

07 What Is JEPA, and Why Not Predict Pixels?

BY NILUSHANAN KULASINGHAM

12 MIN READ • INTERACTIVE

When a deterministic pixel predictor meets an open future, its best answer can be a picture of something that cannot happen. Yann LeCun's case against generation, what sampling fixes, what embeddings avoid, and what each still owes.

The figures in this chapter are interactive. Press and drag them online at worldmodels101.com/chapters/jepa.

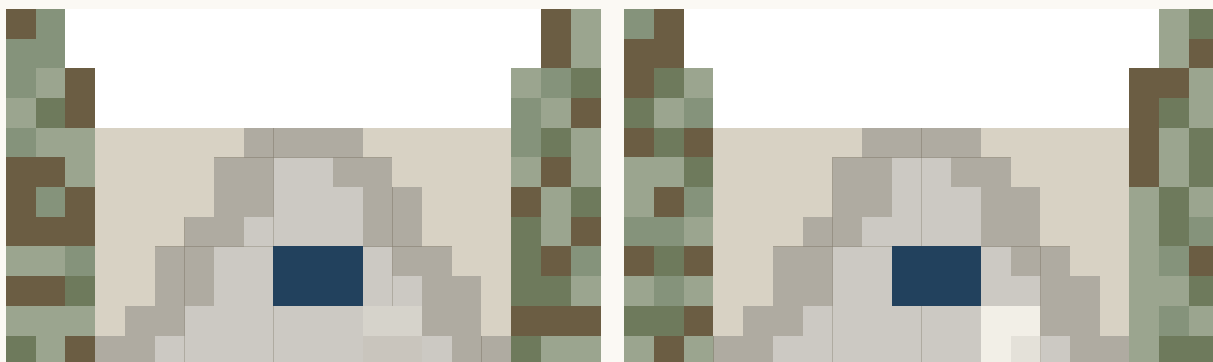
If you've followed the world model argument online, you'll have run into a slogan: stop predicting pixels. It comes from Yann LeCun. He was Meta's chief AI scientist at the time, and he is a Turing Award winner. He is also one of the people who made convolutional networks work, the kind of network that reads images.

In 2022 he published *A Path Towards Autonomous Machine Intelligence*. It is a position paper, meaning a case for how machines should learn rather than a result. One charge in it was aimed at the reflex then governing learned world models: when in doubt, predict the next picture.

His smaller charge starts with a camera pointed at a road. Ask what the next second looks like: a leaf at the edge of the frame will move, and which way is unknowable. It is also the least useful fact on offer, since nothing you might do about the road depends on it. A model asked to predict the picture has to predict that leaf anyway.

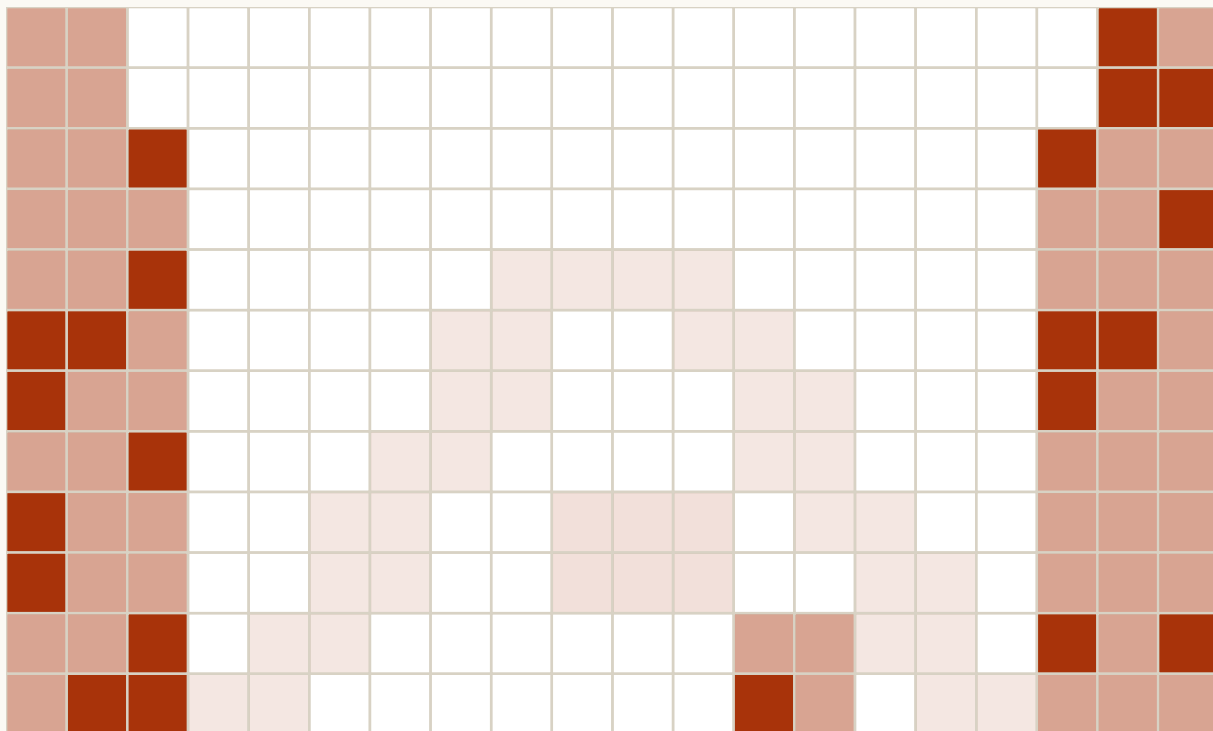
The leaf is made of pixels, and pixels are what the model is marked on. Multiply that by every leaf, every ripple, every flicker of light on wet tarmac. Most of the capacity has gone to things that are hard to predict and do not matter. That is the leaf

problem, the smaller of the two charges.



WHAT THE MODEL PREDICTED

WHAT HAPPENED NEXT



WHERE THE ERROR LANDED

each cell tinted by its squared error

PIXELS A DRIVER CARES ABOUT

the car, the road edge

10% OF THE ERROR

15% OF THE FRAME

PIXELS NO DECISION DEPENDS ON

leaves, puddle, light

90% OF THE ERROR

30% OF THE FRAME

Most of the error, and so most of the push on the weights, is on pixels no decision depends on.

SHADES, ERRORS AND SHARES ARE ILLUSTRATIVE

SECOND 1	SHARE OF THE ERROR ON THE CAR AND ROAD EDGE 10%	SHARE ON LEAVES AND WATER 90%
-------------	--	-------------------------------------

FIG. 7.1 A road scene, scored the way a pixel loss scores it. Press Next second and watch where the error lands: the car ahead moves the way the model expected, and the leaves and the puddle flicker the way nobody could expect. Now drag the foliage slider up and read the ledger on the right. The share of the error on things a driver would care about keeps shrinking, and every one of those pixels pulls on the weights just as hard. The numbers are illustrative.

JEPA is a family of models from Meta. It predicts a description of what it cannot see, instead of the pixels. An embedding, in everything that follows, is a short list of numbers standing in for a picture.

An average of two futures is neither

The larger charge is a car coming to a junction. It will turn left or right. You do not know which, and no amount of staring at the current frame will tell you.

Now let's ask a deterministic model, one that gives a single answer every time, for one next picture. Score it by squared pixel error, the gap at each pixel squared and added up. The best answer in the maths is the conditional average of the two turns. That is the left and right futures blended at their odds, rather than either one on its own.

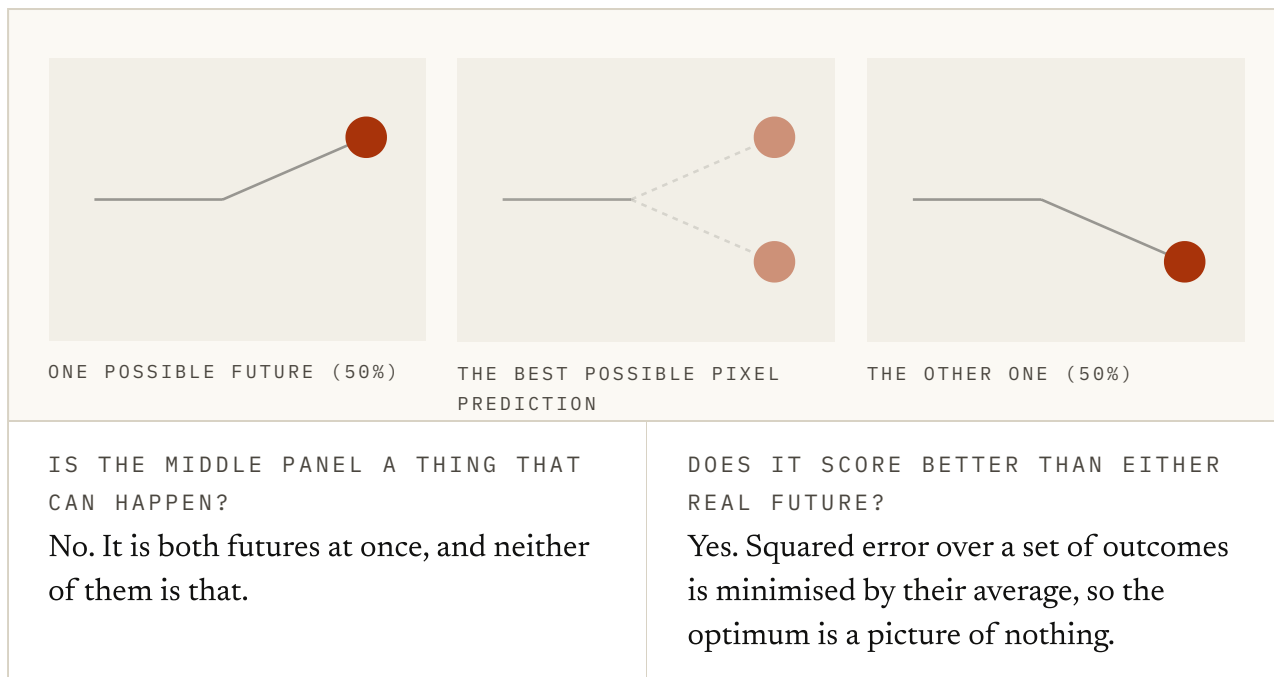


FIG. 7.2 The middle panel is the two futures drawn at their odds, and that average is what squared error rewards. Drag the chance to either end and you should see the best answer settle on one turn. Leave it near the middle and the best prediction becomes a picture of something that cannot happen.

Have a look at the middle panel. It scores better than either real future. A model that predicted the left turn would be punished half the time. The ghost in the middle is punished a little all the time, and under squared error a little all the time wins.

So that training signal asks for the mean rather than for a plausible future. When the future is open, the mean is a smear that never occurs.

A deterministic predictor trained with a mean-seeking pixel loss goes vague just when something interesting is about to happen. Modern generators avoid that failure by drawing samples rather than aiming at a mean.

The ghost had an answer five years early

One complication, though. The pixel camp had dealt with the ghost by 2017, five years before the position paper. SV2P, short for *Stochastic Variational Video Prediction*, is Mohammad Babaeizadeh and colleagues' video model from that year, and Emily Denton and Rob Fergus built one like it. Both learn a spread of futures and draw one at random.

The trick is a hidden variable, a few numbers standing in for whatever the frame cannot tell you, such as which way the car turns. Draw the variable first, then the frame that goes with it. Each draw can be sharp even when the model has no idea which turn will happen.

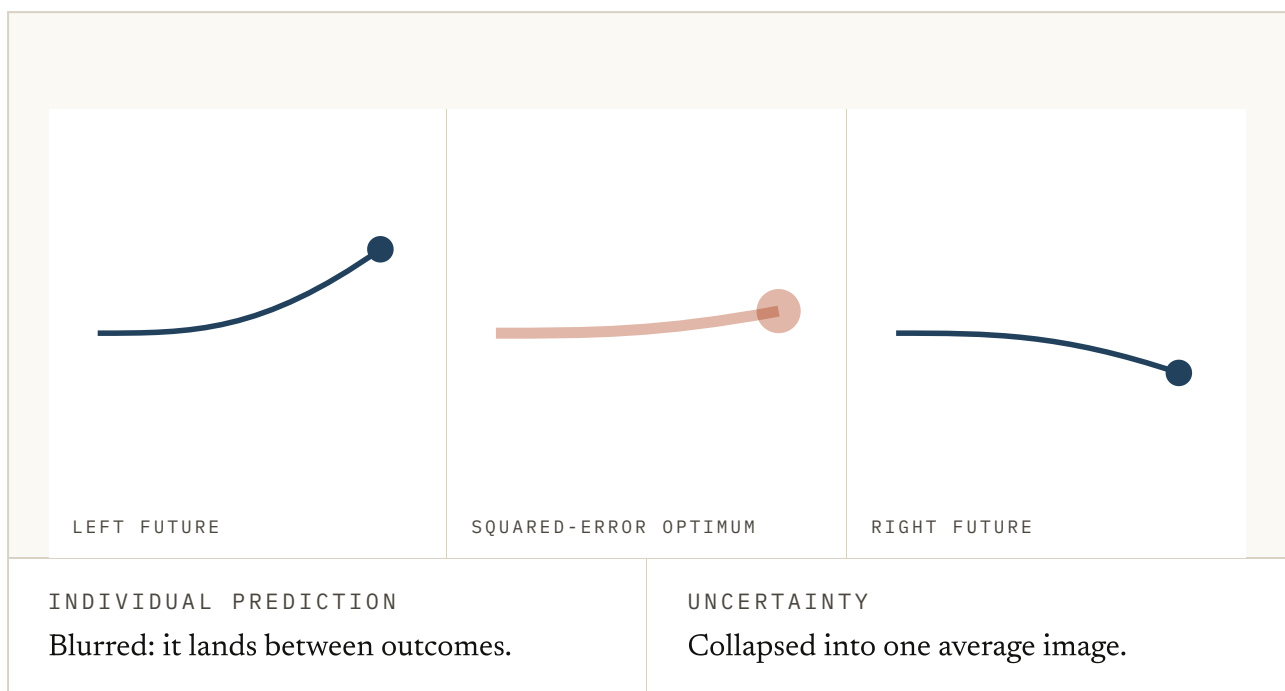


FIG. 7.3 Switch from the mean to samples of the same two futures, and the in-between frame disappears. Press Draw a few times. The uncertainty survives as variation across draws. What the model still does not say is which draw the world will choose, so a planner has to value risk across the whole spread.

That kills the blur objection, though not the rest of the case.

Sampling fixes the form of the output. It does not make useless texture cheap, make rare futures well estimated, or tell a decision maker which possible future matters. Stochastic generation keeps the pixel target, and embedding prediction changes it.

Predict the description instead

LeCun's proposal is to stop predicting the picture. Encode what you can see into a compact description (an *embedding*, a short list of numbers that stands in for something bigger). Similar things land near each other in that list. Encode the part you are predicting into one too.

Then train the model to predict the second description from the first. Nothing is ever decoded back into pixels. I-JEPA is the simplest example: a 2023 image model from Mahmoud Assran and colleagues in LeCun's group at Meta. Hide part of the picture and predict the embedding of the missing piece. What changes with the target is what counts as a bad prediction.

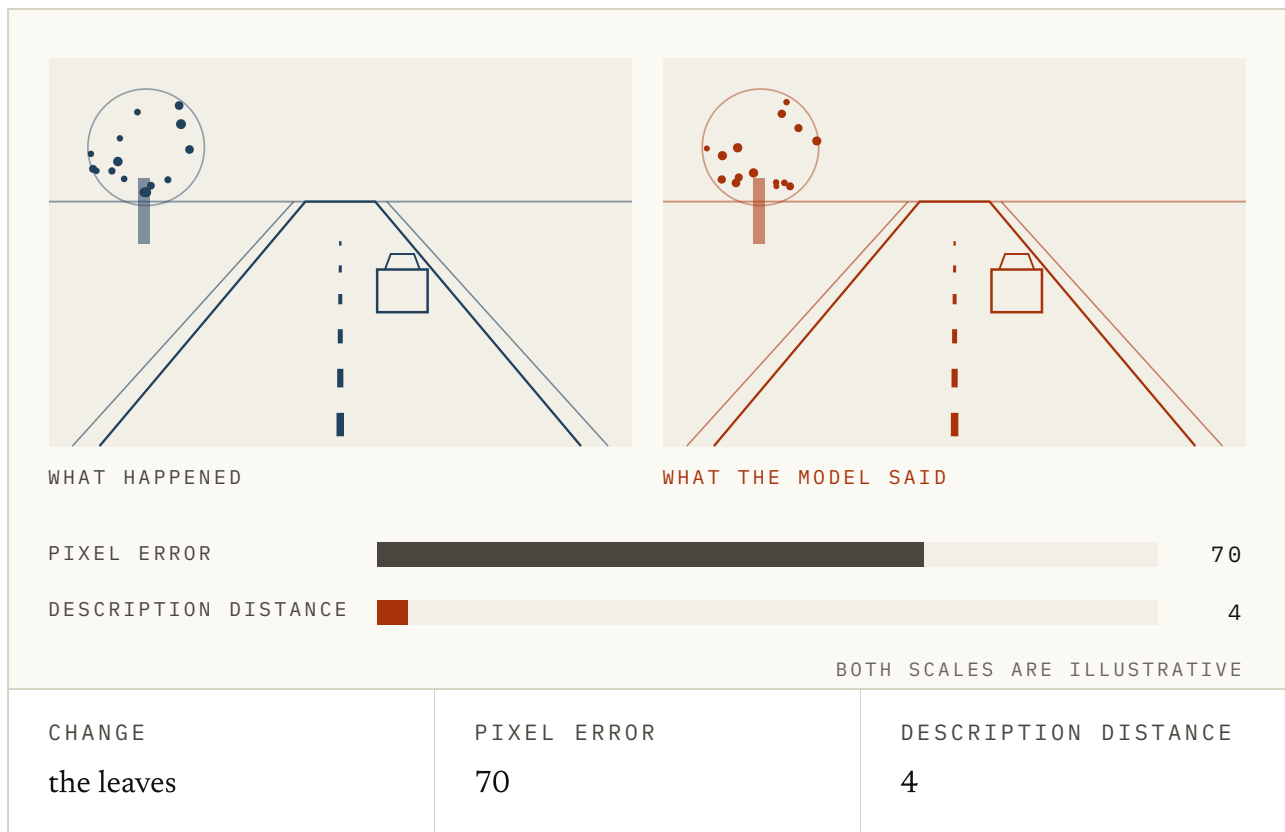


FIG. 7.4 Two road frames, and one change between them. Pick a change and drag it up, then read the two scales. Move every leaf, or shift the whole frame sideways, and the pixel scale charges a fortune for something no decision turns on. Move the car ahead into your lane and a few hundred pixels change what you would do, and the pixel scale barely notices. Both scales are illustrative.

V-JEPA 2, from Meta AI in 2025, does the same for video. It is trained first on plain video, with no actions. Then it is trained again on a robot's actions, for control.

Two things follow. The leaf problem goes away, because the description never had to record which way the leaf went. If a detail does not survive the encoding, nothing downstream is graded on it.

The ghost shrinks as well, because you are no longer averaging pictures. The descriptions of a car turning left and one turning right can share most of their entries. The part still open is either kept explicit or dropped as irrelevant. One caution: a space of descriptions can still hold two separate answers, and changing the representation does not get rid of uncertainty.

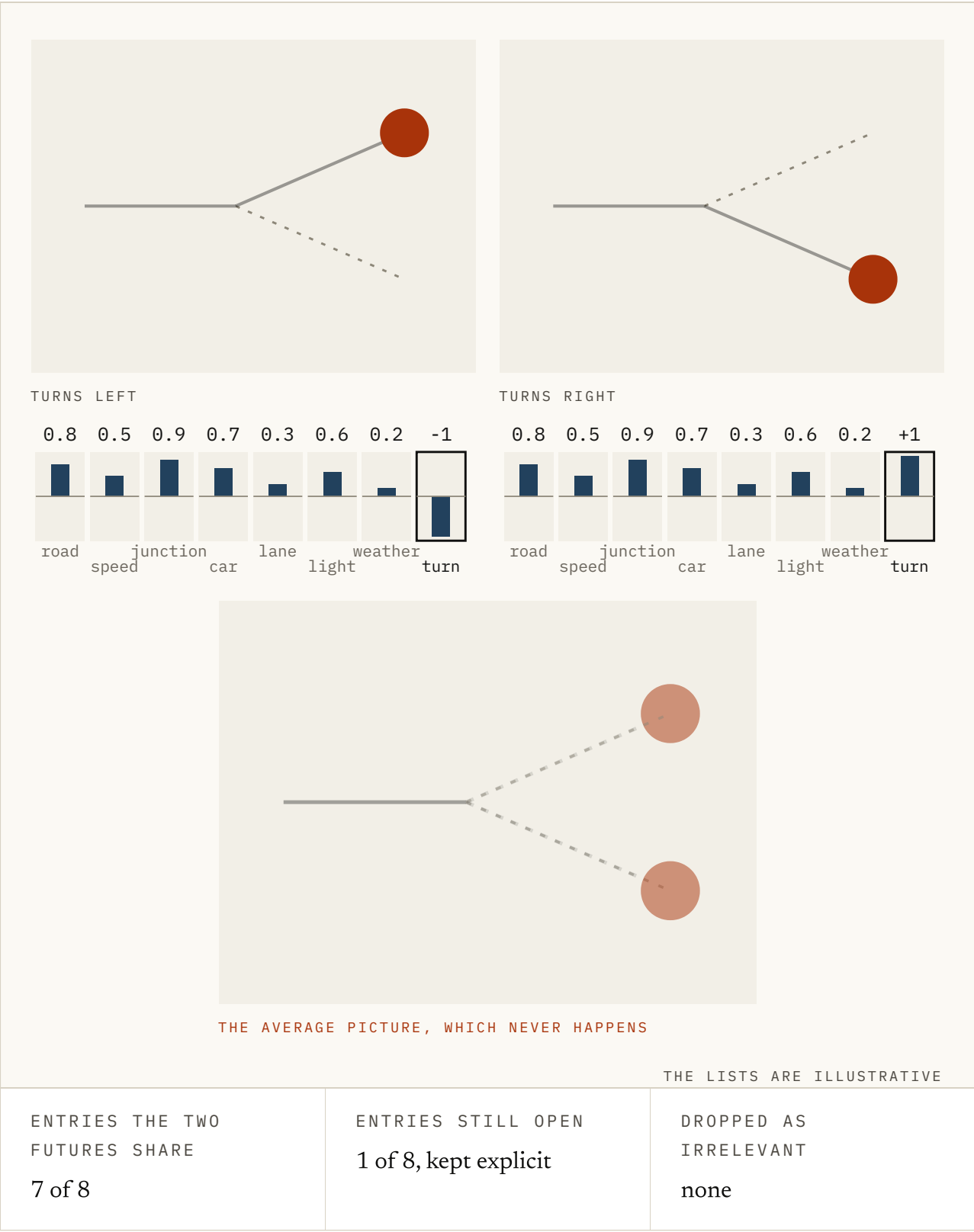


FIG. 7.5 The same junction as Figure 7.2, this time described rather than drawn. Each future is a short list of numbers, and most of the entries agree: road, speed, a car at a junction. Leave the switch on pixels and the prediction is the ghost again. Flip it to description and watch what happens: the shared entries are predicted outright, and the one entry that differs is the only place the two futures still disagree. The lists are illustrative.

This is the JEPA family, and the name describes the architecture. It is joint embedding because both sides get embedded, and predictive because the training signal is one embedding predicting another.

The bill for that

Predicting pixels has one large virtue, and the JEPA side gives it up. The target is fixed, and the model cannot make the next frame easier to predict. Predict an embedding and that stops being true.

The target now comes from the encoder, the network that turns pictures into descriptions, and that network is also being trained. It can get a perfect score while learning nothing, by making every description the same. The prediction is then always right, the loss goes to zero, and the descriptions say nothing about anything. That failure is called collapse.

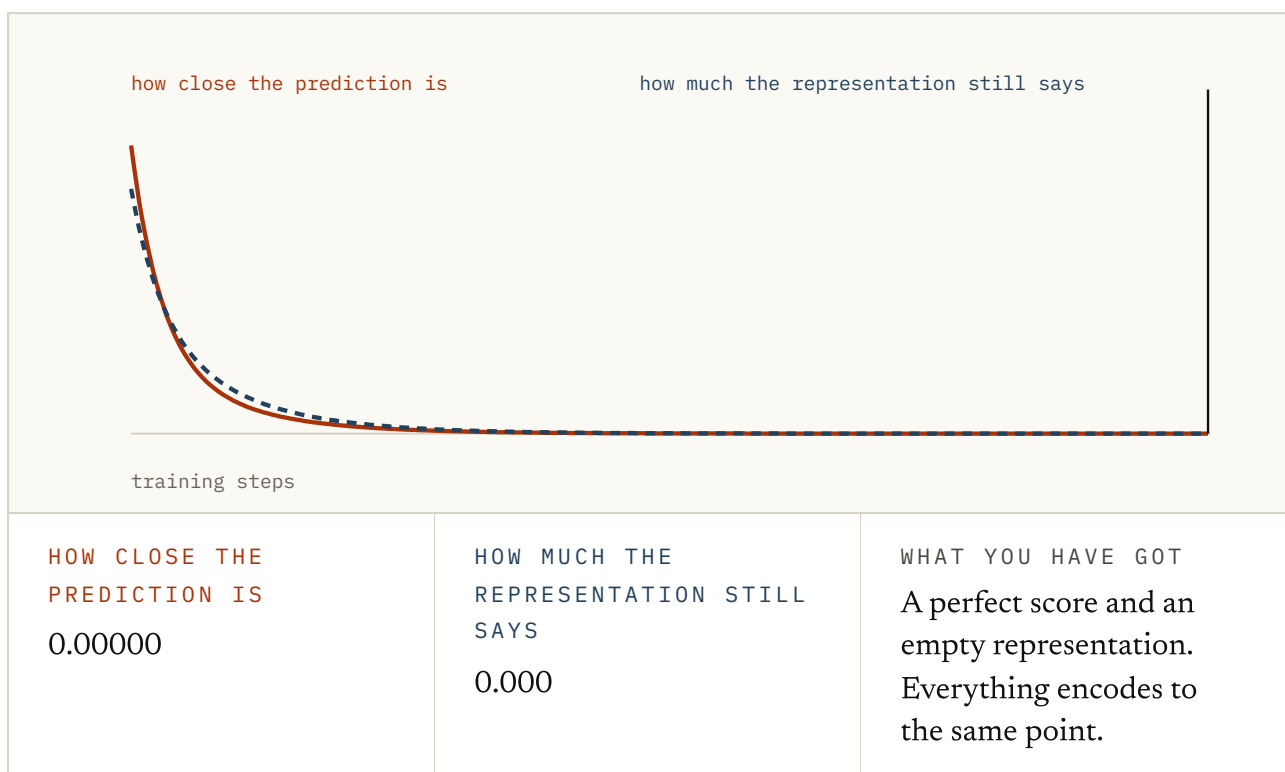


FIG. 7.6 A tiny encoder, two numbers in and two out, trained on four pairs by gradient descent. Leave the safeguard off and drag the step forward: you should see the loss and the spread of the representation reach zero together. Now turn the safeguard on and drag again. The loss never gets close to zero, and that is the run you want.

Most of the difficulty in these methods is preventing that, and the fixes form a lineage. In 2020 the received wisdom for learning descriptions without labels was contrastive, meaning you learn by comparing pairs. Pull two views of the same image together. Push everything else apart, so the descriptions cannot all pile up in one place.

BYOL, short for *Bootstrap Your Own Latent*, is a 2020 method from Jean-Bastien Grill and colleagues at DeepMind. It dropped the pushing apart. Instead it keeps a second copy of the encoder that updates slowly (usually called a *target encoder*, it trails the one being trained), and it predicts that copy's output. The target cannot run away from the predictor.

By the standard account it should have collapsed. It did not, and that convinced people collapse could be avoided without pushing anything apart.

VICReg, a 2021 method from Adrien Bardes, Jean Ponce and LeCun, went the explicit route instead. The name spells out its three terms: variance, invariance, covariance. It penalises descriptions whose parts have collapsed or copied each other. That is Figure 7.6's safeguard done properly.

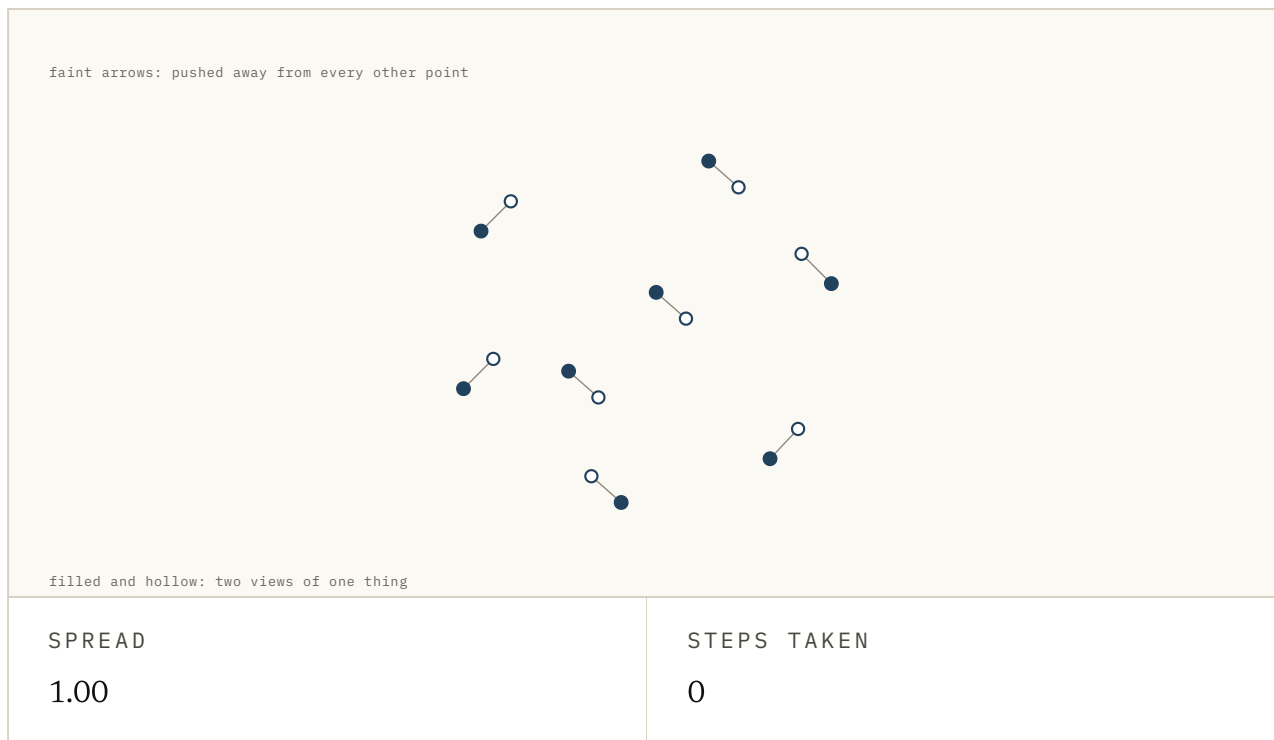
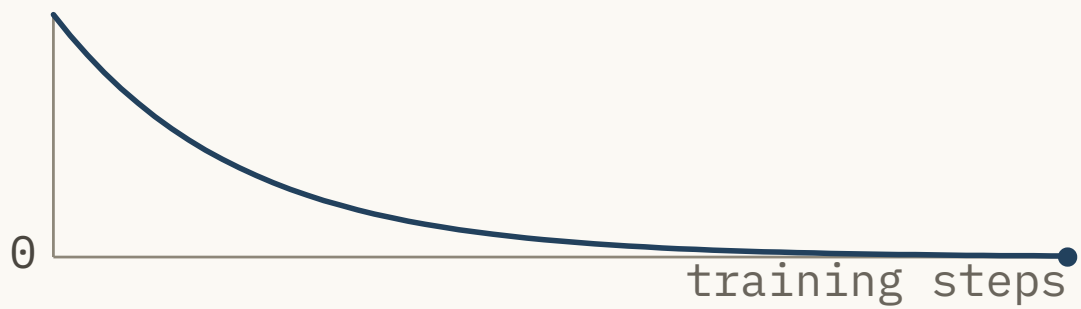


FIG. 7.7 Three ways of stopping the pile-up, each run on the same little cloud of descriptions. Pick one and press Step a few times. Contrastive pushes every pair apart. BYOL pushes nothing apart, and the cloud still holds its shape because the target it chases trails behind. VICReg leaves the points alone and penalises the cloud itself when its spread drops or its two axes copy each other. The cloud is illustrative; watch the spread readout.

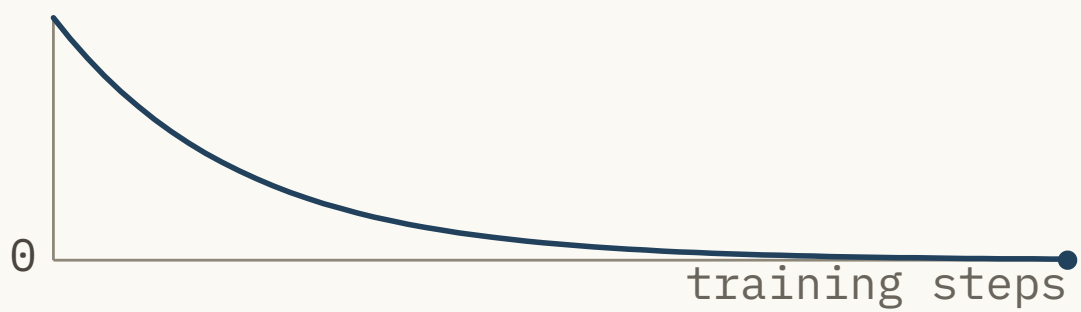
Randall Balestrieri and colleagues wrote a 2023 survey, *A Cookbook of Self-Supervised Learning*. It is frank that much of the field is machinery for stopping the trivial solution from winning.

The loss is also no longer a number you can read. A pixel loss of zero means the model predicted the frame. An embedding loss of zero might mean it understands everything, or nothing at all. The number will not tell you which.

PIXEL LOSS



EMBEDDING LOSS



CURVES ARE ILLUSTRATIVE

PIXEL LOSS AT THE END

WHAT THAT ZERO TELLS YOU

0	not yet revealed
EMBEDDING LOSS AT THE END	WHAT THAT ZERO TELLS YOU
0	not yet revealed

FIG. 7.8 Two training runs, both ending with a loss of zero. Press Reveal on the pixel run and there is nothing to reveal: zero means the frame was predicted. Press Reveal on the embedding run and the same zero splits two ways, one where the descriptions still tell left from right and one where they have all become the same. Only the probe underneath can tell you which you got. The curves are illustrative.

So Quentin Garrido and colleagues tested what the embeddings could do. Their 2024 paper is *Learning and Leveraging World Models in Visual Representation Learning*. They found that the strength of the predictor decides what the encoder keeps. Give the predictor an easy job and the encoder throws away whatever the transformation changed. Make it work, and the encoder keeps enough to undo the change, which is more than the loss ever asked for.

What this does not settle

None of this makes generation a mistake. The version of the argument that says it does is the overclaim. Systems that predict pixels are the ones making worlds you can walk around in, and they work. Whatever they waste on leaves, they deliver a picture, and no embedding does that.

The awkward counter-example comes from LeCun's own lab. Kaiming He, best known for ResNet, the network design computer vision ran on for years, was at that lab in 2021. His paper that year was *Masked Autoencoders Are Scalable Vision Learners*. It hides most of an image and rebuilds its pixels, the very thing JEPA refuses to do.

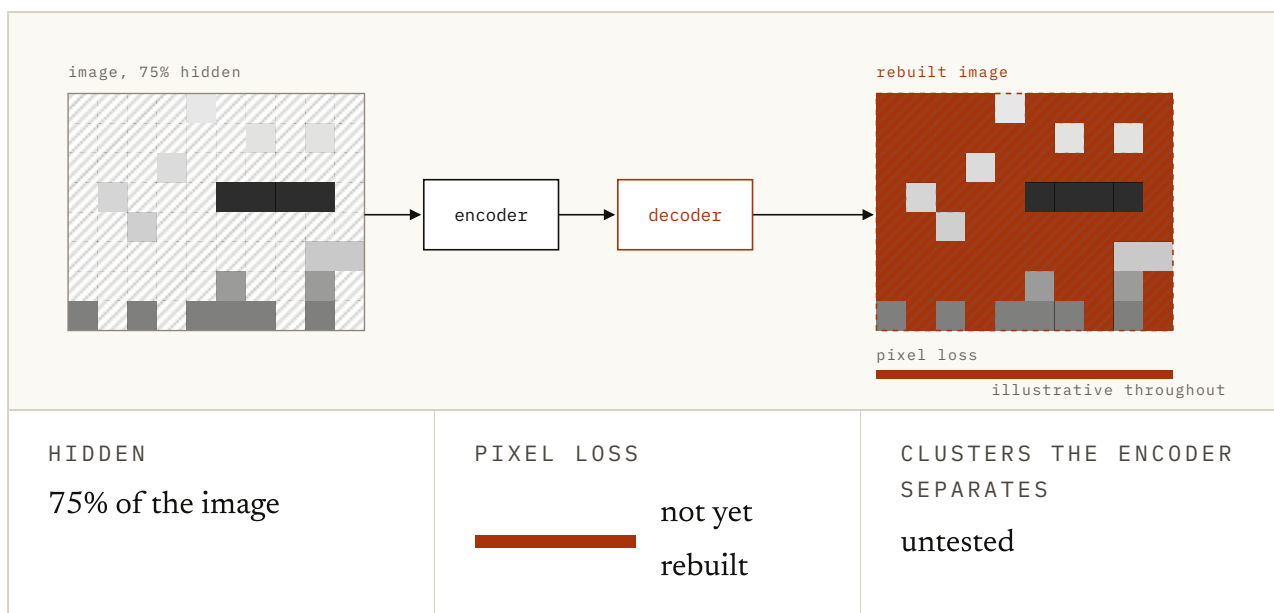


FIG. 7.9 Drag the mask up to three quarters of the image and press Rebuild. The decoder redraws the hidden patches, roughly, which is exactly the job the previous section said was wasteful. Then press Test the encoder. The descriptions it learned still sort the shapes cleanly, and that is the result to keep in mind next time you hear the slogan. Illustrative throughout.

The descriptions it learns are very good anyway. It came out the year before the position paper.

The version of the argument that holds up is narrower than the slogan, but it is right. Predicting appearances spends most of its capacity on things no decision depends on. When the future is open, a deterministic predictor is trained towards a picture of something that will not happen.

If what you want is a compact state to act on, that is a bad way to get one. If what you want is a picture, it is the only way anyone has. Most of the confusion in this corner of the field comes from comparing systems built for different goals.

Hopefully this chapter helped. There is a short quiz below if you want to check the argument stuck, including the part about what it does not settle. If I have got something wrong, or you know a better source than the ones I used, the About page has a corrections section. I would be glad to hear from you.

Try it

1 A deterministic predictor uses squared pixel error and a car turns left or right with equal probability. What is its optimum?

- a. The average of the two, which is a picture of neither
- b. Whichever was more common in training
- c. The left turn
- d. The right turn

ANSWER a. The average of the two, which is a picture of neither. Squared error over a set of outcomes is minimised by their mean. Committing to one turn is punished half the time; the smear is punished a little all the time, and that wins.

2 A stochastic video model now draws sharp left and right turns instead of their blur. What has it fixed, and what has it not?

- a. It removed the need for a pixel target
- b. It proved embedding prediction is unnecessary
- c. It has learned which turn the real car will choose
- d. It fixed the impossible average but still has to represent probabilities and decision risk

ANSWER d. It fixed the impossible average but still has to represent probabilities and decision risk. Sampling is a real solution to blur: each draw can be plausible. It does not identify which draw reality will select or tell a planner how to value the distribution.

3 A leaf at the edge of the frame moves unpredictably. Why does a pixel-predicting model spend capacity on it?

- a. Leaves are important for scene understanding
- b. The leaf is made of pixels, and pixels are what the model is marked on
- c. It cannot tell leaves from cars
- d. The encoder forces it to

ANSWER b. The leaf is made of pixels, and pixels are what the model is marked on. Nothing tells the objective which pixels matter. Hard-to-predict and irrelevant is the worst combination, and it is most of the frame.

4 What does predicting an embedding rather than a picture do to the leaf problem?

- a. Nothing, the leaf is still in the input
- b. It makes the leaf easier to predict
- c. If a detail does not survive the encoding, nothing downstream is graded on it
- d. It moves the problem to the decoder

ANSWER c. If a detail does not survive the encoding, nothing downstream is graded on it. The description was never obliged to record which way the leaf went, so the model is not punished for failing to say.

5 Predicting pixels has one large virtue that predicting embeddings gives up. What is it?

- a. The target is fixed: the model cannot make the next frame easier
- b. It needs less data
- c. It generalises better
- d. It is faster to compute

ANSWER a. The target is fixed: the model cannot make the next frame easier. The moment the target is produced by a network that is also being trained, the target can move, and there is a way to make it move somewhere very convenient.

6 What is collapse?

- a. The model forgetting earlier training
- b. Gradients vanishing in deep layers
- c. The loss diverging to infinity
- d. Every input encoding to the same description, so the prediction is always right and the representation says nothing

ANSWER d. Every input encoding to the same description, so the prediction is always right and the representation says nothing. It is a perfect score obtained by learning nothing, and it is the cheapest solution available unless something is specifically stopping it.

7 In Figure 7.6, the run with the safeguard has a worse loss. What does that tell you?

- a. The safeguard is badly tuned
- b. An embedding loss is no longer a number you can read off as quality
- c. The model needs more training
- d. The safeguard should be removed once training stabilises

ANSWER b. An embedding loss is no longer a number you can read off as quality. A pixel loss of zero means the frame was predicted. An embedding loss of zero might mean everything or nothing, and the number cannot tell you which.

8 Does this argument show that generating pixels is a mistake?

- a. No, because collapse makes embeddings unusable
- b. Yes, it is strictly worse
- c. No. It is a bad way to get a compact state to act on, and the only way anyone has to get an actual picture
- d. Yes, except for very short videos

ANSWER c. No. It is a bad way to get a compact state to act on, and the only way anyone has to get an actual picture. The two goals were never the same goal, and most of the confusion here is people comparing systems built for different ones.

FIG. 7.10 Eight questions on the argument and its bill. The last two are there to stop the argument being read as a knock-down case, which is the mistake I'd most like you to avoid.

Sources

Stochastic Variational Video Prediction (SV2P) BABAEIZADEH ET AL., 2017 ↗

The real counterargument to deterministic blur: learn a distribution and draw distinct plausible futures instead of returning their pixelwise mean.

<https://arxiv.org/abs/1710.11252>

A Path Towards Autonomous Machine Intelligence LECUN, 2022 ↗

The position paper the whole argument comes from, including why predicting appearances is the wrong job for a system meant to act.

<https://openreview.net/forum?id=BZ5a1r-kVsf>

I-JEPA ASSRAN ET AL., 2023 ↗

Hide part of an image and predict the embedding of the missing piece rather than redrawing it. The clean statement of the method.

<https://arxiv.org/abs/2301.08243>

V-JEPA 2 META AI, 2025 ↗

The video version, pre-trained without actions and then post-trained for control, which is where the argument meets a robot.

<https://ai.meta.com/blog/v-jepa-2-world-model-benchmarks/>

Bootstrap Your Own Latent (BYOL) GRILL ET AL., 2020 ↗

The slowly-updating target copy, and the result that made people believe you could avoid collapse without pushing things apart.

<https://arxiv.org/abs/2006.07733>

VICReg BARDES, PONCE & LECUN, 2021 ↗

The explicit approach: penalise a representation whose components have collapsed or duplicated each other. Figure 7.6's safeguard, done properly.

<https://arxiv.org/abs/2105.04906>

A Cookbook of Self-Supervised Learning BALESTRIERO ET AL., 2023 ↗

The survey that says how much of this field is machinery for stopping the trivial solution from winning.

<https://arxiv.org/abs/2304.12210>

Masked Autoencoders Are Scalable Vision Learners HE ET AL., 2021 ↗

The counter-example worth holding on to: reconstruct the pixels of the masked part, and it works very well anyway.

<https://arxiv.org/abs/2111.06377>

Learning and Leveraging World Models in Visual Representation Learning

GARRIDO ET AL., 2024 ↗

What the embedding-prediction objective turns out to have learned, tested rather than asserted.

<https://arxiv.org/abs/2404.08471>

FIG. 7.11 I've ordered these for someone building on this rather than for historical completeness, and preferred first-party material throughout.