

04 What Is Latent Space?

BY NILUSHANAN KULASINGHAM

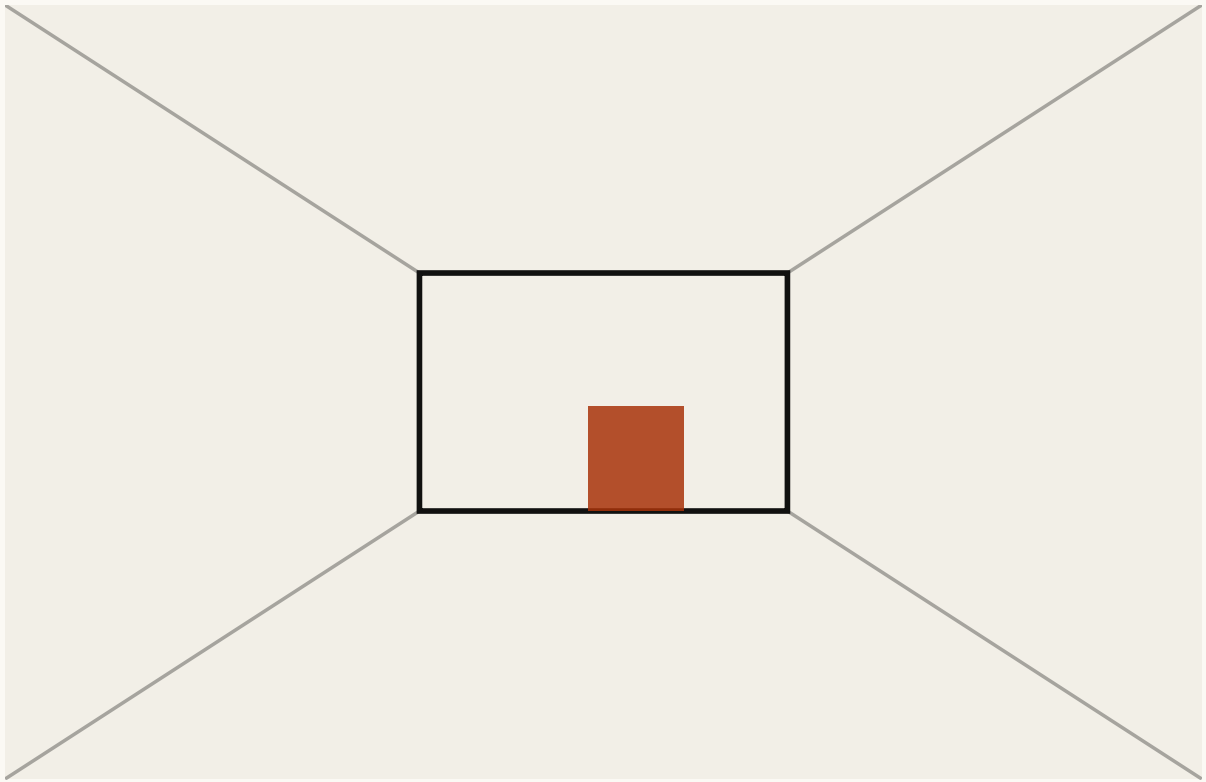
11 MIN READ · INTERACTIVE

A camera measures tens of thousands of numbers and the decision needs two or three. What happens at the squeeze between them, and why that narrow point sets the ceiling on everything downstream.

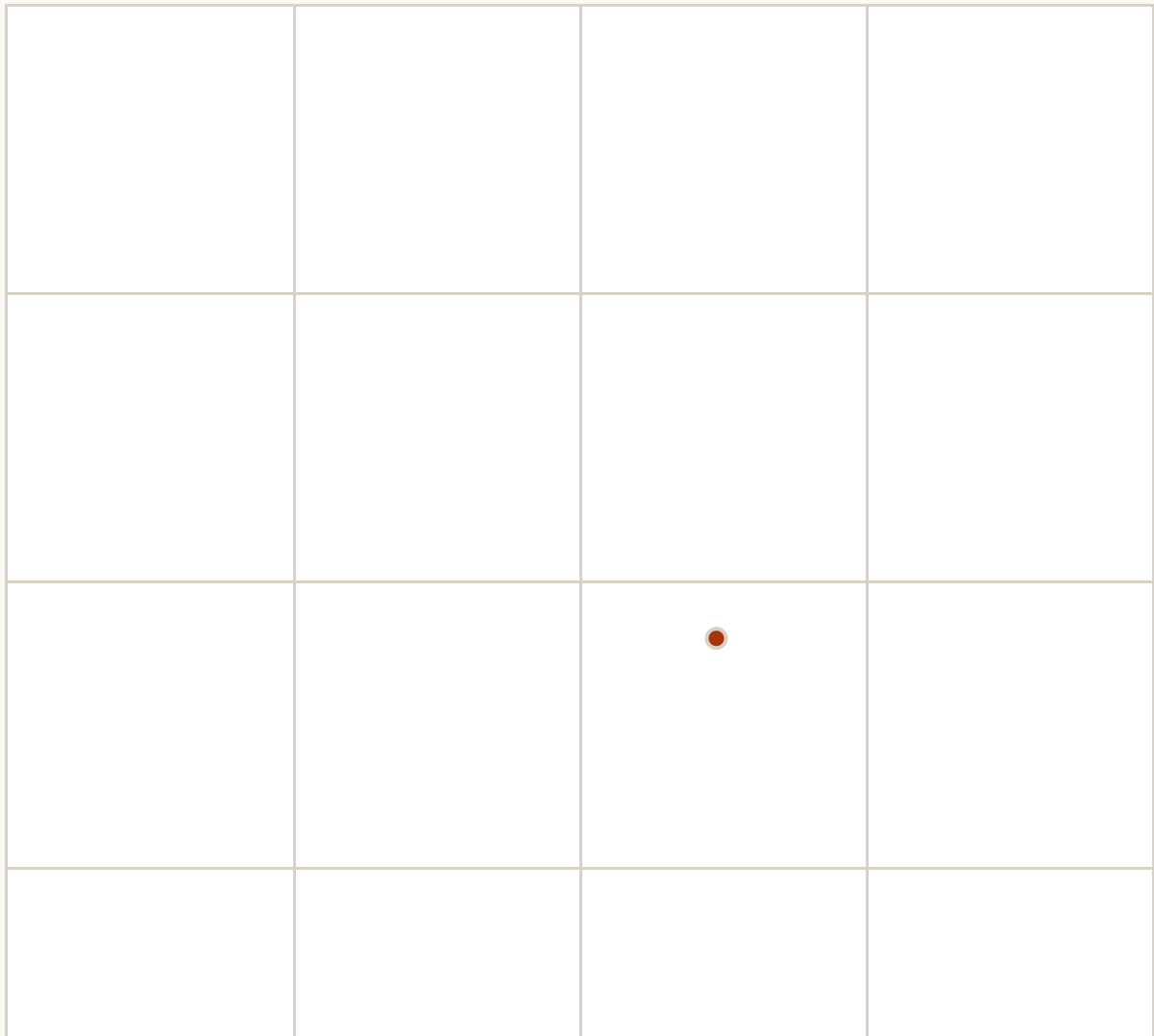
The figures in this chapter are interactive. Press and drag them online at worldmodels101.com/chapters/latents.

Stand in a doorway and look into a room. Your eyes are taking in something like a hundred million measurements a second. Now decide whether to walk forward. For that decision you need almost none of them.

How far away is the far wall? Is there a way through, and roughly where? That is two or three numbers, and everything else your eyes collected was, for this decision, decoration.



THE ROOM THOSE TWO NUMBERS DECODE TO



THE SPACE ITSELF. DRAG ANYWHERE IN IT	
HOW FAR THE WALL IS	0.45
WHERE THE WAY THROUGH IS	0.62
NUMBERS IN THE PICTURE 74,800	NUMBERS IN THE DESCRIPTION 2

FIG. 4.1 Have a go at this one before reading on. Drag anywhere in the square on the right. Every point in it decodes to a room, and the room is computed from those two numbers rather than looked up. Watch how little drag it takes to change what you would do. The picture is tens of thousands of numbers, and the description that produced it is two.

Those two or three numbers have a name. They are the *latent* (latent means hidden, or not directly visible. The numbers are never shown to you and nothing labels them, but the picture is what it is because of them) **description of the scene**. You will meet the phrase "latent space" in nearly every paper from here on.

Somewhere between the camera and the decision there has to be a squeeze, and whatever comes out the other side is all the rest of the system ever gets. That is the case for latents, and the same fact is the case against them.

The squeeze

The architecture is a narrow opening in the middle of a system. One half turns the picture into a short list of numbers. The other half takes the list and tries to rebuild the picture. Train the pair together and there is only one way to succeed: the short list has to carry what the picture was made of.

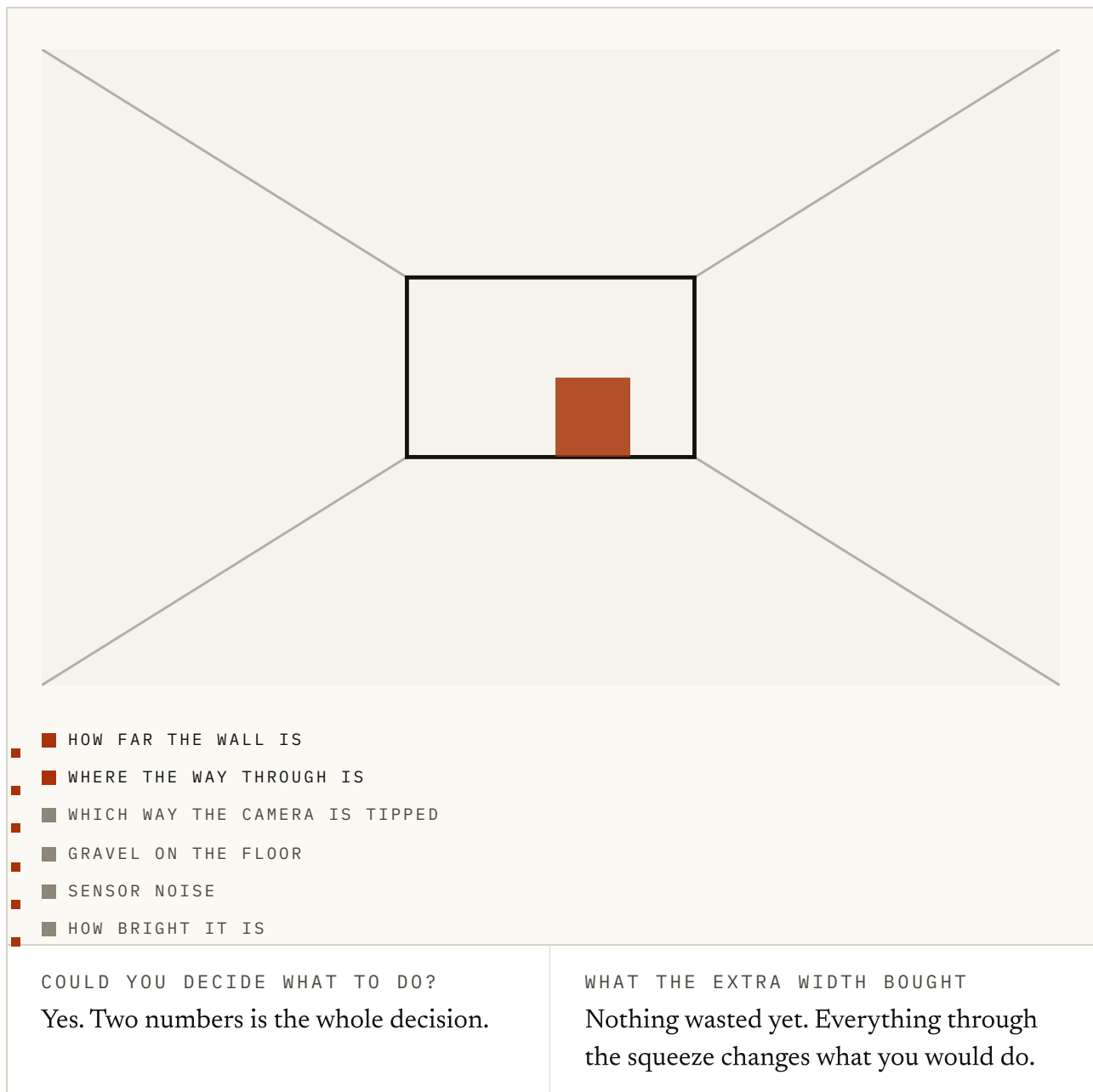


FIG. 4.2 This is that narrow opening with a dial on it. The scene is generated from six numbers and the squeeze keeps the first few. Slide the width up from zero and watch the two readouts underneath. Two numbers in, the decision is settled. The rest is gravel and sensor noise, real but beside the point.

The figure is rigged: the two factors the decision turns on sit in its first two slots. A learned system gets no such ordering for free.

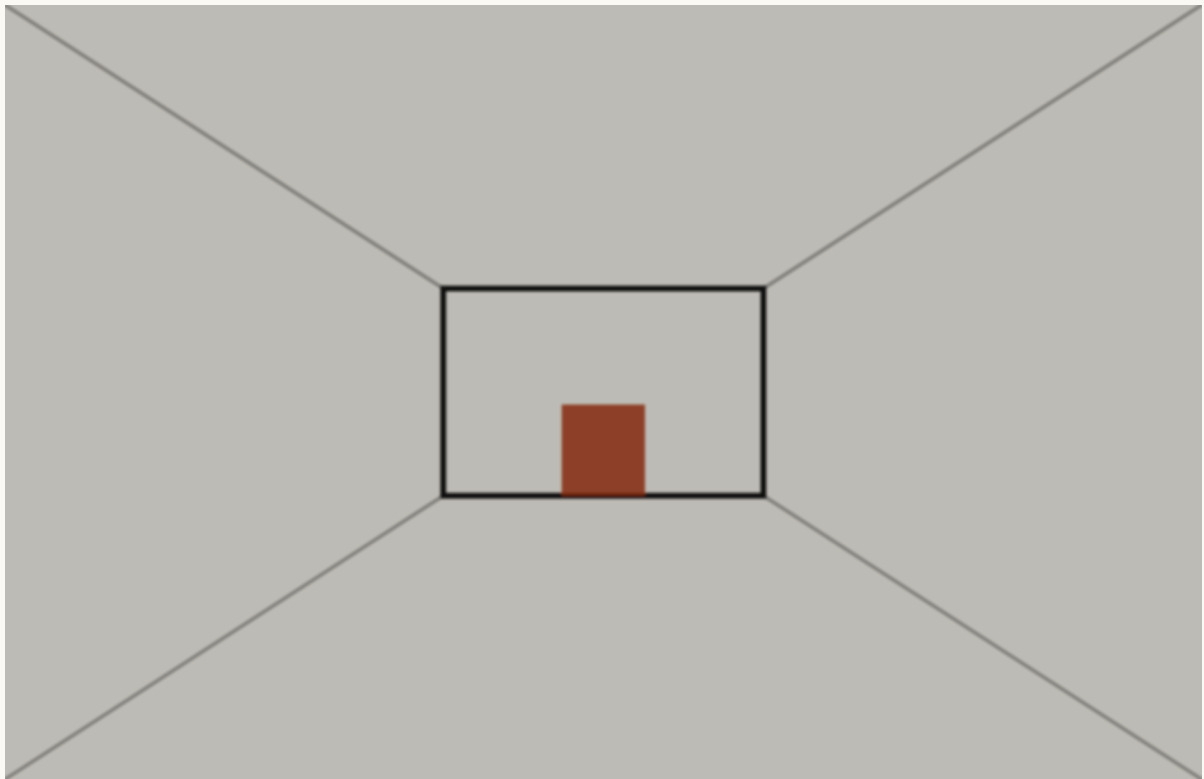
Geoffrey Hinton and Ruslan Salakhutdinov made the case for the squeeze in 2006, in *Reducing the Dimensionality of Data with Neural Networks*. Hinton taught in Toronto and had stuck with neural nets through the lean years, when most of the

field had moved on. Salakhutdinov was his student. They put the paper in *Science* rather than a machine learning journal, which tells you who they wanted to win over.

The two halves have names, and every paper uses them, so let's get them down now. The one that takes the picture in is the *encoder*, and the one that rebuilds it is the *decoder*. The narrow bit between them is the *bottleneck*. The pair together is an *autoencoder*: it encodes its own input and decodes it again.

Everything since has been that argument with better kit. But a bottleneck creates pressure without naming the answer. Many equally short lists of numbers can rebuild the same pictures, and nothing makes one number mean distance and another mean lighting. You need a bias for that, in the architecture, the data or the task.

Irina Higgins and colleagues at DeepMind, Google's London lab, made the case for such a bias in 2016. The paper was *Early Visual Concept Learning with Unsupervised Deep Learning*. A good representation, they argued, is one whose directions each mean one thing: turn one number and only the lighting changes.



THE ROOM, REDRAWN FROM THE TWO NUMBERS

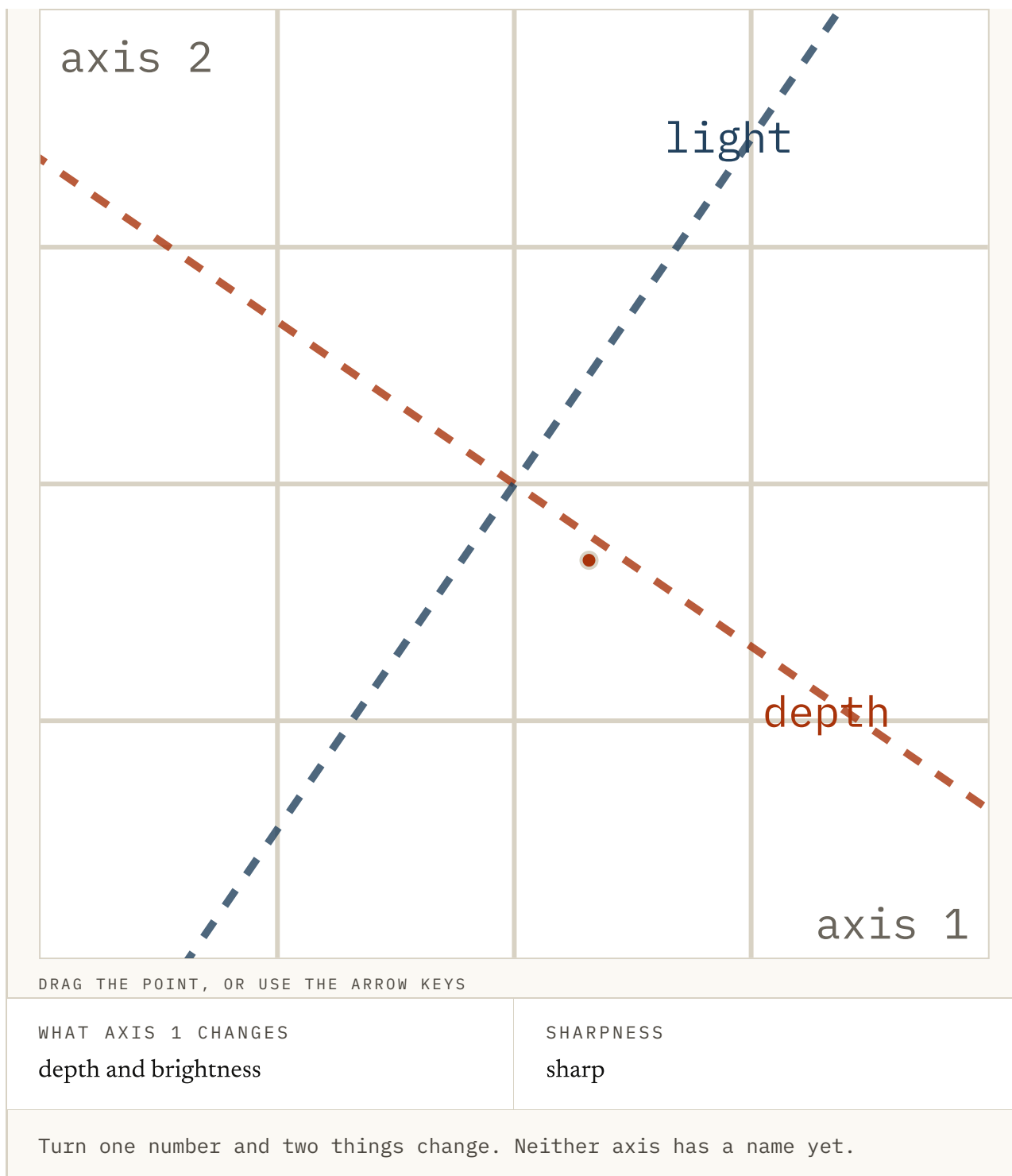


FIG. 4.3 This is Higgins's test, with the dial from the next paragraph added. Leave the pressure low and drag the point around the square: the wall moves and the light changes together, so neither axis has a name. Now turn the pressure up and drag again. One direction moves the wall and the other changes the light, and the room has gone soft. Everything here is illustrative.

Their *beta-VAE* a year later was an autoencoder of this shape with one dial added. The dial (the beta in the name) turns the pressure on the bottleneck up, and past one the axes start lining up with nameable things, the turn of a face or the size of a

shape. The same paper shows the cost, which you just saw: press harder and the pictures go blurry.

By 2018 the squeeze was inside a working agent. David Ha and Jürgen Schmidhuber published *World Models* that year, an agent that learned a driving game through a model of it. Every frame was crushed to thirty-two numbers, and the parts that predicted and chose never saw a pixel.

The same year Danijar Hafner and colleagues built PlaNet, with Ha among the authors. The paper was *Learning Latent Dynamics for Planning from Pixels*. It squeezed raw pixels the same way, then planned inside what came out, and never rebuilt a frame to decide anything. At decision time the short list was all it had, the bolder of the two designs.

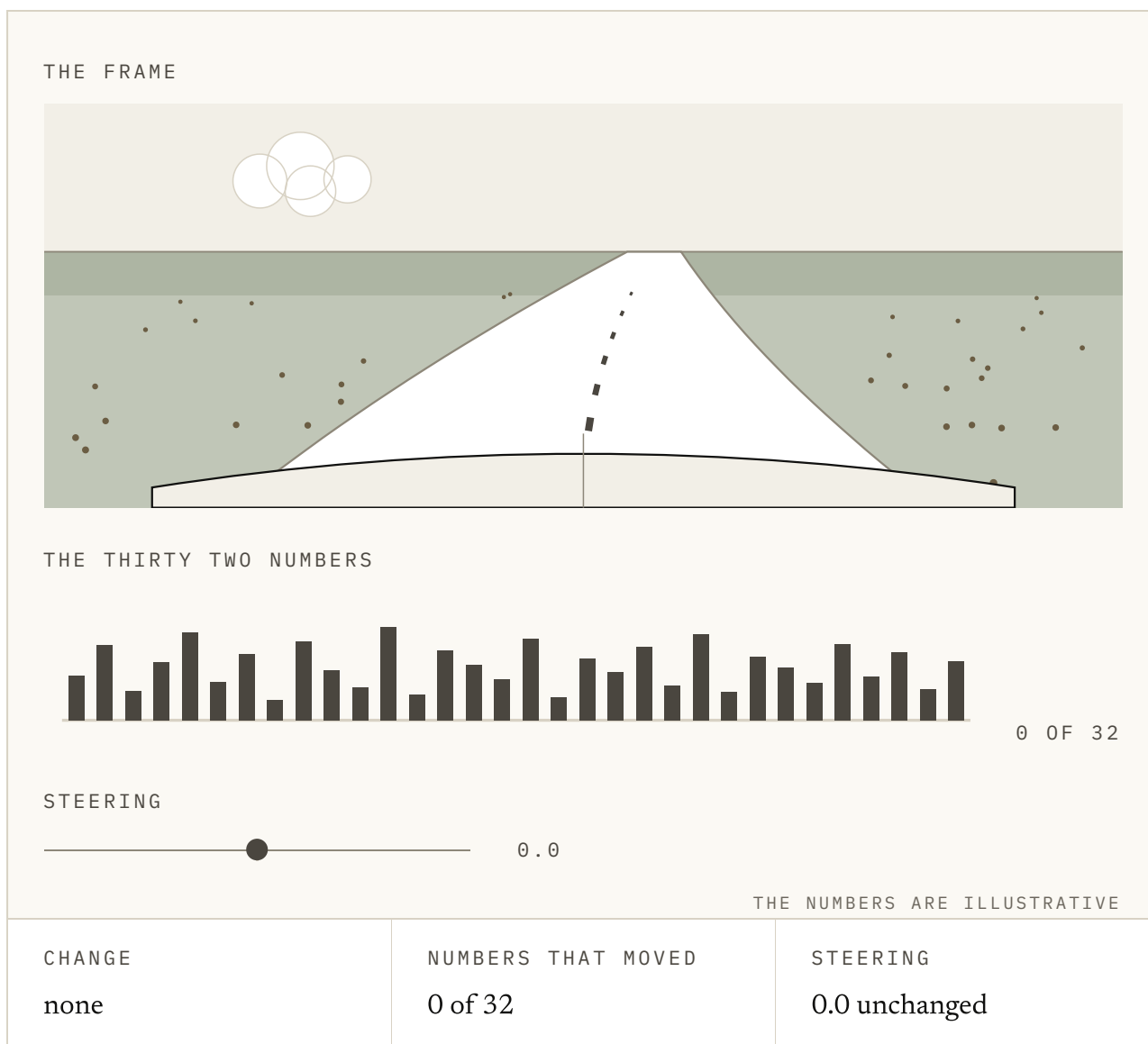


FIG. 4.4 One frame from the driving game, and the thirty-two numbers it was crushed to. Press a change and watch which of the numbers move and whether the steering moves with them. Move the bend or the car and several do. Reshuffle the gravel or reshape the cloud and the picture is plainly different while not one number moves, so nothing after the squeeze ever hears about it. The numbers are illustrative.

Then in 2019 Francesco Locatello and colleagues closed the question Higgins had opened. Pulling the hidden factors apart, one per axis, is called disentanglement. With no labels and no built-in bias, they proved, it cannot be done. For any squeeze whose axes mean something, another draws the same pictures just as well with axes that mean nothing.

They then trained thousands of models, and which axes you got came down mostly to the random seed. The paper was *Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations*. It took the best paper award at the ICML conference that year. A lot of latent-space writing has not caught up with it.

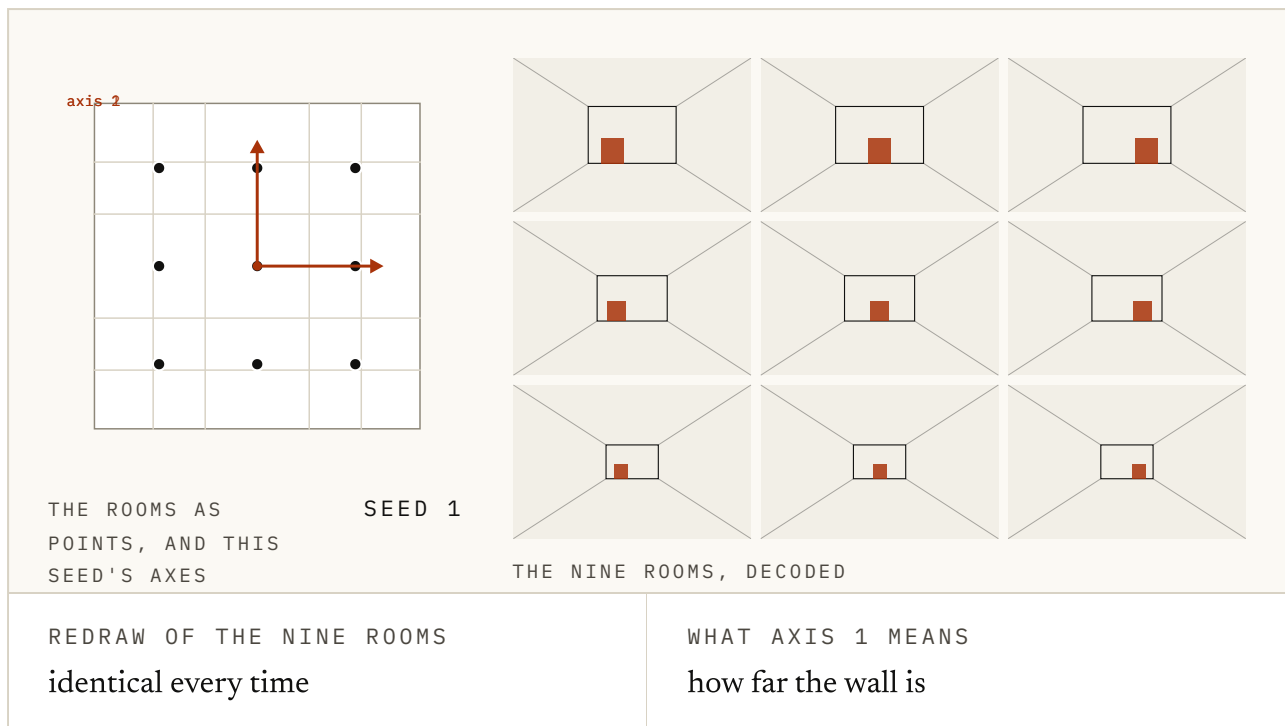


FIG. 4.5 Press Retrain a few times. Each press is a fresh random seed, and each one hands you a different pair of axes over the same rooms. Read the two lines underneath: the pictures come back the same every time, and what the first axis means does not. That is Locatello's result, and it is also why Figure 4.2 had to be rigged. Illustrative.

Look at who was unaffected. Ha and Schmidhuber's agent never needed its thirty-two numbers to mean anything a person could name, and neither did PlaNet's planner. The squeeze carried on, because the agents built on it had only ever asked that it keep what the next step turns on.

A place, not a list

Those numbers are coordinates. In a latent space that training has shaped well, they name a point with useful neighbours, directions and distances. Nothing guarantees that; a plain autoencoder can learn a twisted lookup table instead.

Once it has been shaped, you can do things there that make no sense in pixels. You can ask what is nearby, move in one direction and watch the world change smoothly in one way, or take two rooms and stand between them. That last one is the figure below.

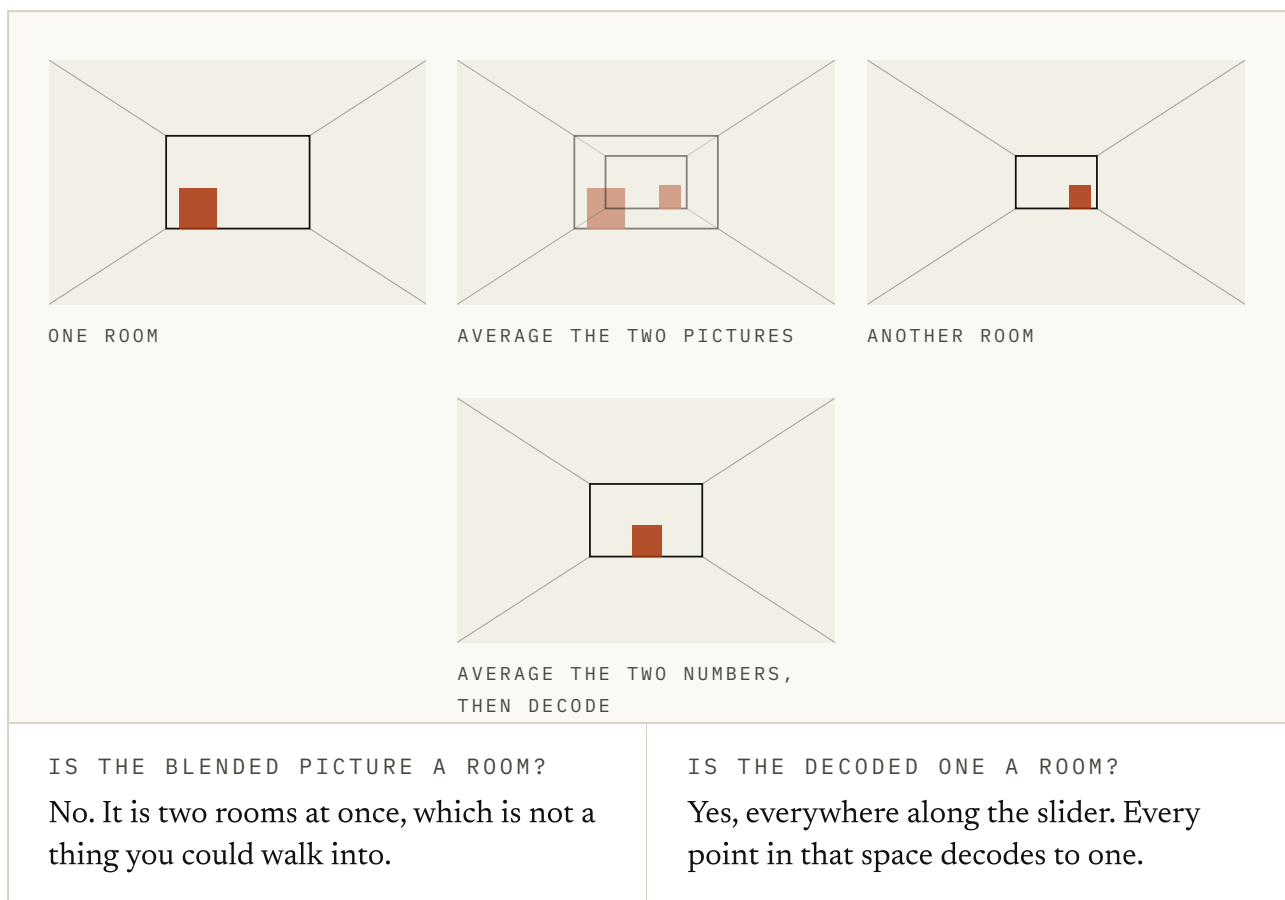


FIG. 4.6 Both rows are real output from this constructed decoder. Drag the blend slider and watch the two middle panels. Averaging two pictures gives you both rooms at once, faintly. Averaging the two coordinates and decoding stays among the rooms the decoder has learned to draw. A real latent space only does this if training has made the region between examples mean something.

Halfway between two pictures is a ghost, and halfway between two descriptions is a room. That matters because prediction, planning and search all move somewhere you have not been, and if every step lands on nonsense none of them work.

This is also why the noise in a *variational* autoencoder, the VAE that beta-VAE builds on, is deliberate. Diederik Kingma and Max Welling built it in 2013, in *Auto-Encoding Variational Bayes*; Kingma was Welling's doctoral student. It blurs where each picture lands so neighbouring points decode to similar things. That is what makes the space navigable, and why Ha and Schmidhuber used one as their agent's eyes five years later.

What makes one good

Being small is not the goal in itself. A single number is very small and useless. What you want is that the things which matter are easy to get at, and the things that do not are gone. Yoshua Bengio, Aaron Courville and Pascal Vincent set that down in 2013, in a survey called *Representation Learning: A Review and New Perspectives*.

The three were colleagues in Montreal. Bengio had kept a neural network lab going through the same lean years as Hinton, and would later share the Turing Award with Hinton and Yann LeCun. The survey is still the best single statement of what a learned representation is for. So let's take three tests from it, in rough order of how often they fail.

Does it keep what the future turns on? Take two situations that lead to different outcomes and land on the same point: nothing downstream can tell them apart. This is the failure that shows least, because the reconstructions can look fine while it happens. Ha and Schmidhuber's thirty-two numbers were trained only to redraw frames, and the authors flagged the risk themselves.

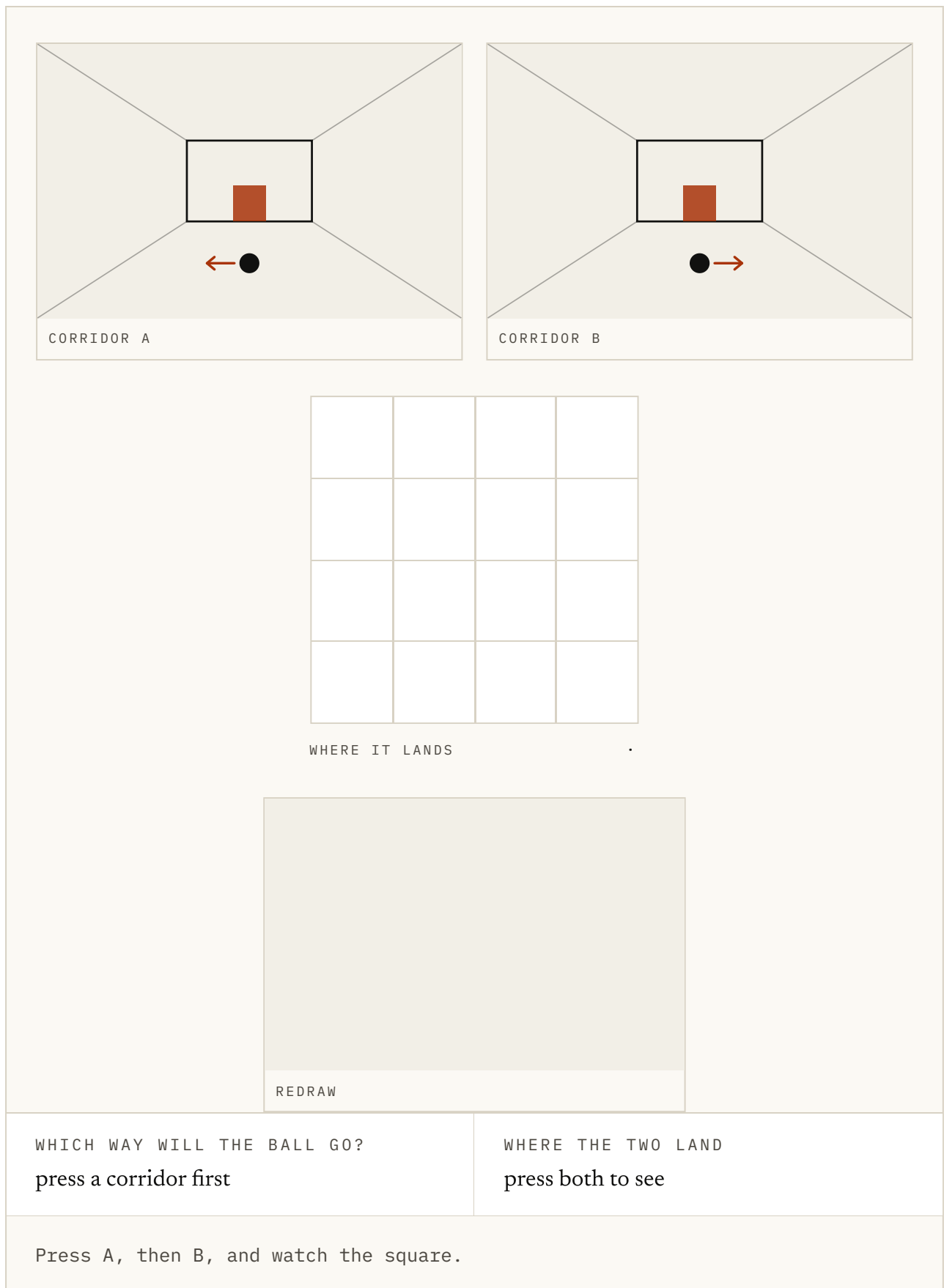


FIG. 4.7 Two corridors, and the only difference is which way the ball is rolling. Press each one in turn and watch where it lands in the square: the same point, because a squeeze scored on redrawing the frame found nothing worth keeping in a few pixels of motion. Now look at the redraw, which is fine, and at the question underneath, which cannot be answered. Flip to scoring on the next frame and the two land apart. Illustrative.

Do nearby points mean nearby things? A space where a small step can land anywhere is just a lookup table. Smoothness lets you move through it on purpose, and it is what the VAE's noise was built for. As you saw in Figure 4.6, it is also what lets you stand between two rooms.

Is it easy to predict forward? A description can be perfectly faithful and still be a nightmare to run forward in time. The reason anyone compressed was to get something they could roll forward cheaply.

What the squeeze costs

Rebuilding the picture is a proxy, and proxies drift. The test asks whether the short list can redraw what the camera saw. What you wanted to know is whether it kept what the decision turns on.

The things that take up the most pixels are usually not the things that matter. A description scored on rebuilding pixels spends itself on texture and weather, and loses the small fast object you needed to avoid. Figure 4.2 was rigged to hide this, so let's take the rigging off.

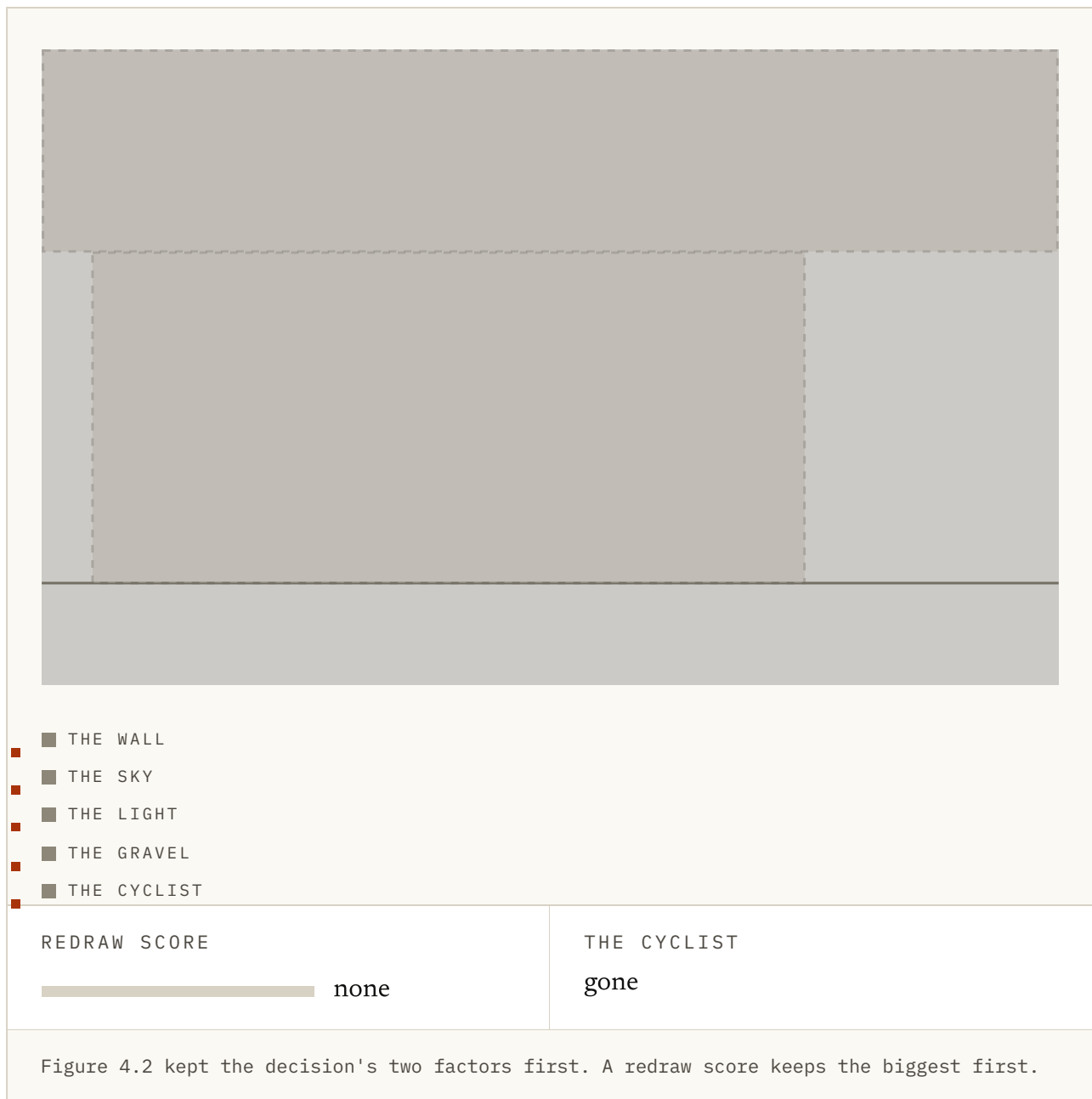


FIG. 4.8 The same kind of squeeze as Figure 4.2, but this time the slots go to whatever covers the most pixels, which is what a redraw score rewards. Slide the budget up and watch the order things come through: the wall, the sky, the light, the gravel, and only then the cyclist. Now read the two lines underneath. The redraw score is high long before the one thing you needed arrives. Illustrative.

The second cost is subtler, and I think more serious. Anything the squeeze drops is gone for good from that point on. Everything after the bottleneck works from the short list and nothing else.

That choice is made early, by a part of the system that has no idea what task is coming, and it sets the ceiling on everything after it.

That is why the argument since has been about the middle of the pipe. PlaNet trained the squeeze together with the forward model, and LeCun's JEPA line (chapter 7) asks whether to rebuild pixels at all.

Hopefully this chapter helped. There is a short quiz below if you want to check the ideas stuck. If I have got something wrong, or you know a better source than the ones I used, the About page has a corrections section and I would be glad to hear from you.

Try it

1 A camera hands you tens of thousands of numbers per frame. Deciding whether to walk forward needs two or three. What is the bottleneck for?

- a. Making the model smaller so it runs faster
- b. Forcing everything through a narrow opening, so what survives is what the pictures were actually made of
- c. Removing noise from the camera
- d. Compressing the file on disk

ANSWER b. Forcing everything through a narrow opening, so what survives is what the pictures were actually made of. Speed and file size are side effects. The reason to squeeze is that succeeding at the squeeze requires recovering the things that generated the picture in the first place.

2 Nobody hand-picks what the short list should contain. Why does it end up holding the right things anyway?

- a. The architecture has one unit per concept
- b. If the pictures were generated by a few things changing, recovering those things is the cheapest way to rebuild them
- c. The training labels say so
- d. The decoder is told the answer

ANSWER b. If the pictures were generated by a few things changing, recovering those things is the cheapest way to rebuild them. The ordering falls out of the pressure rather than being designed. That is the appeal, and also why the ordering is only roughly the one you wanted.

3 Average two pictures of two different rooms. What do you get?

- a. A room halfway between them
- b. Both rooms at once, faintly, which is not a room
- c. The first room
- d. An empty picture

ANSWER b. Both rooms at once, faintly, which is not a room. This is what pixel-space blending does. It is the clearest reason to want a space where the point between two valid things is itself valid.

4 Average the two short descriptions instead, then decode. What do you get?

- a. The same ghost
- b. A room, because every point in that space decodes to one
- c. Nothing, the numbers do not add
- d. A picture of both rooms side by side

ANSWER b. A room, because every point in that space decodes to one. Prediction, planning and search all involve moving somewhere you have not been. That only works if the places in between mean something.

5 Why is the noise in a variational autoencoder not just a nuisance?

- a. It makes training faster
- b. It hides the training data
- c. Blurring where each picture lands forces neighbouring points to decode to similar things
- d. It reduces the number of parameters

ANSWER c. Blurring where each picture lands forces neighbouring points to decode to similar things. Smoothness is the property that makes the space navigable. Without it you have a lookup table with extra steps.

6 Which of these is NOT what makes a compact description a good one?

- a. It keeps what the future turns on
- b. Nearby points mean nearby things
- c. It is as small as possible
- d. It is easy to step forward in time

ANSWER c. It is as small as possible. A single number is very small and useless. Small is a consequence of keeping the right things, never the goal on its own.

7 A description is scored on how well it can rebuild the camera image. What can go wrong?

- a. Nothing, that is exactly the right test
- b. It spends itself on whatever occupies the most pixels, which is usually not what the decision turns on
- c. It becomes too small to be useful
- d. It stops being differentiable

ANSWER b. It spends itself on whatever occupies the most pixels, which is usually not what the decision turns on. Reconstruction is a proxy. Texture and weather are most of the pixels; the small fast object you needed to avoid is not.

8 Why is the width of the bottleneck an uncomfortable place to put an important decision?

- a. It is hard to tune
- b. Everything after it works only from the short list, so whatever was dropped is gone, and it was dropped before anyone knew the task
- c. It uses too much memory
- d. It has to be chosen before training

ANSWER b. Everything after it works only from the short list, so whatever was dropped is gone, and it was dropped before anyone knew the task. The loss is not recoverable downstream. A part of the system with no idea what is coming sets the ceiling on everything after it.

FIG. 4.9 Eight questions on the squeeze and the space it produces. The last two are about the parts that are easy to nod along to and hard to keep hold of.

Everything here has happened one moment at a time. A picture goes in, a short description comes out, and nothing has moved yet.

Sources

Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations

LOCATELLO ET AL., 2019 ↗

The result that prevents a bottleneck from being magic: without inductive biases, unsupervised disentanglement is impossible in general and the axes are not identifiable.

<https://arxiv.org/abs/1811.12359>

Auto-Encoding Variational Bayes KINGMA & WELLING, 2013 ↗

The variational autoencoder. The noise it adds is the reason the space ends up navigable instead of a scatter of unrelated addresses.

<https://arxiv.org/abs/1312.6114>

Reducing the Dimensionality of Data with Neural Networks

HINTON & SALAKHUTDINOV, 2006 ↗

The bottleneck argument before it had modern machinery behind it. Paywalled.

<https://doi.org/10.1126/science.1127647>

World Models HA & SCHMIDHUBER, 2018 ↗

Every frame crushed to thirty-two numbers, and everything after that working only from those. The clearest example of the squeeze in a working agent.

<https://arxiv.org/abs/1803.10122>

Learning Latent Dynamics for Planning from Pixels (PlaNet) HAFNER ET AL., 2018 ↗

Encodes pixels to a compact state and then plans in it, without ever rebuilding a frame in order to decide anything.

<https://arxiv.org/abs/1811.04551>

Neural Discrete Representation Learning (VQ-VAE) VAN DEN OORD ET AL., 2017 ↗

What happens when the short list is forced to be a handful of discrete symbols rather than continuous numbers.

<https://arxiv.org/abs/1711.00937>

beta-VAE HIGGINS ET AL., 2017 ↗

Turn the pressure on the bottleneck up and the axes start to line up with things you can name. Also a good demonstration of what that costs.

<https://openreview.net/forum?id=Sy2fzU9gl>

Early Visual Concept Learning with Unsupervised Deep Learning

HIGGINS ET AL., 2016 ↗

The argument that a good representation is one whose directions mean something, written before it was a crowded field.

<https://arxiv.org/abs/1606.05579>

Representation Learning: A Review and New Perspectives

BENGIO, COURVILLE & VINCENT, 2013 ↗

Still the best single statement of what a learned representation is supposed to be for, and of how many of these questions were already open.

<https://arxiv.org/abs/1206.5538>

FIG. 4.10 I've ordered these for someone building on this rather than for historical completeness.