Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

UNIVERSITÄT BONN

FOR 2131

IMPRS-ESM International Max Planck Research School on Earth System Modelling

cen

09.01.2019

Deep assimilation: Adversarial variational Bayes for non-linear data assimilation

Tobias Sebastian Finn

What is the connection between my PhD thesis and machine learning?

Writing his PhD thesis

?

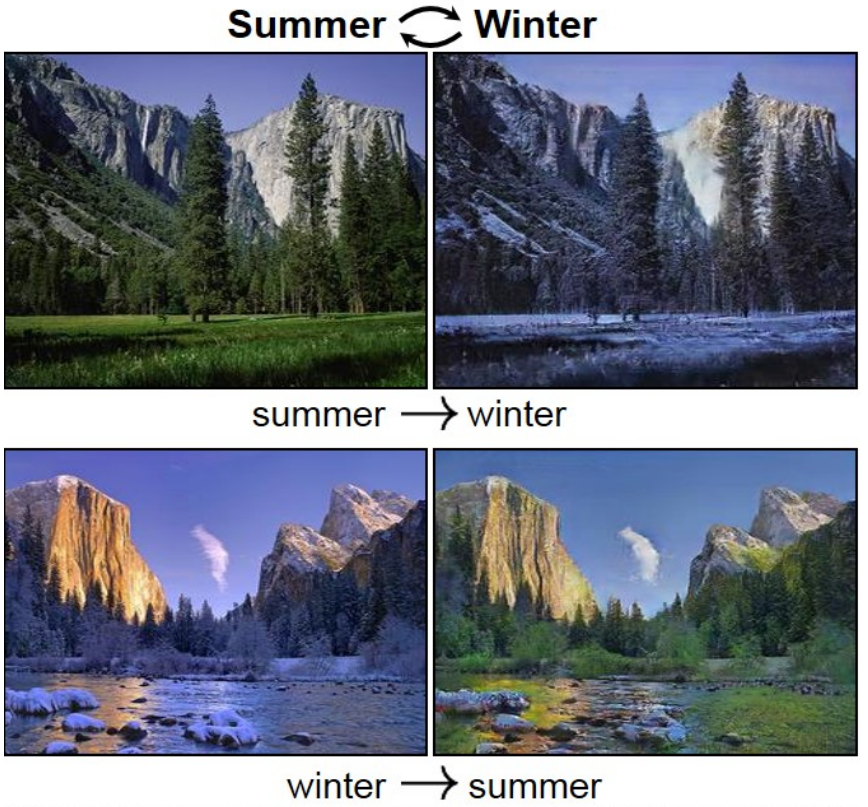
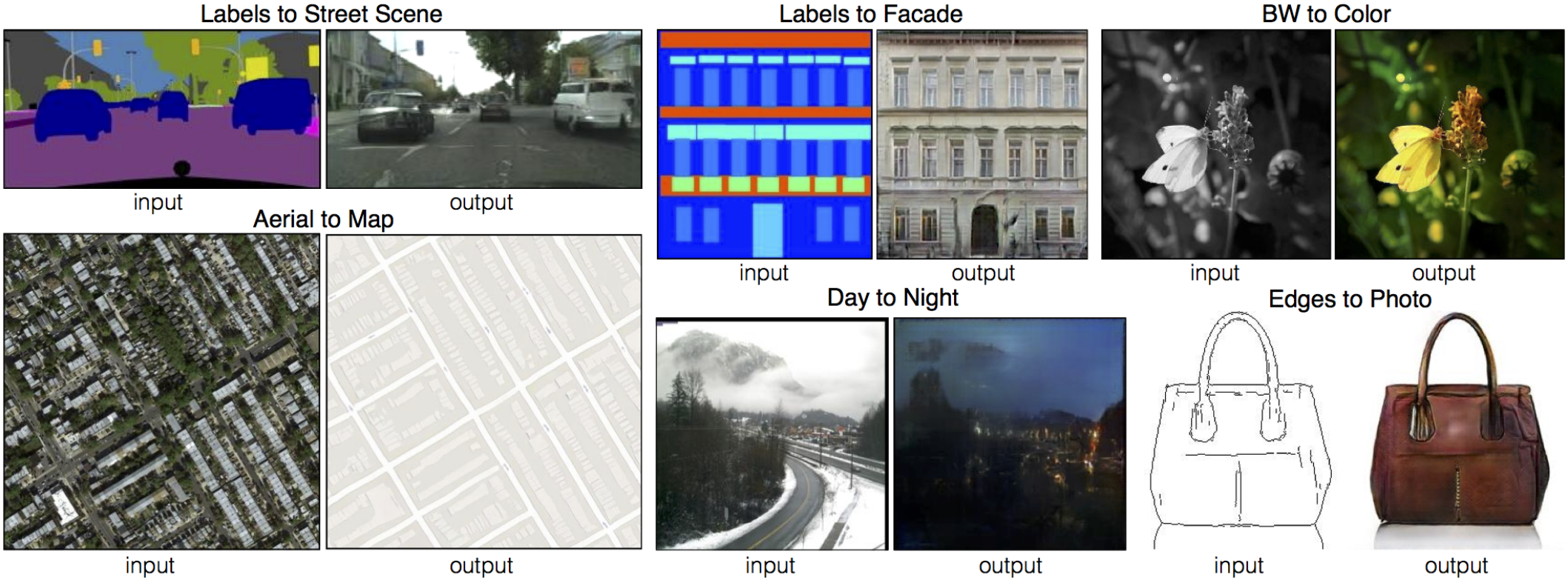


Image-to-Image translation

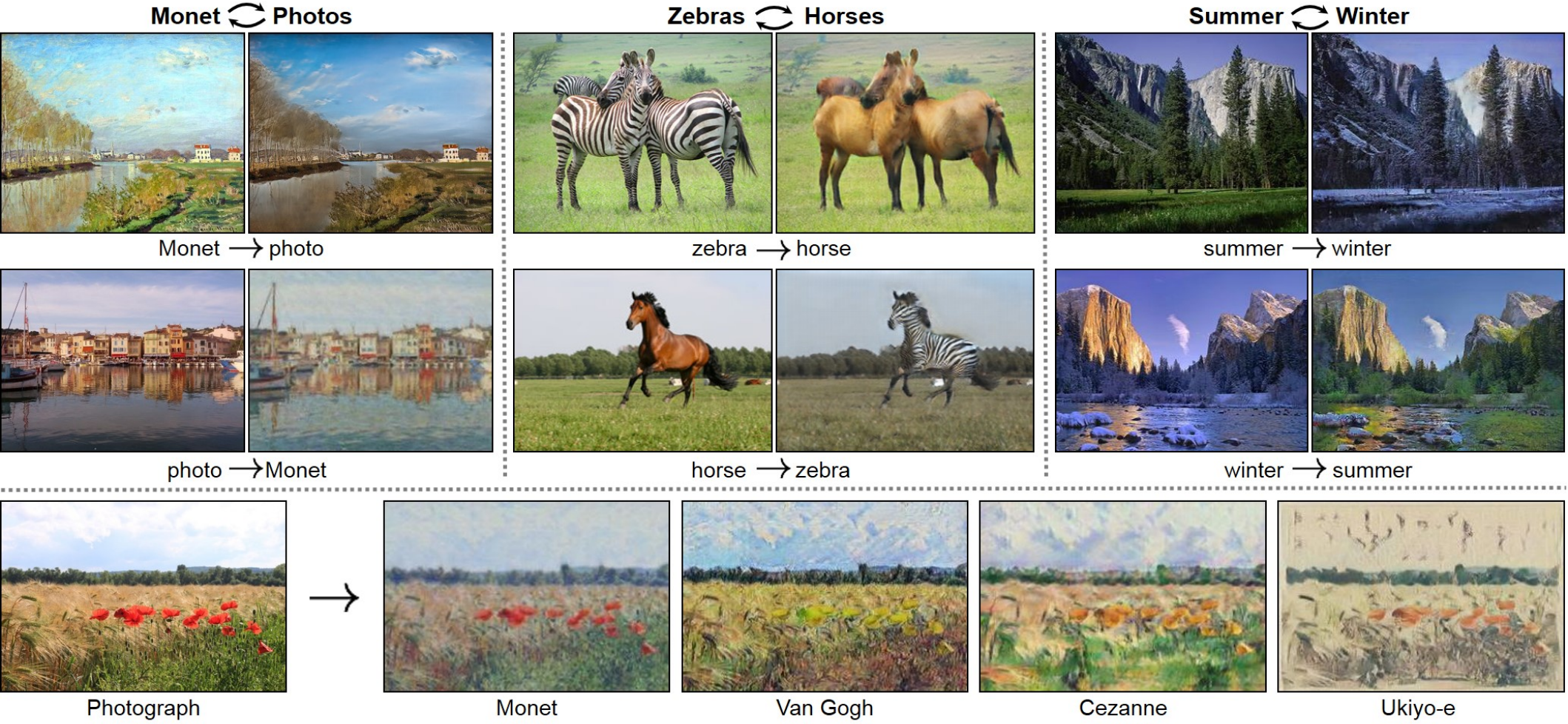
Sources:
https://junyanz.github.io/CycleGAN/
https://magazin.spiegel.de/SP/2018/52/161498484/index.html

How can we use machine learning for data assimilation?



Source: https://phillipi.github.io/pix2pix/ (pix2pix)

How can we use machine learning for data assimilation?



Source: https://junyanz.github.io/CycleGAN/ (CycleGAN)

How can we use machine learning for data assimilation?

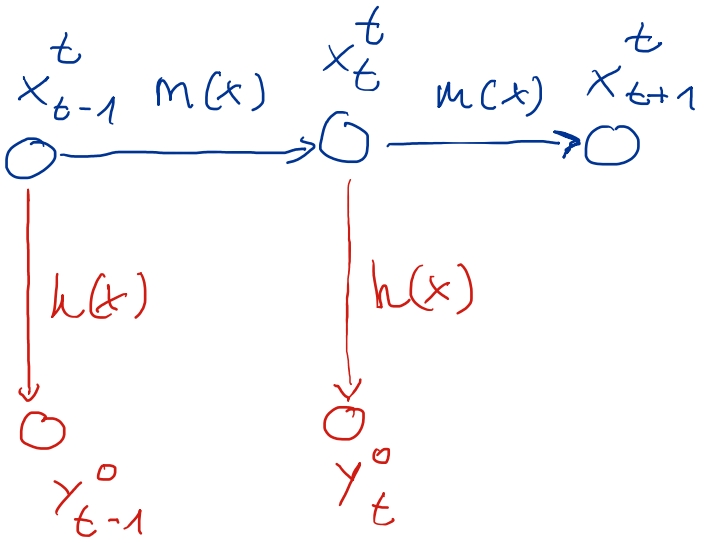
Figure 3: Street scene image translation results. For each pair, left is input and right is the translated image

Source: https://arxiv.org/abs/1703.00848 (UNIT)

What's new?

- First application + theoretical foundation of deep learning for non-linear data assimilation
- No assumption about state space
  $\rightarrow$ flexible distributions + non-Gaussian assimilation
- Efficient and fast inference
- Unclear: assimilation of not yet used observables (e.g. cloud camera, radar)
- Unclear: unification of data assimilation algorithms

Data assimilation problem



How to get x^t_t based on observations $y^o_{0:t}$?

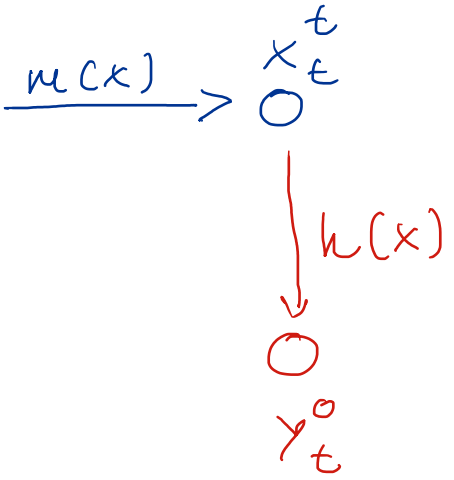
$$y^o_t = h(x^t_t) + \varepsilon^o \qquad \varepsilon^o \sim p(\varepsilon^o)$$

Assumption: Markov-1 process, we know $m(x)$ and $h(x)$ up to errors ε.

↳ look at time t → update problem → simplifies to static problem

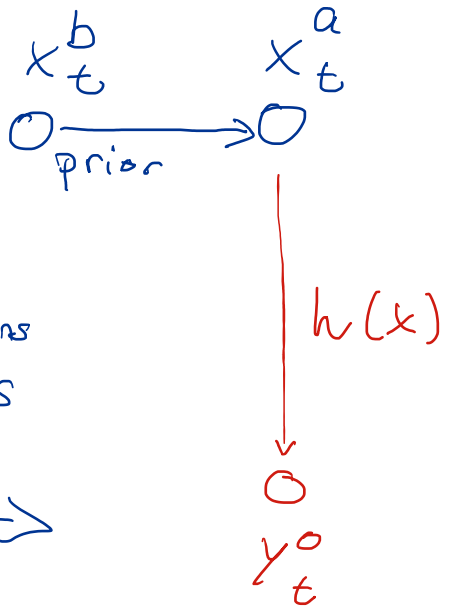
Latent variable model for data assimilation

Let's look at time t



We can only estimate an approximation to x^t: x^a.
Based on an old analysis and model $u(x)$, we have a prior x_t^b for x_t^a. This prior contains all information from previous observations.

Static



We can estimate x_t^a with Bayes' formula:

$$p(x_t^a \mid y_{0:t}^o) = \frac{p(y_t^o \mid x_t^a)\, p(x_t^a \mid y_{0:t}^o)}{p(y_{0:t}^o)} = \frac{p(y_t^o \mid x_t^a) \cdot p(x_t^b)}{p(y_{0:t}^o)}$$

$p(y_{0:t}^o)$ intractable!

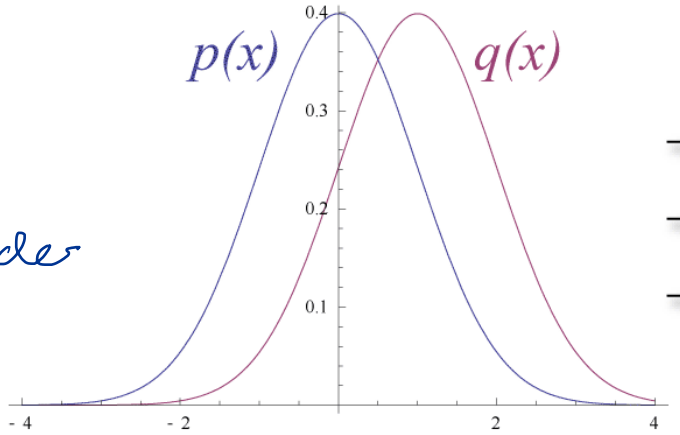
Let's approximate with $q_\phi(x_t^a)$

↳ we want $q_\phi(x_t^a) \approx p(x_t^a \mid y_{0:t}^o)$ → Kullback-Leibler Divergence

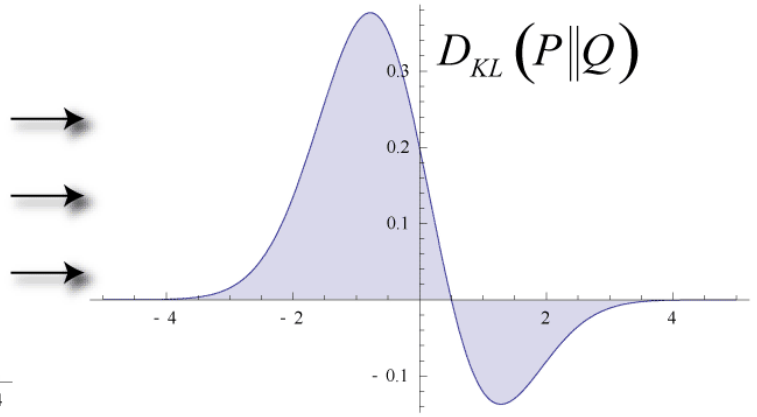
Kullback-Leibler divergence

$$\mathrm{KL}(q(x)||p(x)) = -\sum_{x\in\mathcal{X}} q(x)\log\left(\frac{q(x)}{p(x)}\right) = \mathbb{E}_{q(x)}\log\left(\frac{q(x)}{p(x)}\right)$$

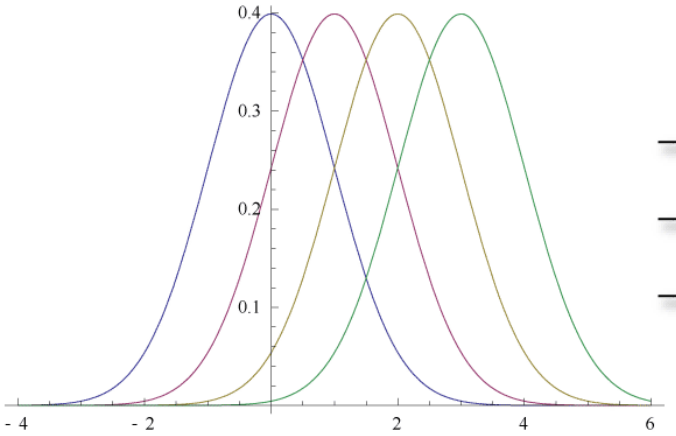
Integral under KL-area

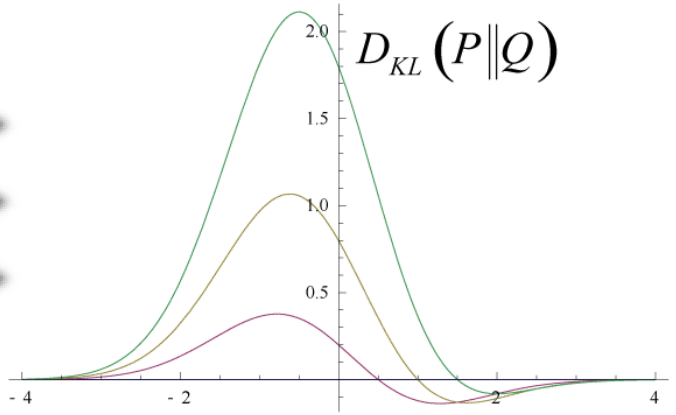


Original Gaussian PDF's



KL Area to be Integrated

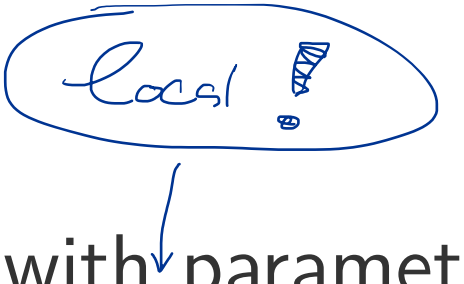




T. Nathan Mundhenk, PhD thesis appendix C.

Variational inference

- Minimize $KL(q_{\phi_t}(\mathbf{x}_t^a)||p(\mathbf{x}_t^a|\mathbf{y}_{0:t}^o))$



- Search for approximated $q_{\phi_t}(\mathbf{x}_t^a)$ with parameters ϕ_t from variational family, e.g. Gaussian assumption
- $p(\mathbf{x}_t^a|\mathbf{y}_{0:t}^o)$ intractable, we cannot optimize!
- We need to circumvent this problem

  $\rightarrow$ rephrase the minimization into a simpler one!

  ↳ like done in first two slides!

Overview: Blei, D. M., A. Kucukelbir, and J. D. McAuliffe, 2017: Variational Inference: A Review for Statisticians.

Evidence lower bound (ELBO)

use of Bayes + rearrange

$$\mathrm{KL}(q_{\phi_t}(\mathbf{x}_t^a)||p(\mathbf{x}_t^a|\mathbf{y}_{o:t}^o)) \geq KL(q_{\phi_t}(\mathbf{x}_t^a)||p(\mathbf{x}_t^b)) - \mathbb{E}_{\mathbf{x}\sim q_{\phi_t}(\mathbf{x}_t^a)} \log p(\mathbf{y}_t^o|\mathbf{x})$$

Similarity to first guess + Reconstruction loss (e.g. MSE)

- Can be optimized by minimization
- Similar loss to standard data assimilation losses
- Computational expensive (can be slower than 4dVar!)
- Every new analysis has to be optimized

↳ online learning like classical DA

only local parameters ϕ_t

e.g. Vrettas, M. D., Opper, M., & Cornford, D. (2015). Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations.

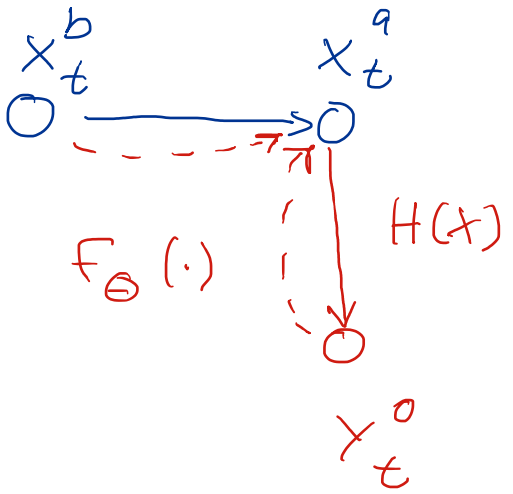
Amortized inference

We want to get global fixed parameters for inference

$q_{\phi_t}(x_t^a)$ —(make time-dependent by using conditional probabilities)→ $q_\theta(x_t^a \mid y_{0:t}^o) \rightarrow q_\theta(x_t^a \mid y_t^o, x_t^b)$

$\Leftrightarrow \quad x_t^a = f_\theta(y_t^o, x_t^b, \varepsilon_t), \quad \varepsilon_t \sim p(\varepsilon)$

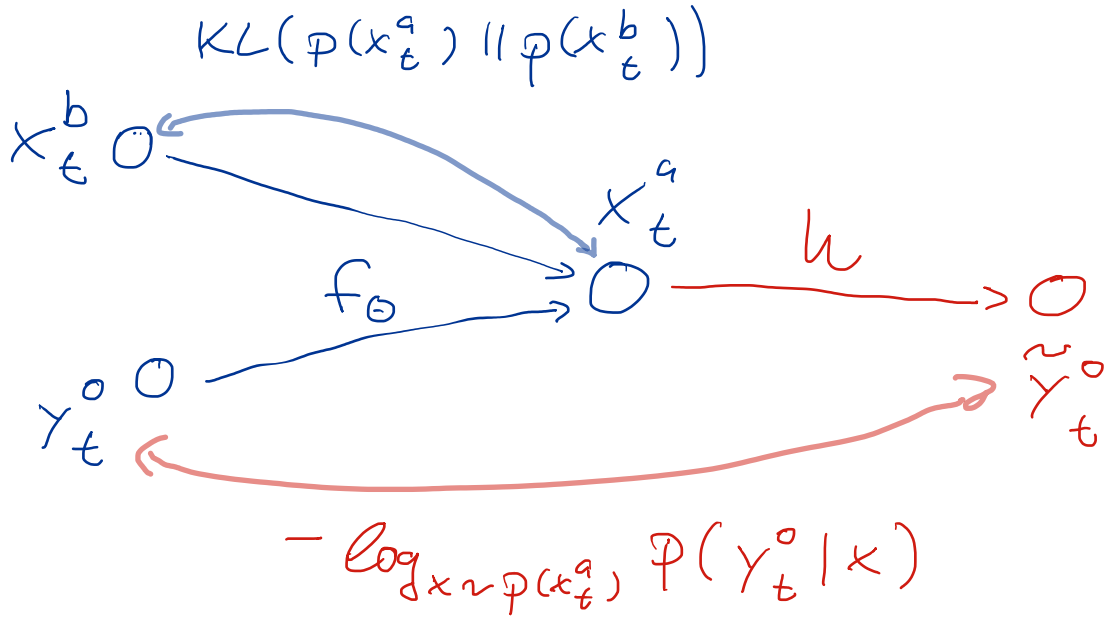
$\Rightarrow$



Roweis, S., and Z. Ghahramani, 1999: A Unifying Review of Linear Gaussian Models

Gershman, S., and N. Goodman, 2014: Amortized Inference in Probabilistic Reasoning

Variational autoencoder (VAE)



$\Rightarrow$ normal assumption: $p(x_t^a \mid y_t^o, x_t^b) = \mathcal{N}\left[f_\theta^\mu(y_t^o, x_t^b), f_\theta^\sigma(y_t^o, x_t^b) \right]$

↳ for backpropagation: reparametrization trick

Kingma, D. P., and M. Welling, 2013: Auto-Encoding Variational Bayes

Rezende, D. J., S. Mohamed, and D. Wierstra, 2014: Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Variational autoencoder (VAE)

- Computational expensive parts are moved to training
  → Efficient at inference!
- (L)ETKF: Linearized VAE + Gaussian assumption
  → analytical solution
- Candidate to unify
  variational (not the same) + sequential data assimilation
- Needs closed solution of Kullback-Leibler
  → Explicit modelling of prior and posterior

For connection ETKF (PPCA) to VAE:
Dai, B., Y. Wang, J. Aston, G. Hua, and D. Wipf, 2017:
Hidden Talents of the Variational Autoencoder.

Variational autoencoder (VAE)

- Computational expensive parts are moved to training
  $\rightarrow$ Efficient at inference!
- (L)ETKF: Linearized VAE + Gaussian assumption
  $\rightarrow$ analytical solution
- Candidate to unify
  variational (not the same) + sequential data assimilation
- Needs closed solution of Kullback-Leibler
  $\rightarrow$ Explicit modelling of prior and posterior 

We want implicit modelling!

Implicit representation of density ratio

- Kullback-Leibler divergence is density ratio between $q_\phi(\mathbf{x}_t^a|\mathbf{x}_t^b, \mathbf{y}_t^o)$ and $p(\mathbf{x}_t^b)$
- We can sample from $p(\mathbf{x}_t^b)$
- Approximation via classification:

  Samples from $p(\mathbf{x}_t^b)$: True

  Samples from $q_\phi(\mathbf{x}_t^a|\mathbf{x}_t^b, \mathbf{y}_t^o)$: False
- Use another NN for classification

new classifier for every new sample!

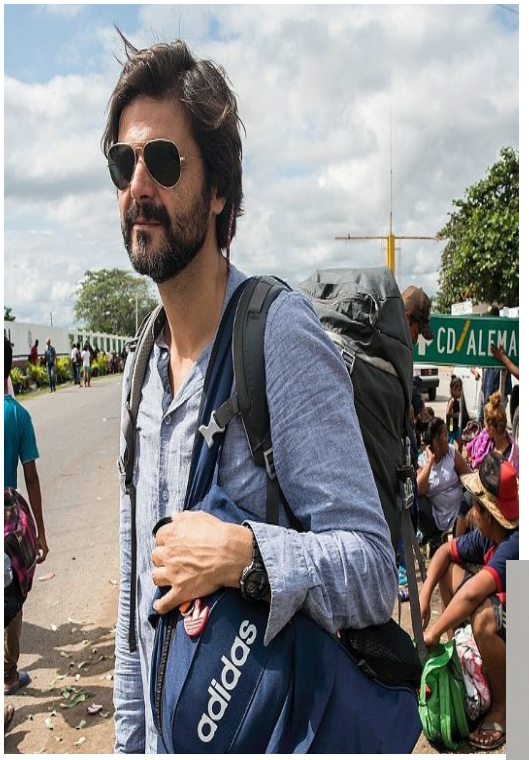
$D_t(x)$

we have to amortize too!

$D(x, y)$

Friedman, J., T. Hastie, and R. Tibshirani, 2001: The elements of statistical learning.
Sugiyama, M., 2012: Density Ratio Estimation in Machine Learning.

Search: Find real Tobiases



Sources: Spiegel

Search: Find real Tobiases

Sources: Spiegel

Generative adversarial network

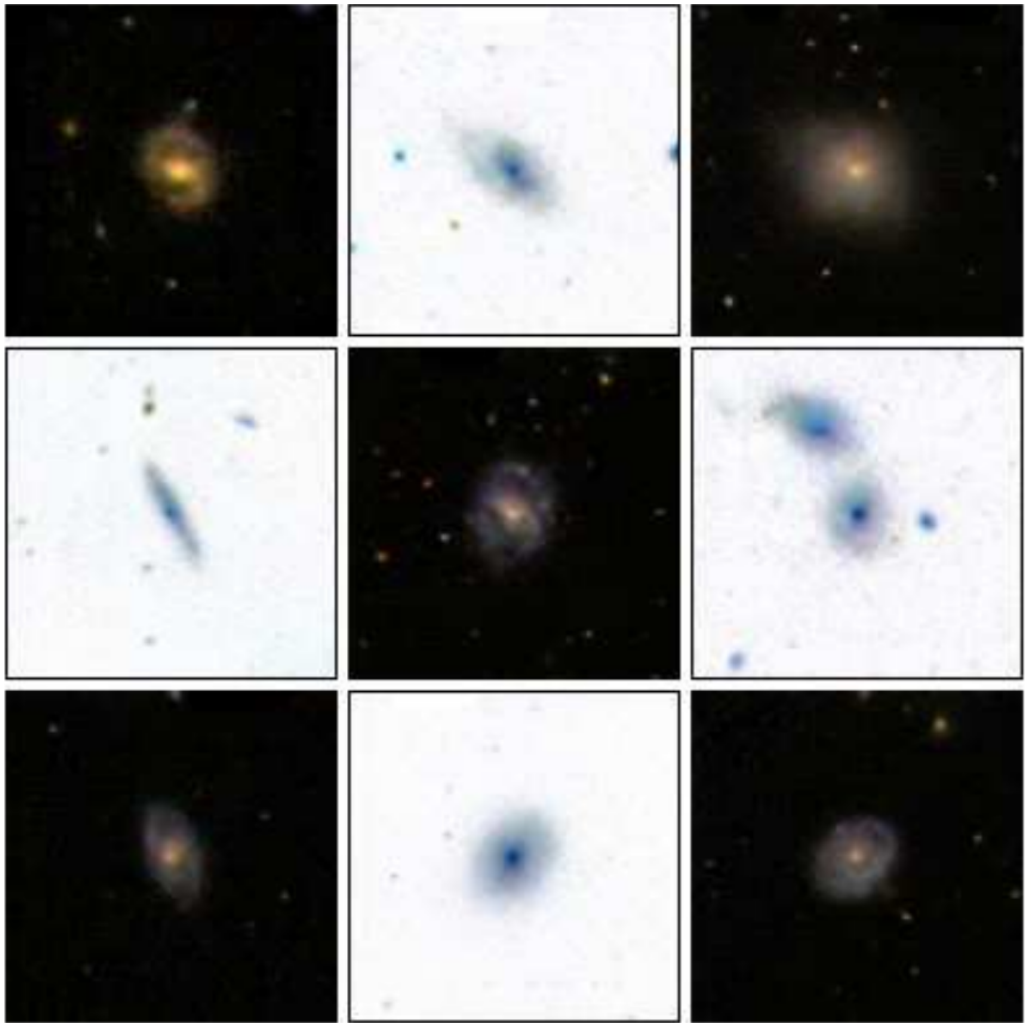
- Used as amortized approximation to Kullback-Leibler divergence
- MinMax game → Nash equilibrium
- Unstable training (ongoing work in CS)
- Many different variants: (Wasserstein GAN, boundary equilibrium GAN, relativistic GAN, gradient regularization)

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2014: Generative Adversarial Nets.

Examples: Generative adversarial networks (GANs)

from Karras, T., S. Laine, and T. Aila, 2018: A Style-Based Generator Architecture for Generative Adversarial Networks.

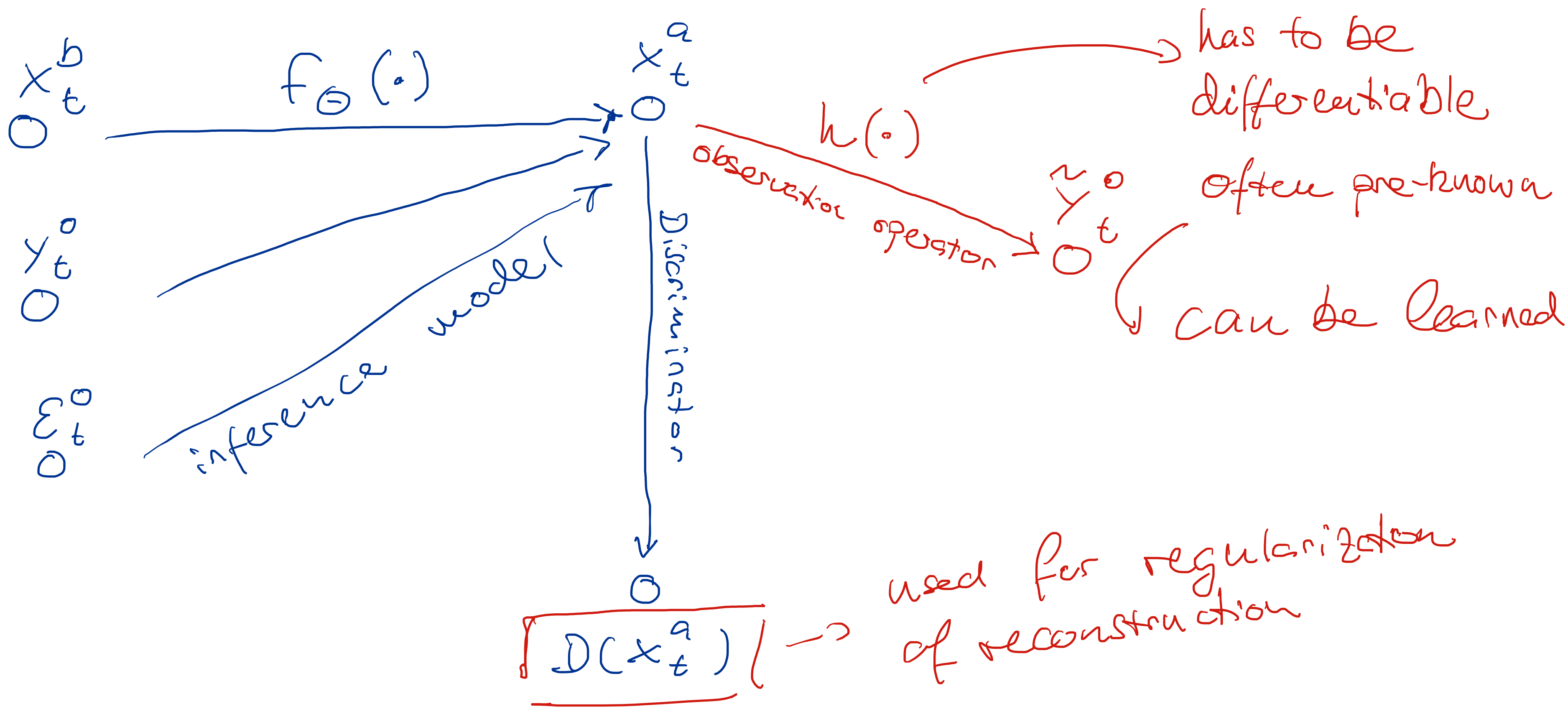
Examples: Generative adversarial networks (GANs) in physics



also partially used for CMC at CERN

from Fussell, L., and B. Moews, 2018: Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks.

Adversarial variational Bayes for DA



⇒ Discriminator is here trained by ensemble trick:

$x_{t,i}^a = f_\Theta(y_t^o, x_{t,i}^b, \varepsilon_t^o)$ → fake sample

$x_{t,j}^b$ → real sample

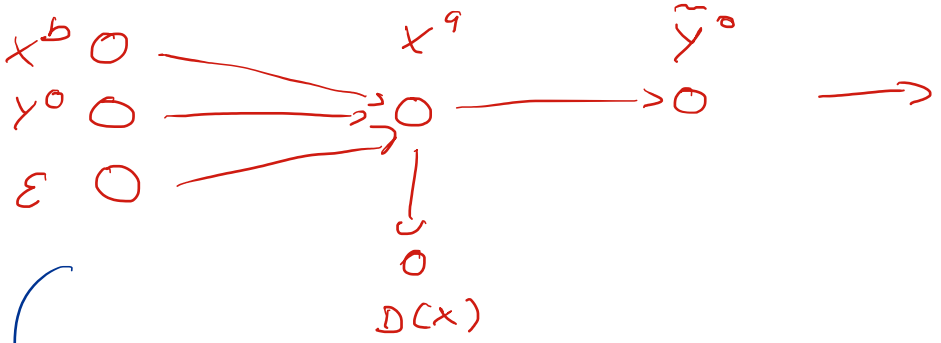
} like adaptive contrast

Based on – Mescheder, L., S. Nowozin, and A. Geiger, 2017:
Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks.

Experiment

- Model: Lorenz 1996, 40 grid points, RK4
- Virtual reality: F=8 (train), 8.1 (eval), 7.9 (test)
- Ensemble: 50 member, $\mathsf{F} = \mathcal{N}(8, 0.25)$, $\mathsf{IC} = \mathsf{VR} + \mathcal{N}(0, 4)$
- Obs: dt $= 8$ ts, 20 observations, $\mathsf{VR} + \mathcal{N}(0, 0.25)$
- LETKF, Gaspari-Cohn 5 grid points, mult. inflation 1.1

Used neural network



Stochastic inference network, no ensemble information!

- prescribed obs operator
- gradient penalty for real data only (Mescheder 2018)
- Multiplicative concatenation of ε and (y^o, x^b)
- linear layers only
  - batch normalization
  - leaky relu
- RMS Prop

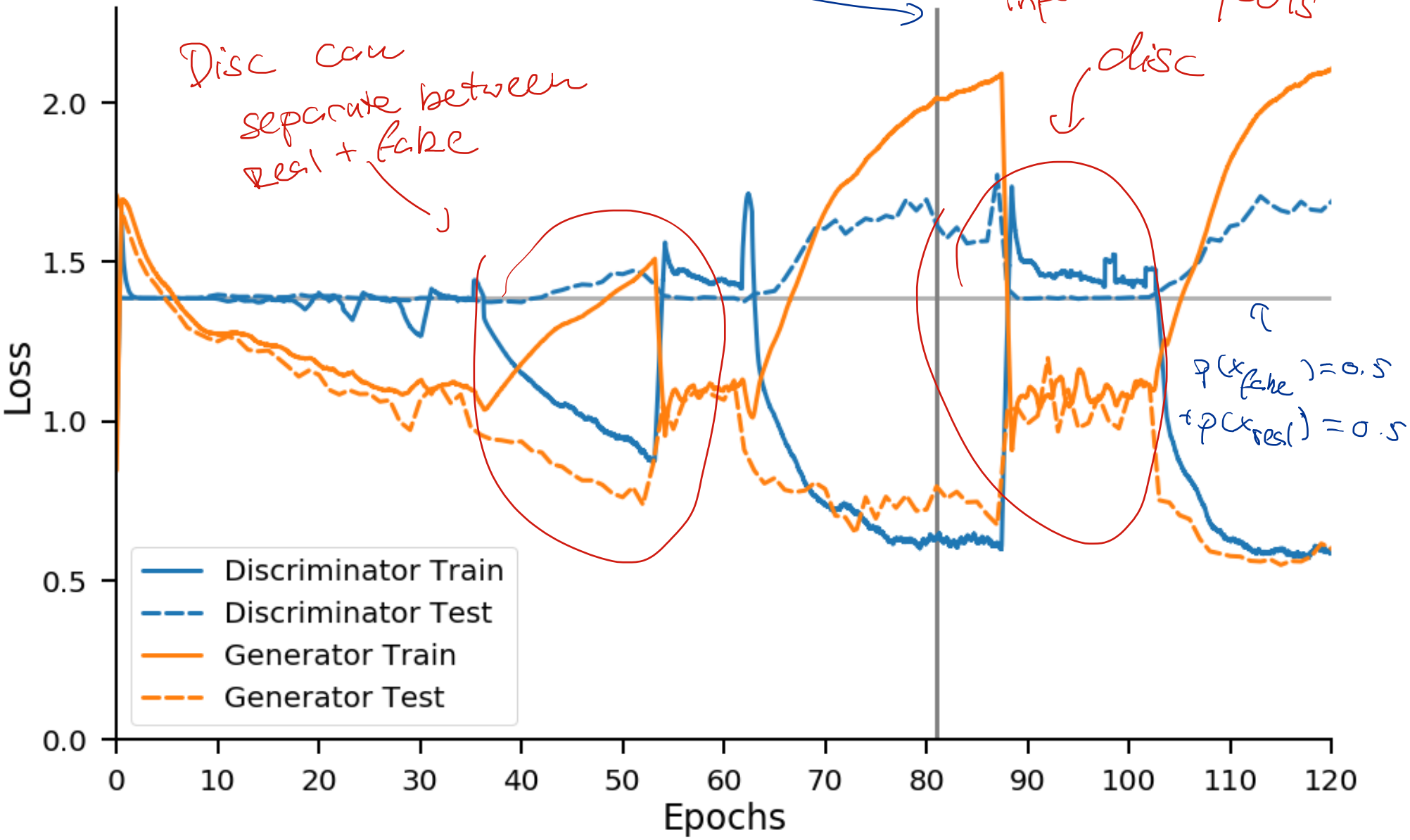
TRAINING:

Per epoch random ensemble member selection

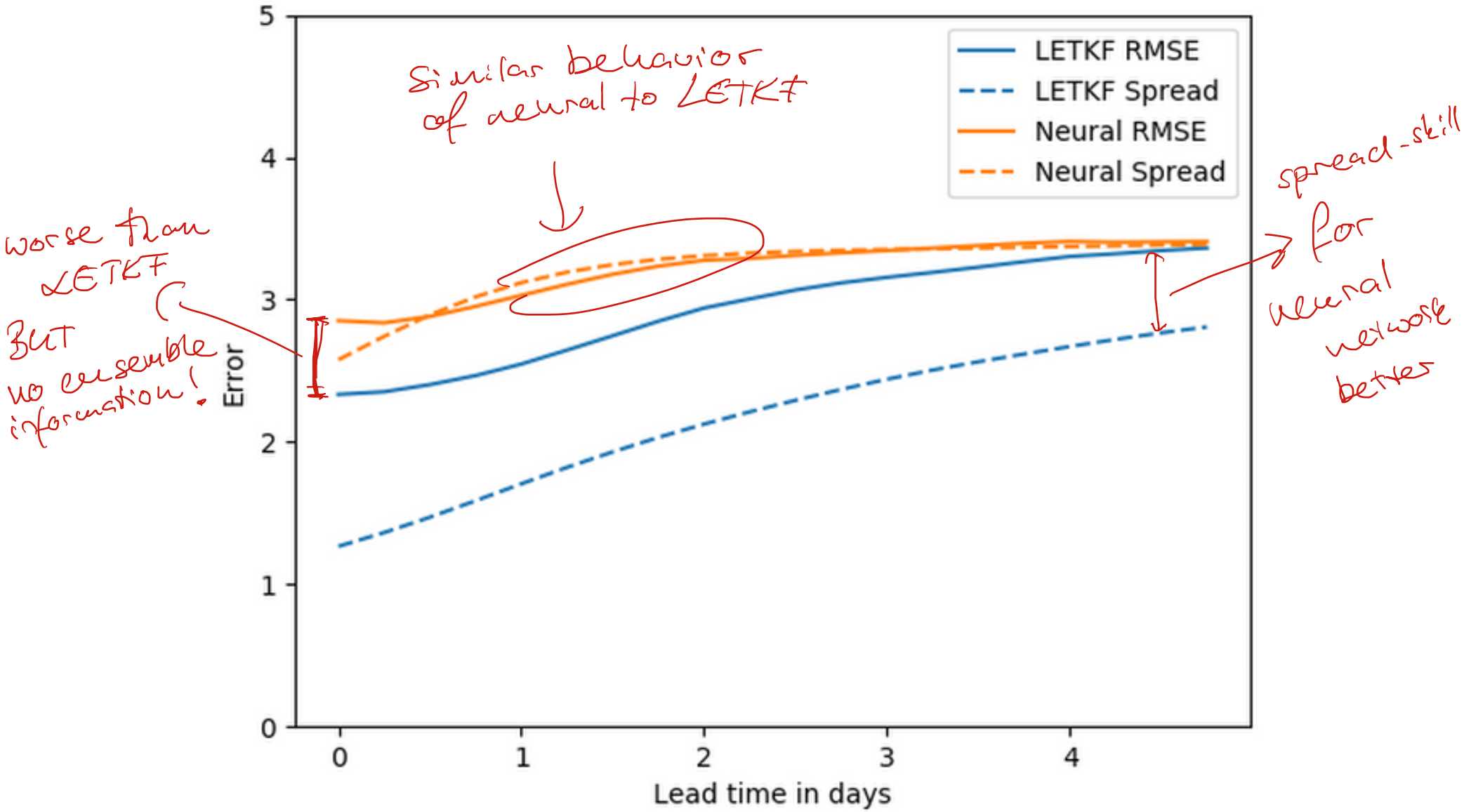
Assimilation:

Inference for every ens member independently → like Stochastic ensemble Kalman filter

Losses oscillate & diverge

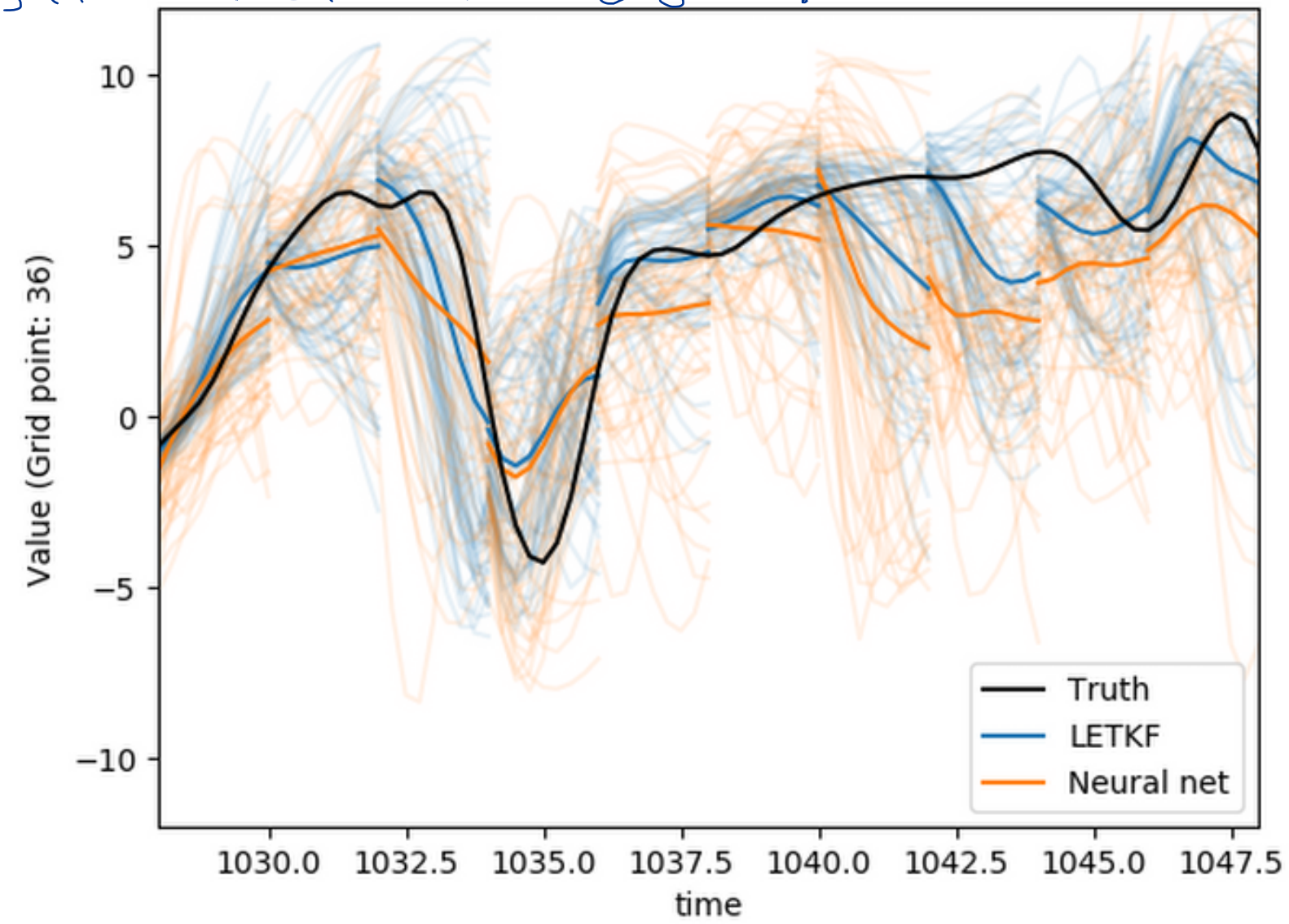


No problems with spread

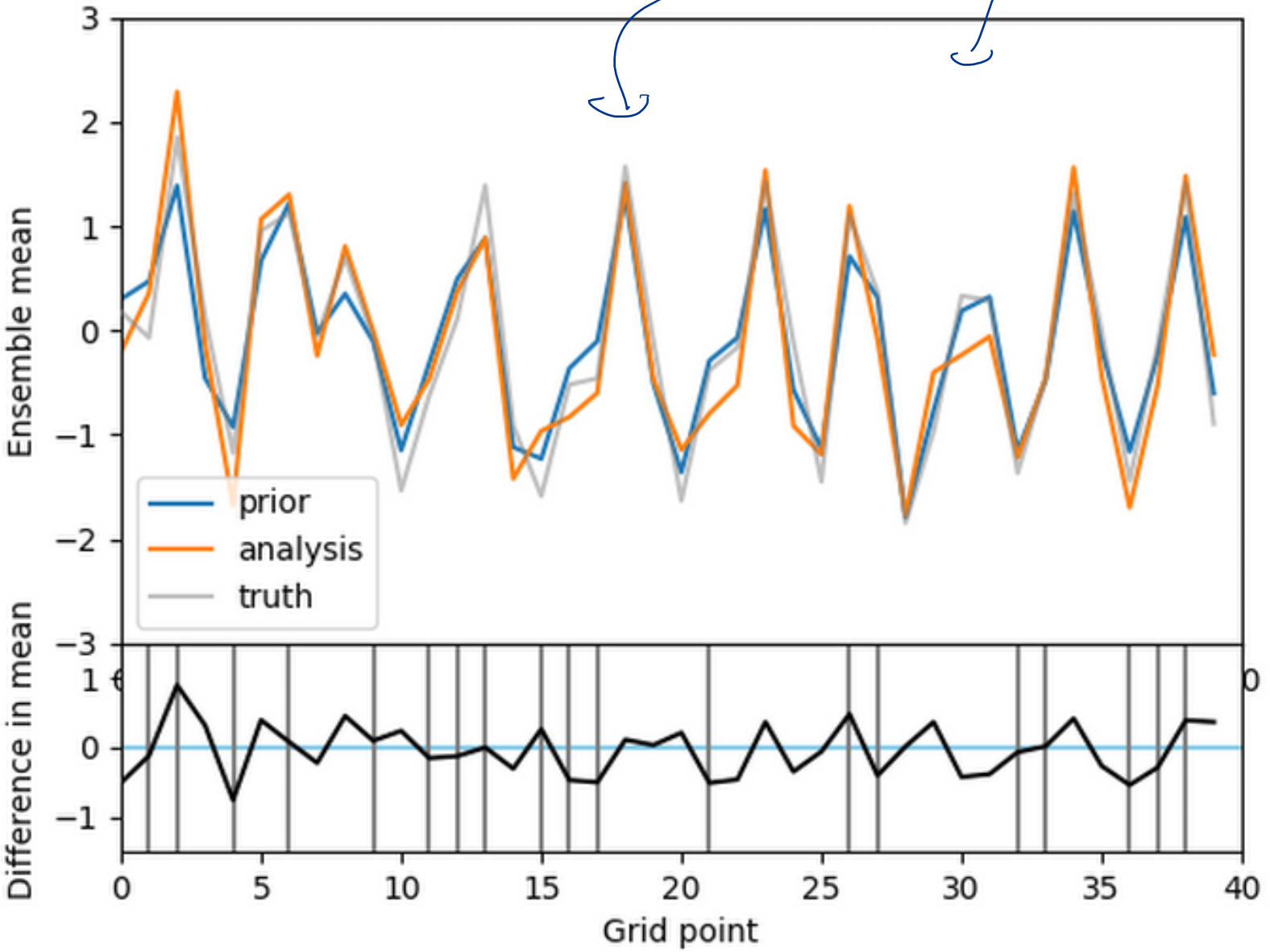


Captures time evolution

Neural assimilation updates every ens member independently
↳ in most cases it looks good!



Nudged to truth



Which questions have to be answered

- How to stabilize training (open question for GANs)?
- Analytical solution for simplified problem (DiracGAN)?
  - Is unsupervised training possible?
    - ↳ w/o overlapping times in samples between prior and obs during training
  - What happens if observation has no impact?
    - ↳ analysis should then be the prior
  - Can we use ensemble for flow-dependent inference?
    - ↳ include ensemble data?
  - How to train in operational environment?
    - ↳ transfer learning and recursive fitting of layers
  - Can we train new observation operators?
    - ↳ do we need to predict an observation operator?
- Can we generalize other data assimilation algorithms?
  - ↳ this has to be proven mathematically for optimal discriminator!

Related

Mescheder 2018

Summary

- Combined deep learning & data assimilation
- Amortized variational inference
  $+$ adversarial training (most problems)
- Many open questions remain
  $\rightarrow$ developments in computer science
- Can be a candidate for coupled non-linear data assimilation $+$ to solve normal problems

4DVar as variational autoencoder

$$\mathcal{L} = KL(p(x_0^a) \,||\, p(x_0^b)) - \sum_{i=0}^{t} \log_{x \sim x_i^a} p(y_i^o | x)$$

learn this

$x_{0:t}^a$ x_0^a

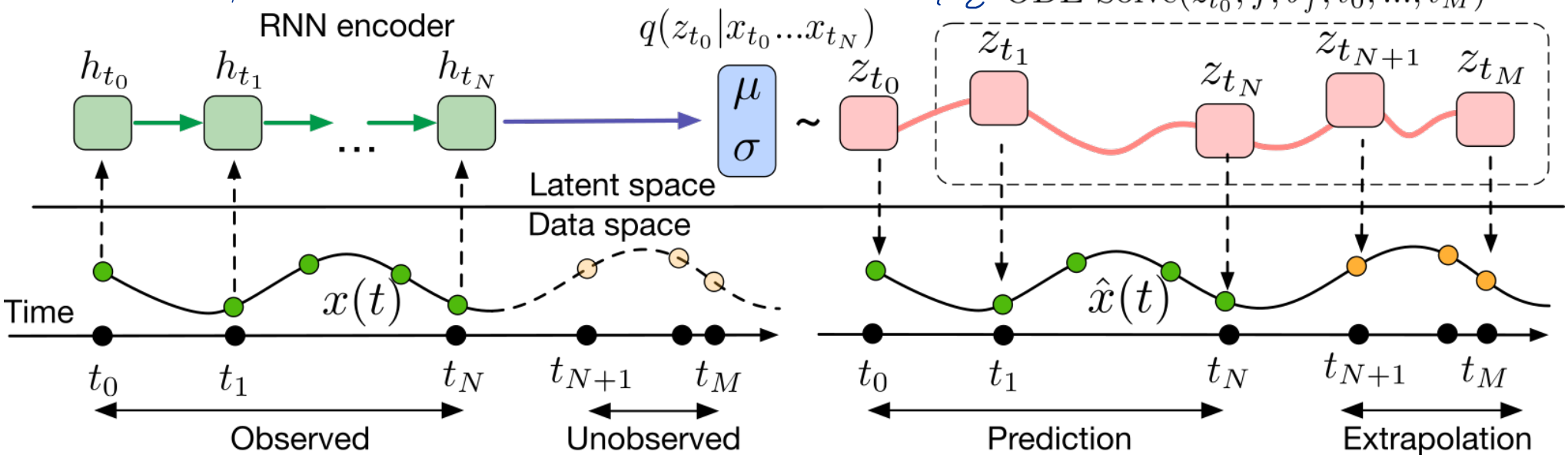
y_0^o y_1^o y_2^o

inverse adjoint

adjoint

$\tilde{y}_0^o$ $\tilde{y}_1^o$ $\tilde{y}_2^o$

almost same loss as 4D-Var (Recurrent neural network like)



Chen, R. T. Q., Y. Rubanova, J. Bettencourt, and D. Duvenaud, 2018: Neural Ordinary Differential Equations

Generative adversarial network

Source: arxiv.org/abs/1612.00005

Generative adversarial network