APPRENTISSAGE DE REPRÉSENTATIONS ET MODÈLES GÉNÉRATIFS PROFONDS DANS LES SYSTÈMES DYNAMIQUES

Jean-Yves Franceschi

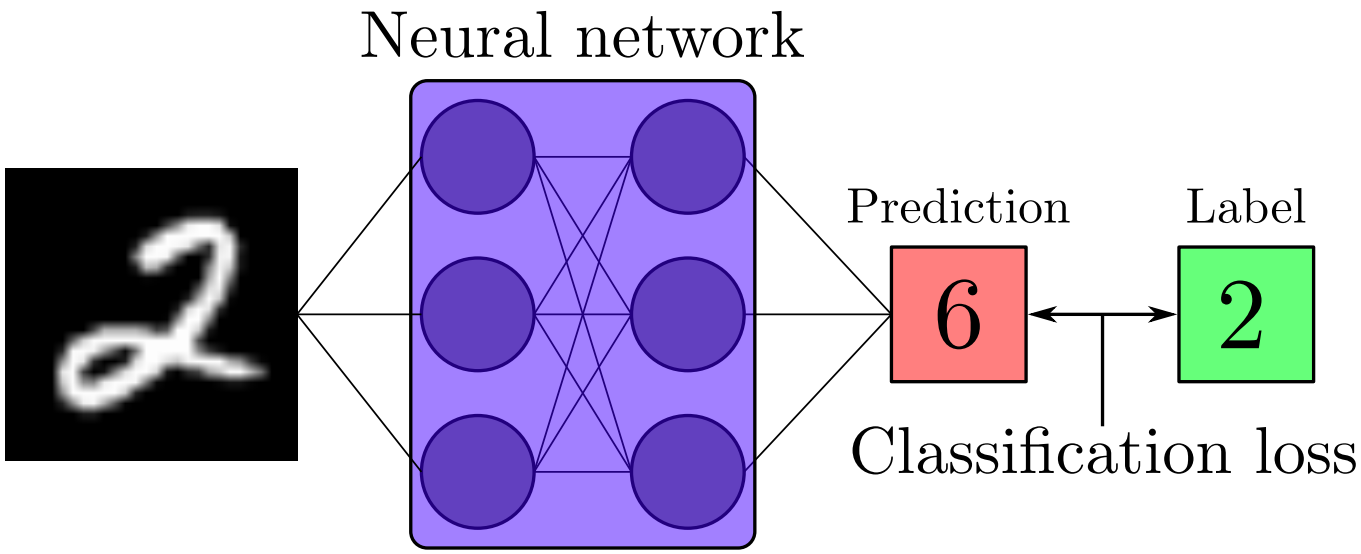
14 février 2022

Soutenance de thèse de doctorat
devant le jury composé de :

MM. Xavier Alameda-Pineda & Alexandre Gramfort	Rapporteurs
Mmes Catherine Achard & Camille Couprie	Examinatrices
MM. Sylvain Lamprier & Patrick Gallinari	Directeurs de thèse

Machine Learning & Deep Learning for Information Access

Motivation

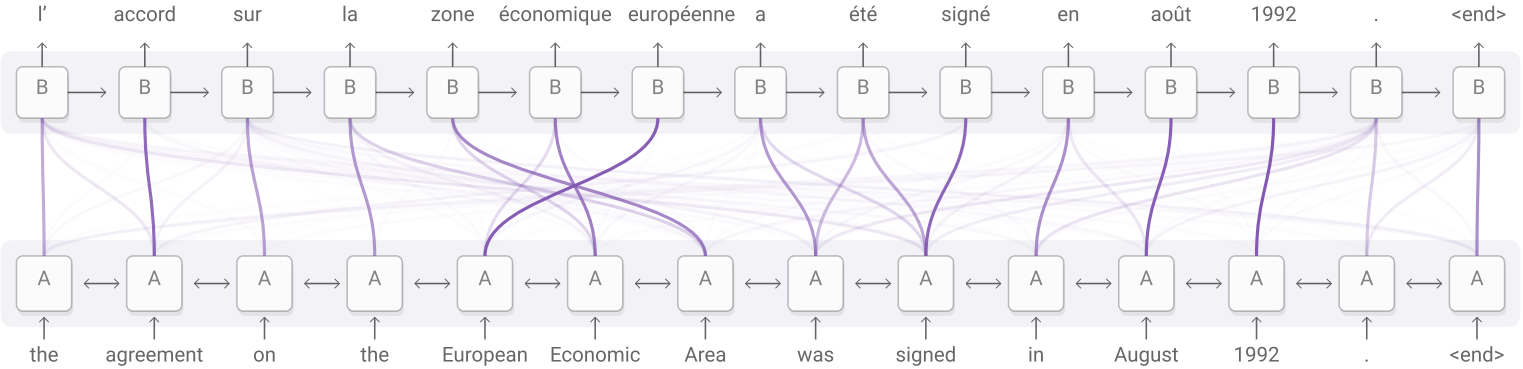


- Deep learning has driven many scientific breakthroughs since 2012, including in:
  - computer vision (Krizhevsky et al., 2012);

AlphaGo

Source: Google DeepMind, via Wikimedia Commons. Public domain.

- Deep learning has driven many scientific breakthroughs since 2012, including in:
  - computer vision (Krizhevsky et al., 2012);
  - games (Silver et al., 2017);



Source: Olah & Carter, Attention and Augmented Recurrent Neural Networks, Distill, 2016. CC-BY 2.0.

- Deep learning has driven many scientific breakthroughs since 2012, including in:
  - computer vision (Krizhevsky et al., 2012);
  - games (Silver et al., 2017);
  - natural language processing (Brown et al., 2020).

AlphaGo	Elo
Master (sup.)	4858
Zero (unsup.)	5185

Few-shot language correction with GPT-3:

Input:	The patient was died.
Output:	The patient died.
Input:	Janet broke Bill on the finger.
Output:	Janet broke Bill's finger.

- Deep learning has driven many scientific breakthroughs since 2012, including in:
  - computer vision (Krizhevsky et al., 2012);
  - games (Silver et al., 2017);
  - natural language processing (Brown et al., 2020).
- Many advances nowadays are achieved in an unsupervised learning setting.
- This is necessary for the development of autonomous systems.

Deep Learning

Source: Dllu, via Wikimedia Commons. CC BY-SA 4.0

- Deep learning has driven many scientific breakthroughs since 2012, including in:
  - computer vision (Krizhevsky et al., 2012);
  - games (Silver et al., 2017);
  - natural language processing (Brown et al., 2020).
- Many advances nowadays are achieved in an unsupervised learning setting.
- This is necessary for the development of autonomous systems.

Representation Learning & Deep Generative Models

Image source: Geiger, Lenz, Stiller & Urtasun, KITTI Vision Benchmark Suite. CC BY-NC-SA 3.0.

Representation Learning

- Learning vectorial embeddings of raw data.
- Applications:
  - pre-training from unlabeled samples;
  - dimensionality reduction;
  - disentanglement, etc.

Generative Modeling

- Learning to synthesize new data points.
- Applications:
  - data augmentation;
  - image & text generation;
  - procedural generation;
  - sequence prediction, etc.

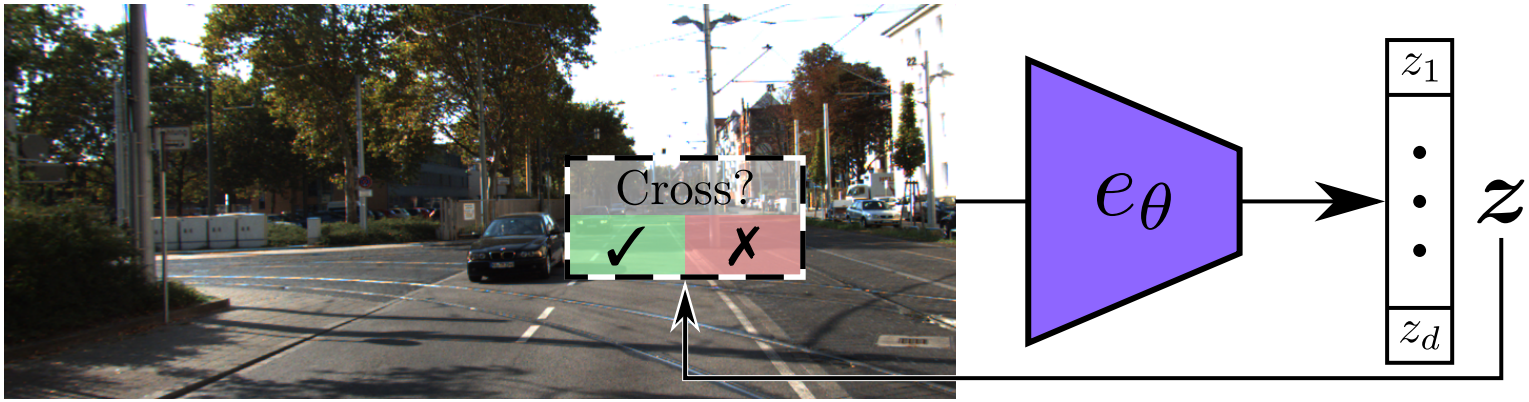


Image source: Geiger, Lenz, Stiller & Urtasun, KITTI Vision Benchmark Suite. CC BY-NC-SA 3.0.

Representation Learning

- Learning vectorial embeddings of raw data.
- Applications:
  - pre-training from unlabeled samples;
  - dimensionality reduction;
  - disentanglement, etc.

Generative Modeling

- Learning to synthesize new data points.
- Applications:
  - data augmentation;
  - image & text generation;
  - procedural generation;
  - sequence prediction, etc.

Image source: Geiger, Lenz, Stiller & Urtasun, KITTI Vision Benchmark Suite. CC BY-NC-SA 3.0.

Representation Learning

- Learning vectorial embeddings of raw data.
- Applications:
  - pre-training from unlabeled samples;
  - dimensionality reduction;
  - disentanglement, etc.

Generative Modeling

- Learning to synthesize new data points.
- Applications:
  - data augmentation;
  - image & text generation;
  - procedural generation;
  - sequence prediction, etc.

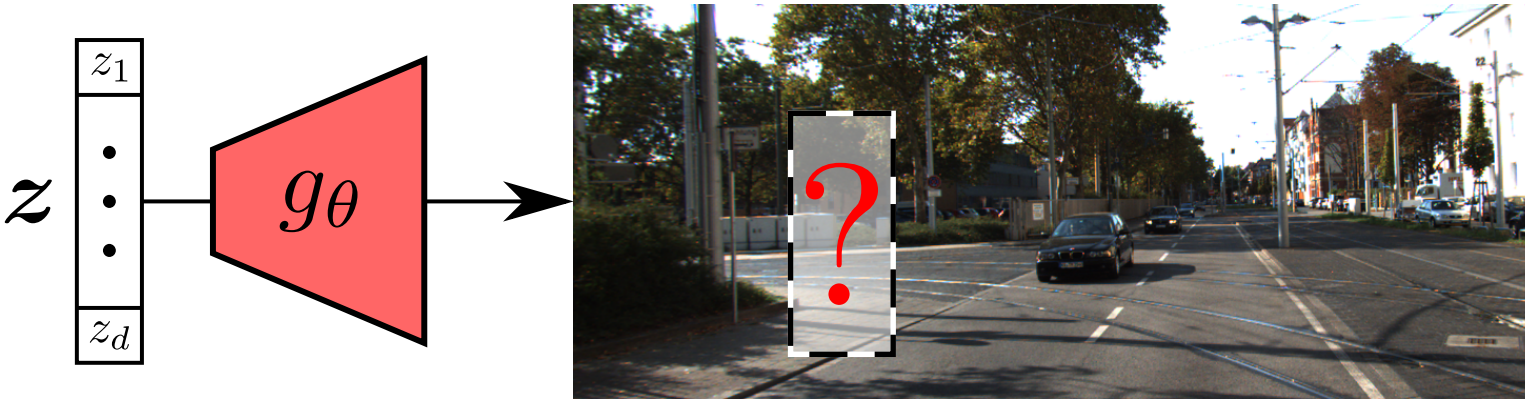


Image source: Geiger, Lenz, Stiller & Urtasun, KITTI Vision Benchmark Suite. CC BY-NC-SA 3.0.

Representation Learning

- Learning vectorial embeddings of raw data.
- Applications:
  - pre-training from unlabeled samples;
  - dimensionality reduction;
  - disentanglement, etc.

Generative Modeling

- Learning to synthesize new data points.
- Applications:
  - data augmentation;
  - image & text generation;
  - procedural generation;
  - sequence prediction, etc.

Temporality

Source: NVIDIA Corporation, Wang, Liu & Zhu, vid2vid. CC BY-NC-SA 4.0.

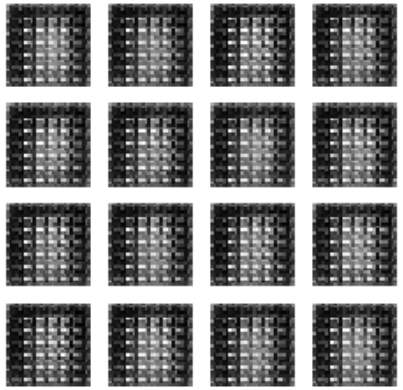
Data Dynamics

- Data evolve naturally.
- Important for autonomous systems in evolving environments.

Training Dynamics

- Evolution of neural networks during training.
- Important to understand their behavior.

Temporality



Source: The TensorFlow Authors, TensorFlow Documentation. Apache.

Data Dynamics

- Data evolve naturally.
- Important for autonomous systems in evolving environments.

Training Dynamics

- Evolution of neural networks during training.
- Important to understand their behavior.

Neural Networks and Dynamical Systems

- Dynamical systems have been increasingly used in deep learning.
- Example with ODEs:

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = f(x_t).$$

ODEs in Data Dynamics

Replacing RNNs by neural ODEs (Chen et al., 2018):

$$y_{t+1} = \mathrm{RNN}_\theta(y_t)$$

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = f_\theta(y_t).$$

ODEs in Training Dynamics

From discrete-time to continuous-time (with a loss $\mathcal{L}_\theta$):

$$\theta_{k+1} = \theta_k - \lambda \frac{\partial \mathcal{L}_{\theta_k}}{\partial \theta_k}$$

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\lambda \frac{\partial \mathcal{L}_{\theta_t}}{\partial \theta_t}.$$

- Dynamical systems have been increasingly used in deep learning.
- Example with ODEs:

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = f(x_t).$$

ODEs in Data Dynamics

Replacing RNNs by neural ODEs (Chen et al., 2018):

$$y_{t+1} = \mathrm{RNN}_\theta(y_t)$$

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = f_\theta(y_t).$$

ODEs in Training Dynamics

From discrete-time to continuous-time (with a loss $\mathcal{L}_\theta$):

$$\theta_{k+1} = \theta_k - \lambda \frac{\partial \mathcal{L}_{\theta_k}}{\partial \theta_k}$$

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\lambda \frac{\partial \mathcal{L}_{\theta_t}}{\partial \theta_t}.$$

- Dynamical systems have been increasingly used in deep learning.
- Example with ODEs:

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = f(x_t).$$

ODEs in Data Dynamics

Replacing RNNs by neural ODEs (Chen et al., 2018):

$$y_{t+1} = \mathrm{RNN}_\theta(y_t)$$

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = f_\theta(y_t).$$

ODEs in Training Dynamics

From discrete-time to continuous-time (with a loss $\mathcal{L}_\theta$):

$$\theta_{k+1} = \theta_k - \lambda \frac{\partial \mathcal{L}_{\theta_k}}{\partial \theta_k}$$

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\lambda \frac{\partial \mathcal{L}_{\theta_t}}{\partial \theta_t}.$$

Contributions

Temporal Data, Representation Learning

- Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. "**Unsupervised Scalable Representation Learning for Multivariate Time Series**". NeurIPS 2019.
- *Jean-Yves Franceschi*, *Edouard Delasalles*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari (July 2020). "**Stochastic Latent Residual Video Prediction**". ICML 2020.
- *Jérémie Donà*, *Jean-Yves Franceschi*, Sylvain Lamprier, and Patrick Gallinari (2021). "**PDE-Driven Spatiotemporal Disentanglement**". ICLR 2021.
- *Jean-Yves Franceschi*, *Emmanuel de Bézenac*, *Ibrahim Ayed*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. "**A Neural Tangent Kernel Perspective of GANs**". arXiv: 2106.05566, 2021.

Dynamical Systems, Generative Models

Contributions

Temporal Data, Representation Learning

- Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. "**Unsupervised Scalable Representation Learning for Multivariate Time Series**". NeurIPS 2019.
- *Jean-Yves Franceschi*, *Edouard Delasalles*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari (July 2020). "**Stochastic Latent Residual Video Prediction**". ICML 2020.
- *Jérémie Donà*, *Jean-Yves Franceschi*, Sylvain Lamprier, and Patrick Gallinari (2021). "**PDE-Driven Spatiotemporal Disentanglement**". ICLR 2021.
- *Jean-Yves Franceschi*, *Emmanuel de Bézenac*, *Ibrahim Ayed*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. "**A Neural Tangent Kernel Perspective of GANs**". arXiv: 2106.05566, 2021.

Dynamical Systems, Generative Models

Contributions

Temporal Data, Representation Learning

- Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. "**Unsupervised Scalable Representation Learning for Multivariate Time Series**". NeurIPS 2019.
- *Jean-Yves Franceschi*, *Edouard Delasalles*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari (July 2020). "**Stochastic Latent Residual Video Prediction**". ICML 2020.
- *Jérémie Donà*, *Jean-Yves Franceschi*, Sylvain Lamprier, and Patrick Gallinari (2021). "**PDE-Driven Spatiotemporal Disentanglement**". ICLR 2021.
- *Jean-Yves Franceschi*, *Emmanuel de Bézenac*, *Ibrahim Ayed*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. "**A Neural Tangent Kernel Perspective of GANs**". arXiv: 2106.05566, 2021.

Dynamical Systems, Generative Models

Contributions

Temporal Data, Representation Learning

- Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. "**Unsupervised Scalable Representation Learning for Multivariate Time Series**". NeurIPS 2019.
- *Jean-Yves Franceschi*, *Edouard Delasalles*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari (July 2020). "**Stochastic Latent Residual Video Prediction**". ICML 2020.
- *Jérémie Donà*, *Jean-Yves Franceschi*, Sylvain Lamprier, and Patrick Gallinari (2021). "**PDE-Driven Spatiotemporal Disentanglement**". ICLR 2021.
- *Jean-Yves Franceschi*, *Emmanuel de Bézenac*, *Ibrahim Ayed*, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. "**A Neural Tangent Kernel Perspective of GANs**". arXiv: 2106.05566, 2021.

Dynamical Systems, Generative Models

State-Space Predictive Models for Spatiotemporal Data

Franceschi, Delasalles et al. "**Stochastic Latent Residual Video Prediction**". ICML'20.

Donà, Franceschi et al. "**PDE-Driven Spatiotemporal Disentanglement**". ICLR'21.

First Contribution: Video Prediction

Applications:

- Reinforcement Learning (Gregor et al., 2019).
- Robotics (Babaeizadeh et al., 2018).

Challenges

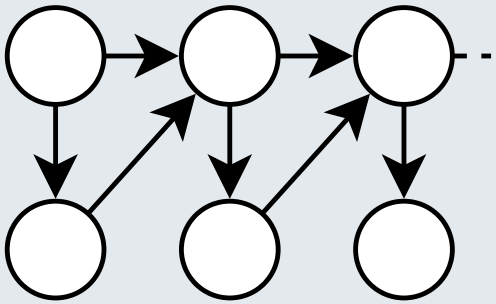
- Generation of realistic images.
- Long-term prediction.
- Account for uncertainty in the future.

Stochastic Video Prediction Literature Taxonomy

Autoregressive Models

+ Easy to learn, powerful.
– Temporal model tied to generation.

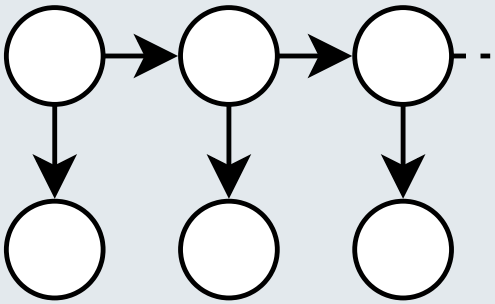
Ex.: Denton and Fergus (2018)



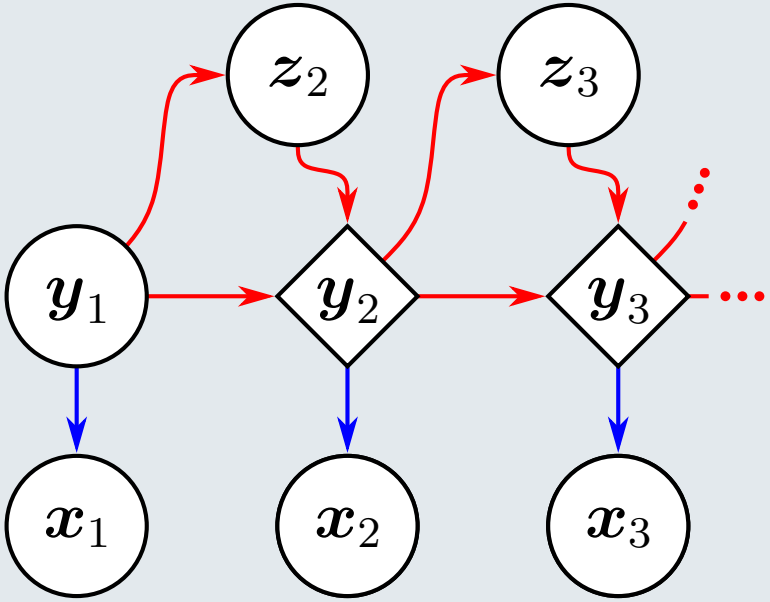
State-Space Models

+ Decoupled dynamics and prediction, interpretable.
– Harder to train.

Ex.: Fraccaro et al. (2017)



Model



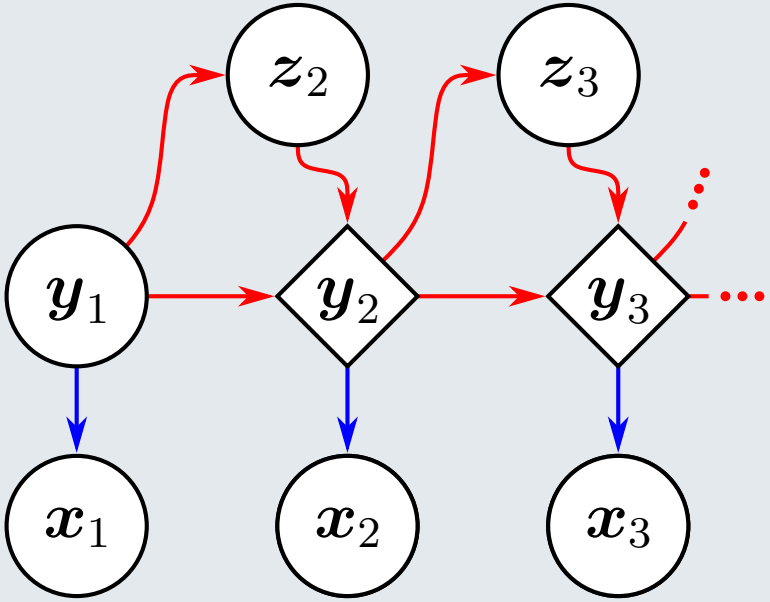
Update Rule:

$$\begin{cases} z_{t+1} \sim \mathcal{N}\big(\mu_\theta(y_t), \sigma_\theta(y_t) I\big) \\ y_{t+1} = y_t + f_\theta(y_t, z_{t+1}) \end{cases}$$

Key points:

- VAE state-space model.
- Residual updates.
- ODE inspiration.
- Generation at arbitrary frame rates.

Model



Update Rule:

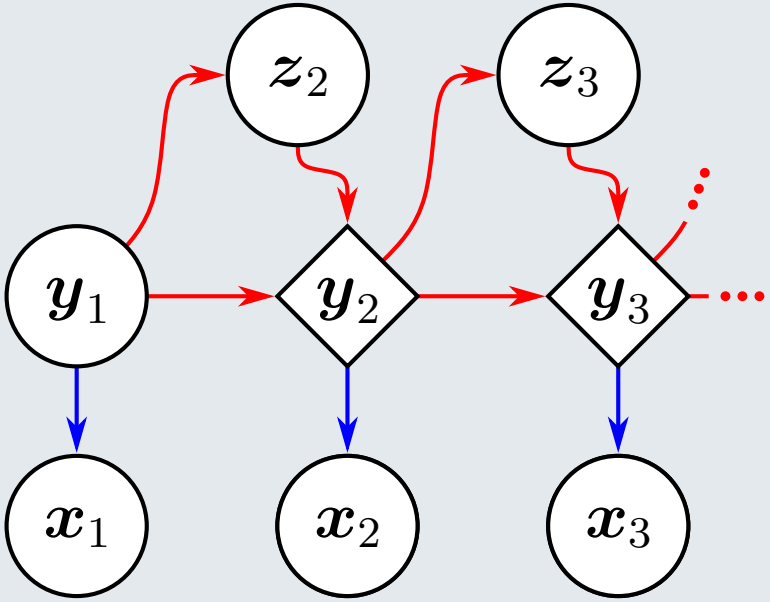
$$\begin{cases} z_{t+1} \sim \mathcal{N}\big(\mu_\theta(y_t), \sigma_\theta(y_t) I\big) \\ y_{t+1} = y_t + f_\theta(y_t, z_{t+1}) \end{cases}$$

Key points:

- VAE state-space model.
- Residual updates.
- ODE inspiration.
- Generation at arbitrary frame rates.

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f_\theta\Big(y_t, z_{\lfloor t \rfloor + 1}\Big)$$

Model

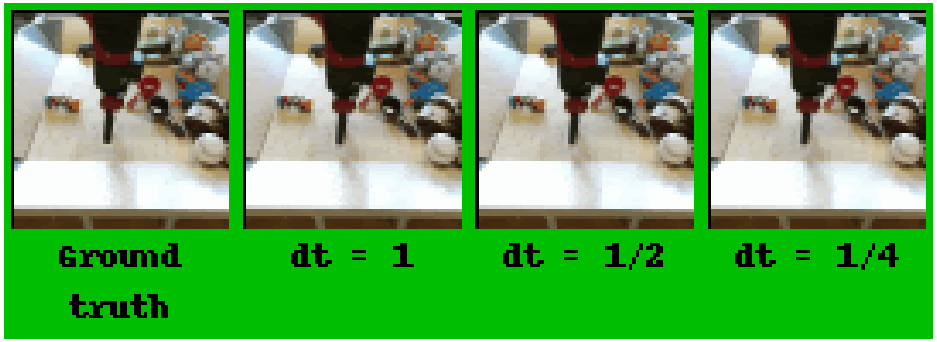


Update Rule:

$$\begin{cases} z_{t+1} \sim \mathcal{N}\big(\mu_\theta(y_t), \sigma_\theta(y_t) I\big) \\ y_{t+1} = y_t + f_\theta(y_t, z_{t+1}) \end{cases}$$

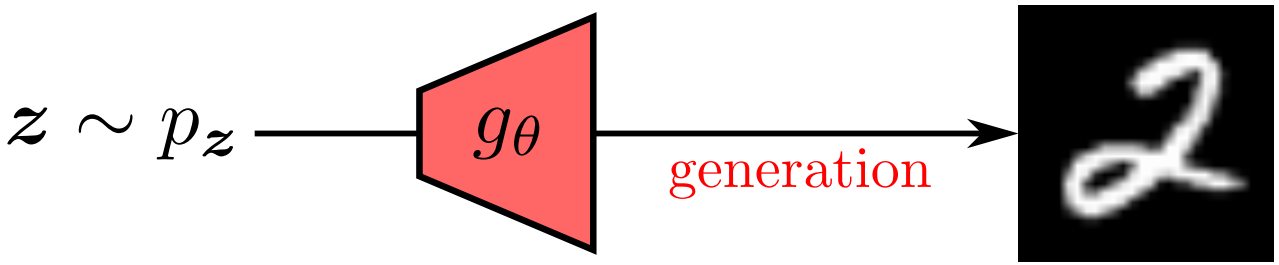
Key points:

- VAE state-space model.
- Residual updates.
- ODE inspiration.
- Generation at arbitrary frame rates.



- Generative model $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ for $\boldsymbol{x}$ from a latent $\boldsymbol{z} \sim p_\theta(\boldsymbol{z}) = p_{\boldsymbol{z}}$:

$$p_\theta(\boldsymbol{x}) = \int_{\boldsymbol{z}} p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\theta(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}$$



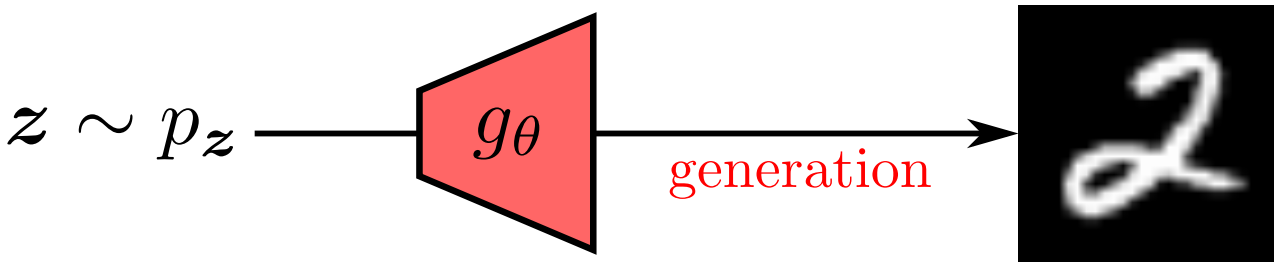
Variational Autoencoders (VAEs)

- Generative model $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ for $\boldsymbol{x}$ from a latent $\boldsymbol{z} \sim p_\theta(\boldsymbol{z}) = p_{\boldsymbol{z}}$:

$$p_\theta(\boldsymbol{x}) = \int_{\boldsymbol{z}} p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\theta(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}$$

- Intractable objective.
- Introducing $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ to approximate $p_\theta(\boldsymbol{z} \mid \boldsymbol{x})$:

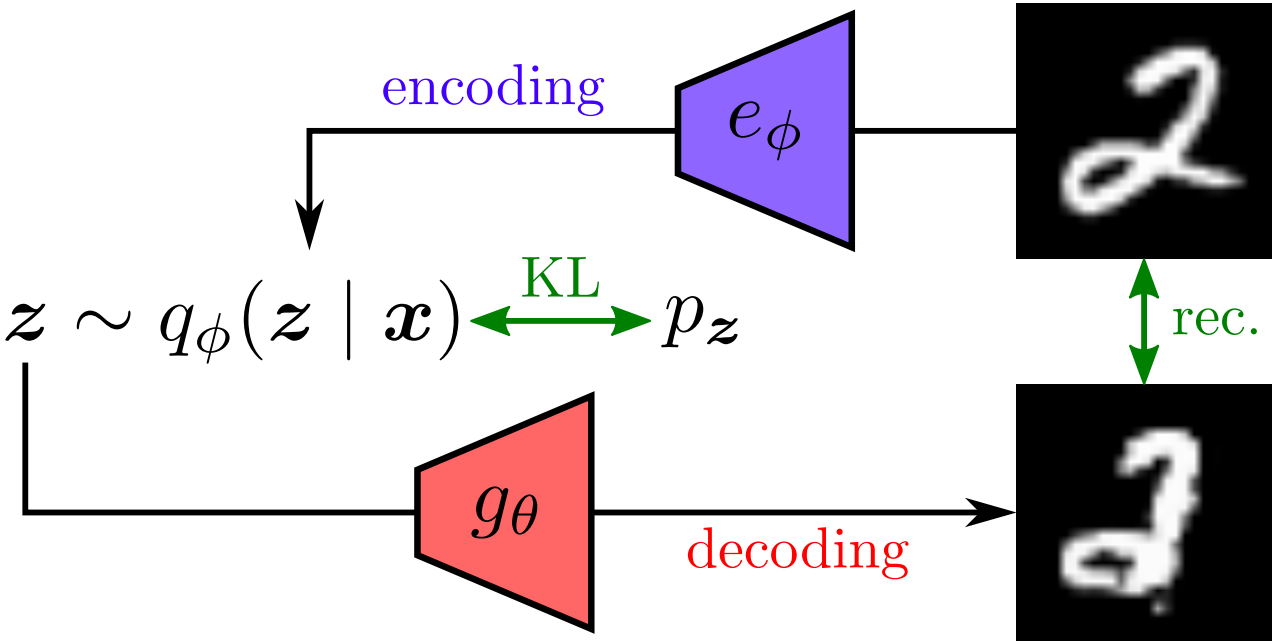
$$\log p_\theta(\boldsymbol{x}) \geq \mathbb{E}_{\boldsymbol{z} \sim q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) \right] - D_{\mathrm{KL}} \left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \,\middle\|\, p_{\boldsymbol{z}} \right).$$



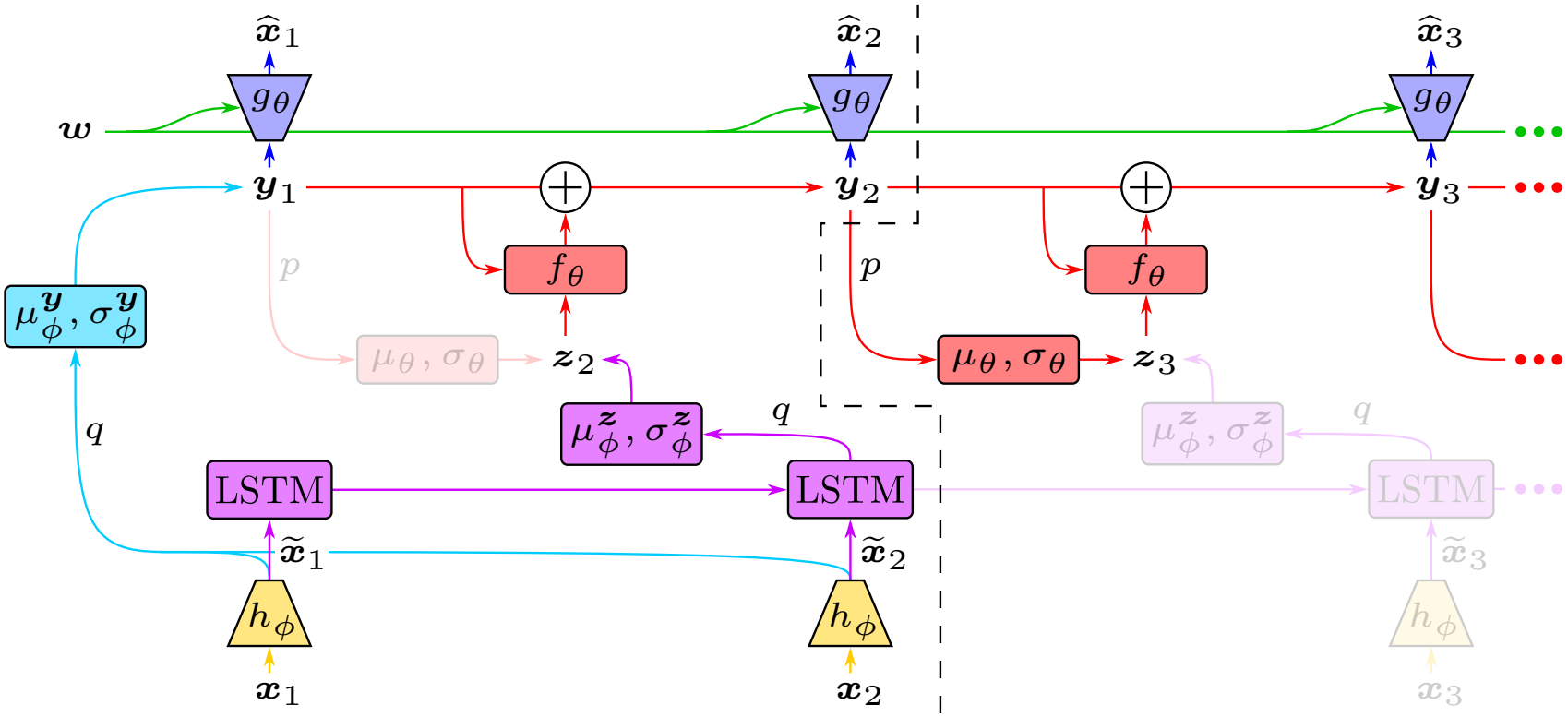
Evidence Lower Bound (ELBO)

Optimized objective:

$$\log p_\theta(\boldsymbol{x}) \geq \underbrace{\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}\mid\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}\mid\boldsymbol{x}) \,\|\, p_{\boldsymbol{z}}\right)}_{\text{KL term}}.$$



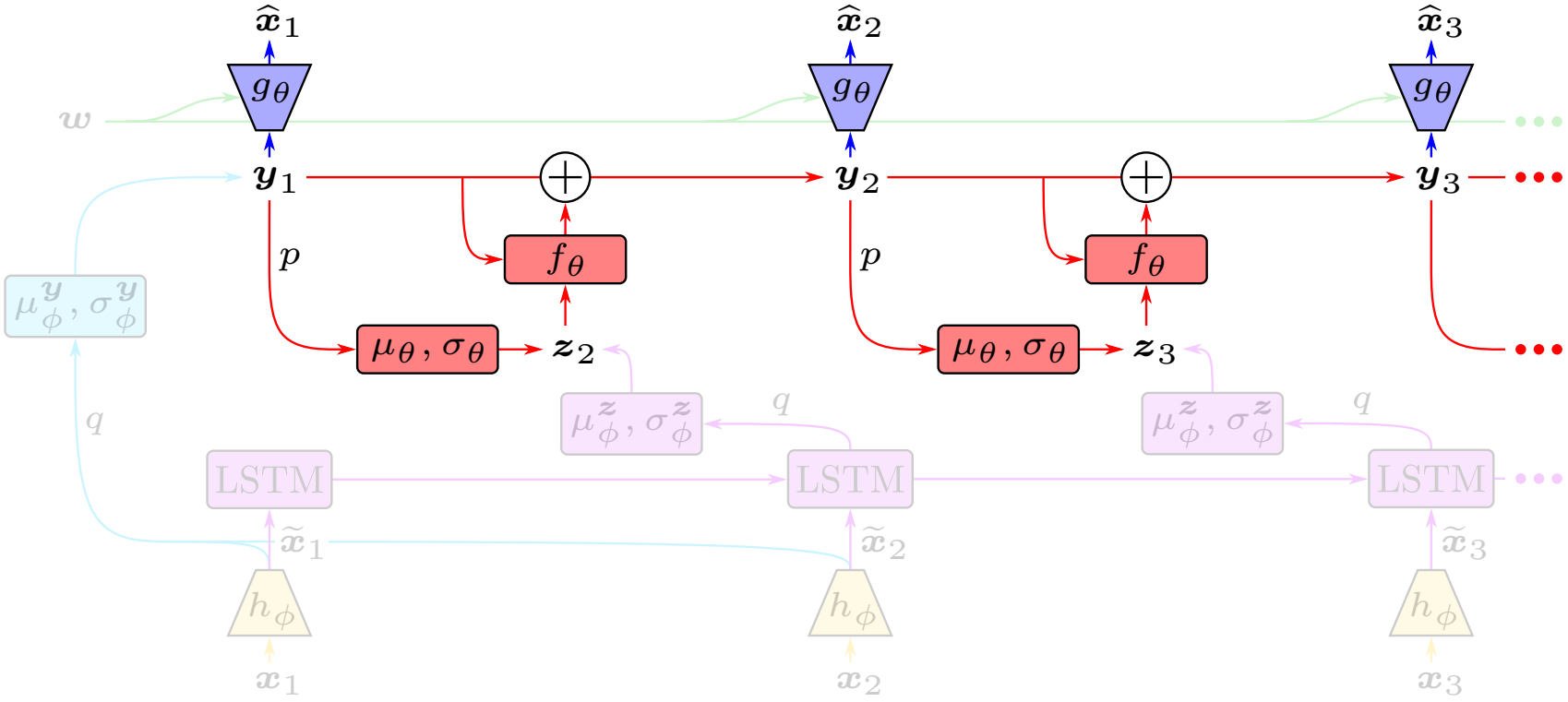
Prediction Model



Two operating modes:

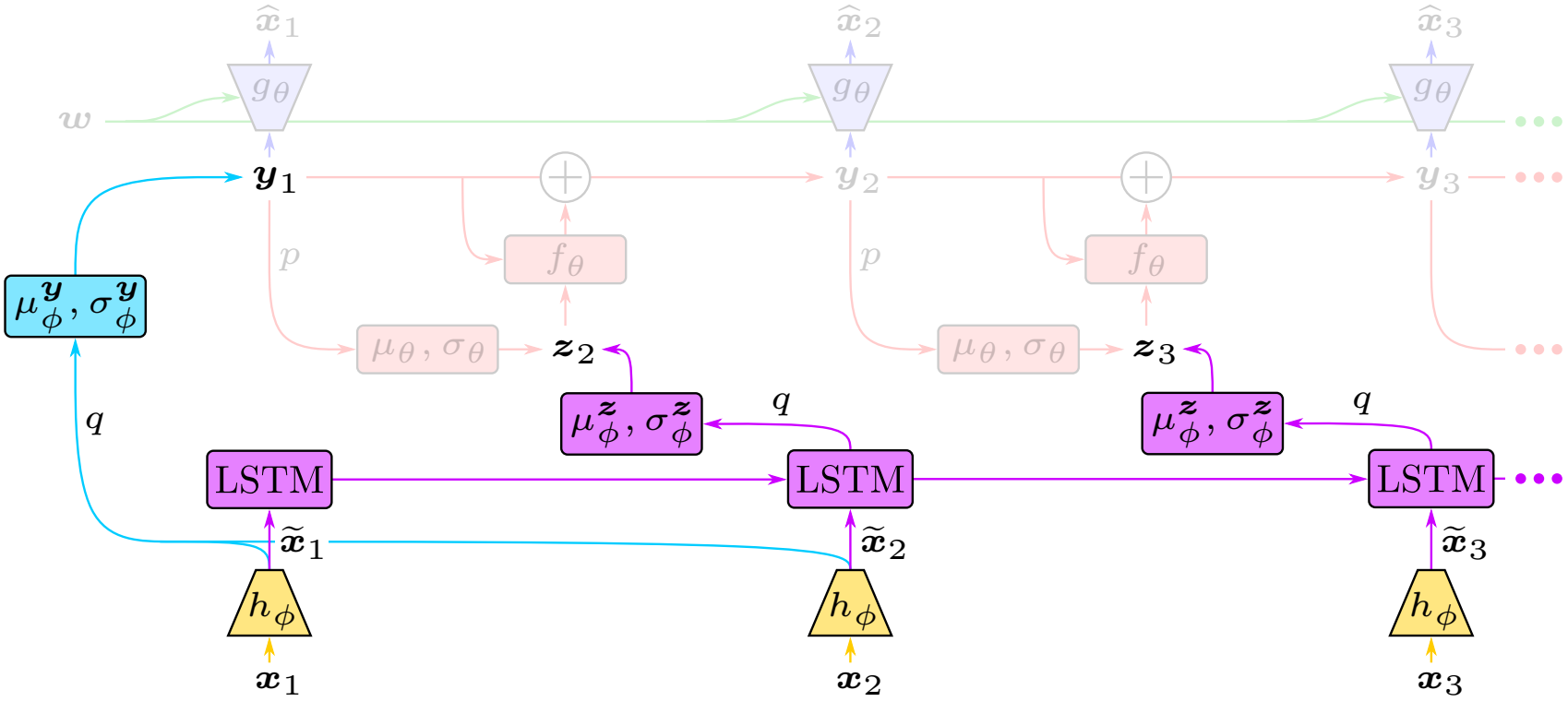
- conditioning, via q_ϕ;
- prediction / generation, via p_θ.

Generation Model



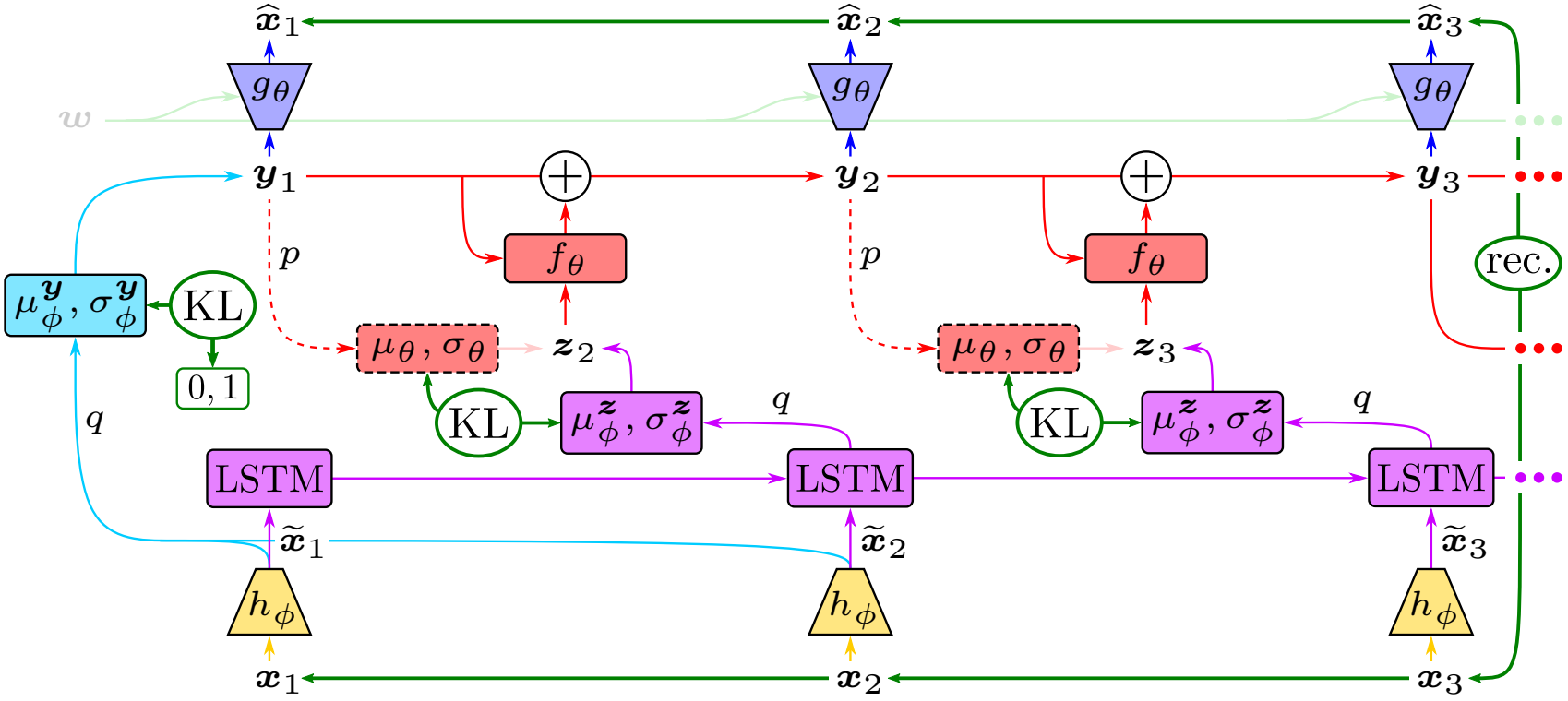
$$p_\theta\colon \begin{cases} \boldsymbol{y}_1 \sim \mathcal{N}(\mathbf{0}, I) & \text{(initial condition)} \\ \boldsymbol{z}_{t+1} \sim \mathcal{N}\big(\mu_\theta(\boldsymbol{y}_t), \sigma_\theta(\boldsymbol{y}_t) I\big) & \text{(random prediction)} \\ \boldsymbol{y}_{t+1} = \boldsymbol{y}_t + f_\theta(\boldsymbol{y}_t, \boldsymbol{z}_{t+1}) & \text{(latent state prediction)} \\ \boldsymbol{x}_t \sim \mathcal{N}\big(g_\theta(\boldsymbol{y}_t), \nu I\big) & \text{(decoding)} \end{cases}$$

Inference / Conditioning Model



$$q_\phi(\boldsymbol{z}_{2:T}, \boldsymbol{y}_1 \mid \boldsymbol{x}_{1:T}) = \underbrace{q_\phi(\boldsymbol{y}_1 \mid \boldsymbol{x}_{1:k})}_{\text{Init. Cond.}} \prod_{t=2}^{T} \underbrace{q_\phi(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t})}_{\text{LSTM}}$$

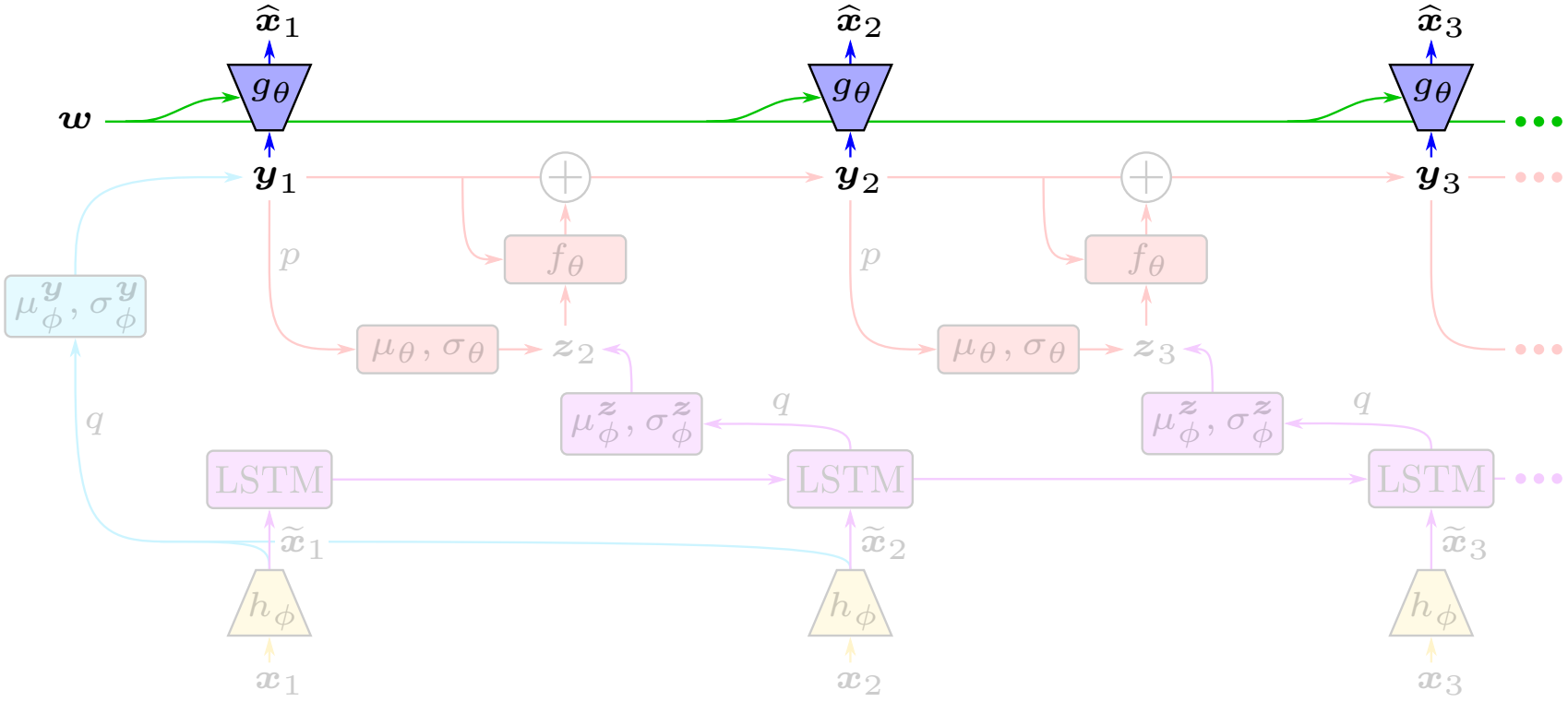
Inference / Conditioning Model



$$q_\phi(\boldsymbol{z}_{2:T}, \boldsymbol{y}_1 \mid \boldsymbol{x}_{1:T}) = \underbrace{q_\phi(\boldsymbol{y}_1 \mid \boldsymbol{x}_{1:k})}_{\text{Init. Cond.}} \prod_{t=2}^{T} \underbrace{q_\phi(\boldsymbol{z}_t \mid \boldsymbol{x}_{1:t})}_{\text{LSTM}}$$

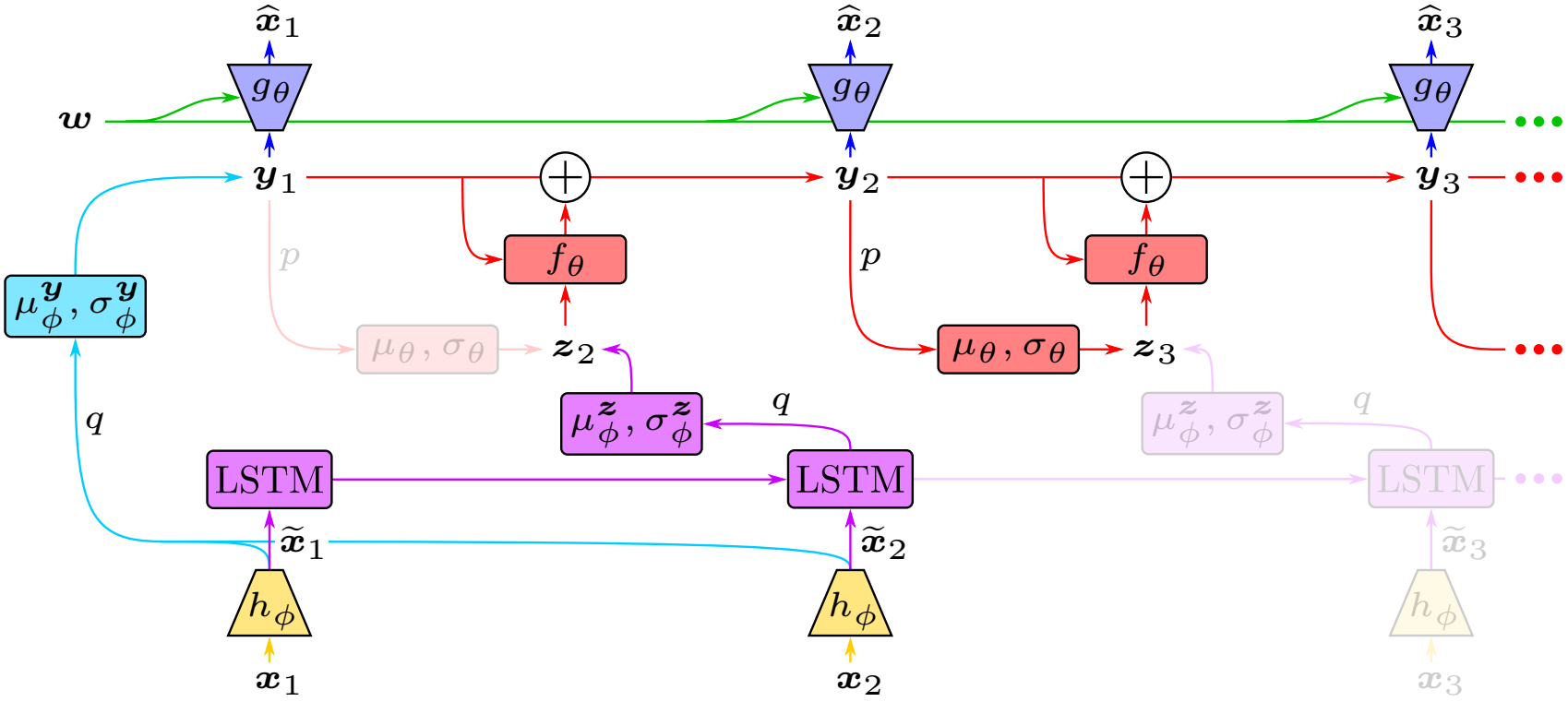
$\rightarrow$ Training consists in optimizing the corresponding ELBO.

Content Variable $\boldsymbol{w}$



- Store static information (e.g., background and object shapes).
- Outside ELBO (we consider $p_\theta(\boldsymbol{x} \mid \boldsymbol{w})$).
- Computed from randomly sampled frames $\rightarrow$ temporal invariance.

Evaluation

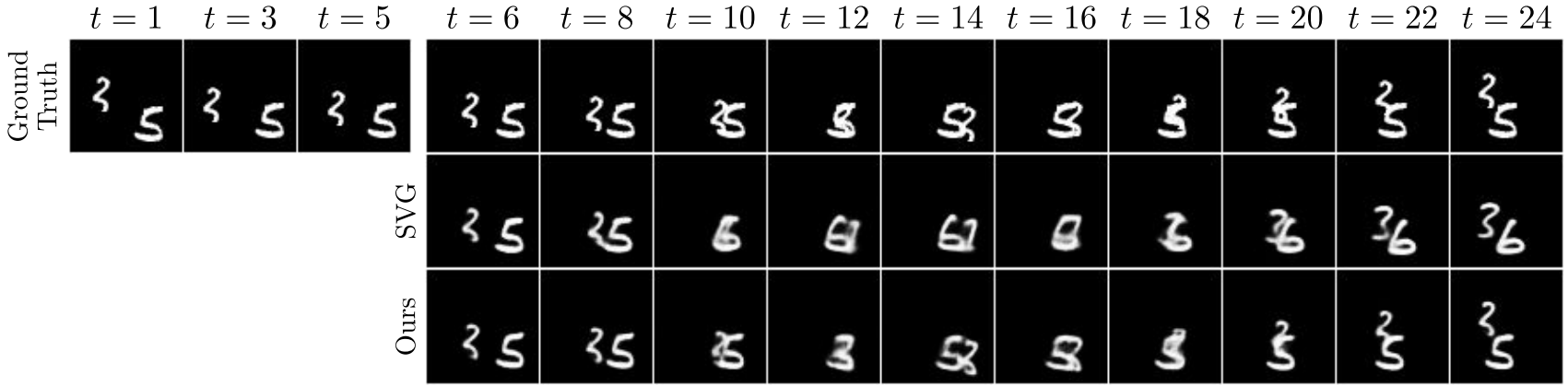


- Conditioning frames are used to infer dynamic variables and w.
- Prediction follows using the forward model.

Experimental Results: Moving MNIST

Models	Stochastic		Deterministic	
	PSNR	SSIM	PSNR	SSIM
SVG	14.50	0.7090	12.85	0.6185
Ours	**16.93**	**0.7799**	**18.25**	**0.8300**
Ours - GRU	15.80	0.7464	13.17	0.6237
Ours - MLP	16.55	0.7694	16.70	0.7876
Ours - w/o z	—	—	14.99	0.4757

SVG: Denton and Fergus (2018).

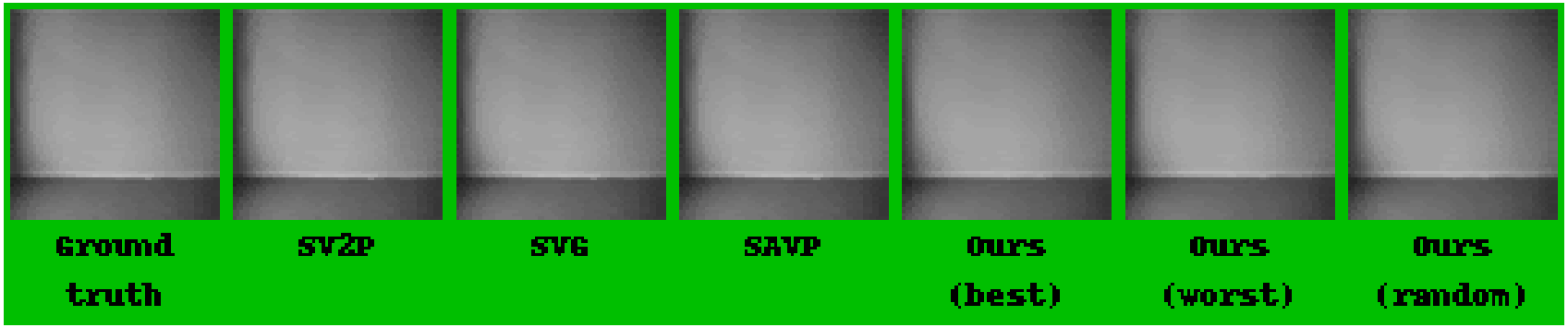


Experimental Results: KTH, Human3.6M and BAIR

Metric	Dataset	SV2P	SAVP	SVG	SVRNN	Ours
FVD (↓)	KTH	636	374	377	—	**222**
	H3.6M	—	—	—	556	**416**
	BAIR	965	**152**	255	—	163
LPIPS (↓)	KTH	0.2049	0.1120	0.0923	—	**0.0736**
	H3.6M	—	—	—	0.0557	**0.0509**
	BAIR	0.0912	0.0634	0.0609	—	**0.0574**
PSNR (↑)	KTH	28.19	26.51	28.06	—	**29.69**
	H3.6M	—	—	—	24.46	**25.30**
	BAIR	**20.39**	18.44	18.95	—	19.59

SV2P: Finn et al. (2016), SAVP: Babaeizadeh et al. (2018), SVRNN: Minderer et al. (2019).

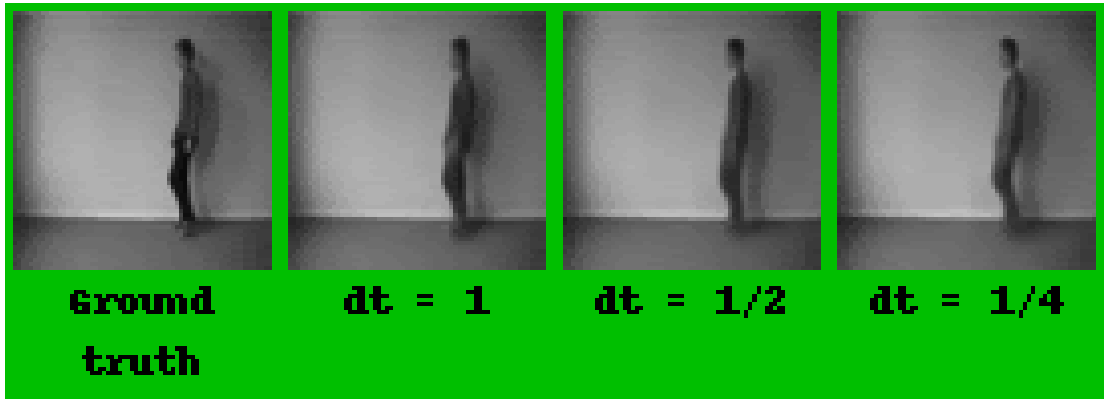
Experimental Results: Samples

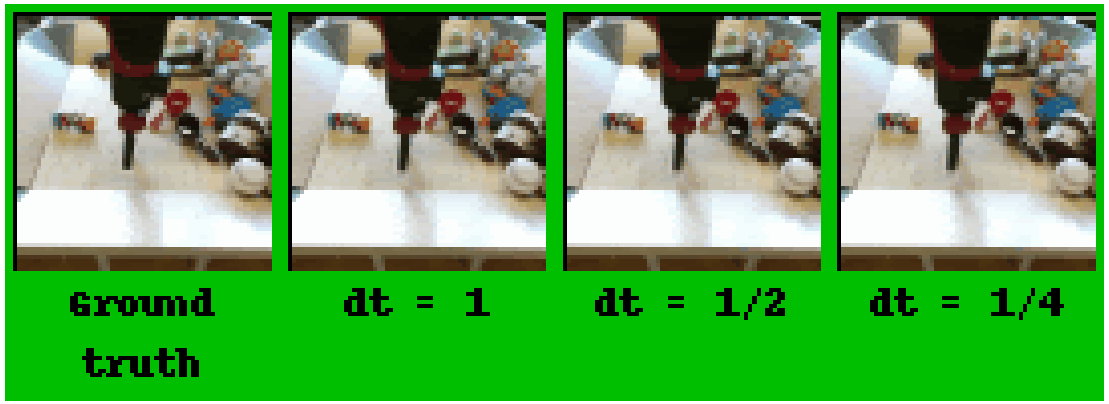


Experimental Results: Generation at Various Frame Rates

Using the Euler approximation scheme:

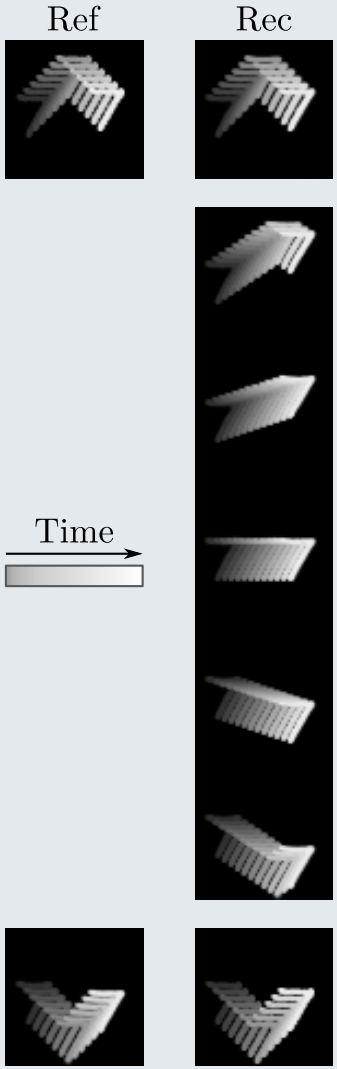
$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = f_\theta\Big(\boldsymbol{y}_t, \boldsymbol{z}_{\lfloor t \rfloor + 1}\Big) \quad \xRightarrow{\text{discr.}} \quad \boldsymbol{y}_{t+\Delta t} = \boldsymbol{y}_t + \Delta t \cdot f_\theta\Big(\boldsymbol{y}_t, \boldsymbol{z}_{\lfloor t \rfloor + 1}\Big).$$



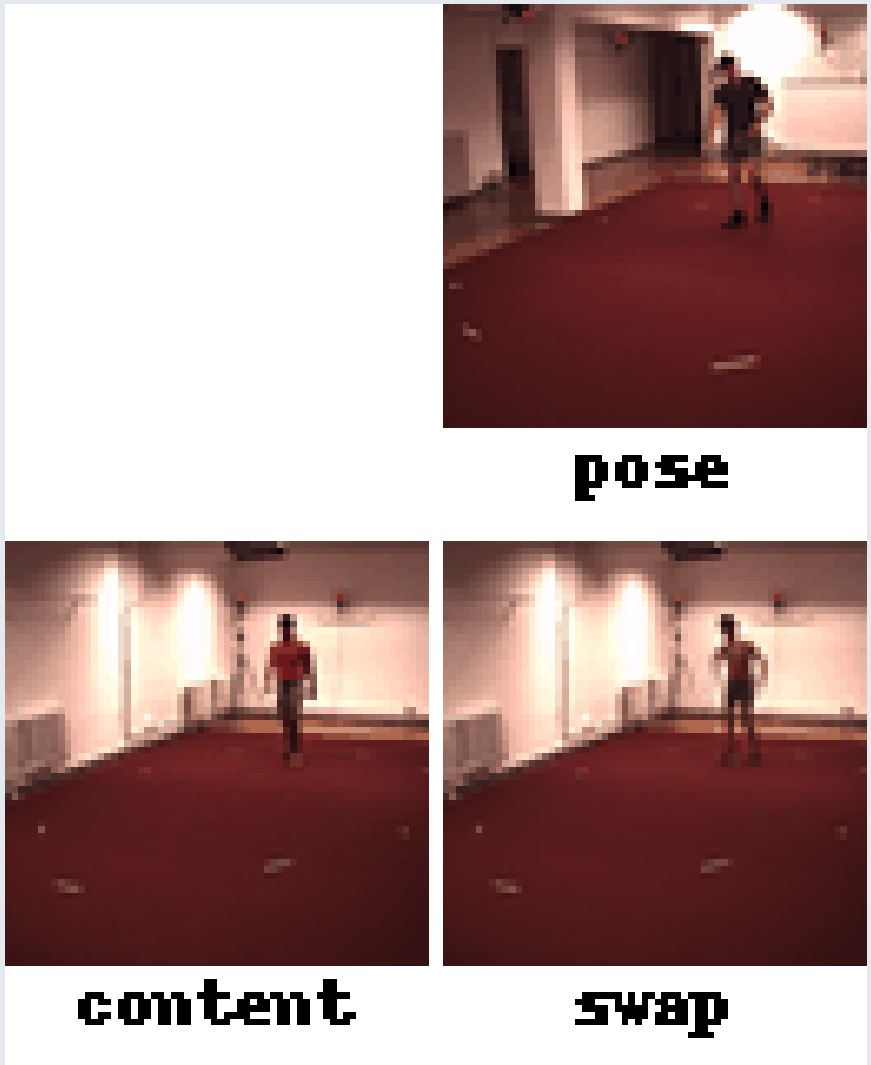


Experimental Results: Latent Space Properties

Dynamics interpolation



Content Swap



Second Contribution: Spatiotemporal Disentanglement

- The separation of dynamic variables $\boldsymbol{y}$ and the content variable $\boldsymbol{w}$ can be seen as spatiotemporal disentanglement.

Second Contribution: Spatiotemporal Disentanglement

- The separation of dynamic variables $\boldsymbol{y}$ and the content variable $\boldsymbol{w}$ can be seen as spatiotemporal disentanglement.
- Spatiotemporal disentanglement provides interpretability and can improve prediction performance.

Second Contribution: Spatiotemporal Disentanglement

- The separation of dynamic variables $\boldsymbol{y}$ and the content variable $\boldsymbol{w}$ can be seen as spatiotemporal disentanglement.
- Spatiotemporal disentanglement provides interpretability and can improve prediction performance.

Question

What is spatiotemporal disentanglement?

Second Contribution: Spatiotemporal Disentanglement

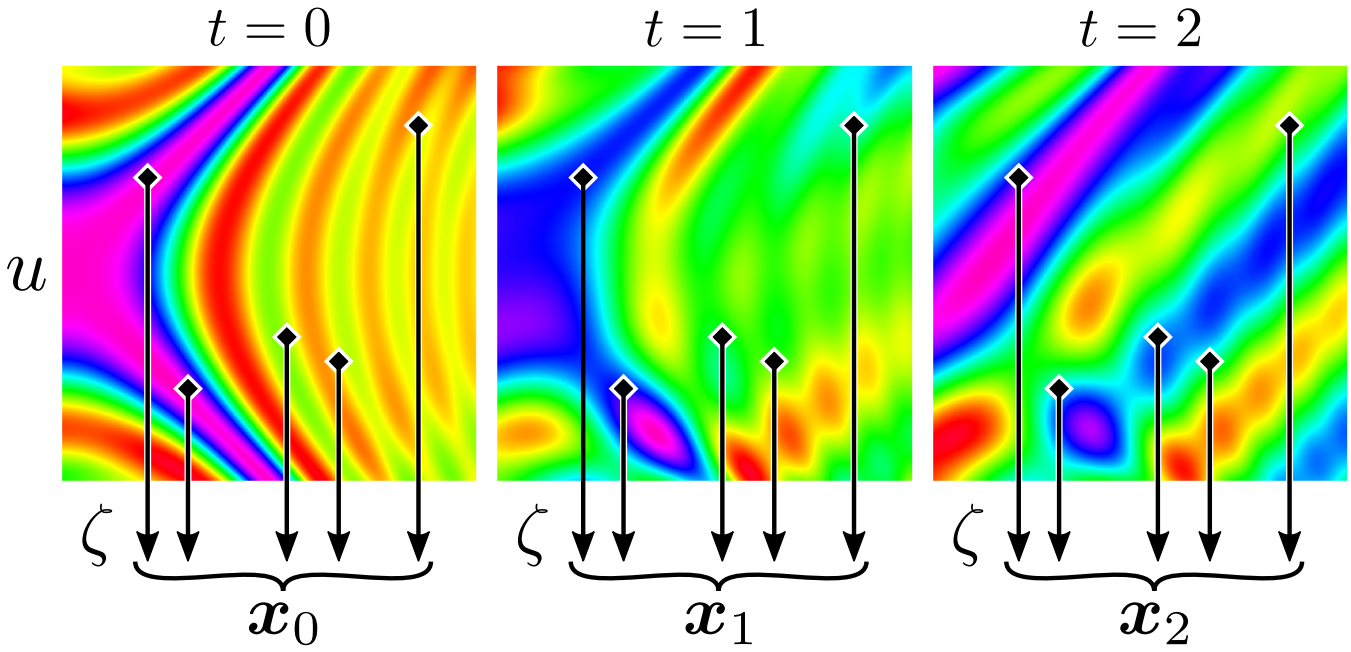
- The separation of dynamic variables $\boldsymbol{y}$ and the content variable $\boldsymbol{w}$ can be seen as spatiotemporal disentanglement.
- Spatiotemporal disentanglement provides interpretability and can improve prediction performance.

Question

What is spatiotemporal disentanglement?

- Often complex and seldom analyzed, achieved through:
  - KL-based separation in VAEs: ours, Hsieh et al. (2018) and Yingzhen et al. (2018);
  - adversarial losses: Villegas, Yang, et al. (2017) and Denton and Birodkar (2017).
- We aim at grounding spatiotemporal disentanglement on stronger foundations, with fewer implicit assumptions.

Deterministic Setting



Hidden State u

Function of continuous coordinates following a PDE:

$$u\colon (p, t) \mapsto u(p, t).$$

Ex.: physical state of an action, ocean temperature.

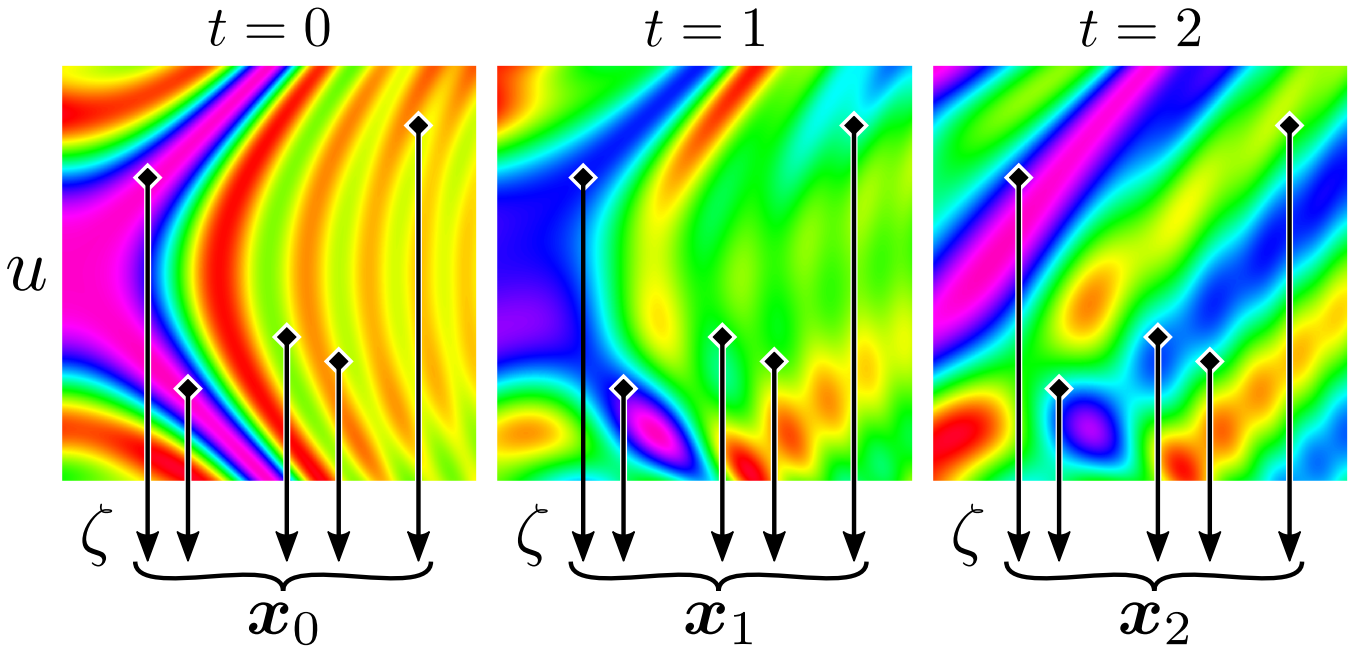
Observations $\boldsymbol{x}$

Vectorial $\boldsymbol{x}_{t_0}, \ldots, \boldsymbol{x}_{t_1}$, spatial measurements of u:

$$\boldsymbol{x}_t = \zeta \circ u(\cdot, t).$$

Ex.: pixel values, punctual surface temperatures.

Deterministic Setting



Hidden State u

Function of continuous coordinates following a PDE:

$$u\colon (p, t) \mapsto u(p, t).$$

Ex.: physical state of an action, ocean temperature.

Observations $\boldsymbol{x}$

Vectorial $\boldsymbol{x}_{t_0}, \ldots, \boldsymbol{x}_{t_1}$, spatial measurements of u:

$$\boldsymbol{x}_t = \zeta \circ u(\cdot, t).$$

Ex.: pixel values, punctual surface temperatures.

Spatiotemporal Disentanglement as Separation of Variables

Heat Equation

PDE: $$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial p^2}.$$

Separation Hypothesis

$$u(p,t) = w(p) \cdot y(t).$$

Spatiotemporal Disentanglement as Separation of Variables

SCIENCES SORBONNE UNIVERSITÉ

Heat Equation

PDE: $\dfrac{\partial u}{\partial t} = c^2 \dfrac{\partial^2 u}{\partial p^2}.$

Separation Hypothesis

$$u(p,t) = w(p) \cdot y(t).$$

- Hence:

$$c^2 \frac{w''(p)}{w(p)} = \frac{y'(t)}{y(t)} = \text{cst}.$$

ODE / PDE on w over p

$$w(p) = \mu \sin\left(\frac{n\pi}{L} p\right).$$

ODE on y over t

$$y(t) = \lambda \exp\left(-\left(\frac{cn\pi}{L}\right)^2 t\right).$$

Spatiotemporal Disentanglement as Separation of Variables

SCIENCES SORBONNE UNIVERSITÉ

Heat Equation

PDE: $\dfrac{\partial u}{\partial t} = c^2 \dfrac{\partial^2 u}{\partial p^2}$.

Separation Hypothesis

$$u(p,t) = w(p) \cdot y(t).$$

- Hence:

$$c^2 \frac{w''(p)}{w(p)} = \frac{y'(t)}{y(t)} = \text{cst}.$$

ODE / PDE on w over p

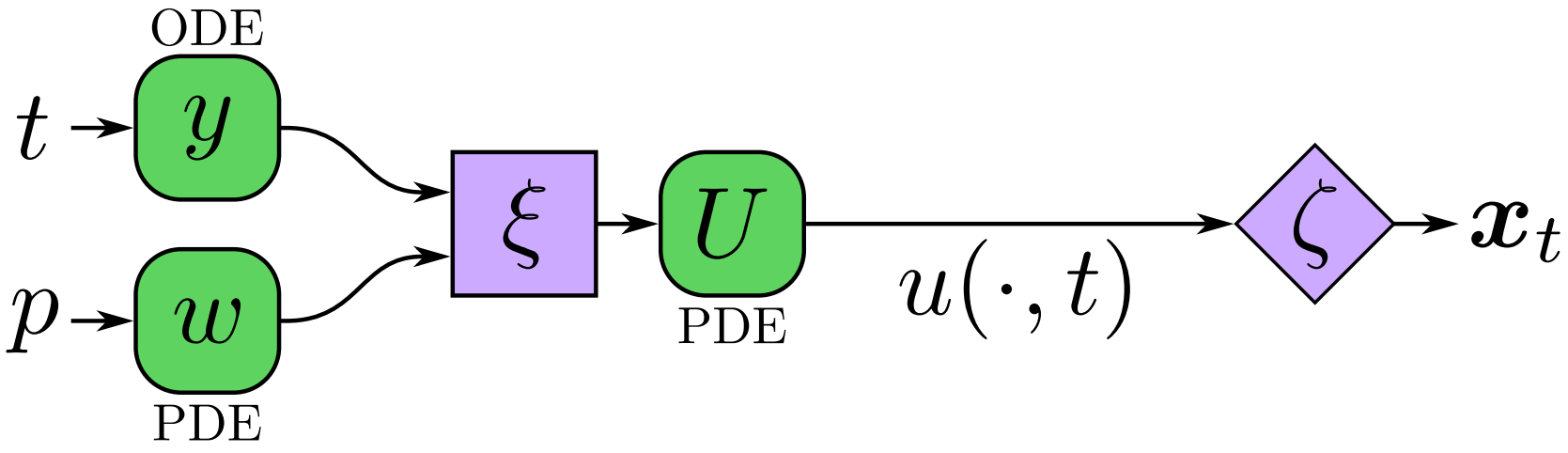
$$w(p) = \mu \sin\left(\frac{n\pi}{L} p\right).$$

ODE on y over t

$$y(t) = \lambda \exp\left(-\left(\frac{cn\pi}{L}\right)^2 t\right).$$

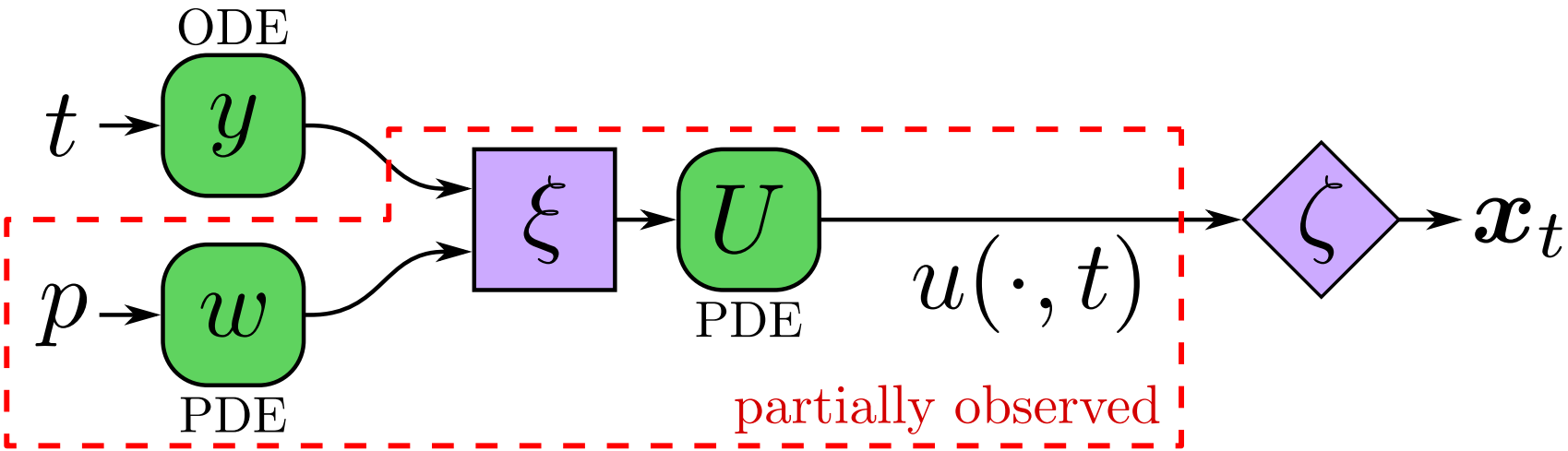
- Spatiotemporal disentanglement is a separation of variables in a PDE.
- We take inspiration from this technique to design a new model.

Partially Observable Scenario



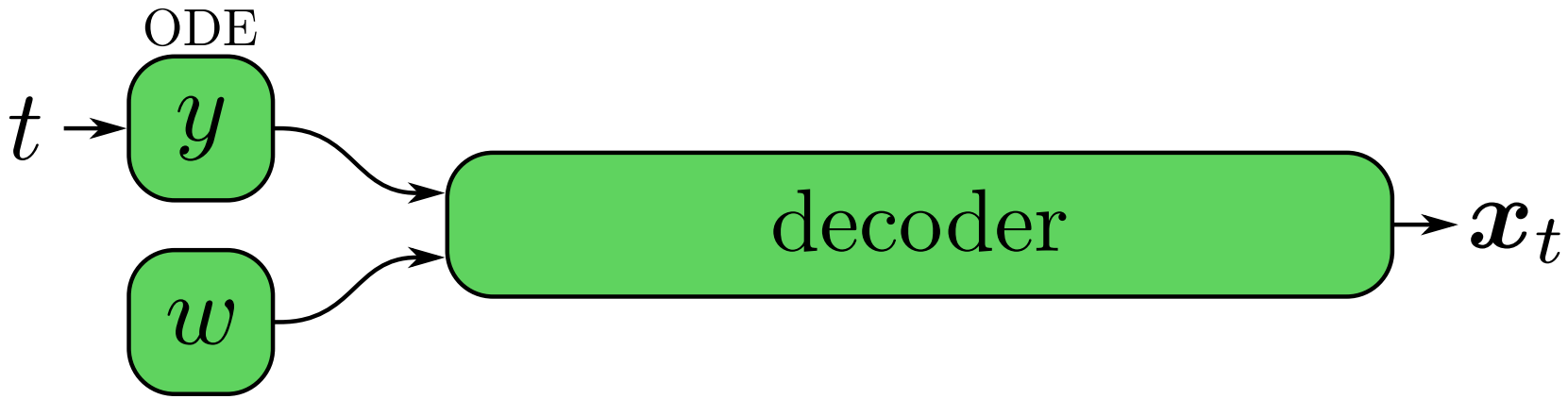
- Several functions to learn in the separation explicitly depend on spatial coordinates p.

Partially Observable Scenario



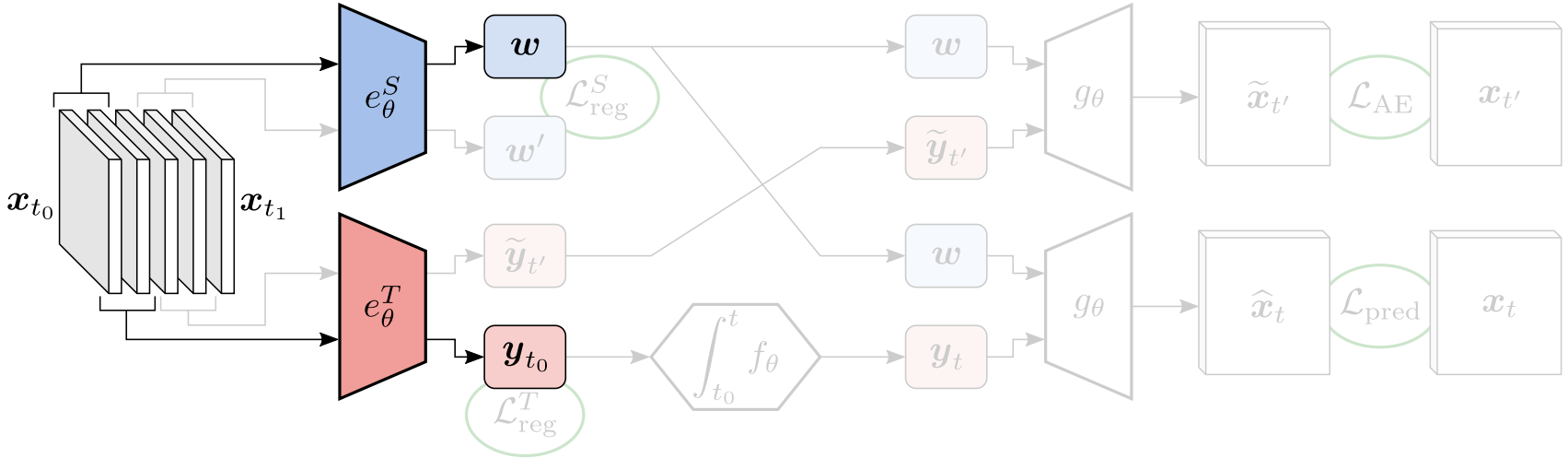
- Several functions to learn in the separation explicitly depend on spatial coordinates p.
- In a partially observed setting (through ζ), p are unknown.

Partially Observable Scenario



- Several functions to learn in the separation explicitly depend on spatial coordinates p.
- In a partially observed setting (through ζ), p are unknown.
- We choose to implicitly learn p-dependent functions through:
  - a representation $\boldsymbol{w}$ of $p \mapsto w(p)$;
  - a decoding step learning how to combine temporal and spatial solutions and the spatial measurement step.

Model: Encoding



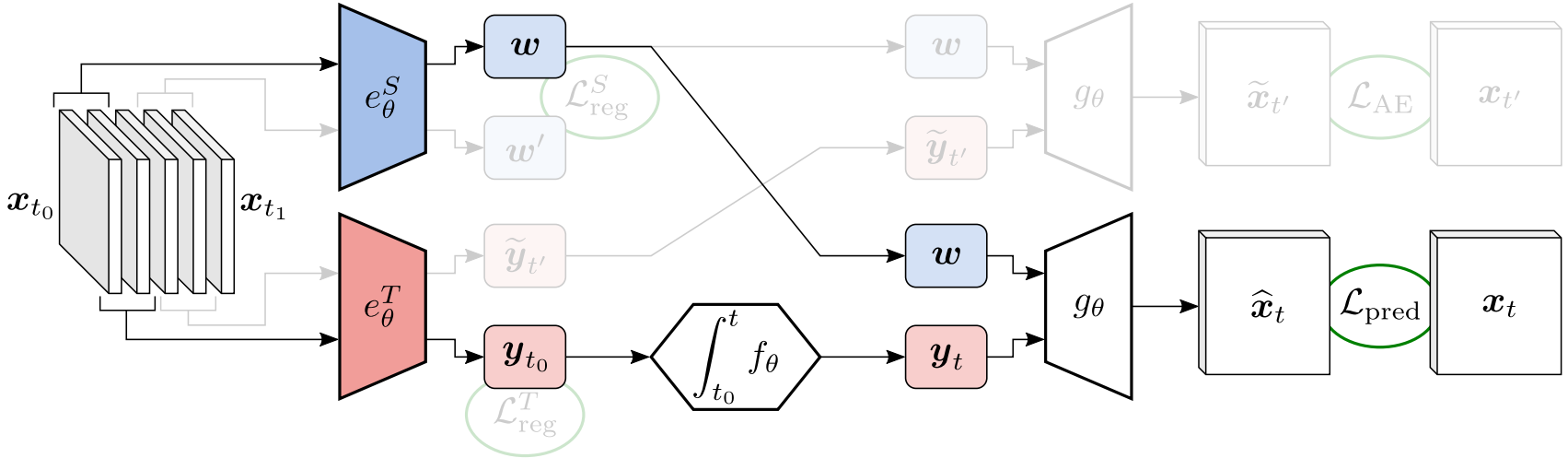
- $\boldsymbol{w}$ and $\boldsymbol{y}_{t_0}$ are inferred from conditioning frames $\boldsymbol{x}_{t_0::\tau} = \left(\boldsymbol{x}_{t_0}, \ldots, \boldsymbol{x}_{t_0+\tau}\right)$:

$$\boldsymbol{w} = e_\theta^S\left(\boldsymbol{x}_{t_0::\tau}\right), \qquad \boldsymbol{y}_{t_0} = e_\theta^T\left(\boldsymbol{x}_{t_0::\tau}\right).$$

- A decoder g_θ then outputs the predicted observations:

$$\widehat{\boldsymbol{x}}_t = g_\theta(\boldsymbol{w}, \boldsymbol{y}_t).$$

Model: Forecasting



- From the separation of variables, $\boldsymbol{y}_t$ follows an ODE:

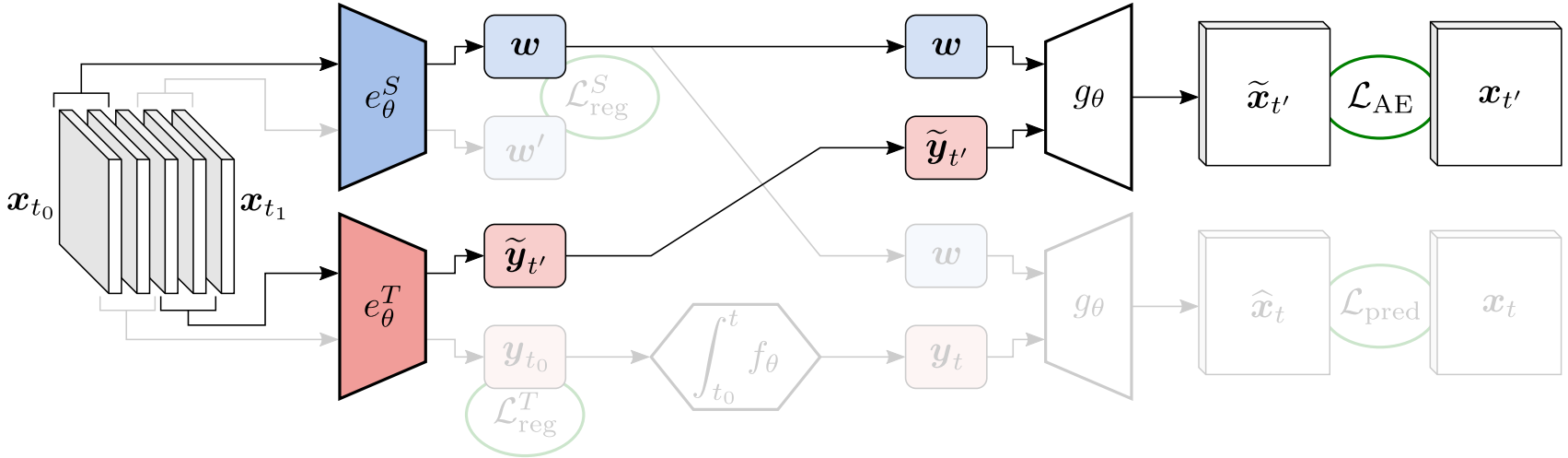
$$\frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = f_\theta(\boldsymbol{y}_t).$$

- Yields forecasting and alignment losses:

$$\mathcal{L}_{\text{pred}} = \sum_t \|\widehat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2^2, \quad \mathcal{L}_{\text{AE}} = \left\| g_\theta\left(\boldsymbol{w}, e_\theta^T(\boldsymbol{x}_{t'::\tau})\right) - \boldsymbol{x}_{t'} \right\|_2^2$$

- Similar structure as in our video prediction model.

Model: Forecasting



- From the separation of variables, $\boldsymbol{y}_t$ follows an ODE:

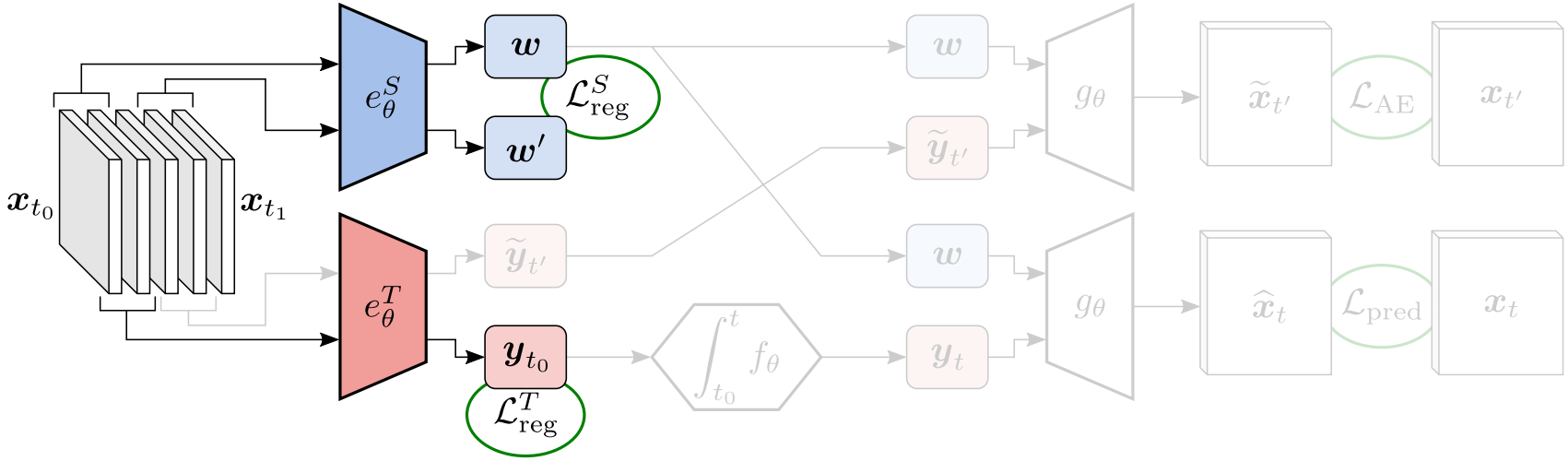
$$\frac{\mathrm{d}\boldsymbol{y}_t}{\mathrm{d}t} = f_\theta(\boldsymbol{y}_t).$$

- Yields forecasting and alignment losses:

$$\mathcal{L}_{\text{pred}} = \sum_t \|\widehat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2^2, \quad \mathcal{L}_{\text{AE}} = \left\| g_\theta\left(\boldsymbol{w}, e_\theta^T(\boldsymbol{x}_{t'::\tau})\right) - \boldsymbol{x}_{t'} \right\|_2^2$$

- Similar structure as in our video prediction model.

Model: Spatiotemporal Disentanglement



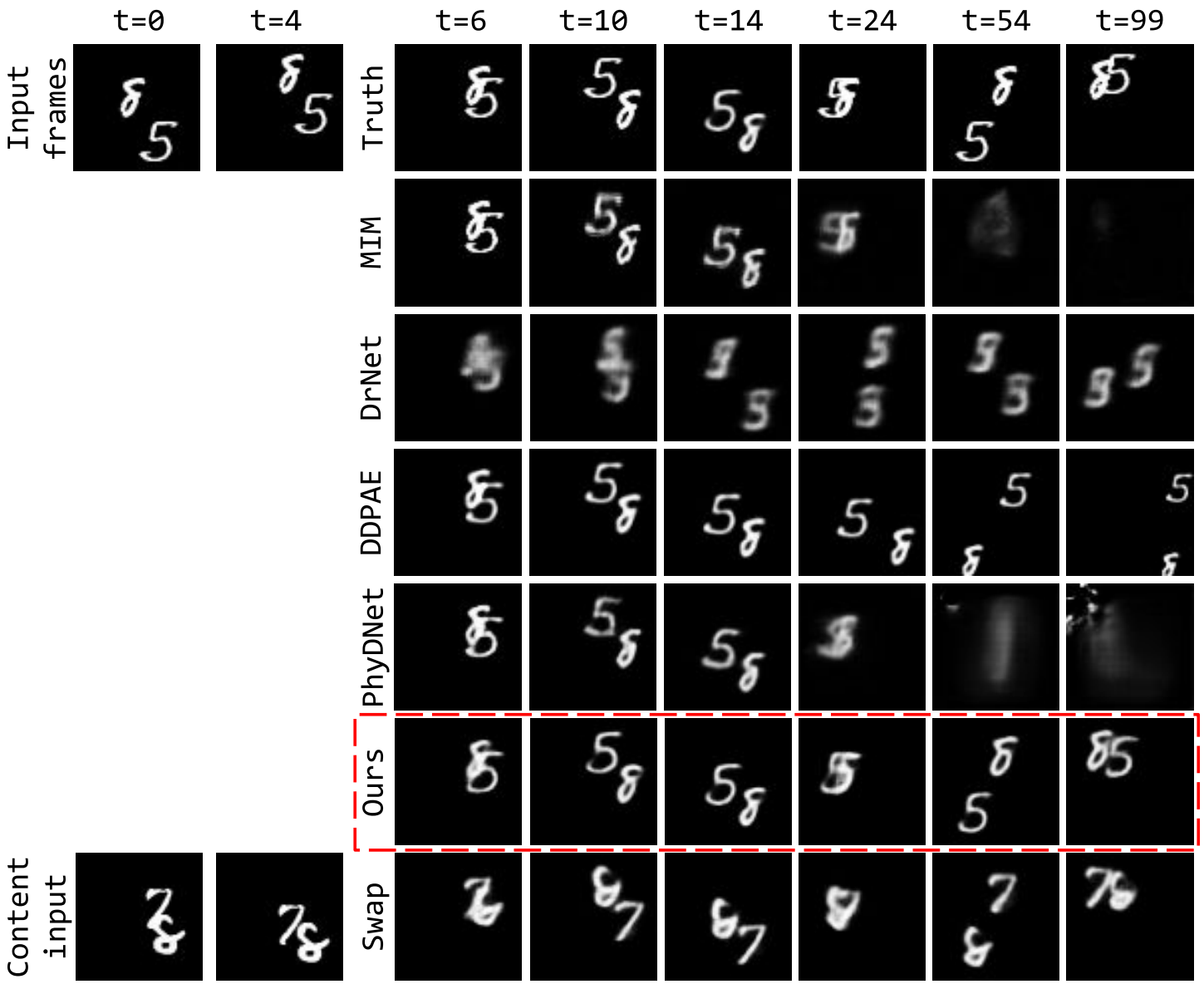
- From a strict $\boldsymbol{w}$ invariance constraint to a weaker one to take into account variations of observable content:

$$\frac{\partial e_\theta^S(\boldsymbol{x}_{t::\tau})}{\partial t} = 0 \Rightarrow \mathcal{L}_{\text{reg}}^S = \left\| e_\theta^S\left(\boldsymbol{x}_{t_0::\tau}\right) - e_\theta^S\left(\boldsymbol{x}_{t_1-\tau\Delta t::\tau}\right) \right\|_2^2$$

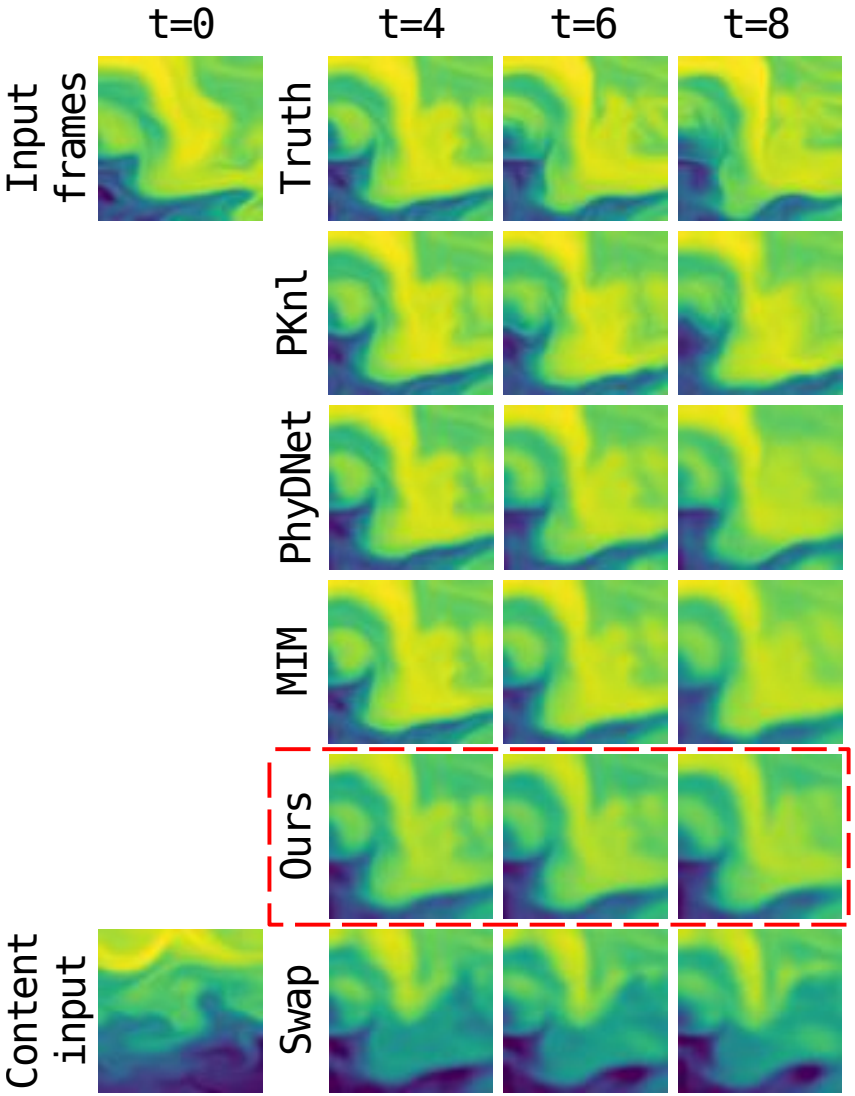
- Disentanglement loss:

$$\mathcal{L}_{\text{reg}}^T = \left\| \boldsymbol{y}_{t_0} \right\|_2^2 = \left\| e_\theta^T\left(\boldsymbol{x}_{t_0::\tau}\right) \right\|_2^2$$

Experimental Results: Prediction & Disentanglement



Experimental Results: Prediction & Disentanglement



State-Space Predictive Models: Conclusion

- We study the relevance of state-space models for spatiotemporal data, with a focus on:
  - stochastic generation for videos;
  - variable separation for physical phenomena.
- Such models allow us to simulatenously perform accurate prediction and learn meaningful representations.
- We show the important role of modeling via dynamical systems for the performance of these temporal models.

Further Resources: code, models, animated samples

- `https://sites.google.com/view/srvp/`
- `https://github.com/JeremDona/spatiotemporal_variable_separation`

Analyzing of the Training Dynamics of Generative Models

Franceschi, de Bézenac, Ayed et al. "**A Neural Tangent Kernel Perspective of GANs**". 2021. Preprint. Under review.

Goal

Theoretical analysis of a popular class of generative models, GANs, to better understand them.

Our approach:

1. **problem:** flaws of previous analyses;
2. **proposed solution:** modeling the discriminator's training dynamics;
3. **results**, including:
   - generator training dynamics;
   - IPMs and NTK MMD;
   - architecture comparisons.

Goal

Theoretical analysis of a popular class of generative models, GANs, to better understand them.

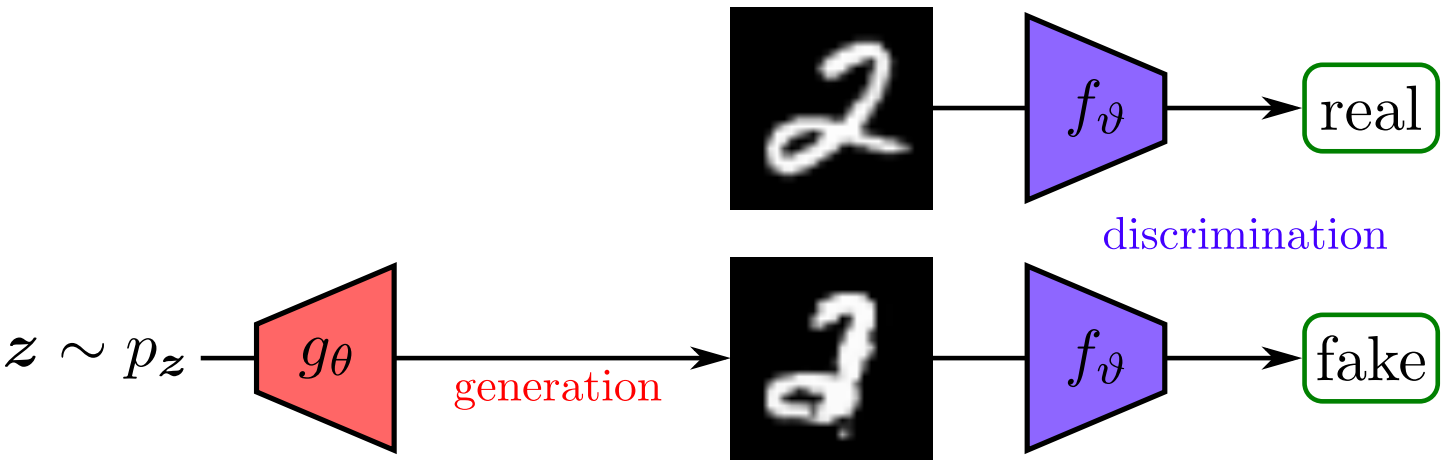
Our approach:

1. **problem:** flaws of previous analyses;
2. **proposed solution:** modeling the discriminator's training dynamics;
3. **results**, including:
   - generator training dynamics;
   - IPMs and NTK MMD;
   - architecture comparisons.

Principle

- Generator g generating a fake distribution α_g that should imitate a target β.
- Competing with a discriminator f trained to distinguish between fake and target samples:

$$\inf_g \sup_f \mathcal{L}(f, g).$$



- Vanilla GAN (Goodfellow et al., 2014), with $\mathcal{F} = L^2(\Omega)$:

$$\inf_g \sup_{f \in \mathcal{F}} \quad \mathbb{E}_{\boldsymbol{x} \sim \alpha_g}\Big[\log\Big(1 - \sigma\big(f(\boldsymbol{x})\big)\Big)\Big] + \mathbb{E}_{\boldsymbol{y} \sim \beta}\Big[\log \sigma\big(f(\boldsymbol{y})\big)\Big]$$
$$= \inf_g \mathrm{JS}\big(\alpha_g, \beta\big).$$

The Vanilla and Wasserstein GAN Analyses

- Vanilla GAN (Goodfellow et al., 2014), with $\mathcal{F} = L^2(\Omega)$:

$$\inf_g \sup_{f \in \mathcal{F}} \quad \mathbb{E}_{\boldsymbol{x} \sim \alpha_g} \Big[\log \Big(1 - \sigma \big(f(\boldsymbol{x}) \big) \Big) \Big] + \mathbb{E}_{\boldsymbol{y} \sim \beta} \Big[\log \sigma \big(f(\boldsymbol{y}) \big) \Big]$$
$$= \inf_g \mathrm{JS}\big(\alpha_g, \beta\big).$$

- Wasserstein GAN reasoning (Arjovsky et al., 2017):
  - $\mathrm{JS}\big(\alpha_g, \beta\big)$ is constant in many cases;
  - no gradient-based optimization of g;
  - alternative criterion to solve the problem ($\mathcal{W}^1$ instead of JS).

The Vanilla and Wasserstein GAN Analyses

- Vanilla GAN (Goodfellow et al., 2014), with $\mathcal{F} = L^2(\Omega)$:

$$\inf_g \sup_{f \in \mathcal{F}} \quad \mathbb{E}_{\boldsymbol{x} \sim \alpha_g}\Big[\log\Big(1 - \sigma\big(f(\boldsymbol{x})\big)\Big)\Big] + \mathbb{E}_{\boldsymbol{y} \sim \beta}\Big[\log \sigma\big(f(\boldsymbol{y})\big)\Big]$$
$$= \inf_g \mathrm{JS}\big(\alpha_g, \beta\big).$$

- Wasserstein GAN reasoning (Arjovsky et al., 2017):
  - $\mathrm{JS}\big(\alpha_g, \beta\big)$ is constant in many cases;
  - no gradient-based optimization of g;
  - alternative criterion to solve the problem ($\mathcal{W}^1$ instead of JS).

Problem

Flawed analysis of GAN practice: vanilla GAN is trainable by gradient descent.

First Issue: Min / Max vs. Alternating Optimization

Min / Max: Optimal Discriminator f_{g_θ} (Metz et al., 2017)

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\Big[\big(a\circ f_{g_\theta}\big)(\boldsymbol{x})\Big] - \mathbb{E}_{\boldsymbol{y}\sim\beta}\Big[\big(b\circ f_{g_\theta}\big)(\boldsymbol{y})\Big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f_{g_\theta}}\right)(\boldsymbol{x})\right] - \nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f_{g_\theta}}\right)(\boldsymbol{y})\right].$$

∇_θ

Discriminator and Generator Optimized Independently

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\big[(a\circ f)(\boldsymbol{x})\big] - \mathbb{E}_{\boldsymbol{y}\sim\beta}\big[(b\circ f)(\boldsymbol{y})\big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f}\right)(\boldsymbol{x})\right] - \nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f}\right)(\boldsymbol{y})\right].$$

∇_θ

Consequence

Altering the gradient changes the loss $\mathscr{C}$ minimized by the generator.

First Issue: Min / Max vs. Alternating Optimization

Min / Max: Optimal Discriminator f_{g_θ} (Metz et al., 2017)

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\Big[\big(a\circ f_{g_\theta}\big)(\boldsymbol{x})\Big]-\mathbb{E}_{\boldsymbol{y}\sim\beta}\Big[\big(b\circ f_{g_\theta}\big)(\boldsymbol{y})\Big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f_{g_\theta}}\right)(\boldsymbol{x})\right]-\nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f_{g_\theta}}\right)(\boldsymbol{y})\right].$$

∇_θ

Discriminator and Generator Optimized Independently

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\big[(a\circ f)(\boldsymbol{x})\big]-\mathbb{E}_{\boldsymbol{y}\sim\beta}\big[(b\circ f)(\boldsymbol{y})\big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f}\right)(\boldsymbol{x})\right]-\nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f}\right)(\boldsymbol{y})\right].$$

∇_θ

Consequence

Altering the gradient changes the loss $\mathscr{C}$ minimized by the generator.

First Issue: Min / Max vs. Alternating Optimization

Min / Max: Optimal Discriminator f_{g_θ} (Metz et al., 2017)

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\Big[\big(a\circ f_{g_\theta}\big)(\boldsymbol{x})\Big] - \mathbb{E}_{\boldsymbol{y}\sim\beta}\Big[\big(b\circ f_{g_\theta}\big)(\boldsymbol{y})\Big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f_{g_\theta}}\right)(\boldsymbol{x})\right] - \nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f_{g_\theta}}\right)(\boldsymbol{y})\right].$$

∇_θ

Discriminator and Generator Optimized Independently

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\big[(a\circ f)(\boldsymbol{x})\big] - \mathbb{E}_{\boldsymbol{y}\sim\beta}\big[(b\circ f)(\boldsymbol{y})\big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f}\right)(\boldsymbol{x})\right] - \cancel{\nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f}\right)(\boldsymbol{y})\right]}.$$

∇_θ

Consequence

Altering the gradient changes the loss $\mathscr{C}$ minimized by the generator.

First Issue: Min / Max vs. Alternating Optimization

Min / Max: Optimal Discriminator f_{g_θ} (Metz et al., 2017)

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\Big[\big(a\circ f_{g_\theta}\big)(\boldsymbol{x})\Big] - \mathbb{E}_{\boldsymbol{y}\sim\beta}\Big[\big(b\circ f_{g_\theta}\big)(\boldsymbol{y})\Big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f_{g_\theta}}\right)(\boldsymbol{x})\right] - \nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f_{g_\theta}}\right)(\boldsymbol{y})\right].$$

∇_θ

Discriminator and Generator Optimized Independently

$$\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\big[(a\circ f)(\boldsymbol{x})\big] - \mathbb{E}_{\boldsymbol{y}\sim\beta}\big[(b\circ f)(\boldsymbol{y})\big]$$

$$\nabla_\theta\mathbb{E}_{\boldsymbol{x}\sim\alpha_{g_\theta}}\left[\left(a\circ \underline{f}\right)(\boldsymbol{x})\right] - \cancel{\nabla_\theta\mathbb{E}_{\boldsymbol{y}\sim\beta}\left[\left(b\circ \underline{f}\right)(\boldsymbol{y})\right]}.$$

∇_θ

Consequence

Altering the gradient changes the loss $\mathscr{C}$ minimized by the generator.

Question

- Generator gradients correspond to some loss $\mathscr{C}$.
- $\mathscr{C}$ is neither the Jensen-Shannon nor $\mathcal{W}^1$.
- What is the loss $\mathscr{C}$ induced by the discriminator?

Question

- Generator gradients correspond to some loss $\mathscr{C}$.
- $\mathscr{C}$ is neither the Jensen-Shannon nor $\mathcal{W}^1$.
- What is the loss $\mathscr{C}$ induced by the discriminator?

Optimization of g_θ for a discriminator f via the chain rule:

$$\theta_{i+1} = \theta_i - \eta \nabla_{\theta_i} \left(\mathbb{E}_{\boldsymbol{x} \sim \alpha_{g_{\theta_i}}} \left[(a \circ f)(\boldsymbol{x}) \right] \right).$$

Question

- Generator gradients correspond to some loss $\mathscr{C}$.
- $\mathscr{C}$ is neither the Jensen-Shannon nor $\mathcal{W}^1$.
- What is the loss $\mathscr{C}$ induced by the discriminator?

Optimization of g_θ for a discriminator f via the chain rule:

$$\theta_{i+1} = \theta_i - \eta \nabla_{\theta_i} \left(\mathbb{E}_{\boldsymbol{x} \sim \alpha_{g_{\theta_i}}} \left[(a \circ f)(\boldsymbol{x}) \right] \right).$$

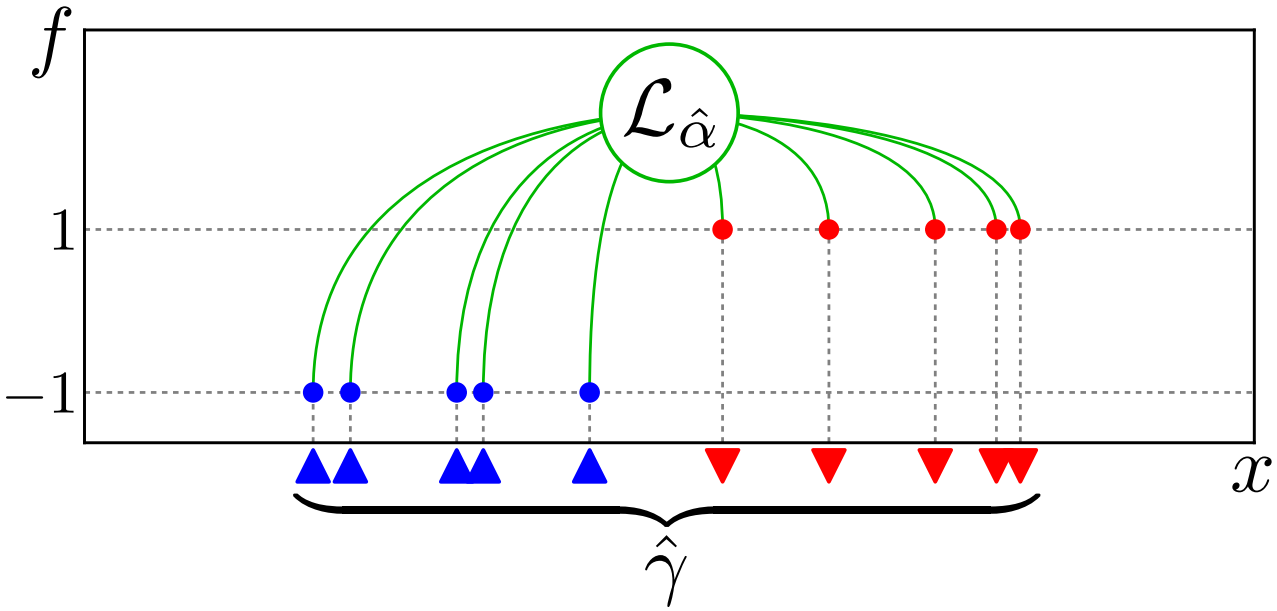
Consequence

∇f is crucial for generator training and should be taken into account in a sound theoretical framework.

Discriminator Inner-Loop Optimization

Discrete training distribution $\hat{\gamma} = \frac{\hat{\alpha}_g + \hat{\beta}}{2}$, with objective:

$$\sup_{f \in L^2(\Omega)} \left\{ \mathcal{L}_{\hat{\alpha}_g}(f) \triangleq \mathbb{E}_{\boldsymbol{x} \sim \hat{\alpha}_g}\left[(a \circ f)(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{y} \sim \hat{\beta}}\left[(b \circ f)(\boldsymbol{y})\right] \right\}.$$



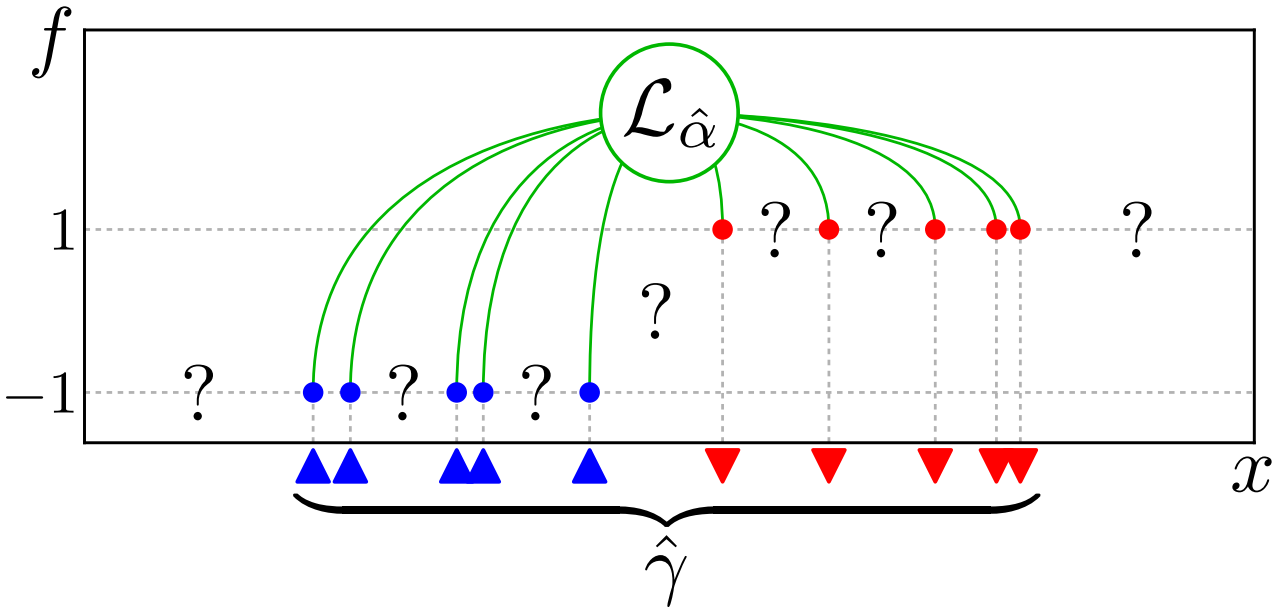
Discriminator Inner-Loop Optimization

Discrete training distribution $\hat{\gamma} = \frac{\hat{\alpha}_g + \hat{\beta}}{2}$, with objective:

$$\sup_{f \in L^2(\Omega)} \left\{ \mathcal{L}_{\hat{\alpha}_g}(f) \triangleq \mathbb{E}_{\boldsymbol{x} \sim \hat{\alpha}_g}\left[(a \circ f)(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{y} \sim \hat{\beta}}\left[(b \circ f)(\boldsymbol{y})\right] \right\}.$$

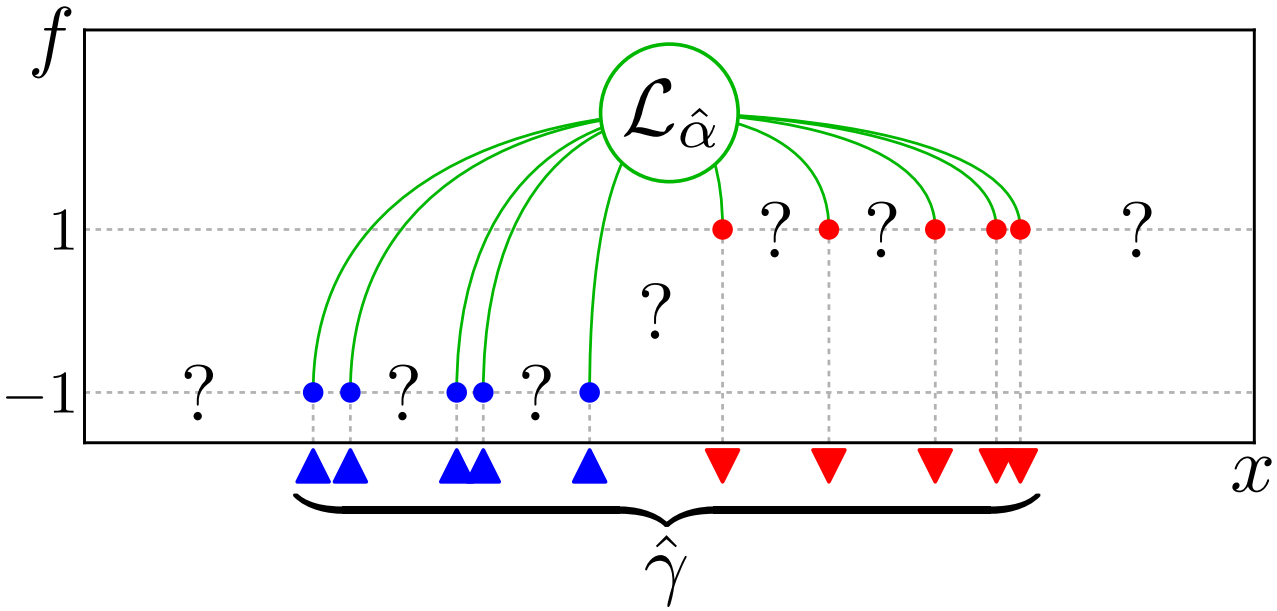
Problem

Since $\hat{\gamma}$ is discrete, ∇f is ill-defined.



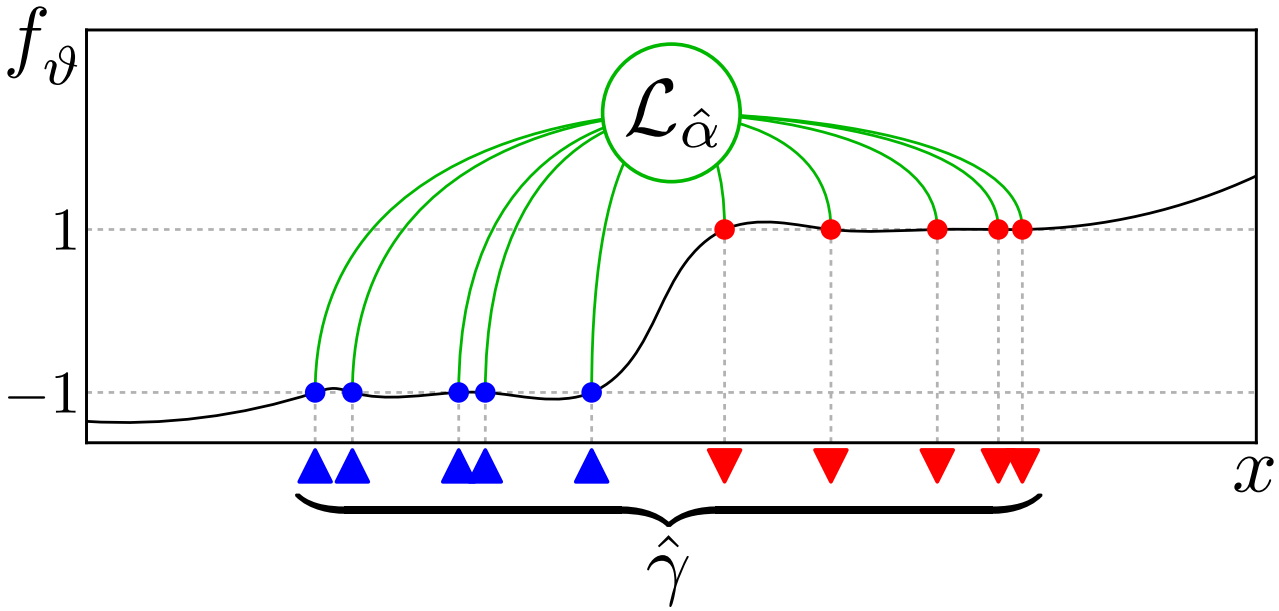
Solution: Modeling Neural Discriminator Training

- In practice, $f = f_\vartheta$ is a neural network of a given architecture trained by gradient ascent to maximize $\mathcal{L}_{\hat{\alpha}}$.



Solution: Modeling Neural Discriminator Training

- In practice, $f = f_\vartheta$ is a neural network of a given architecture trained by gradient ascent to maximize $\mathcal{L}_{\hat{\alpha}}$.
- The training procedure and choice of architecture specify the values and gradients of the discriminator everywhere.

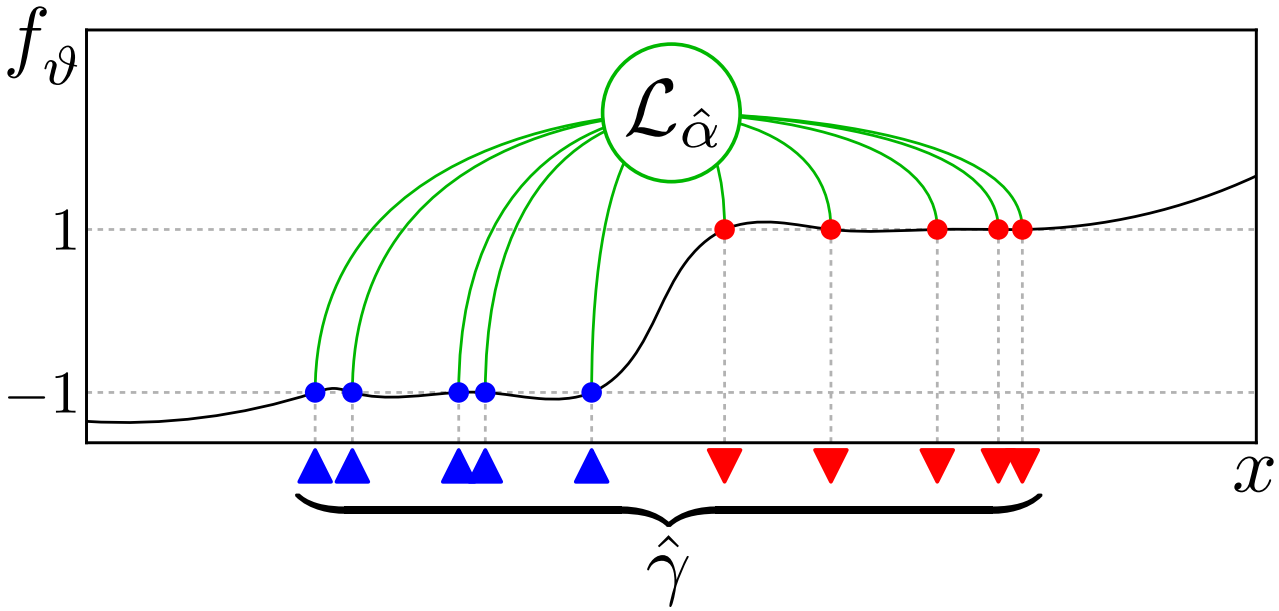


Solution: Modeling Neural Discriminator Training

- In practice, $f = f_\vartheta$ is a neural network of a given architecture trained by gradient ascent to maximize $\mathcal{L}_{\hat{\alpha}}$.
- The training procedure and choice of architecture specify the values and gradients of the discriminator everywhere.

Modeling Choice

We choose to model this complex parameterization with the powerful theory of Neural Tangent Kernels (NTKs).



Goal

Theoretical analysis of a popular class of generative models, GANs, to better understand them.

Our approach:

1. **problem:** flaws of previous analyses;
2. **proposed solution:** modeling the discriminator's training dynamics;
3. **results**, including:
   - generator training dynamics;
   - IPMs and NTK MMD;
   - architecture comparisons.

Infinite-Width Regime

- Kernel k called the NTK of f_ϑ (Jacot et al., 2018).
- Used through the integral operator $\mathcal{T}_{k,\hat{\gamma}}$ (from kernel theory).

Infinite-Width Regime

- Kernel k called the NTK of f_ϑ (Jacot et al., 2018).
- Used through the integral operator $\mathcal{T}_{k,\hat{\gamma}}$ (from kernel theory).

- k is architecture-dependent;
- $\mathcal{T}_{k,\hat{\gamma}}$ encapsulates the inductive biases of the discriminator.

Infinite-Width Regime

- Kernel k called the NTK of f_ϑ (Jacot et al., 2018).
- Used through the integral operator $\mathcal{T}_{k,\hat{\gamma}}$ (from kernel theory).

- k is architecture-dependent;
- $\mathcal{T}_{k,\hat{\gamma}}$ encapsulates the inductive biases of the discriminator.

Discriminator Inner Loop Training (Continuous-Time)

$$\partial_t f_t = \overbrace{\mathcal{T}_{k,\hat{\gamma}}\Big(\underbrace{\nabla^{\hat{\gamma}}\mathcal{L}_{\hat{\alpha}}(f_t)}_{\text{defined over } \hat{\gamma}}\Big)}^{\text{defined everywhere}}.$$

Relevance & Consequences on the Generator Dynamics

The discriminator gradients are well-defined, thereby making this framework closer to GAN practice.

Theorem

f_t is infinitely differentiable (almost) everywhere.

Relevance & Consequences on the Generator Dynamics

The discriminator gradients are well-defined, thereby making this framework closer to GAN practice.

Theorem

f_t is infinitely differentiable (almost) everywhere.

- Evolution of the generator g_ℓ in continuous-time, with NTK k_g:

$$\partial_\ell g_\ell = -\mathcal{T}_{k_g, p_{\boldsymbol{z}}} \left(\boldsymbol{z} \mapsto \nabla_{\boldsymbol{x}} \left(a \circ f_{\hat{\alpha}_{g_\ell}} \right)(\boldsymbol{x}) \Big|_{\boldsymbol{x} = g_\ell(\boldsymbol{z})} \right).$$

- Complex dynamics, cf. Stein gradient flows (Duncan et al., 2019).

Relevance & Consequences on the Generator Dynamics

The discriminator gradients are well-defined, thereby making this framework closer to GAN practice.

Theorem

f_t is infinitely differentiable (almost) everywhere.

- Evolution of the generator g_ℓ in continuous-time, with NTK k_g:

$$\partial_\ell g_\ell = -\mathcal{T}_{k_g, p_{\boldsymbol{z}}} \left(\boldsymbol{z} \mapsto \nabla_{\boldsymbol{x}} \left(a \circ f_{\hat{\alpha}_{g_\ell}} \right)(\boldsymbol{x}) \Big|_{\boldsymbol{x} = g_\ell(\boldsymbol{z})} \right).$$

- Complex dynamics, cf. Stein gradient flows (Duncan et al., 2019).

Consequence on $\mathscr{C}(g_\ell)$

$\mathscr{C}(g_\ell)$ is non-increasing during training.

Goal

Theoretical analysis of a popular class of generative models, GANs, to better understand them.

Our approach:

1. **problem:** flaws of previous analyses;
2. **proposed solution:** modeling the discriminator's training dynamics;
3. **results**, including:
   - generator training dynamics;
   - IPMs and NTK MMD;
   - architecture comparisons.

Case Study: IPM loss

- Suppose that:

$$\mathcal{L}_{\hat{\alpha}}(f) = \mathbb{E}_{\boldsymbol{x}\sim\hat{\alpha}}\big[f(\boldsymbol{x})\big] - \mathbb{E}_{\boldsymbol{y}\sim\hat{\beta}}\big[f(\boldsymbol{y})\big], \qquad f_0 = 0.$$

Case Study: IPM loss

- Suppose that:

$$\mathcal{L}_{\hat{\alpha}}(f) = \mathbb{E}_{\boldsymbol{x}\sim\hat{\alpha}}\big[f(\boldsymbol{x})\big] - \mathbb{E}_{\boldsymbol{y}\sim\hat{\beta}}\big[f(\boldsymbol{y})\big], \qquad f_0 = 0.$$

- Then:

$$\mathcal{L}_{\hat{\alpha}}(f_t) = t \cdot \mathrm{MMD}_k^2\Big(\hat{\alpha}, \hat{\beta}\Big).$$

Consequences

- $\mathscr{C}$ is the squared MMD defined with the NTK of f between fake and target distributions.
- $\mathscr{C}$ depends on the discriminator's architecture.
- Paves the way for convergence results (Arbel et al., 2019; Mroueh et al., 2021).

$\rightarrow$ Novel results made possible by our NTK framework.

Case Study: IPM loss

- Suppose that:

$$\mathcal{L}_{\hat{\alpha}}(f) = \mathbb{E}_{\boldsymbol{x}\sim\hat{\alpha}}\big[f(\boldsymbol{x})\big] - \mathbb{E}_{\boldsymbol{y}\sim\hat{\beta}}\big[f(\boldsymbol{y})\big], \qquad f_0 = 0.$$

- Then:

$$\mathcal{L}_{\hat{\alpha}}(f_t) = t \cdot \mathrm{MMD}_k^2\Big(\hat{\alpha}, \hat{\beta}\Big).$$

Consequences

- $\mathscr{C}$ is the squared MMD defined with the NTK of f between fake and target distributions.
- $\mathscr{C}$ depends on the discriminator's architecture.
- Paves the way for convergence results (Arbel et al., 2019; Mroueh et al., 2021).

$\rightarrow$ Novel results made possible by our NTK framework.

Case Study: IPM loss

- Suppose that:

$$\mathcal{L}_{\hat{\alpha}}(f) = \mathbb{E}_{\boldsymbol{x}\sim\hat{\alpha}}\big[f(\boldsymbol{x})\big] - \mathbb{E}_{\boldsymbol{y}\sim\hat{\beta}}\big[f(\boldsymbol{y})\big], \qquad f_0 = 0.$$

- Then:

$$\mathcal{L}_{\hat{\alpha}}(f_t) = t \cdot \mathrm{MMD}_k^2\Big(\hat{\alpha}, \hat{\beta}\Big).$$

Consequences

- $\mathscr{C}$ is the squared MMD defined with the NTK of f between fake and target distributions.
- $\mathscr{C}$ depends on the discriminator's architecture.
- Paves the way for convergence results (Arbel et al., 2019; Mroueh et al., 2021).

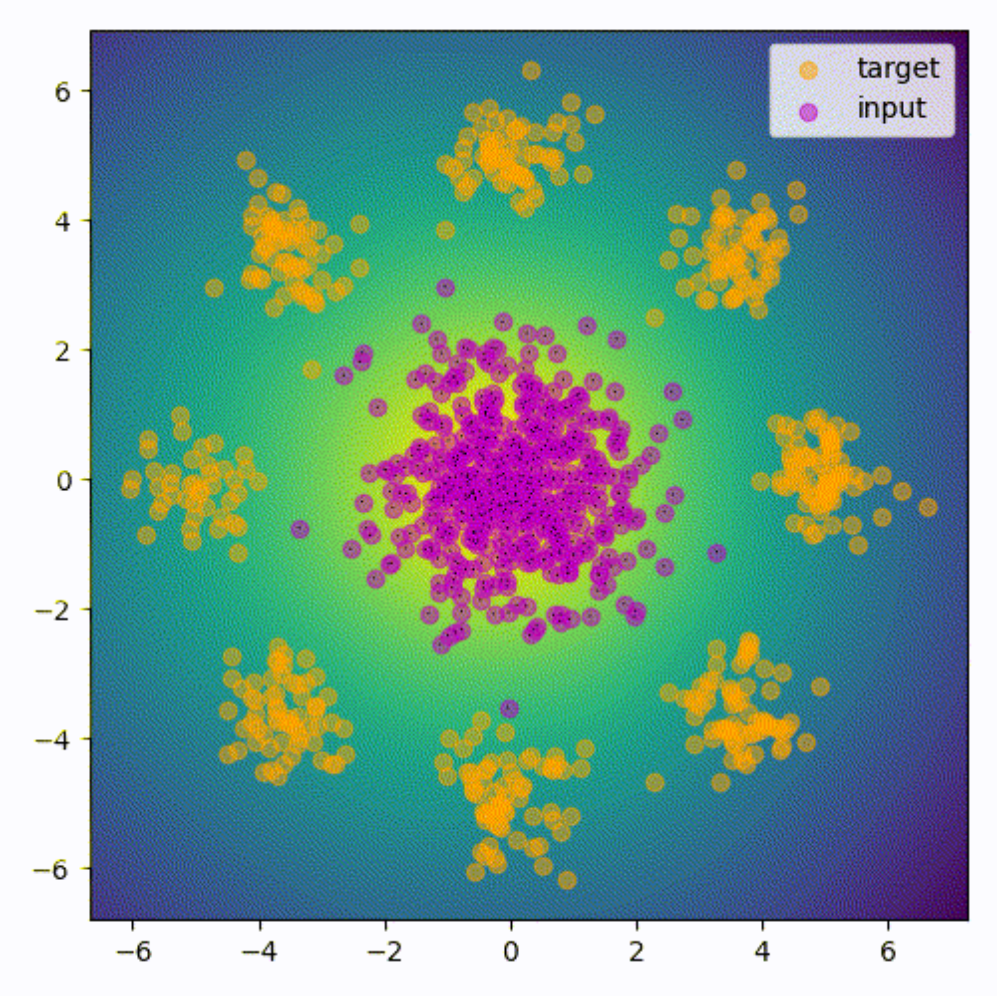
$\rightarrow$ Novel results made possible by our NTK framework.

We simplify the analysis by solving the GAN minimization problem over a point cloud instead of a neural generator.

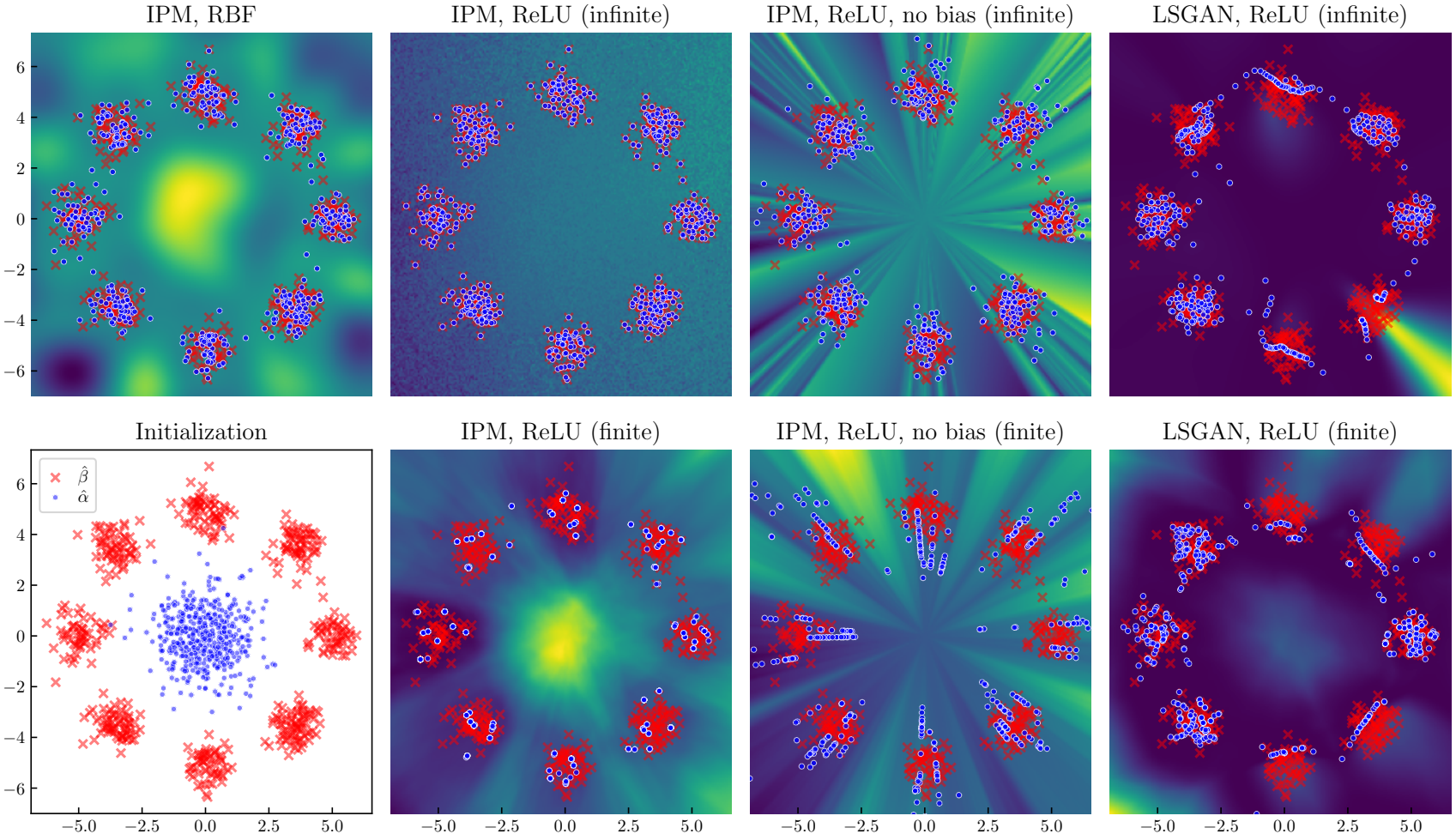
Example for the IPM model:

$$\frac{\mathrm{d}\boldsymbol{x}_\ell}{\mathrm{d}\ell} = -\nabla_{\boldsymbol{x}_\ell}(f_{\alpha_\ell}).$$

Background: discriminator.



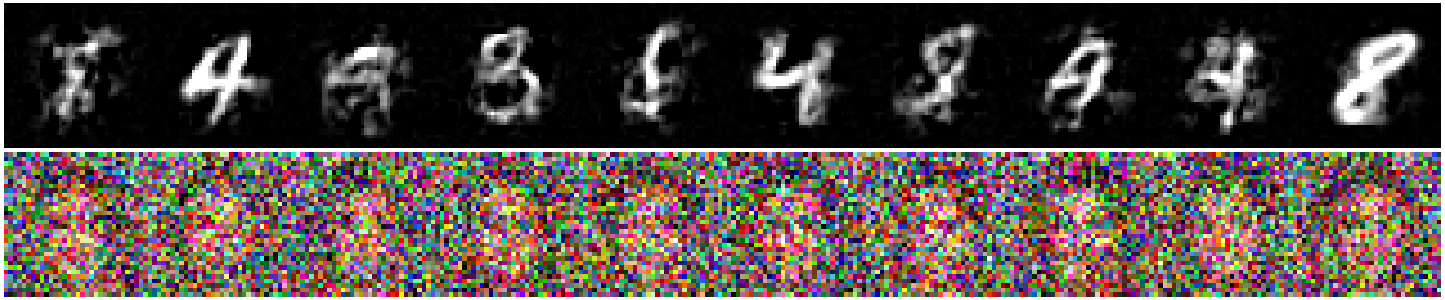
Experimental Study on GAN Convergence



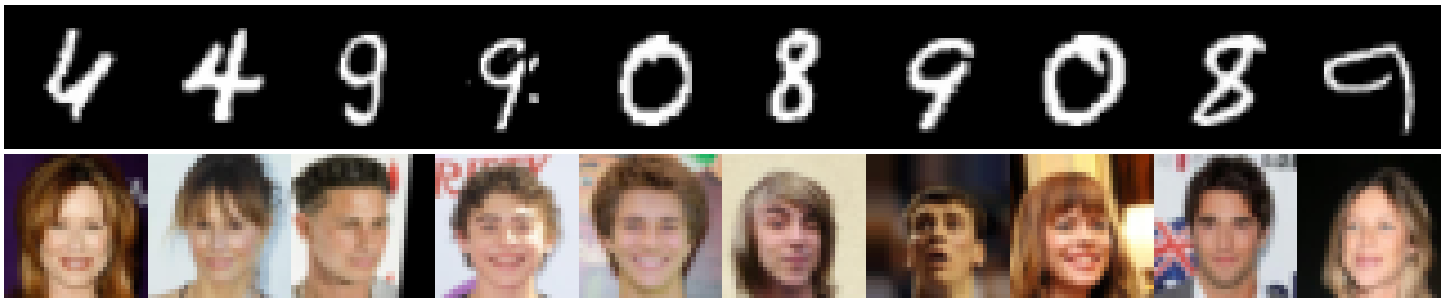
→ Correlation between finite and infinite-width regimes.

Experimental Study on GAN Convergence

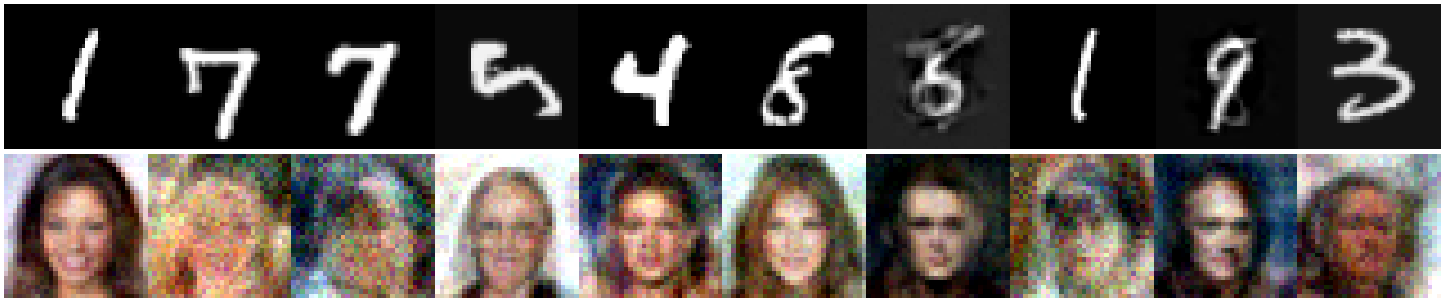
RBF



ReLU



ReLU (no bias)



Analysis of GANs' Training Dynamics: Conclusion

- We identify fundamental flaws in previous GAN analyses.
- We propose a novel NTK-based framework solving these issues.
- This framework provides a new understanding of GAN training, with:
  - the generator's training dynamics;
  - the link between IPMs and the NTK MMD;
  - the performance of ReLU activations for generative modeling.
- Opens new perpectives for further analyses.

Experimental Framework

Code: `https://github.com/emited/gantk2`.

Conclusion & Perspectives

Dynamical Systems and Generative Modeling

We explored two applications:

- for better modeling and representation learning of spatiotemporal sequences;
- to study the training dynamics of generative models.

Dynamical Systems and Generative Modeling

We explored two applications:

- for better modeling and representation learning of spatiotemporal sequences;
- to study the training dynamics of generative models.

- In spatiotemporal modeling, it allowed us:
  - to accurately predict videos and physical phenomena;
  - to improve our interpretation of state-space models.
- By studying training dynamics, we could overcome the issues of prior GAN analyses and better understand these models.

Perpectives: Spatiotemporal Prediction

- Improve modeling hypotheses:
  - merge video and PDE-based models;
  - relax the constancy hypothesis of the content variable $\boldsymbol{w}$;
  - introduce adaptive stochasticity (unfinished project):

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = f_\theta\Big(\boldsymbol{y}, \boldsymbol{z}_{\lfloor t \rfloor + 1}\Big) \quad \Rightarrow \quad \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = f_\theta(\boldsymbol{y}, \boldsymbol{z}_{t_i}), \text{ with } t_i \text{ learned.}$$

- Scale models (Villegas, Pathak, et al., 2019).
- Apply models to other types of sequential data.

Perpectives: Training Dynamics of Generative Models

- Solve open questions:
  - find the loss $\mathscr{C}$ for other models;
  - establish convergence results;
  - understand the performance of the ReLU NTKs.
- Leverage our framework to improve GANs:
  - find better discriminators;
  - develop new models and losses (ongoing work on MMD NTK GAN).
- Apply this analysis to other generative models (links with score-based models).

Thank you for your attention!

References

Arbel, Michael et al. (2019). "Maximum Mean Discrepancy Gradient Flow". In: *Advances in Neural Information Processing Systems*. Ed. by Hanna Wallach et al. Vol. 32. Curran Associates, Inc., pp. 6484–6494.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (Aug. 2017). "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 214–223.

Babaeizadeh, Mohammad et al. (2018). "Stochastic Variational Video Prediction". In: *International Conference on Learning Representations*.

Brown, Tom et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by Hugo Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 1877–1901.

Chen, Ricky T. Q. et al. (2018). "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. Ed. by Samy Bengio et al. Vol. 31. Curran Associates, Inc., pp. 6572–6583.

Denton, Emily and Vighnesh Birodkar (2017). "Unsupervised Learning of Disentangled Representations from Video". In: *Advances in Neural Information Processing Systems*. Ed. by Isabelle Guyon et al. Vol. 30. Curran Associates, Inc., pp. 4417–4426.

Denton, Emily and Rob Fergus (July 2018). "Stochastic Video Generation with a Learned Prior". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm, Sweden: PMLR, pp. 1174–1183.

Duncan, Andrew, Nikolas Nüsken, and Lukasz Szpruch (2019). "On the geometry of Stein variational gradient descent". In: arXiv: 1912.00894.

Finn, Chelsea, Ian Goodfellow, and Sergey Levine (2016). "Unsupervised Learning for Physical Interaction through Video Prediction". In: *Advances in Neural Information Processing Systems*. Ed. by Daniel D. Lee et al. Vol. 29. Curran Associates, Inc., pp. 64–72.

Fraccaro, Marco et al. (2017). "A Disentangled Recognition and Nonlinear Dynamics Model for Unsupervised Learning". In: *Advances in Neural Information Processing Systems 30*. Ed. by Isabelle Guyon et al. Vol. 30. Curran Associates, Inc., pp. 3604–3613.

Goodfellow, Ian et al. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Zoubin Ghahramani et al. Vol. 27. Curran Associates, Inc., pp. 2672–2680.

Gregor, Karol et al. (2019). "Temporal Difference Variational Auto-Encoder". In: *International Conference on Learning Representations*.

Hsieh, Jun-Ting et al. (2018). "Learning to Decompose and Disentangle Representations for Video Prediction". In: *Advances in Neural Information Processing Systems*. Ed. by Samy Bengio et al. Vol. 31. Curran Associates, Inc., pp. 515–524.

Jacot, Arthur, Franck Gabriel, and Clement Hongler (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Samy Bengio et al. Vol. 31. Curran Associates, Inc., pp. 8580–8589.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Fernando Pereira et al. Vol. 25. Curran Associates, Inc., pp. 1097–1105.

Metz, Luke et al. (2017). "Unrolled Generative Adversarial Networks". In: *International Conference on Learning Representations*.

Minderer, Matthias et al. (2019). "Unsupervised Learning of Object Structure and Dynamics from Videos". In: *Advances in Neural Information Processing Systems*. Ed. by Hanna Wallach et al. Vol. 32. Curran Associates, Inc., pp. 92–102.

Mroueh, Youssef and Truyen Nguyen (Apr. 2021). "On the Convergence of Gradient Descent in GANs: MMD GAN As a Gradient Flow". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 1720–1728.

Silver, David et al. (2017). "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: arXiv: 1712.01815.

Villegas, Ruben, Arkanath Pathak, et al. (2019). "High Fidelity Video Prediction with Large Stochastic Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Hanna Wallach et al. Vol. 32. Curran Associates, Inc., pp. 81–91.

Villegas, Ruben, Jimei Yang, et al. (2017). "Decomposing Motion and Content for Natural Video Sequence Prediction". In: *International Conference on Learning Representations*.

Yingzhen, Li and Stephan Mandt (July 2018). "Disentangled Sequential Autoencoder". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm, Sweden: PMLR, pp. 5670–5679.