$ is a probability model for complete images

$\mathcal{K}$ = pixels we want to keep



The important thing is to draw a *coherent completion* from the joint distribution $(Y \mid \text{keep})$.
If we drew separately from $(Y_{x'} \mid \text{keep})$ for each blanked pixel x' we'd get a silly answer.

The First AI-Generated Interactive Game

Experience Oasis - a revolutionary AI-powered game that generates real-time, interactive gameplay. Built with cutting-edge neural networks, every frame is created on the fly.

Start Playing

https://oasis-ai.org/

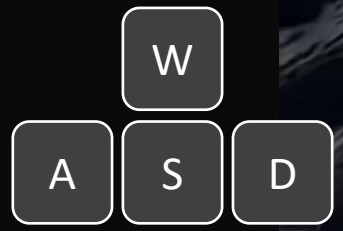


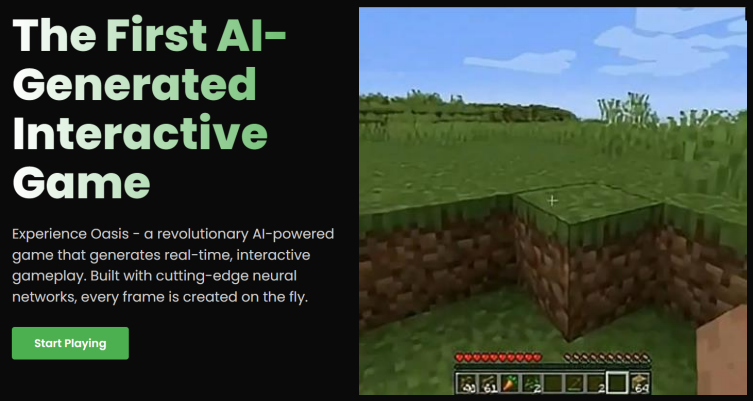


October 27, 2026 Models

Genie 4: A new generation of world simulation

Raia Hadsell, Diego Rivas and Hang Qi



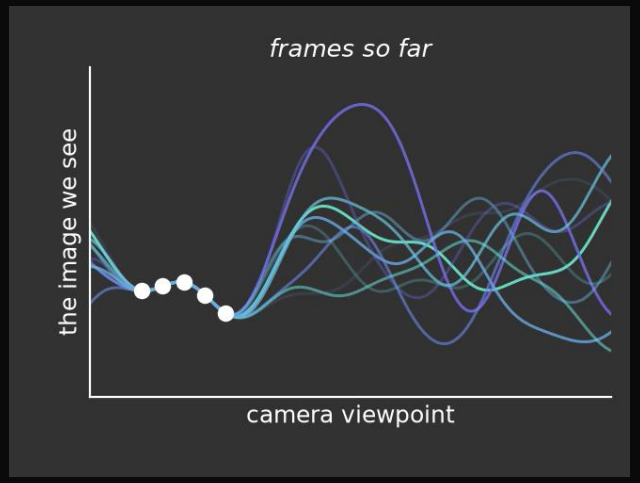


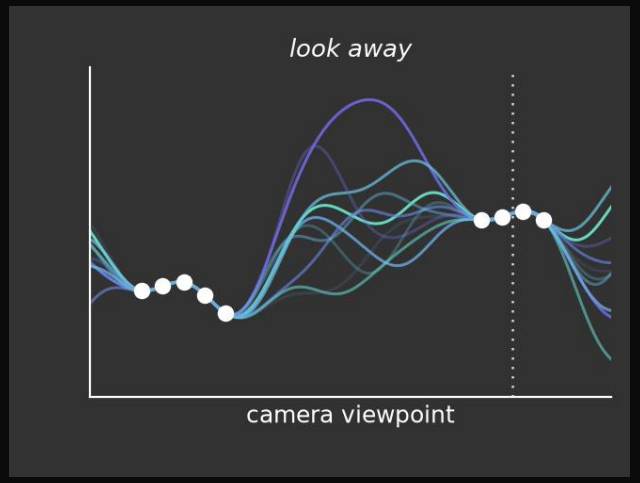


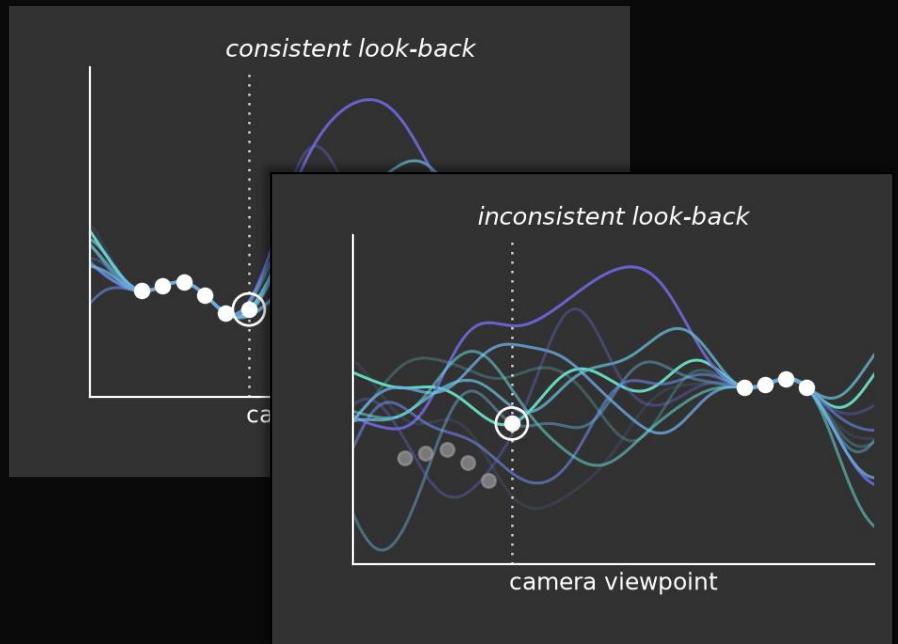
"actual worlds with structure and logic and rules"

How it works: Each frame is generated conditional on previous frames.
Each new frame is generated incrementally – there's no pre-ordained world-model object.

What it needs: Does the world have "consistency"? If I see something in one frame, then turn away, then turn back and look again, is the thing still there?





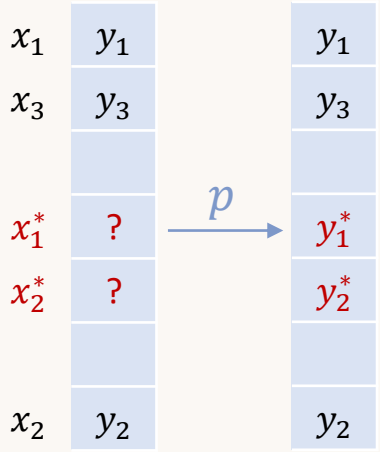


We want to be able to generate readouts as if from a real persistent world

Definition. A *show-ask machine* p is a family of distributions $p(\cdot\,;\,\mathcal{O},\mathcal{T})$

- $\mathcal{O} = [(x_1, y_1), \ldots, (x_m, y_m)]$ is the already-observed (coordinate,value) pairs
- $\mathcal{T} = [x_1^*, \ldots, x_{m^*}^*]$ is the asked-for x-coordinates

specifying a joint distribution of the values $Y_{\mathcal{T}}^* = (Y_1^*, \ldots, Y_{m^*}^*)$ at the asked-for x-coordinates.



Definition. A show-ask machine p is called a *world model* if there is a distribution r over whole worlds such that every member of the p family is consistent with r:

$$p(y_{\mathcal{T}}^*\,;\mathcal{O},\mathcal{T}) = r(y_{\mathcal{T}}^* \mid \text{the readings in } \mathcal{O}) \text{ for all } \mathcal{O},\mathcal{T},y_{\mathcal{T}}^*$$

what the machine generates, given the query

a coherent completion, drawn from the world distribution



Sometimes the x-axis is so huge that we have no choice but to generate incrementally, using show-ask queries

... and if p is a world model, and if we feed back our samples into $\mathcal{O}$, all our looks back will be consistent.

Sometimes we want to draw the full completion in one go

... though knowing r isn't enough. Sampling from $r(\cdot\,|\mathcal{O})$ is a show-ask query, and we still need a mechanism to answer it.



Task. Build a show-ask machine that's a world model.
If we can, then it is as if we're reading from a pre-ordained world.

A simple tractable class of world models: Gaussian processes

A simple way to specify a world model p is
1. specify the distribution r of the full process $Y = (Y_x)_{x \in \mathcal{X}}$
2. define the show-ask machine p to return the conditional distributions of r

Then p will automatically be a world model.

A *Gaussian process* with kernel k is a probability distribution on the full process such that, for every $\mathcal{T} = [x_1, \ldots, x_m]$,

$r(y_{\mathcal{T}}) = \Pr_{\mathrm{MVN}(0,\Sigma)}(y_{\mathcal{T}})$ where $\Sigma_{jj'} := \mathrm{Cov}(Y_j, Y_{j'}) = k(x_j, x_{j'})$ giving $p(y^*_{\mathcal{T}}; \mathcal{O}, \mathcal{T}) =$<messy algebra involving Σ>

Example: a pre-ordained Gaussian process.

First generate $\alpha \sim N(0, \rho_0^2),\ \beta \sim N(0, \tau_0^2)$

Then let $Y_j = \alpha + \beta x_j + E_{x_j}$

where $E_x \sim N(0, \sigma_0^2)$ independently for each x.

The joint distribution of $Y_{\mathcal{T}}$ at $\mathcal{T} = [x_1, \ldots, x_m]$ is

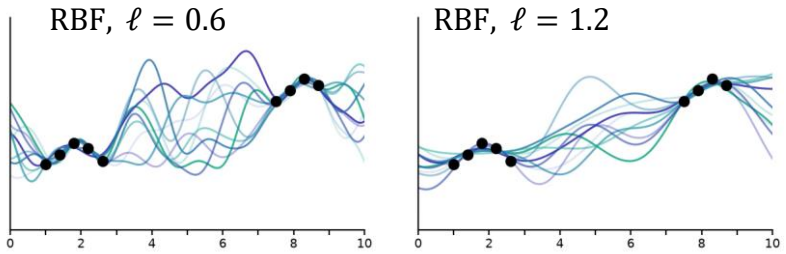
$r(y_1, \ldots, y_m) = \Pr_{\mathrm{MVN}(0,\Sigma)}(y_1, \ldots, y_m)$

where $\Sigma_{jj'} = \mathrm{Cov}(Y_j, Y_{j'}) =$ <algebra>

All that matters here is the covariance structure. The idea of a Gaussian process is that all we need to specify is this covariance structure, rather than a "latent parameter" story.

Example: an incrementally generated Gaussian process.

The RBF_ℓ kernel specifies that $\mathrm{Cov}(Y_x, Y_{x'})$ is large when $|x - x'| \ll \ell$, small when $|x - x'| \gg \ell$. Use the implied show-ask machine to generate Y x-coordinate by x-coordinate.



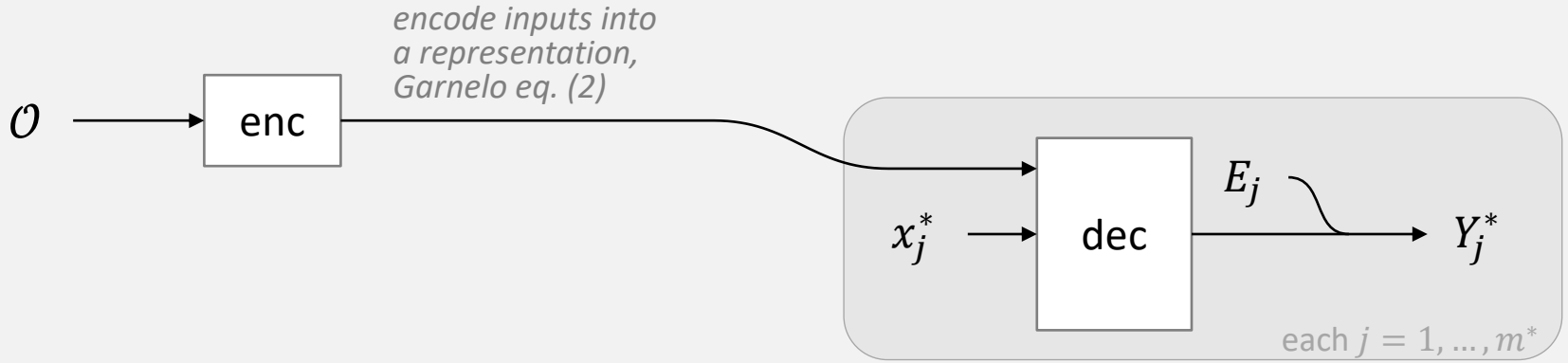
We can generate any Gaussian process incrementally. For the Bayesian linear regression, we can use either pre-ordination or incremental generation.

World-model wish-list

Gaussian processes do lots of things right.
Can we achieve the same effect with a neural network?

1. Respond to $\mathcal{O}$
   like the CNP
2. Generate a coherent completion
   GCVAE does it via a ‘pre-ordained’ latent
   Transformer can do it via auto-regression
3. Generate a completion that’s consistent with what we generated previously
   Needs the right wiring: append generated values to $\mathcal{O}$
   GCVAE: needs the correct posterior. Transformer: ???

The Conditional Neural Process §2.2 is a show-ask machine



What was it designed to achieve?

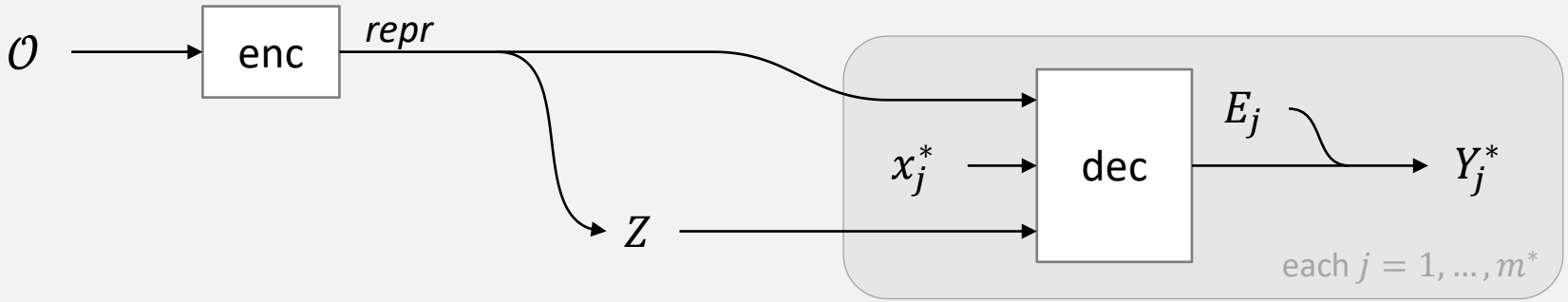
To produce $Y_{\mathcal{T}}^*$ that depend on previous observations $\mathcal{O}$, like a GP does.

Does it generate consistently, i.e. is it a world model?

No. It generates the completion points independently.



The CNP latent-variable model §4.2.3 is a show-ask machine



What was it designed to achieve, that §2.2 did not?

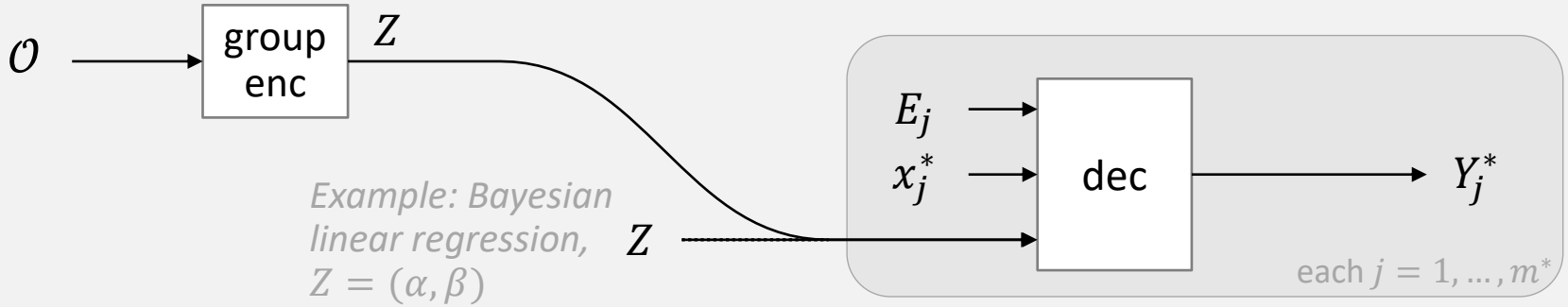
A coherent completion: the values Y_j^* that we generate all depend on each other. The trick is a shared latent Z.

Does it generate consistently, i.e. is it a world model?

No. It's clearly impossible to be consistent, unless we add (x_1^*, Y_1^*) to $\mathcal{O}$ before generating Y_2^* etc. In CNP, they don't bother doing this.

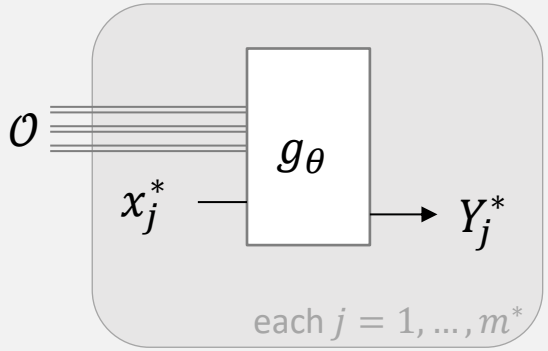
A better show-ask machine is the group-conditional VAE

Edwards and Storkey 2017, "the neural statistician" [lecture 4]



If we feed generated Y_j^* back into $\mathcal{O}$, and if the group encoder returns the true posterior of $(Z|\mathcal{O})$, then this will be a world model.

A network that takes in sequences is also a show-ask machine

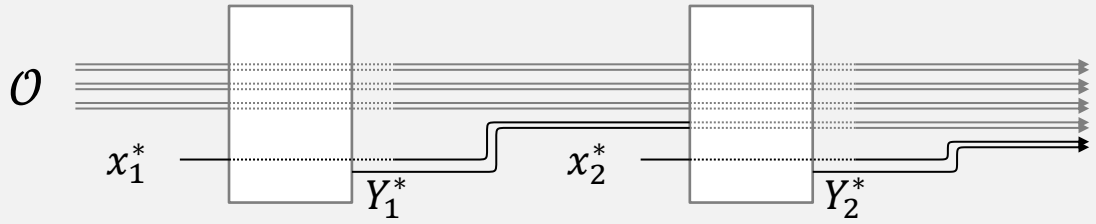


If we replaced equations (1)–(3) by $\phi_* = g_\theta(x_*, \mathcal{O})$, where g_θ is a network that operates on sequences, would it generate consistently?

If Oasis / Genie generated like this, what would go wrong?

How might we fix this?

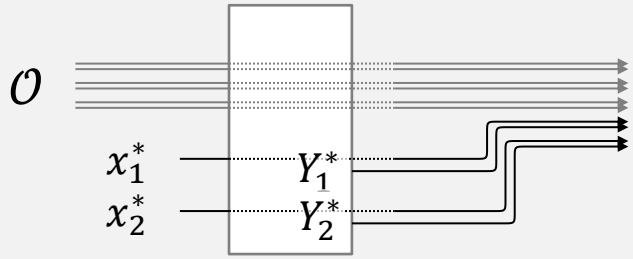
We could generate incrementally, and feed each generated Y_j^* back into the context for the next frame.



Do these generate consistently?

They might or might not be world models, depending on what what the transformer has learnt

Or we could generate a coherent completion in one batch.



What might we do to reduce the computational burden of ever-increasing $\mathcal{O}$?