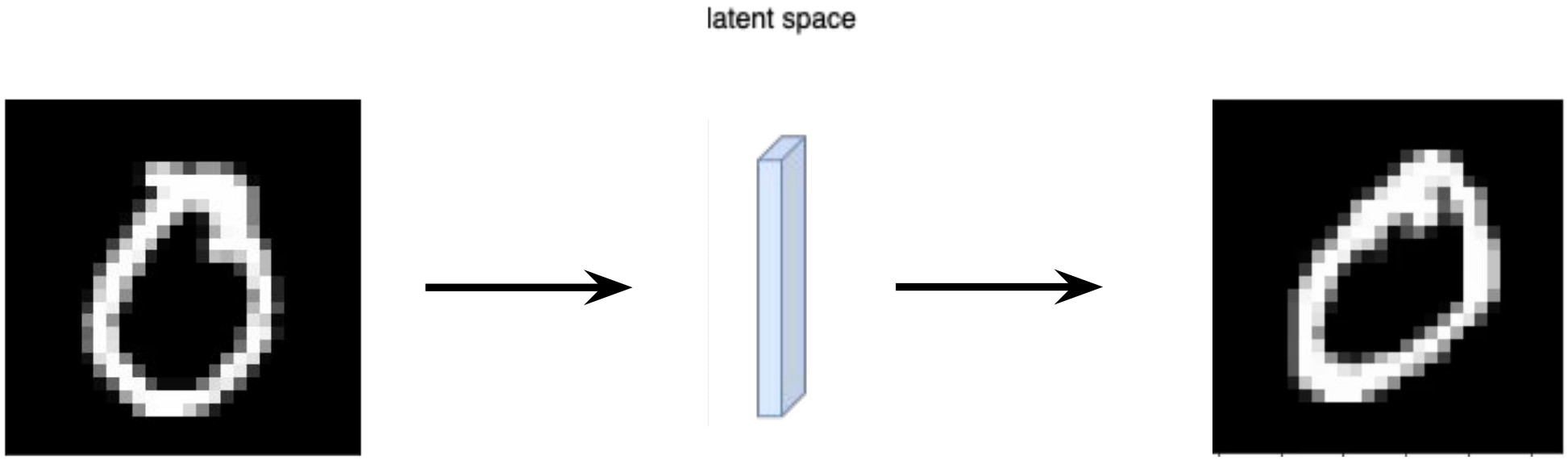
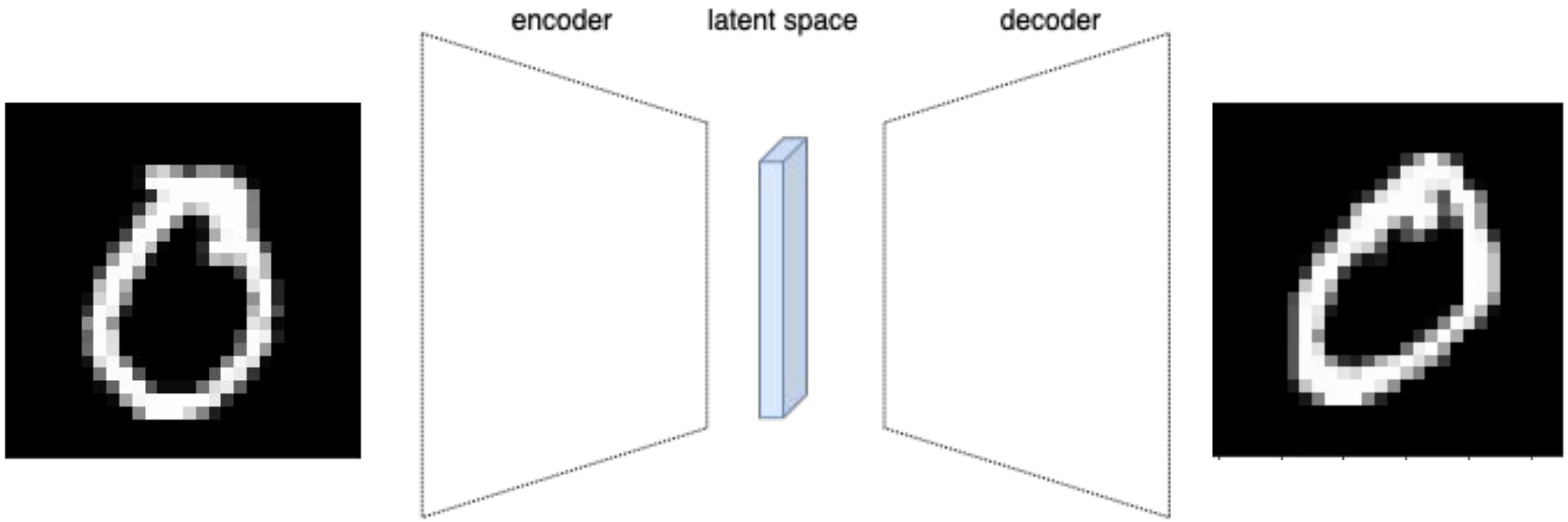
Flow Factorized Representation Learning

Yue Song, T. Anderson Keller, Nicu Sebe, Max Welling

Representation learning

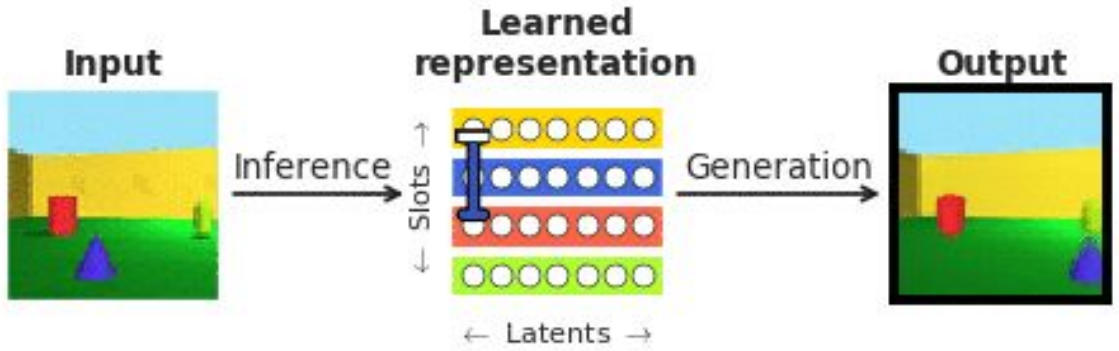


Representation learning



$$\text{ELBO}_i(\lambda) = \mathbb{E}_{q_\lambda}(z|x_i)[log\ p(x_i|z)] - D_{KL}(q_\lambda(z|x_i)||p(z))$$

Disentanglement



[1]

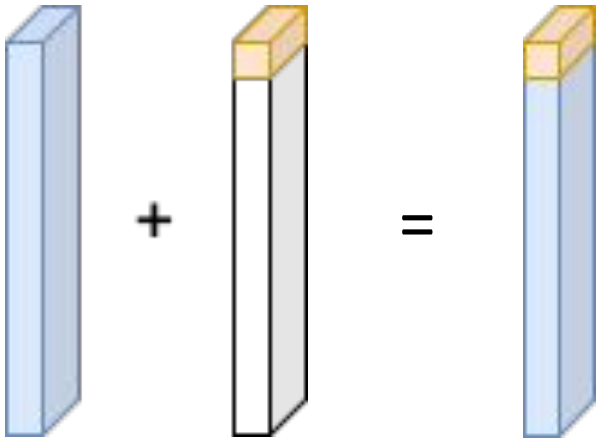
Related work:

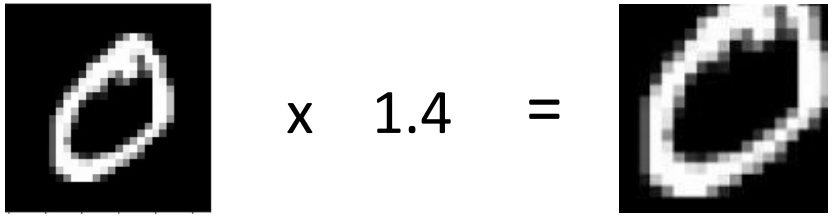
- β-VAE: restructures ELBO with variable β to enforce latent independence
- FactorVAE: uses discriminator to enhance loss for badly factorized latents
- SlowVAE: regularises for temporal sparsity using prior hyperparameters

[1] Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons, DeepMind

Equivariance

$$T'[f(x)] = f(T[x])$$

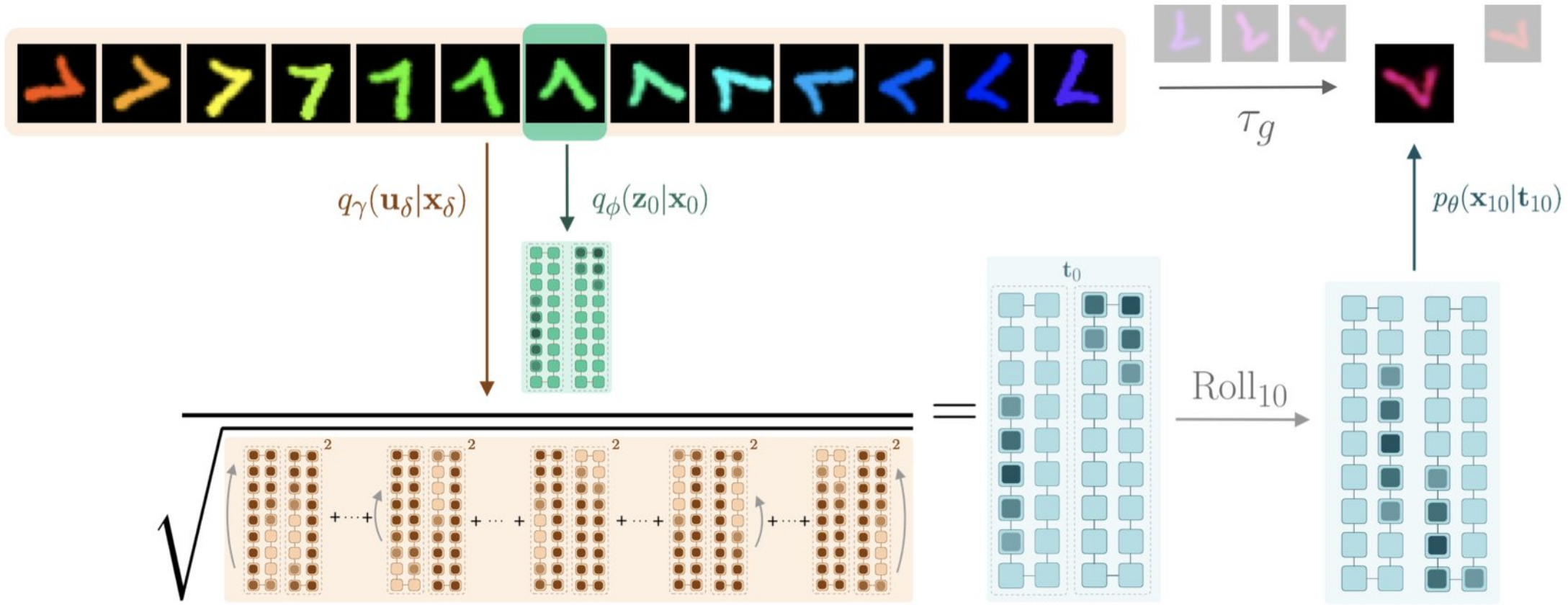




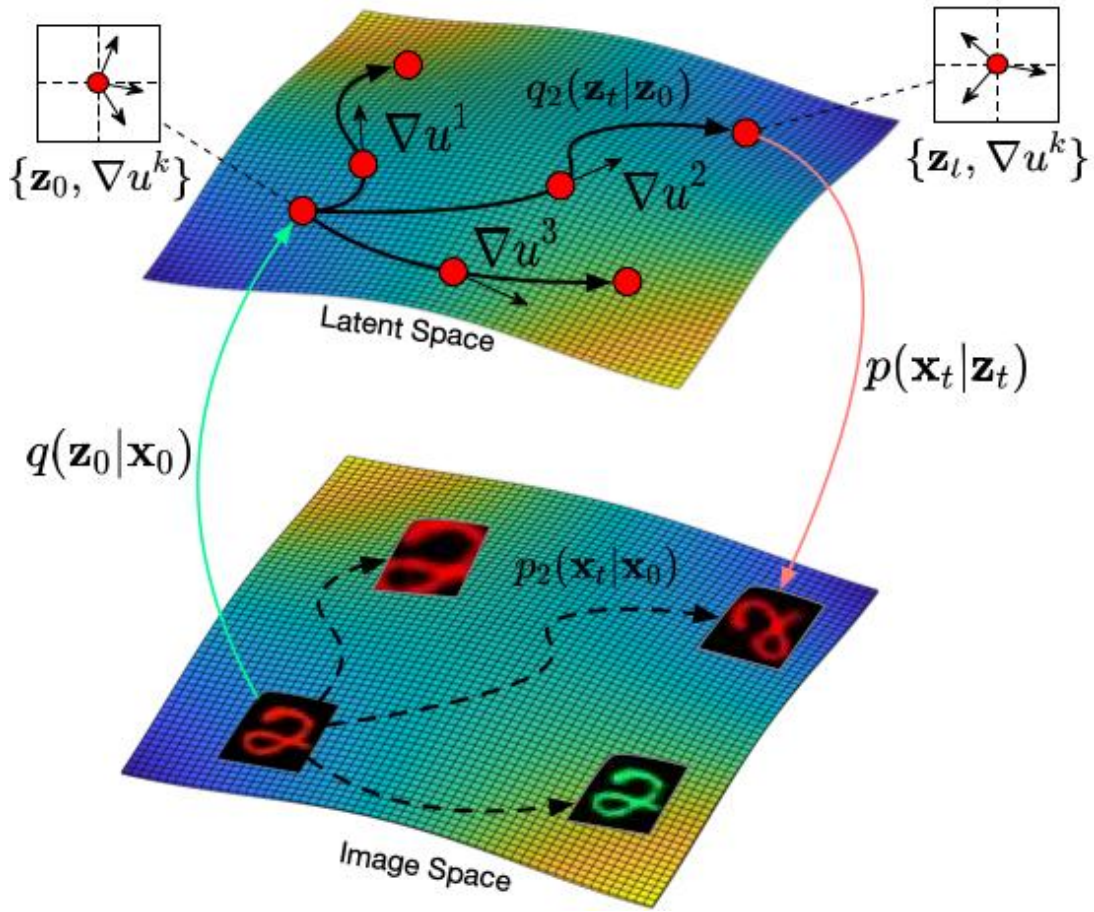
Related work:

- Topographic VAE

Equivariance



Alternative latent perspective



$$\int_{\boldsymbol{z}_0, \boldsymbol{z}_t} q(\boldsymbol{z}_0|\boldsymbol{x}_0) q_k(\boldsymbol{z}_t|\boldsymbol{z}_0) p(\boldsymbol{x}_t|\boldsymbol{z}_t)$$

=

$$p_k(\boldsymbol{x}_t|\boldsymbol{x}_0)$$

Alternative latent perspective

Supervised case

$$p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, k) = p(k)p(\boldsymbol{z}_0)p(\boldsymbol{x}_0|\boldsymbol{z}_0)\prod_{t=1}^{T} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)p(\boldsymbol{x}_t|\boldsymbol{z}_t)$$

$$q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}}, k) = q(\boldsymbol{z}_0|\boldsymbol{x}_0)\prod_{t=1}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)$$

$$\log p(\bar{\boldsymbol{x}}|k) \geq \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\left[\log p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{z}}, k)\right] + \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\left[\log \frac{p(\bar{\boldsymbol{z}}|k)}{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}}, k)}\right]$$

$$= \sum_{t=0}^{T} \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\log p(\boldsymbol{x}_t|\boldsymbol{z}_t, k)\right] - \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\mathrm{D}_{\mathrm{KL}}\left[q_\theta(\boldsymbol{z}_0|\boldsymbol{x}_0)||p(\boldsymbol{z}_0)\right]\right] - \sum_{t=1}^{T} \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\mathrm{D}_{\mathrm{KL}}\left[q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)||p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)\right]\right]$$

Alternative latent perspective

Weakly supervised
case

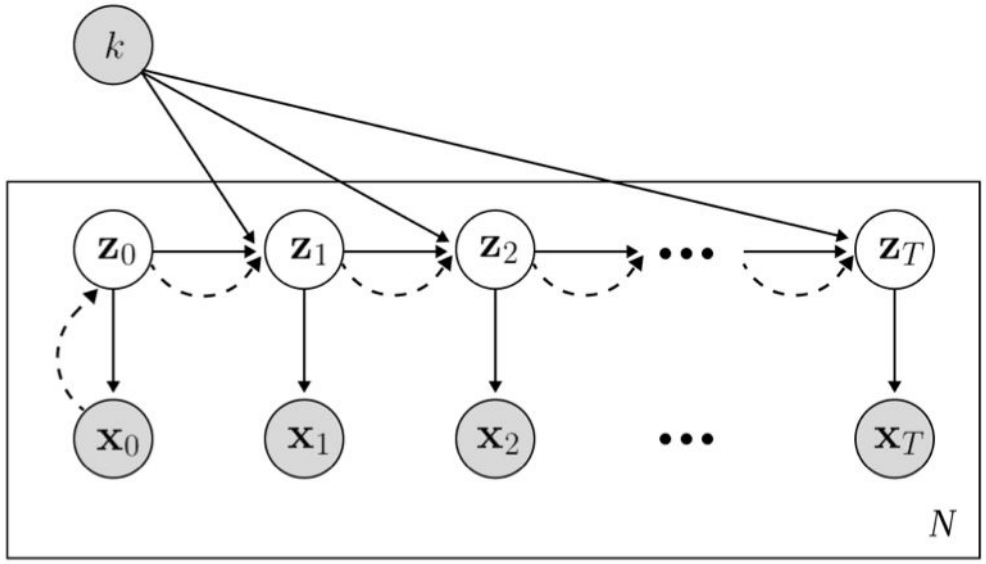
$$p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, k) = p(k)p(\boldsymbol{z}_0)p(\boldsymbol{x}_0|\boldsymbol{z}_0)\prod_{t=1}^{T} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)p(\boldsymbol{x}_t|\boldsymbol{z}_t)$$

$$q(\bar{\boldsymbol{z}}, k|\bar{\boldsymbol{x}}) = q(k|\bar{\boldsymbol{x}})q(\boldsymbol{z}_0|\boldsymbol{x}_0)\prod_{t=1}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)$$

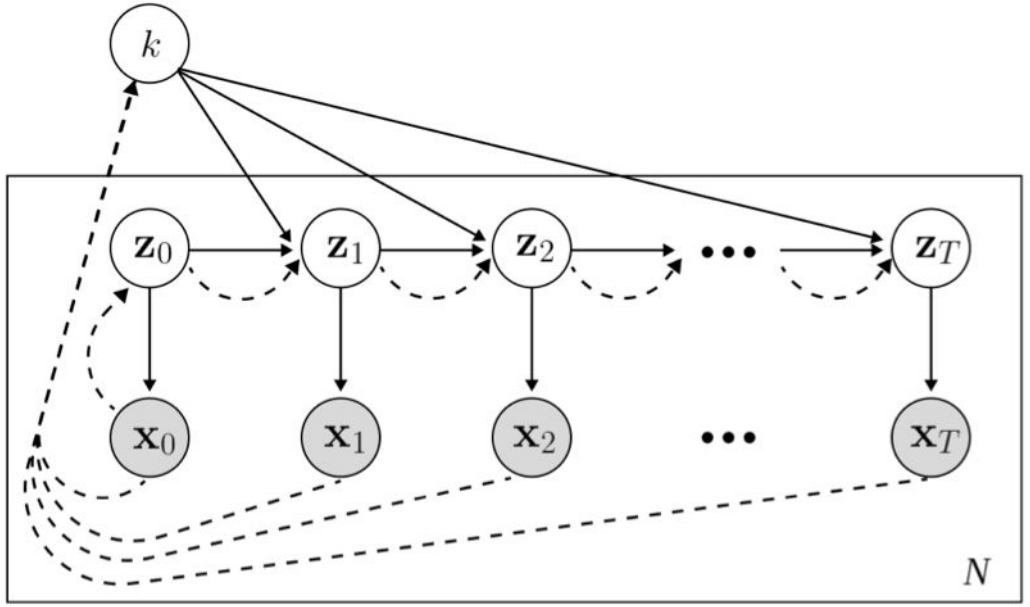
$$\log p(\bar{\boldsymbol{x}}) \geq \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\left[\log p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{z}}, k)\right] + \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\left[\log \frac{p(\bar{\boldsymbol{z}}|k)}{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}}, k)}\right] + \mathbb{E}_{q_\gamma(k|\bar{\boldsymbol{x}})}\left[\log \frac{p(k)}{q(k|\bar{\boldsymbol{x}})}\right]$$

Alternative latent perspective

Supervised case



Weakly supervised case



White nodes: latent variables
Grey nodes: observed data
Solid lines: generative model
Dashed lines: approximate posterior

Time evolving priors & posteriors

$$\mathrm{D}_{\mathrm{KL}}\left[q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k)||p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k)\right] \ ?$$

- Continuity equation: $\partial_t p(\boldsymbol{z}) = -\nabla \cdot (p(\boldsymbol{z})\nabla u^k(\boldsymbol{z}))$
- Discrete particle evolution: $\boldsymbol{z}_t = g(\boldsymbol{z}_{t-1},k) = \boldsymbol{z}_{t-1} + \nabla_{\boldsymbol{z}} u^k$

Prior:

$$u^k = -D_k \log p(\boldsymbol{z}_t)$$

$$\partial_t p(\boldsymbol{z}_t) = D_k \nabla^2 p(\boldsymbol{z}_t)$$

$$p(\boldsymbol{z}_0) = \mathcal{N}(0,1)$$

Posterior:

$$\log q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k) = \log q(\boldsymbol{z}_{t-1}) - \log|1+\nabla^2_{\boldsymbol{z}} u^k|$$

Optimal Transport for posterior flow

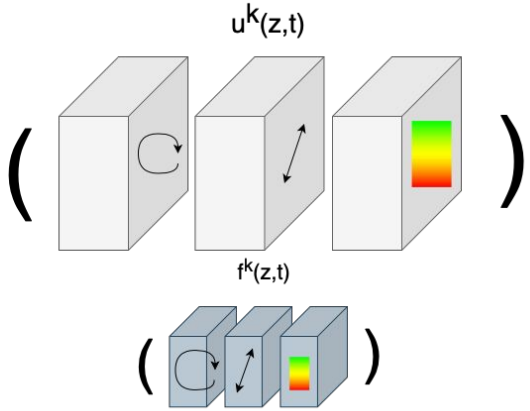
$$\frac{\partial}{\partial t}u^k(\boldsymbol{z},t) + \frac{1}{2}||\nabla_{\boldsymbol{z}}u^k(\boldsymbol{z},t)||^2 = f(\boldsymbol{z},t) \quad \text{subject to} \quad f(\boldsymbol{z},t) \leq 0$$

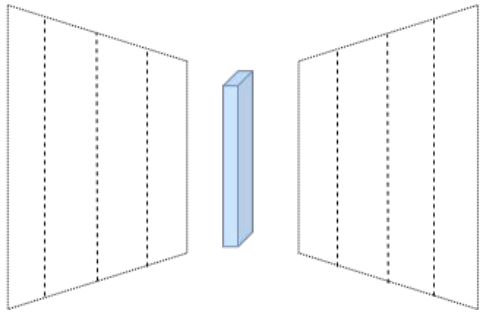
$$\longrightarrow \mathcal{L}_{HJ} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial}{\partial t}u^k(\boldsymbol{z},t) + \frac{1}{2}||\nabla_{\boldsymbol{z}}u^k(\boldsymbol{z},t)||^2 - f(\boldsymbol{z},t)\right)^2 + ||\nabla u^k(\boldsymbol{z}_0,0)||^2$$

[1]

[1] Information Geometry and Optimal Transport, Gabe Khan

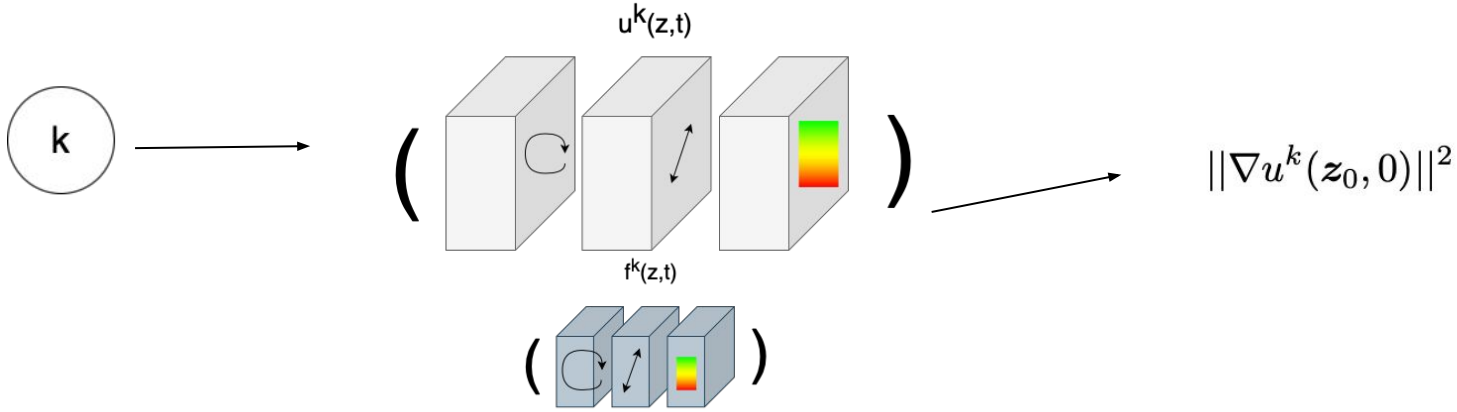
Implementation

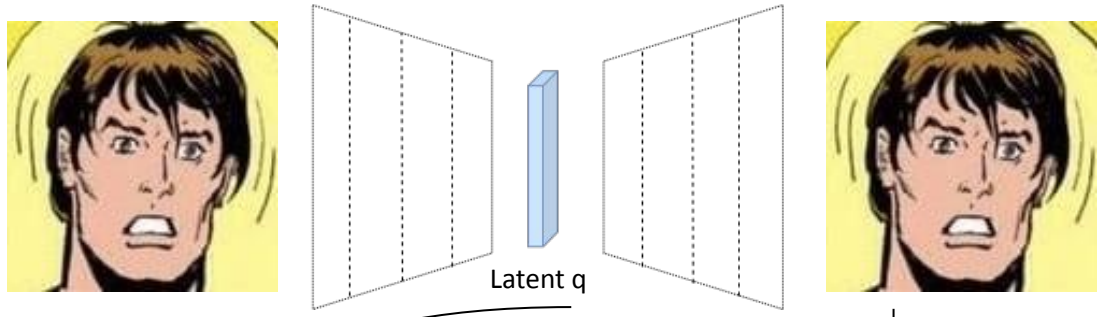




Implementation

t= 0



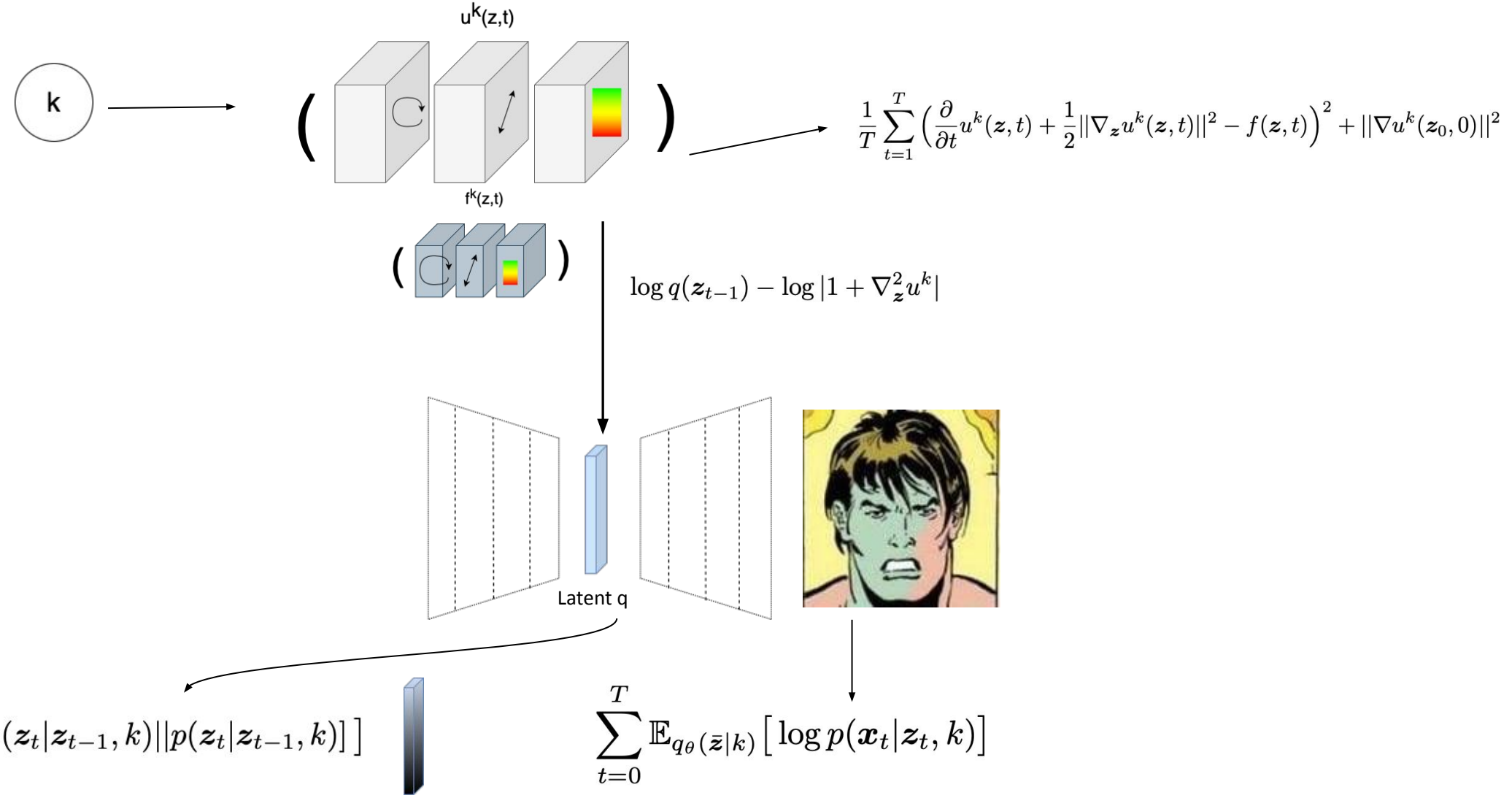


$$\mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\mathrm{D}_{\mathrm{KL}}\left[q_\theta(\boldsymbol{z}_0|\boldsymbol{x}_0)||p(\boldsymbol{z}_0)\right]\right]$$

$$\sum_{t=0}^{T} \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\log p(\boldsymbol{x}_t|\boldsymbol{z}_t, k)\right]$$

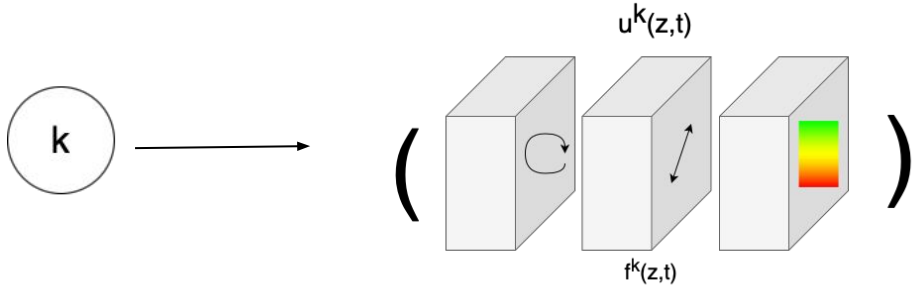
Implementation

t= 1



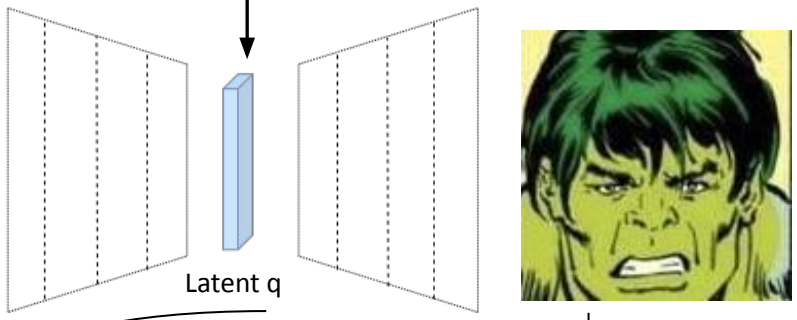
Implementation

t= 2 = T



$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial}{\partial t}u^k(\boldsymbol{z},t)+\frac{1}{2}||\nabla_{\boldsymbol{z}}u^k(\boldsymbol{z},t)||^2-f(\boldsymbol{z},t)\right)^2+||\nabla u^k(\boldsymbol{z}_0,0)||^2$$

$$\log q(\boldsymbol{z}_{t-1})-\log|1+\nabla_{\boldsymbol{z}}^2 u^k|$$



$$\mathcal{L} =$$

$$\mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\mathrm{D}_{\mathrm{KL}}\left[q_\theta(\boldsymbol{z}_0|\boldsymbol{x}_0)||p(\boldsymbol{z}_0)\right]\right]$$

$$+\sum_{t=1}^{T}\mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\mathrm{D}_{\mathrm{KL}}\left[q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k)||p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k)\right]\right]$$

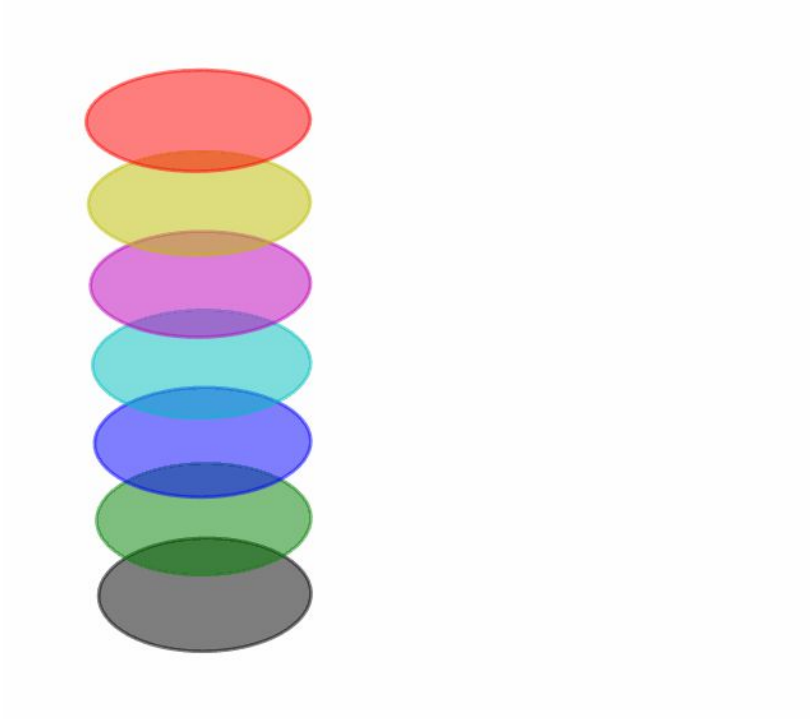
$$-\sum_{t=0}^{T}\mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\left[\log p(\boldsymbol{x}_t|\boldsymbol{z}_t,k)\right]$$

$$+\quad\mathcal{L}_{HJ}$$

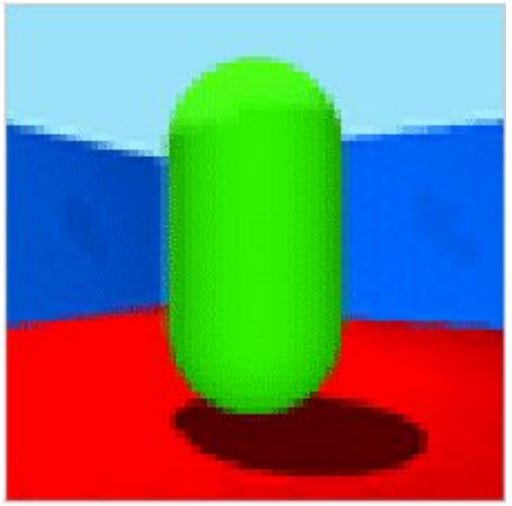
Implementation

: one latent dimension, corresponding to a normal distribution

Color in latent corresponds to distinct ground truth factor



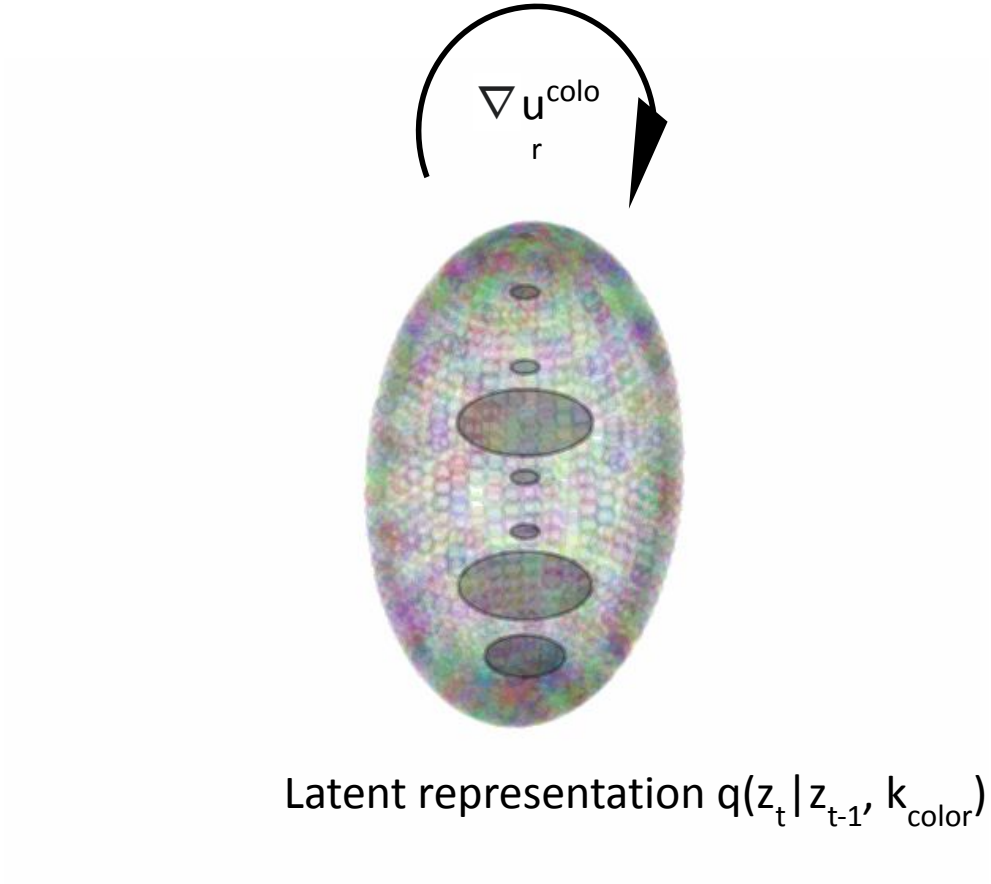
Latent representation q



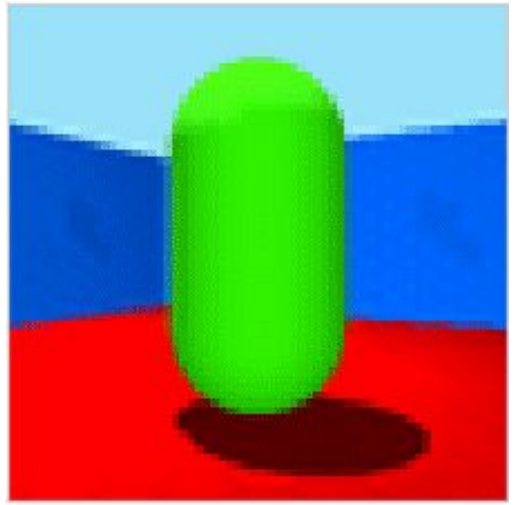
Implementation

: one latent dimension, corresponding to a time-evolving normal distribution

Color in latent corresponds to distinct ground truth factor



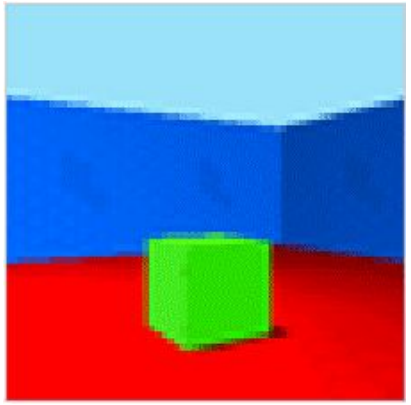
Latent representation $q(z_t | z_{t-1}, k_{color})$



Experiment setting

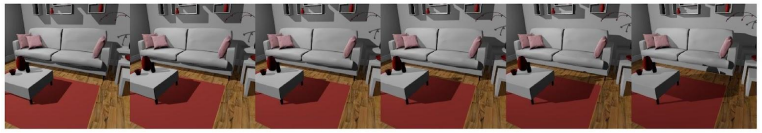
- Datasets:
  - MNIST: color, rotation, scale

  - Shapes3D: wall-, floor- & object hue, scale



lighting intensity

lighting x-dir



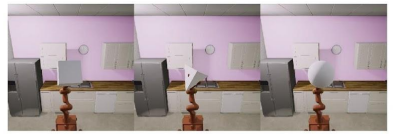
lighting y-dir

lighting z-dir

camera x-pos

camera y-pos

camera z-pos



object shape

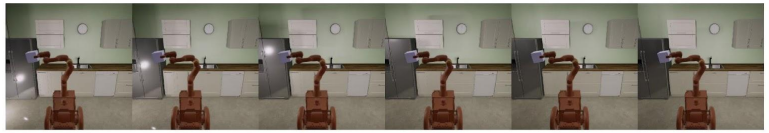
robot x-movement

robot y-movement

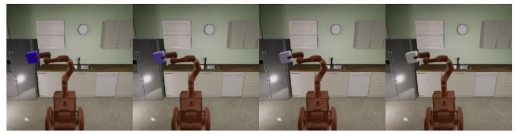
camera height

object scale

lighting intensity



lighting y-dir



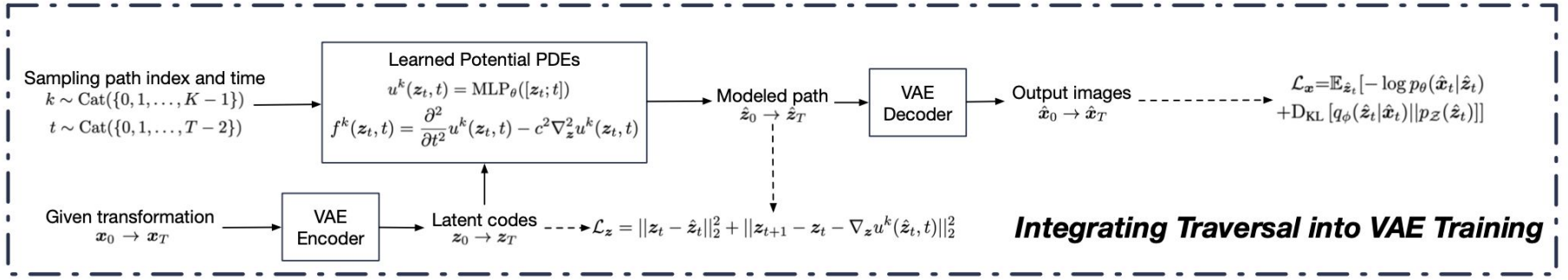
object color

wall color

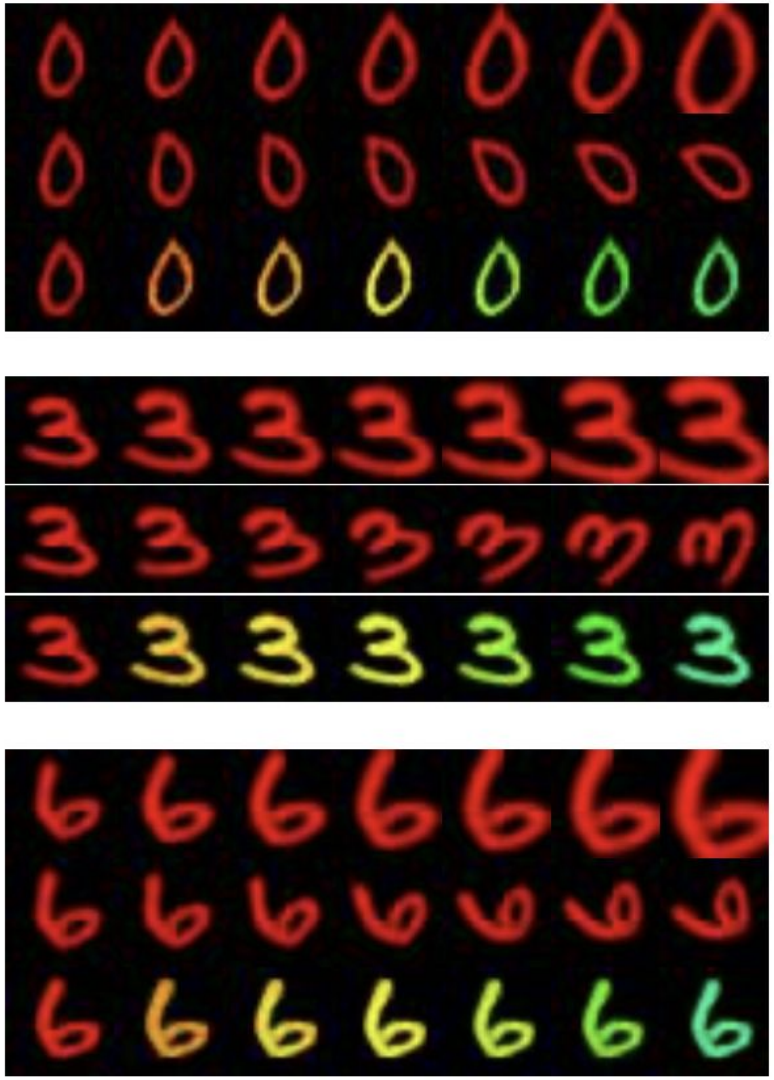
Experiment setting

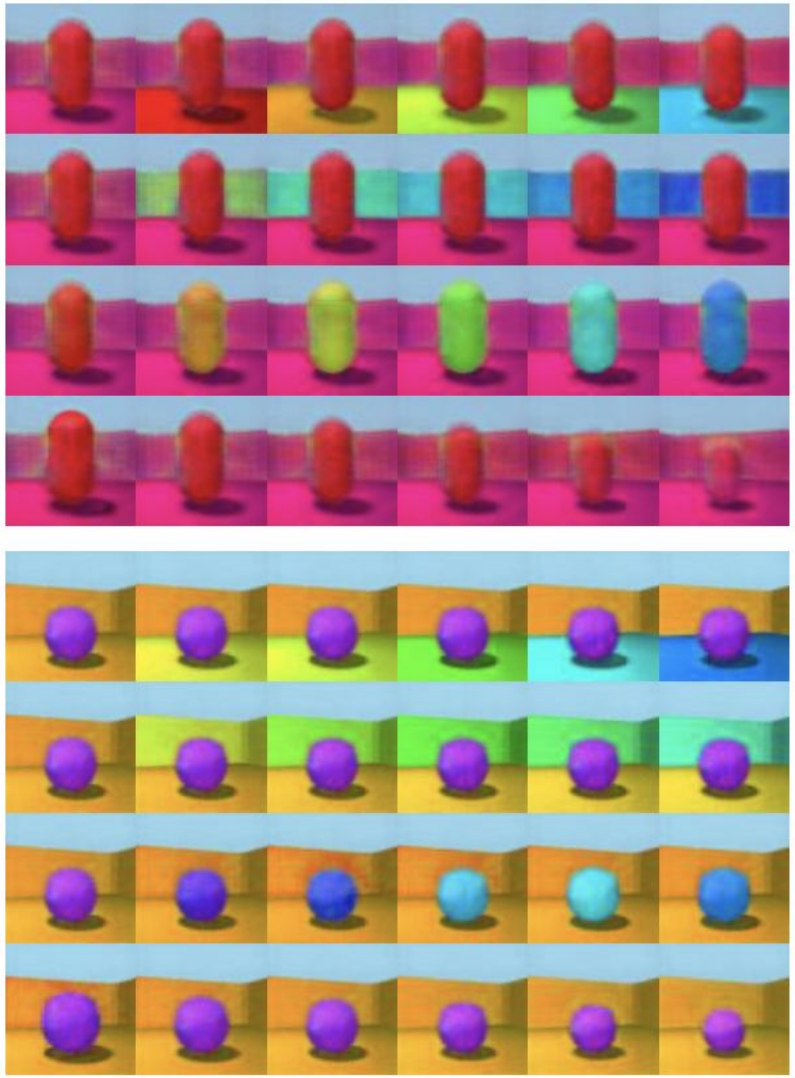
- Metrics:
  - Equivariance error: $\mathcal{E}_k = \sum_{t=1}^{T} |\boldsymbol{x}_t - \texttt{Decode}(\boldsymbol{z}_t)|$, $\boldsymbol{z}_t = \boldsymbol{z}_0 + \sum_{t=1}^{T} \nabla_{\boldsymbol{z}} u^k$
  - Log-likelihood: $\log p(\boldsymbol{x}_t)$
  - VP score: $(\%)$ of correctly classified pairs $[\boldsymbol{x}_0, \boldsymbol{x}_T]$ by a lightweight NN
- Baselines:
  - PoFlow
  - Topographic VAE
  - SlowVAE
  - β-VAE
  - FactorVAE
  - VAE

PoFlow implementation

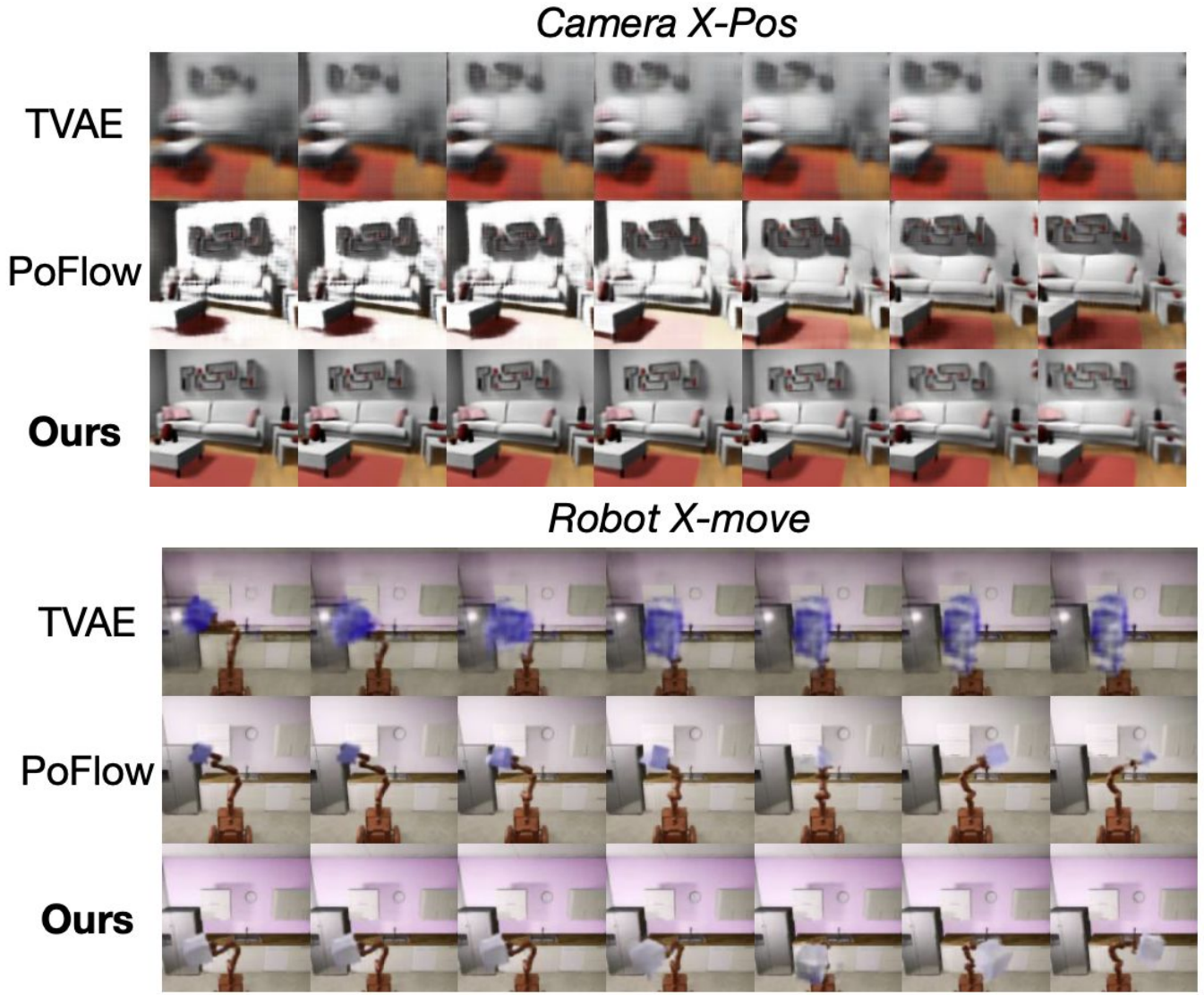


Qualitative results





Qualitative results



Quantitative results

Methods	Supervision?	Equivariance Error (↓)			Log-likelihood (↑)
		Scaling	Rotation	Coloring	
VAE [47]	No (✗)	1275.31±1.89	1310.72±2.19	1368.92±2.33	-2206.17±1.83
β-VAE [35]	No (✗)	741.58±4.57	751.32±5.22	808.16±5.03	-2224.67±2.35
FactorVAE [46]	No (✗)	659.71±4.89	632.44±5.76	662.18±5.26	-2209.33±2.47
SlowVAE [49]	Weak (✓)	461.59±5.37	447.46±5.46	398.12±4.83	-2197.68±2.39
TVAE [45]	Yes (✓)	505.19±2.77	493.28±3.37	451.25±2.76	-2181.13±1.87
PoFlow [79]	Yes (✓)	234.78±2.91	231.42±2.98	240.57±2.58	-2145.03±2.01
Ours	Yes (✓)	**185.42±2.35**	**153.54±3.10**	**158.57±2.95**	**-2112.45±1.57**
Ours	Weak (✓)	193.84±2.47	157.16±3.24	165.19±2.78	-2119.94±1.76

Table 1: Equivariance error $\mathcal{E}_k$ and log-likelihood $\log p(\boldsymbol{x}_t)$ on MNIST

Methods	Supervision?	Equivariance Error (↓)				Log-likelihood (↑)
		Floor Hue	Wall Hue	Object Hue	Scale	
VAE [47]	No (✗)	6924.63±8.92	7746.37±8.77	4383.54±9.26	2609.59±7.41	-11784.69±4.87
β-VAE [35]	No (✗)	2243.95±12.48	2279.23±13.97	2188.73±12.61	2037.94±11.72	-11924.83±5.64
FactorVAE [46]	No (✗)	1985.75±13.26	1876.41±11.93	1902.83±12.27	1657.32±11.05	-11802.17±5.69
SlowVAE [49]	Weak (✓)	1247.36±12.49	1314.86±11.41	1102.28±12.17	1058.74±10.96	-11674.89±5.74
TVAE [45]	Yes (✓)	1225.47±9.82	1246.32±9.54	1261.79±9.86	1142.01±9.37	-11475.48±5.18
PoFlow [79]	Yes (✓)	885.46±10.37	916.71±10.49	912.48±9.86	924.39±10.05	-11335.84±4.95
Ours	Yes (✓)	**613.29±8.93**	**653.45±9.48**	**605.79±8.63**	**599.71±9.34**	**-11215.42±5.71**
Ours	Weak (✓)	690.84±9.57	717.74±10.65	681.59±9.02	653.58±9.57	-11279.61±5.89

Table 2: Equivariance error $\mathcal{E}_k$ and log-likelihood $\log p(\boldsymbol{x}_t)$ on Shapes3D

Quantitative results

Methods	Lighting Intensity	Lighting X-dir	Lighting Y-dir	Lighting Z-dir	Camera X-pos	Camera Y-pos	Camera Y-pos
TVAE [45]	11477.81	12568.32	11807.34	11829.33	11539.69	11736.78	11951.45
PoFlow [79]	8312.97	7956.18	8519.39	8871.62	8116.82	8534.91	8994.63
Ours	**5798.42**	**6145.09**	**6334.87**	**6782.84**	**6312.95**	**6513.68**	**6614.27**

Table 3: Equivariance error (↓) on Falcol3D

Methods	Robot X-move	Robot Y-move	Camera Height	Object Scale	Lighting Intensity	Lighting Y-dir	Object Color	Wall Color
TVAE [45]	8441.65	8348.23	8495.31	8251.34	8291.70	8741.07	8456.78	8512.09
PoFlow [79]	6572.19	6489.35	6319.82	6188.59	6517.40	6712.06	7056.98	6343.76
Ours	**3659.72**	**3993.33**	**4170.27**	**4359.78**	**4225.34**	**4019.84**	**5514.97**	**3876.01**

Table 4: Equivariance error (↓) on Isaac3D

Table 5: VP Scores (%) on MNIST.

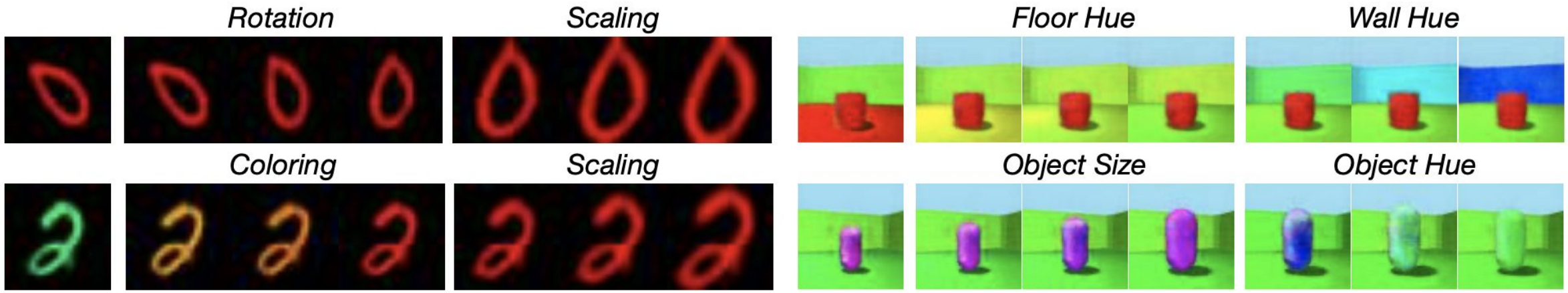
Training Set Split	Ours	PoFlow	TVAE	FactorVAE	β-VAE
10%	**95.69**	93.05	89.91	85.92	87.31
1%	**92.71**	91.27	88.15	84.46	85.25

Table 6: VP Scores (%) on Shapes3D.

Training Set Split	Ours	PoFlow	TVAE	FactorVAE	β-VAE
10%	**95.92**	91.48	88.27	84.49	85.91
1%	**77.03**	72.32	68.39	63.83	65.78

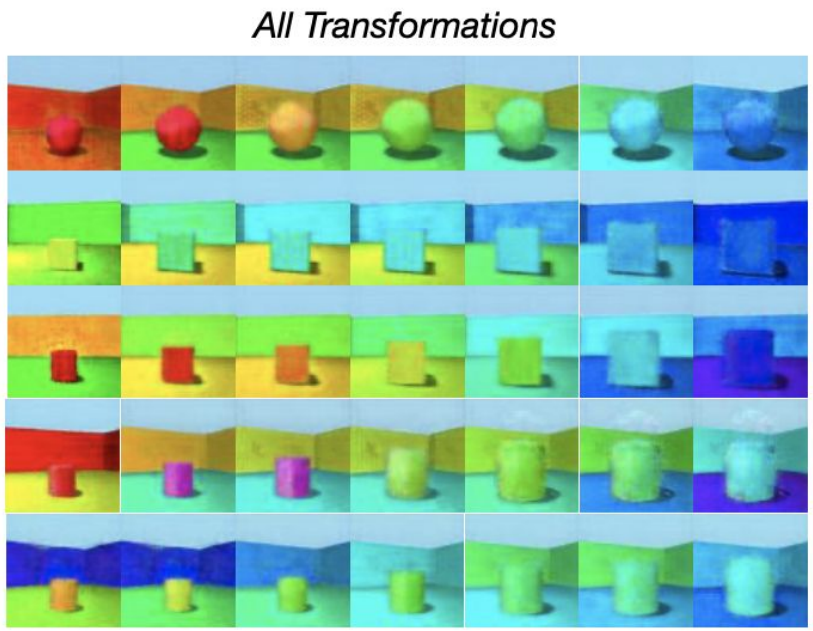
Additional evaluations

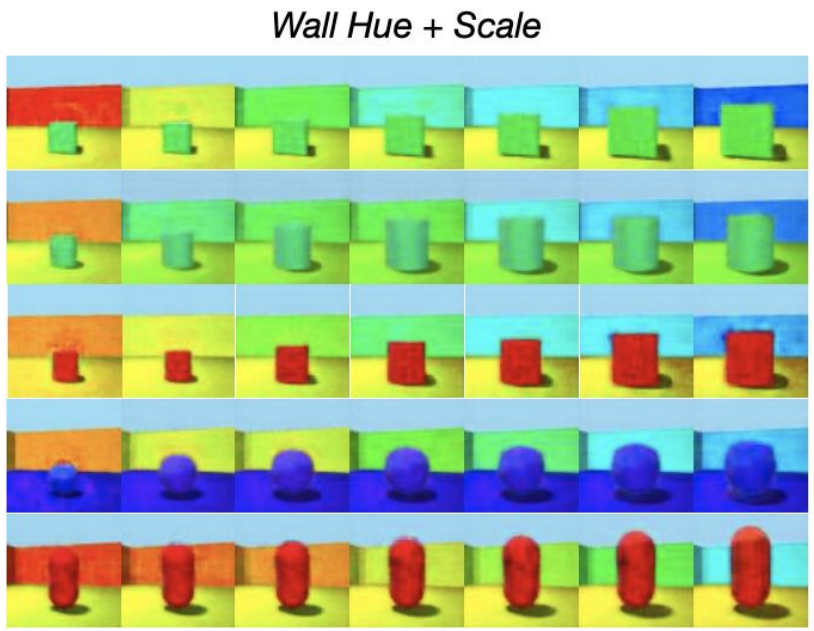
- 1: switching transformations

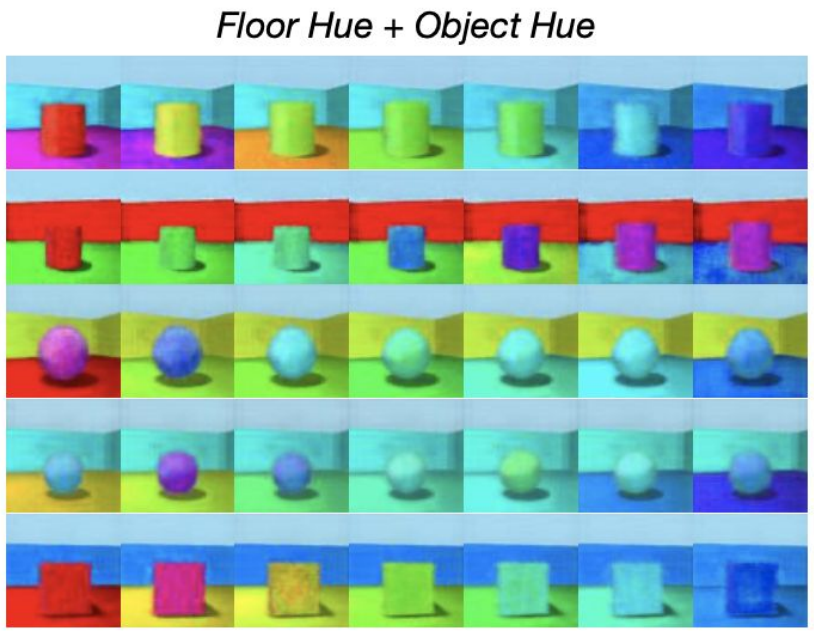


Additional evaluations

- 2: superposing transformations

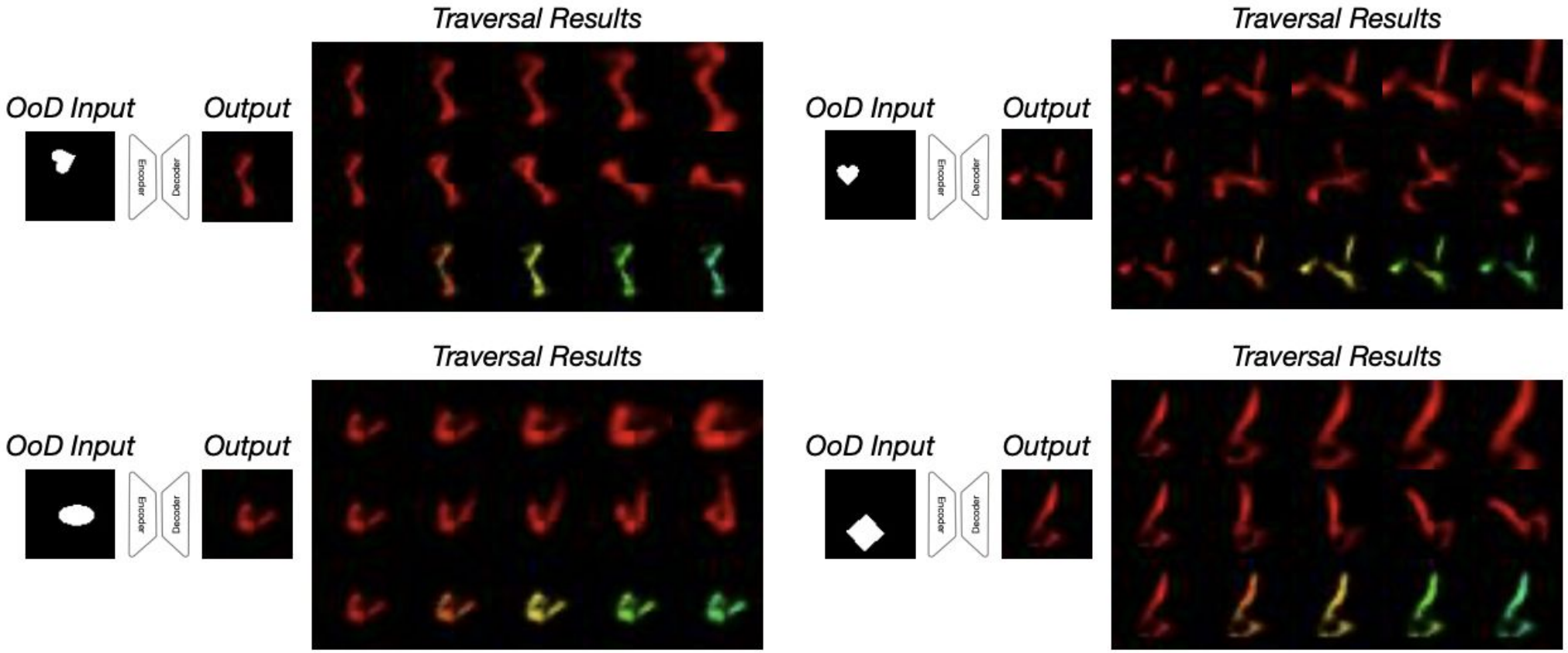






Additional evaluations

- 3: equivariance generalisation to new data



Ablation studies

Table 7: Equivariance error of different priors.

Prior	Scaling	Rotation	Coloring
SG	190.24±2.18	158.93±3.25	164.18±2.77
MoG	188.23±2.45	157.79±2.86	161.49±2.62
VAMP	192.81±3.67	161.47±4.12	162.97±3.89
Diffusion	**185.42±2.35**	**153.54±3.10**	**158.57±2.95**

Table 8: Equivariance error of different PDEs.

Prior	Scaling	Rotation	Coloring
Heat	223.95±3.38	212.47±3.85	207.66±2.91
FP	211.54±3.17	188.59±3.92	194.73±3.09
OHJ	190.43±2.48	163.87±3.03	162.38±2.86
GHJ	**185.42±2.35**	**153.54±3.10**	**158.57±2.95**

Ablation studies

Table 9: Equivariance error on MNIST of a different number of transformations (K).

K	**Scaling**	**Rotation**	**Coloring**
1	185.27±2.59	–	–
2	185.78±2.21	154.29±2.87	–
3	185.42±2.45	153.54±3.10	158.57±2.95

Table 10: Equivariance error on MNIST of different sequence lengths (T).

Sequnce Length (T)	**Scaling**	**Rotation**
9	185.42±2.35	153.54±3.10
12	214.47±2.59	198.72±2.89

Conclusion: positives

- Differentiated approach
- Quantitatively better model
- Intuitive & strong qualitative results

Conclusion: negatives

- Code doesn’t work
- More expensive to train
- Short on generalisation results
- Short on theoretical intuition

Thank you for your attention!

Questions?

```
import torch

#Randomly sample a transformation at each iteration
index = torch.randint(0, potential_number)
x_bar = sequence_generation(index)

#Generating index according to the supervision setting
if training_mode = "supervised":
  index_potential = index
elif training_mode = "weakly-supervised":
  index_potential = q_k(x_bar)

#initial element of the sequence
z, rho_z = flow_vae(x_bar[0])

#Future elements of the sequence obtained by latent flow
for t in range(0,T)
   PDE_loss, delta_z, delat_rho_z = HJ_PDE(index_potential,z,t)

   #Updates in the sample and probability space
   z = z + delta_z
   rho_z = rho_z + delat_rho_z

   #Inference at every intermediate step
   hat_xt = flow_vae.inference(z)

   #Loss: PDE loss + reconstrutction loss + KL div
   loss += PDE_loss + CE(hat_xt,x_bar[t]) + KL(rho_z, prior_rho_z)

#KL div for index prediction (weakly-supervised setting)
if training_mode = "weakly-supervised":
   loss += KL(index_potential,index)

loss.backward()
optimizer.step()
```

Gumbel-Softmax trick

Weakly-supervised setting. For the `Gumbel-Softmax` trick, we re-parameterize $q_\gamma(k|\bar{\boldsymbol{x}})$ by

$$y_i = \frac{e^{\frac{x_i+g_i}{\tau}}}{\sum_i e^{\frac{x_i+g_i}{\tau}}} \tag{15}$$

where x_i is the category prediction, g_i is the sample drawn from Gumbel distributions, and τ is the small temperature to make `softmax` behave like `argmax`. We take the 'hard' binary prediction in the forward pass and use the straight-through gradient estimator [6] during backpropagation. The temperature τ is initialized with 1 and is gradually reduced to 0.05 with the annealing rate $3e{-}5$.

We now turn to introduce why HJ equations could minimize the Wasserstein distance. As stated in [4], the L_2 Wasserstein distance can be re-formulated in the fluid mechanical interpretation as

$$W^2 = \inf \int_D \int_0^1 \frac{1}{2} \rho(x,t) v(x,t)^2 \, dx \, dt \tag{16}$$

where the density satisfies the continuity equation ($\partial_t \rho = -\nabla \cdot (\rho(x,t)v(x,t))$. If we introduce the momentum $m(x,t) = \rho(x,t)v(x,t)$ and two Lagrange multipliers u and λ, the Lagrangian function of the Wasserstein distance would be:

$$L(\rho, m, \phi) = \int_D \int_0^1 \frac{||m||^2}{2\rho} + u(\partial_t \rho + \nabla \cdot m) - \lambda(\rho - s^2) \tag{17}$$

where the second term is the equality constraint, and the third term is an equality constraint with a slack variable s. Using integration by parts formula, the above equation can be re-written as

$$L(\rho, m, \phi) = \int_D \int_0^1 \frac{||m||^2}{2\rho} + \int_D u\rho|_0^1 - \int_D \int_0^1 (\partial_t u \rho + \nabla u \cdot m) - \lambda(\rho - s^2) \tag{18}$$

Based on the set of Karush–Kuhn–Tucker (KKT) conditions ($\partial_m L = 0$, $\partial_u L = 0$, $\partial_\rho L = 0$, and $\lambda \geq 0$), we would have:

$$\begin{cases} \partial_m L = \frac{m}{\rho} - \nabla u = v - \nabla u = 0 \\ \partial_u L = \partial_t \rho + \nabla \cdot m = 0 \\ \partial_\rho L = -\frac{||m||^2}{2\rho^2} - \partial_t u - \lambda = -\frac{1}{2}||v||^2 - \partial_t u - \lambda = 0 \end{cases} \tag{19}$$

where the first condition indicates that the gradient ∇u acts as the velocity field, and the third condition implies the optimal solution is given by the generalized HJ equation:

$$\partial_t u + \frac{1}{2}||\nabla u||^2 = -\lambda \leq 0 \tag{20}$$

We thus apply the generalized HJ equation (*i.e.,* $\partial_t u + \frac{1}{2}||\nabla u||^2 \leq 0$) as the constraints. We further use an extra negative force because this would give more dynamics for modeling the posterior flow.

Time complexity

	Ours (weakly-supervised)	Ours (supervised)	PoFlow	TVAE
Time (s)	1.076	0.723	0.638	0.591

Semi-supervised setting

Q6. Semi-supervised Learning

Thanks for the insightful advice. Of course, our method can be extended to the semi-supervised setting. One straightforward implementation could be switching between inferring the transformation index from sequences and directly enforcing the index during the training process. We test such a possibility on MNIST and present the results below:

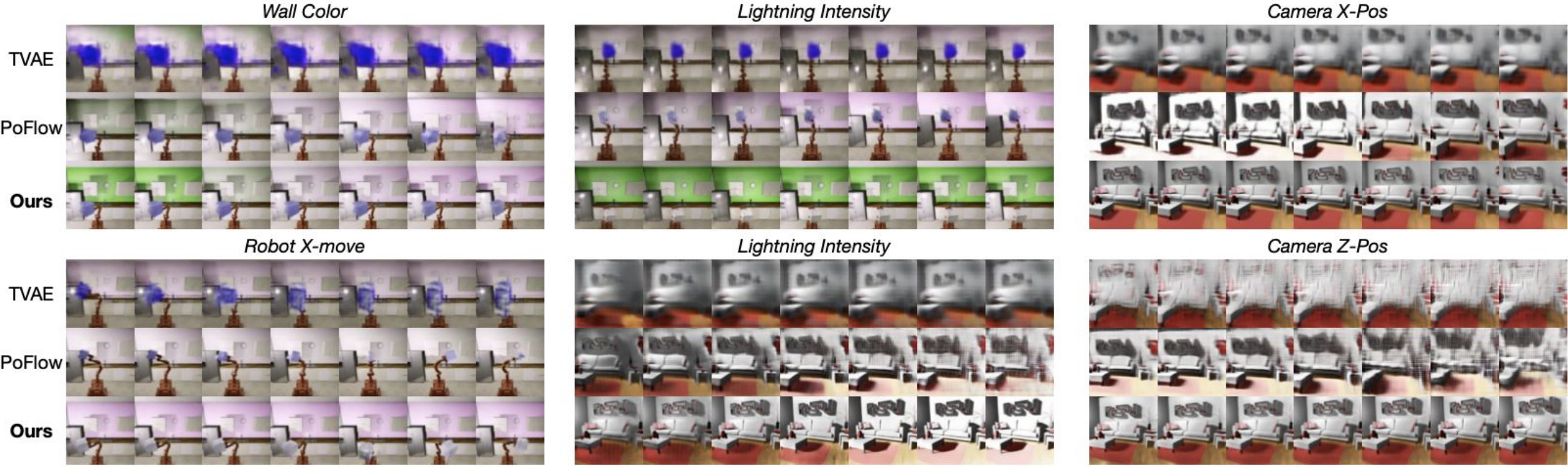
Equvariance error on MNIST under different supervision settings.

Supervision	Scaling	Rotation	Coloring
Weakly	193.84±2.47	157.16±3.24	165.19±2.78
Semi	186.07±2.58	152.87±3.36	160.13±2.82
Full	185.42±2.35	153.54±3.10	158.57±2.95

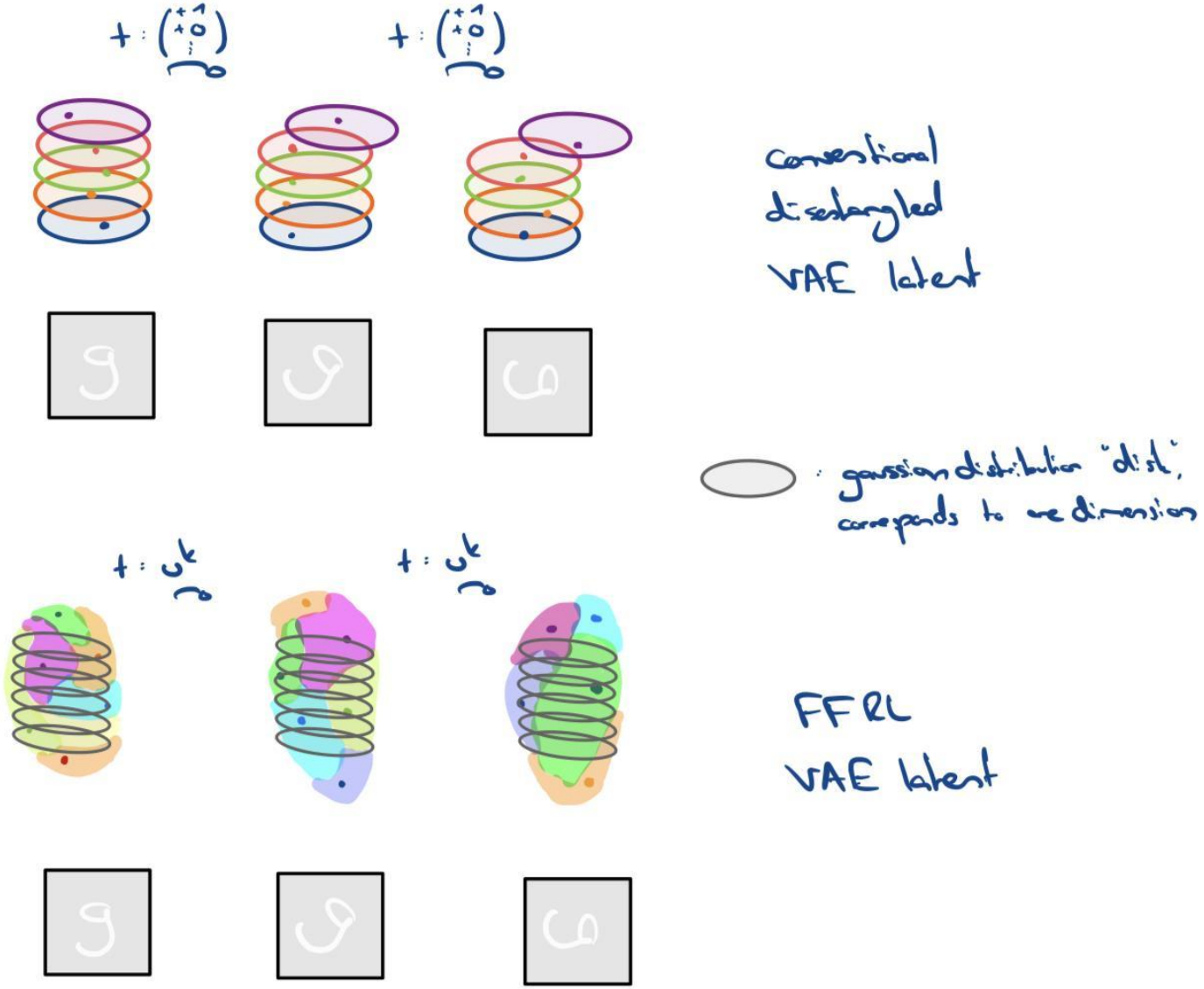
As can be seen, this semi-supervised setting has very close and even competitive performance against the supervised setting. We also observed accelerated convergence of this setting compared with the weakly-supervised fashion.

Another interesting semi-supervised learning setting is directly enforcing the index for some of the transformations while inferring the index for others. This setting might be useful when the number of transformations to model is very large. We would add a paragraph in the revised paper to discuss the many intriguing possibilities!

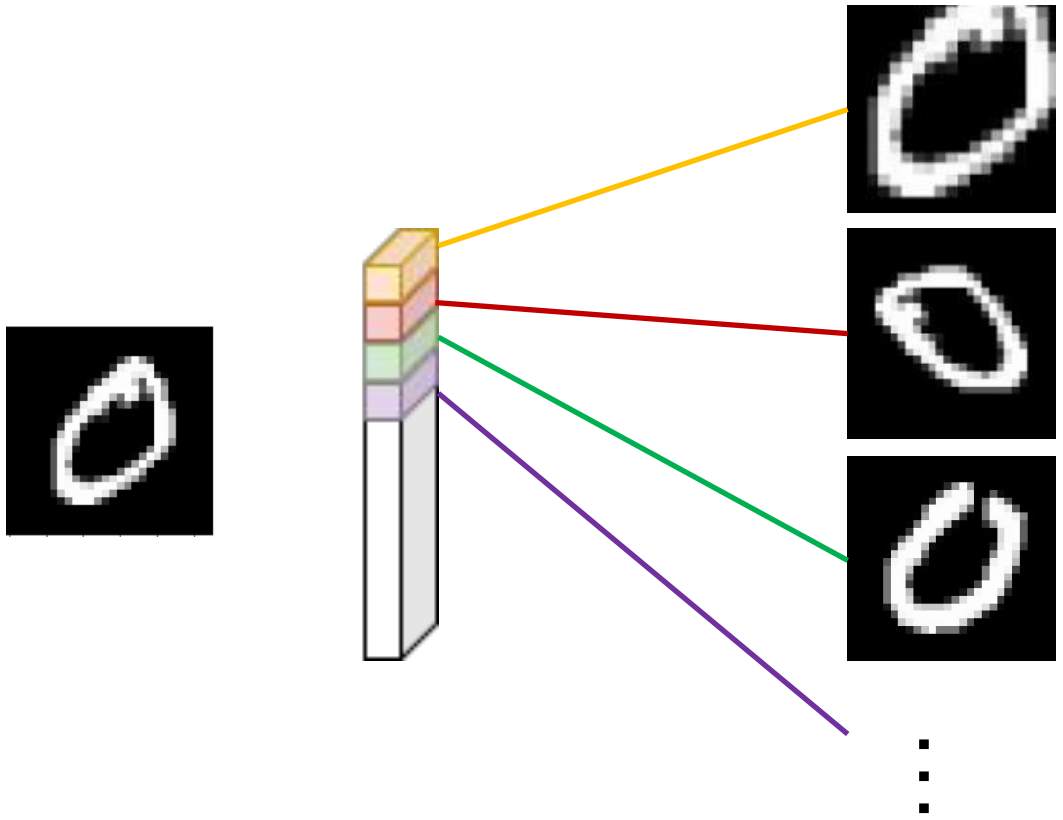
Full qualitative results (Falcor3D, Isaac3D)



Implementation



Disentanglement & Equivariance



$$T'[f(x)] = f(T[x])$$