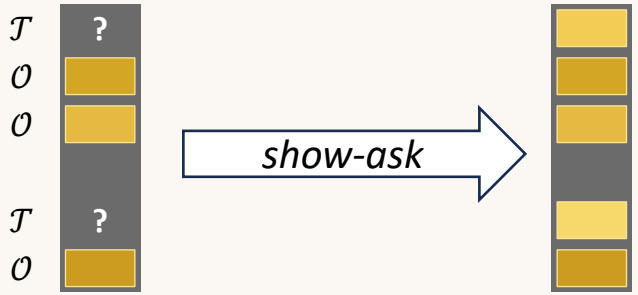
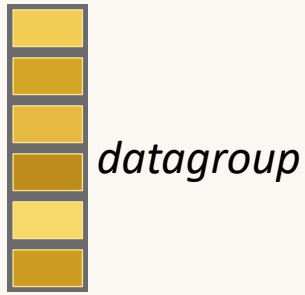
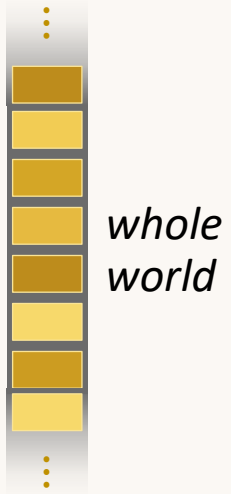
Fitting a world model

Suppose we have a dataset
$\left(x_j^{(i)}, y_j^{(i)}\right)$ for $j = 1, \ldots, m_i,\ i = 1, \ldots, n$
i.e. of datagroups $\left(\mathcal{T}^{(i)}, y^{(i)}\right)$ for $i = 1, \ldots, n$

- a dataset of photos
- a dataset of video clips
- a dataset of functions

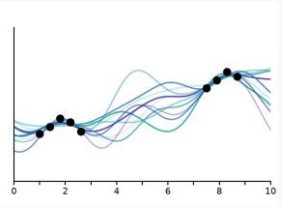






Task: fit a show-ask machine p^θ ... that is a world model
i.e. such that there is some world distribution r^θ that p^θ is consistent with.

I.e. for every query $(\mathcal{O}, \mathcal{T})$, $\underbrace{p^\theta(y^*_\mathcal{T}\,;\,\mathcal{O},\mathcal{T})}_{\text{the machine's answer}} = \underbrace{r^\theta(y^*_\mathcal{T} \mid \text{the values in } \mathcal{O})}_{\text{the world distribution's answer}}$



Fit the world distribution r^θ

then define p^θ to be its conditionals

Fit the show-ask machine p^θ

and try to nudge it to be a world model ...

$$\min_{r^\theta} \mathbb{E}_{\mathcal{T}\sim q_{\mathcal{T}}} \mathrm{KL}\left(q_{Y|\mathcal{T}} \,\|\, r^\theta_{Y|\mathcal{T}}\right)$$

$$p^\theta(y^*; \mathcal{O}, \mathcal{T}) \coloneqq r^\theta(y^*|\mathcal{O}, \mathcal{T})$$

This is conventional supervised learning, with $\mathcal{T}$ as the predictor variable. Here $q_{\mathcal{T}}$ is the predictor's distribution as seen in the dataset.

Gaussian process practice:

$$\max_{r^\theta} \mathbb{E}_{\mathcal{T},Y\sim q} \log r^\theta_{Y|\mathcal{T}}(Y)$$

- *Approximate the expectation with a single draw, which is called the "evidence"*
- *No holdout evaluation*

Learning

all the non-Bayesian parameters
(length scale for RBF_ℓ, variance β, etc.)

$$\theta^* = \mathrm{argmax}_\theta \, p_\theta(\mathcal{D})$$

a single datagroup $(\mathcal{T}, y)$

This is just MLE i.e. KL minimization using a single sample

Why on earth should this work?

In statistics, this would be called
a *mixed-effects model*
(since it’s part Bayesian part MLE)

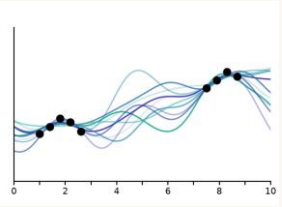
Marginal Log-likelihood

$$\log p(\boldsymbol{y} \mid \mathbf{X}, \theta)$$

$$= \log \int p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{f}|\mathbf{X}, \theta)\; d\boldsymbol{f}$$

$$= -\frac{1}{2}\boldsymbol{y}^\intercal (k_\theta(\mathbf{X},\mathbf{X}) + \beta^{-1}\mathbf{I})^{-1}\boldsymbol{y} - \frac{1}{2}\log\det(k_\theta(\mathbf{X},\mathbf{X}) + \beta^{-1}\mathbf{I}) - \frac{N}{2}\log 2\pi$$

11



Fit the world distribution r^θ

then define p^θ to be its conditionals

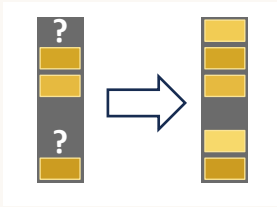
$$\min_{r^\theta} \mathbb{E}_{\mathcal{T}\sim q_{\mathcal{T}}}\ \mathrm{KL}\left(q_{Y|\mathcal{T}} \,\|\, r^\theta_{Y|\mathcal{T}}\right)$$

$$p^\theta(y^*;\mathcal{O},\mathcal{T}) \coloneqq r^\theta(y^*|\mathcal{O},\mathcal{T})$$

Fit the show-ask machine p^θ

How?

and try to nudge it to be a world model ...



... using a critic

minimize $\mathbb{E}_{\mathcal{T}\sim q_{\mathcal{T}}}\ \mathrm{KL}\left(q_{Y|\mathcal{T}} \,\|\, r^\theta_{Y|\mathcal{T}}\right)$

over p^θ, r^θ

such that $\forall \mathcal{O},\mathcal{T},\ p^\theta_{Y^*;\mathcal{O},\mathcal{T}} = r^\theta_{Y^*|\mathcal{O},\mathcal{T}}$

No one has done this AFAIK

... by averaging

$$\min_{p^\theta} \mathbb{E}_{(\mathcal{O},\mathcal{T})\sim q_{\mathcal{O},\mathcal{T}}} \mathrm{KL}\left(q_{Y^*|\mathcal{O},\mathcal{T}} \,\|\, p^\theta_{Y^*;\mathcal{O},\mathcal{T}}\right)$$

If we had infinite capacity and perfect training, then each $p^\theta_{Y^*;\mathcal{O},\mathcal{T}}$ would learn to match the truth q, so p^θ would be a world model.

With finite capacity, we'll fudge it: we'll design corruptions $(\mathcal{O},\mathcal{T}) \sim q_{\mathcal{O},\mathcal{T}}$ and train to do well on those, on average.

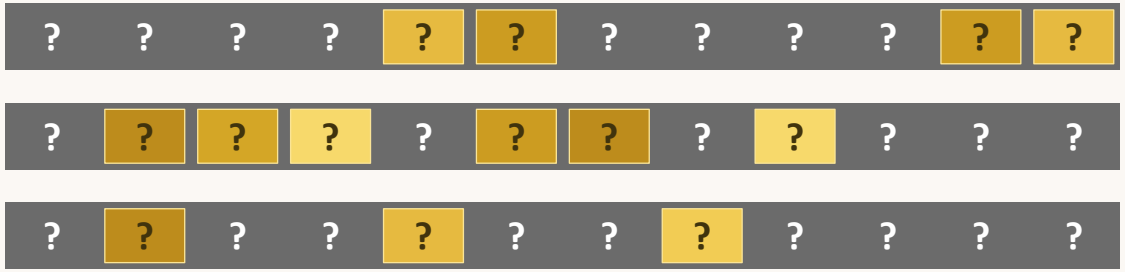
When we use the 'averaging fudge' we have to choose a distribution of queries. Our machine will learn to be good at answering those queries.

Our goal is $\min_{p^\theta} \mathbb{E}_{(\mathcal{O},\mathcal{T})\sim q_{\mathcal{O},\mathcal{T}}} \mathrm{KL}\left(q_{Y^*|\mathcal{O},\mathcal{T}} \,\|\, p^\theta_{Y^*;\mathcal{O},\mathcal{T}}\right)$

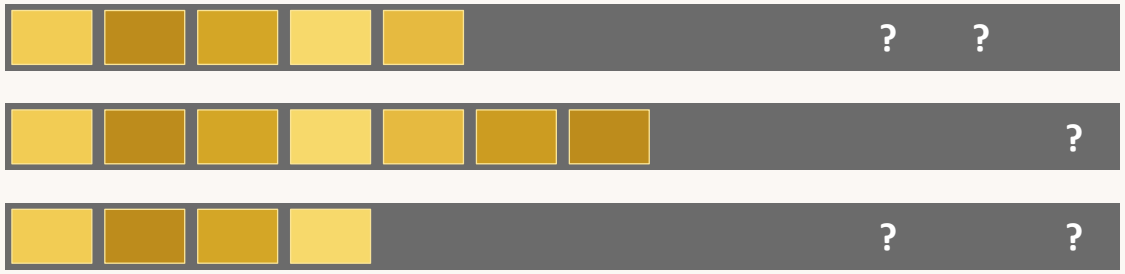
Equivalently, $\max_{p^\theta} \mathbb{E}_{(\mathcal{O},\mathcal{T})\sim q_{\mathcal{O},\mathcal{T}}} \mathbb{E}_{Y^*\sim q_{Y^*|\mathcal{O},\mathcal{T}}} \log p^\theta_{Y^*;\mathcal{O},\mathcal{T}}\,(Y^*)$

} we'll use this as a loss function for training; we might as well use it for holdout evaluation too

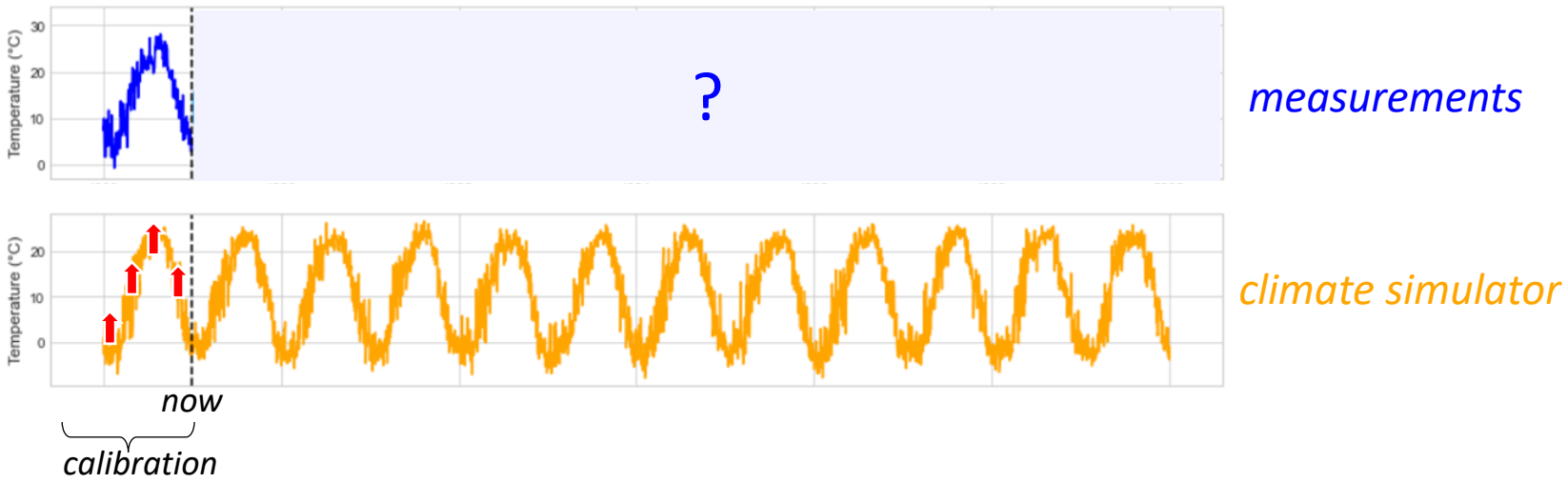
What $q_{\mathcal{O},\mathcal{T}}$ did the CNP paper choose?



To put more weight on extrapolation performance, what $q_{\mathcal{O},\mathcal{T}}$ would you choose?



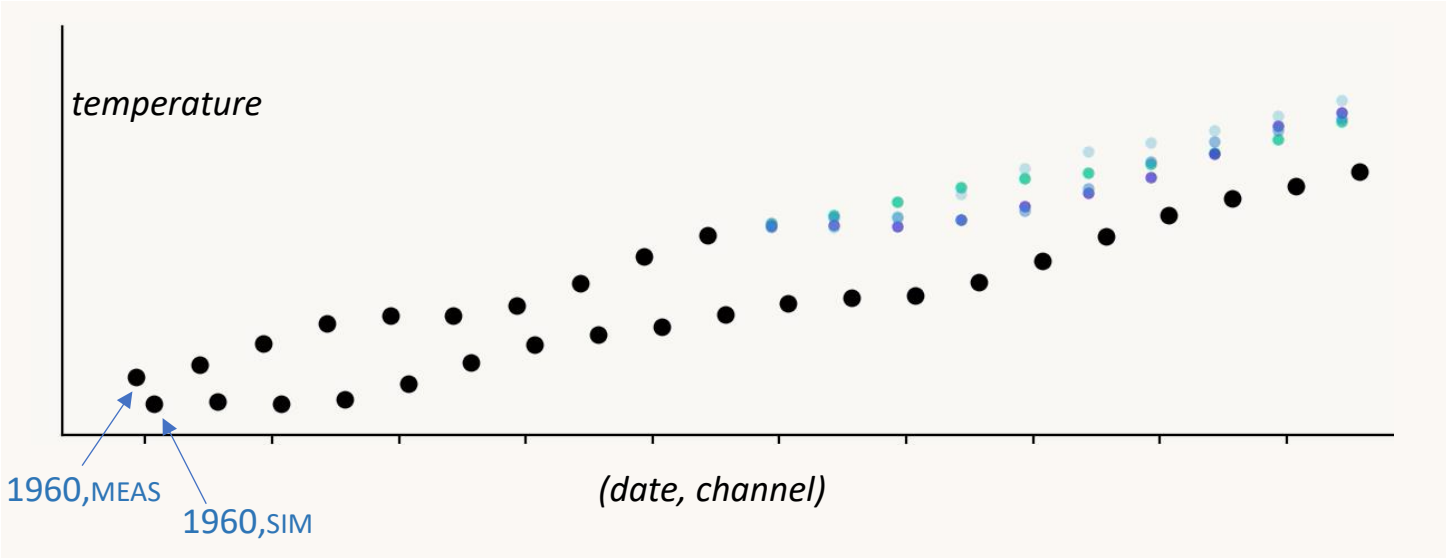
Extended application: correcting the bias in climate models [Omer Nivron 2024]



- Climate simulators are good at modelling long-term trends, e.g. the impact of changing CO_2 emissions
- But they disagree systematically with measurements in the short term
- Can we learn a **bias correction** ⬆ from the calibration period, and use it to correct the simulator's long-term forecasts?

The straightforward approach is to estimate an average offset, "measurements are 2.1°C hotter than simulation".
What might this miss?

Consider both measurement and simulation as **readouts of a single function** (date,channel) → temperature

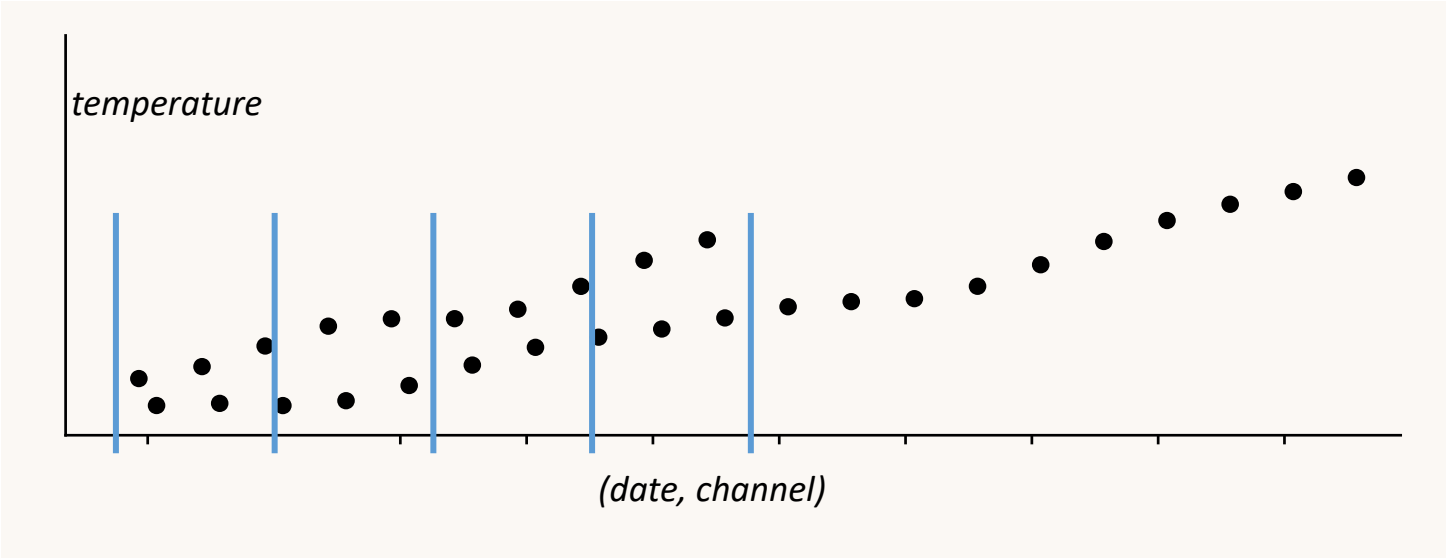


- *Across dates:*
  If we can sample coherent completions of the MEAS channel,
  we'll be able to estimate all sorts of statistics e.g. heatwave durations

- *Across dates and channels:*
  If we can learn a full joint distribution, we'll be able to complete MEAS by conditioning on both calibration data and simulated climate-change data — i.e. we can do bias correction

In other words, we want a **world model.**

Challenge: we only have one long record, not a dataset of datagroups



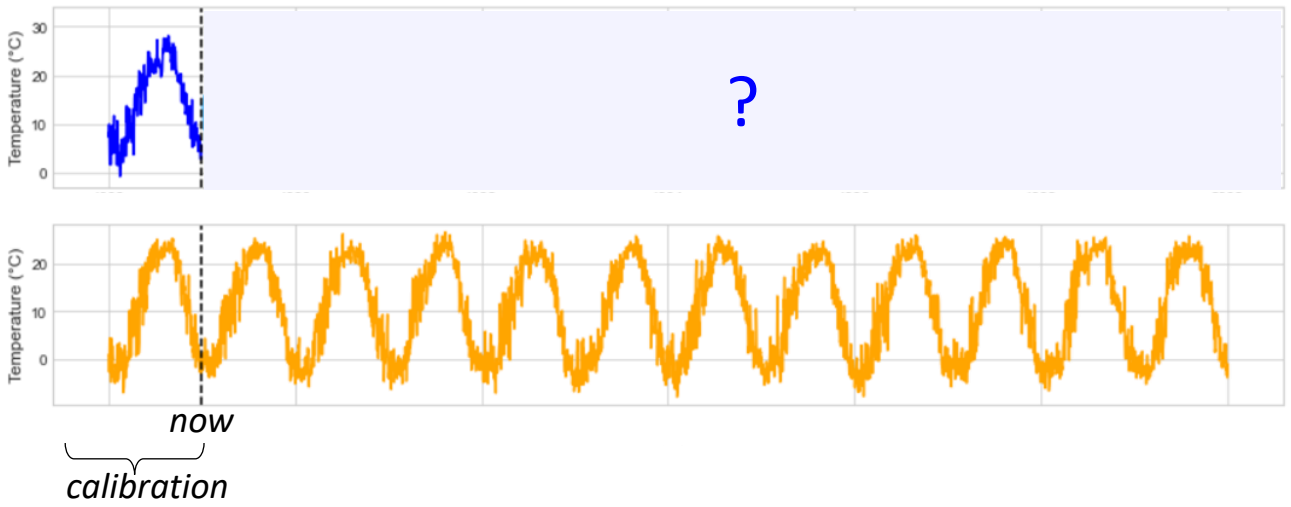
How can we fit a world model, if we don't have a dataset of many datagroups?

- Slice the climate data into windows, and treat each window as a datagroup

- This is making an **inductive assumption:**
  that there's some common law across the windows,
  which we can learn and apply to the future

  Think of a datagroup, read off at x-coordinates $\mathcal{T}$, as a latent-variable generative model $f_\theta(Z) \in \mathbb{R}^{|\mathcal{T}|}$.
  Z is what makes each datagroup different, the common law is f_θ.

Challenge: we might not learn what we're seeking to learn



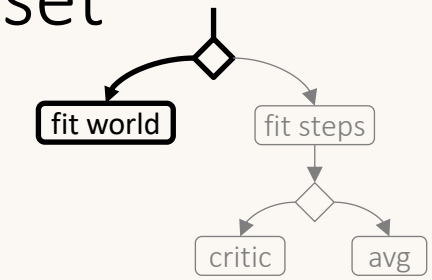
Here are two laws that we might discover while fitting:

- *CONTINUE*: continue the measurements, since future measurements are much like past measurements
- *TRANSLATE*: measurements are bias-corrected versions of simulator output

By choosing the datagroups and queries, we pick which law we learn:

If we fit on queries with a short calibration period and a long forecast period,
then *CONTINUE* will do worse (because it has no access to the climate trend), and *TRANSLATE* will do better.

A world distribution can be described by a formula—or a dataset



There are two equivalent ways to describe a world distribution $r_{Y|\mathcal{T}}$:

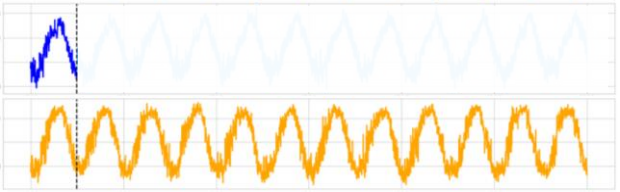
- *algebraically* — via a concrete formula for $r_{Y|\mathcal{T}}$ e.g. a Gaussian-process kernel
- *empirically* — via a dataset of many datagroups to which a distribution $r^{\theta}_{Y|\mathcal{T}}$ can be fitted

Indeed, if we have an algebraic kernel, we can turn it into a dataset by simulating it.

There's no point learning Bayesian methods, when sampling+learning does the same job easier!

Example: climate bias correction. We want to learn the *TRANSLATE* law, not the *CONTINUE* law.

Empirically. Curate a dataset of datagroups, and fit a model. If we've chosen the datagroups carefully, the fitted show-ask machine should embody a *TRANSLATE* law rather than a *CONTINUE* law.

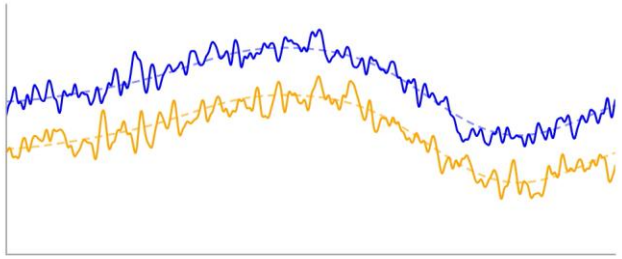


Algebraically. $$Y_{t,\text{channel}} = \begin{cases} C_t + W_t + B & \text{if channel} = \text{MEAS} \\ C_t + W'_t & \text{if channel} = \text{SIM} \end{cases}$$

Slowly-varying climate trend $C_t \sim \text{GP}(\text{RBF}_{30\text{years}})$

Rapidly-varying weather patterns $W_t, W'_t \sim \text{GP}(\text{RBF}_{3\text{days}})$

The bias correction we want to learn, $B \sim N(0,1)$. The posterior for B, on a datagroup, is the bias correction for that datagroup.



Because of the climate trend, *CONTINUE* will be a bad model over longer forecast periods.
Because the trend is shared, *TRANSLATE* will be a better model.

Linear Model of Co-regionalisation

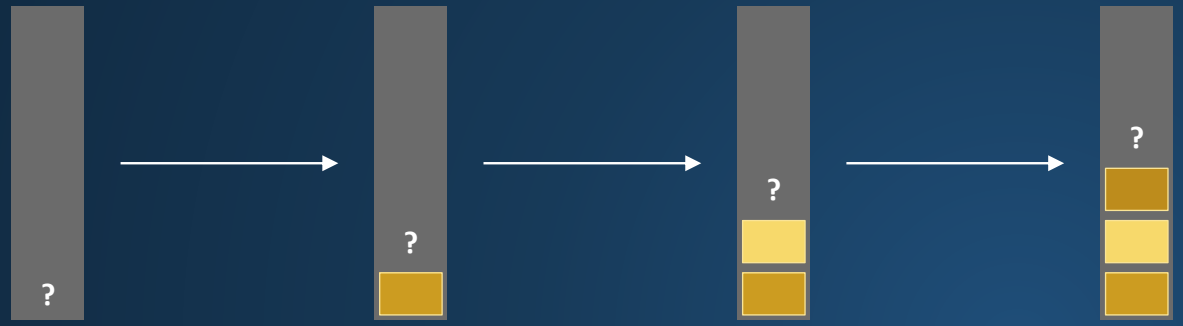
$$\boldsymbol{y}_i = \mathbf{W}\,\boldsymbol{f}_i + \epsilon$$
$$f_\ell \sim \mathrm{GP}(\mathbf{0}, \mathbf{K}_\ell)$$
$$\mathrm{Cov}(y_{iq}, y_{jq'}) = \sum_{\ell=1}^{L} k_\ell(x_i, x_j)\, w_{q\ell} w_{q'\ell}$$

- L latent functions $f_1, \ldots, f_L$
- $\mathbf{W} \in \mathbb{R}^{D \times L}$ loading matrix
- Gaussians are closed under linear operations

12

In my notation,
$\boldsymbol{y}_i$ is $(Y_{t_i,\mathrm{MEAS}}, Y_{t_i,\mathrm{SIM}})$
$f_1, \ldots, f_L$ are the latents (C, W, W', B)
$\mathbf{W}$ says which latents are used for which channel

Simple language-model generation, and diffusion models, are show-ask machines.
But we're not bothered about consistency, because we control the order of queries.



In other settings, we want to be able to answer **random-access show-ask queries.**
For consistency, we want our model to behave <u>as if</u> it had a pre-ordained world.

