Neural Fourier Transform: Learning group representation from data

Kenji Fukumizu

The Institute of Statistical Mathematics / Preferred Networks

Workshop: Mathematics of data streams: signatures, neural differential equations, and diffusion models

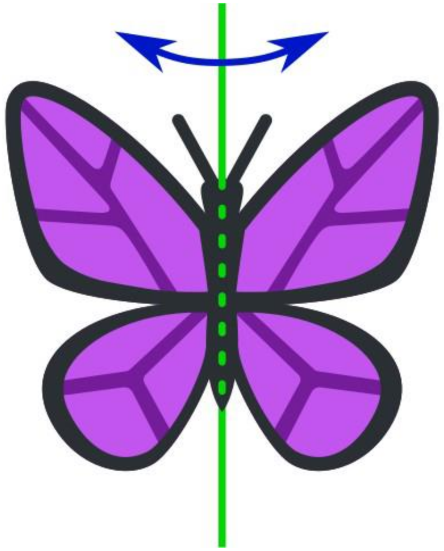
April 11, 2024 @ Greifswald, Germany

Outline

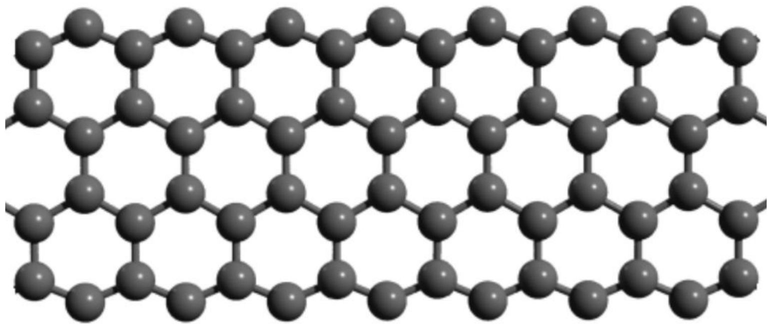
1. Group actions in machine learning
   A quick review of two major existing approaches

2. A new approach: Neural Fourier Transform
   1. Representation learning
   2. Neural Fourier Transform: Equivariant representation learning
   3. Theory
   4. Experiments

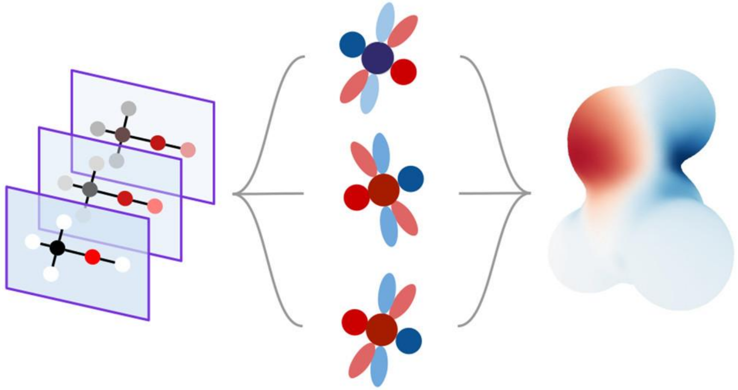
Symmetry/group action exits in nature



Reflection



Crystal lattice



[Thürlemann et al. *J. Chem. Theory Comput.* 2022]

Atomic configuration/static potential

<u>Def.</u>

G: group, X: set. An <u>action</u> of G on X is a mapping $\alpha: G \times X \to X$ such that

i) $\alpha(e, x) = x$

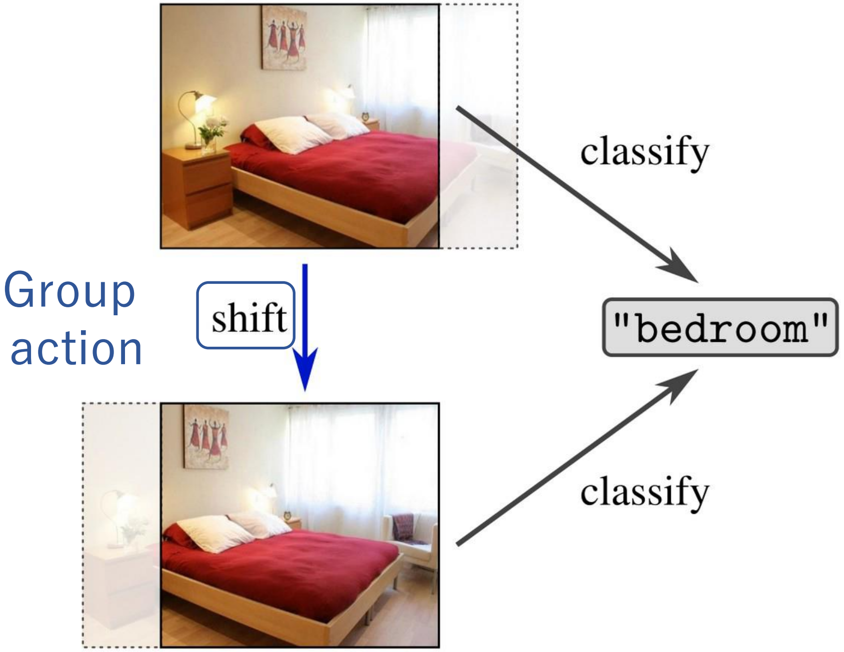
ii) $\alpha(hg, x) = \alpha(h, \alpha(g, x))$

for any $x \in X$ and $g, h \in G$. Denoted by $g \circ x := \alpha(g, x)$.

Invariance and equivariance in machine learning

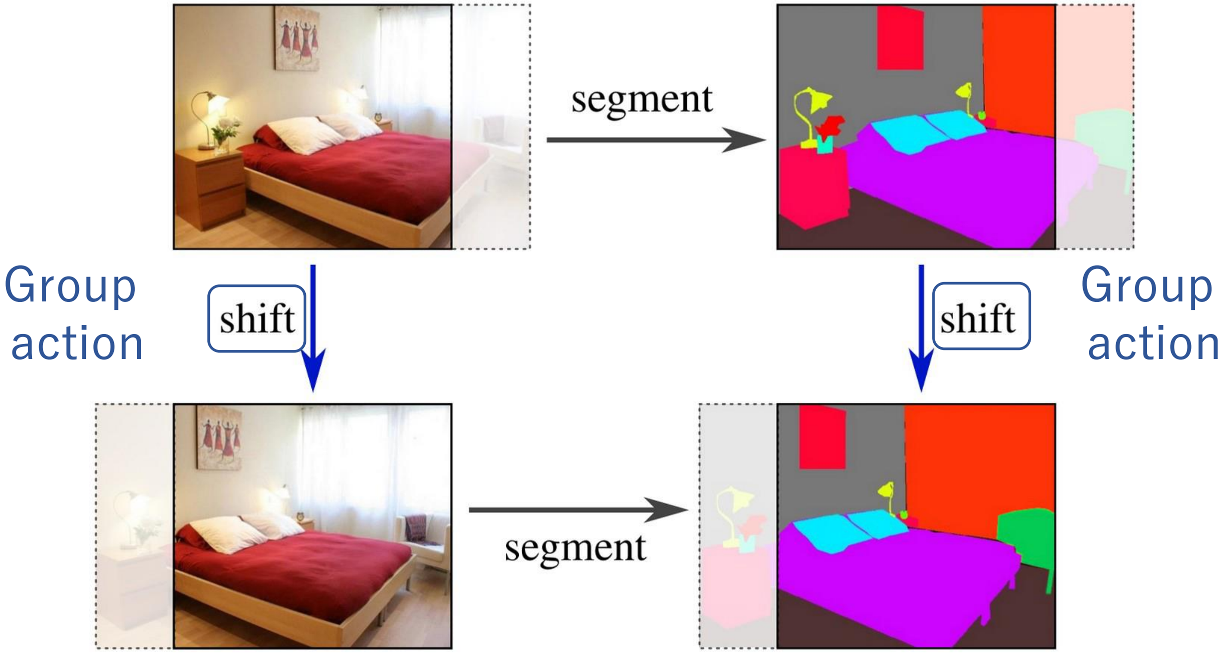
Invariant
Object classification

$\varphi: X \to Y, G \curvearrowright X,$
$\varphi(g \circ x) = \varphi(x)$
$(\forall g \in G, \forall x \in X)$



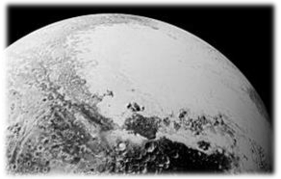
Equivariant
Segmentation

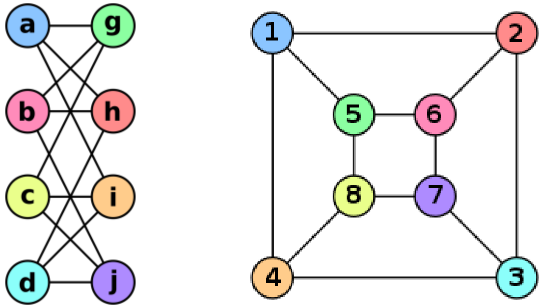
$G \curvearrowright X, G \curvearrowright Y,$
$\varphi(g \circ x) = g \circ \varphi(x)$
$(\forall g \in G, \forall x \in X)$



From “Groups, Representations & Equivariant maps” by Maurice Weiler (University of Amsterdam)

- Group actions in machine learning
  - Various data naturally has symmetry and group actions.
    - Image: shifts, SO(2)-rotation, ⋯
    - Spherical data: SO(3)-rotations
    - Graphs: permutation/graph isomorphism
  - Incorporating such group actions should be useful for the compact representation:
    - Data: Low dimensional expression
    - Model: Smaller models, efficient learning





A remark: Invariance vs Equivariance

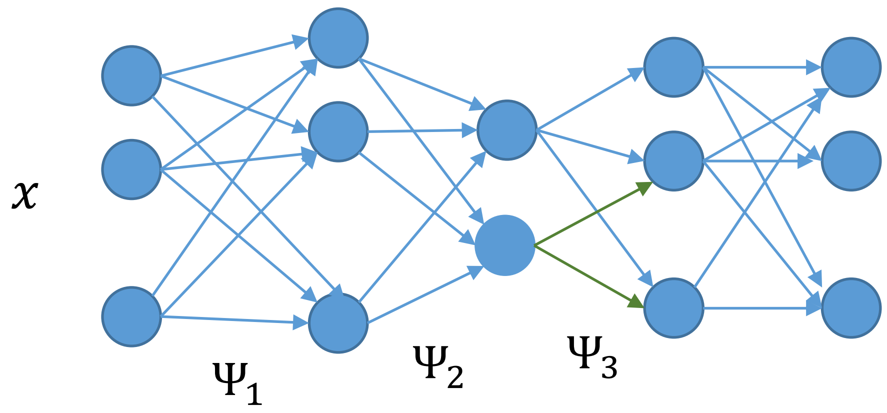
- Equivariance is usually more focused in machine learning/deep learning.

  - Invariance can be added only at the end.

    If Ψ_ℓ's are all equivariant, adding an invariant layer Φ in final layer

    $$\Phi \circ \Psi_L \circ \cdots \circ \Psi_1(x)$$

    makes an invariant mapping.



Two major approaches to equivariance in machine learning

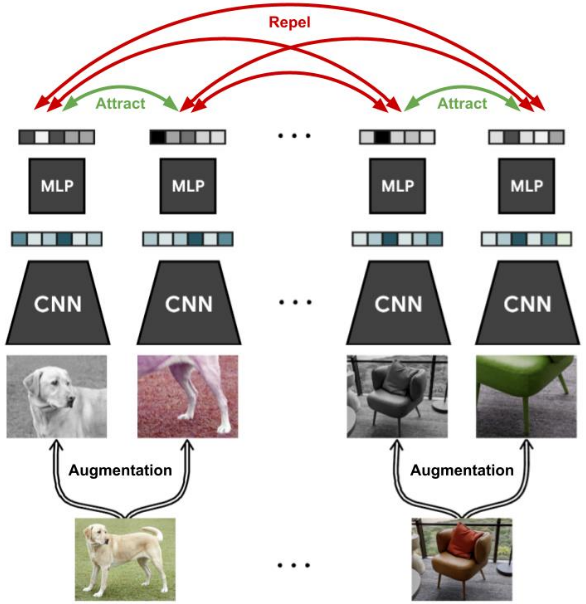
1) Data augmentation:

Augment the training data with the known group actions.

- Easy. Extendable to non-group cases.
- Needs many training data.

- In addition to supervised learning,
  self-supervised learning often uses data augmentation.
  e.g. SimCLR (Chen et al 2020), CPC (Oord et al 2018), etc.

  * Self-supervised learning:
  learning of features without class-labels/teaching data.
  It avoids the cost of labeling/annotation.



SimCLR

2) Architecture:

Convolutional Neural Networks (CNN)

Imposes the symmetry in the architecture.

- Original CNN considers translation equivariance by convolution layers.
- Convolutional layer:
  - 2D gray-scale image f of $W \times H$ pixels.
  - $\psi_{[a,b]}: \mathbb{Z}^2 \to \mathbb{R}$. A spatial filter of small size (e.g. 3×3).

$$h^{out} = \psi * f^{In}$$

i.e., $h^{Out}_{[i,j]} = \sum_{i-a \in \{0,\pm 1\}} \sum_{j-b \in \{0,\pm 1\}} \psi_{[i-a,j-b]} f^{In}_{[a,b]}$

$$f^{Out}_{[i,j]} = \phi\left(h^{Out}_{[i,j]} + \theta\right)$$

  - Equivariance:

Def. $(L_s f)[i,j] \coloneqq f(i - s_W, j - s_H)$, $\left(s = (s_W, s_H)\right)$.

Then, $L_s(\psi * f^{In}) = \psi * (L_s f^{In})$

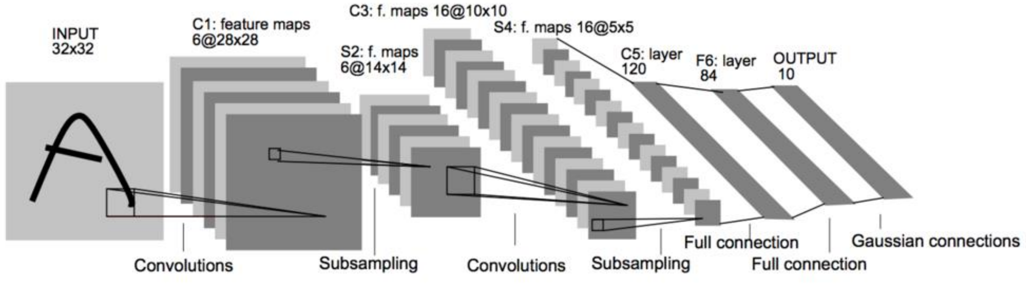
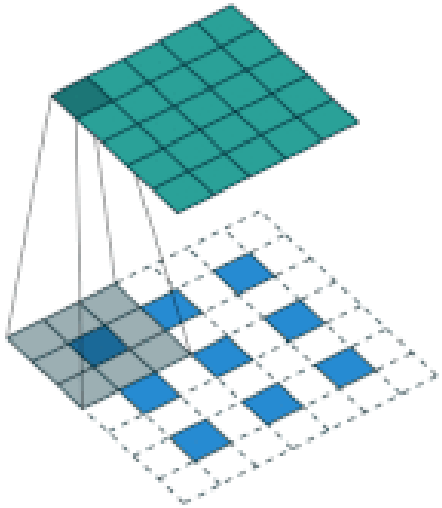
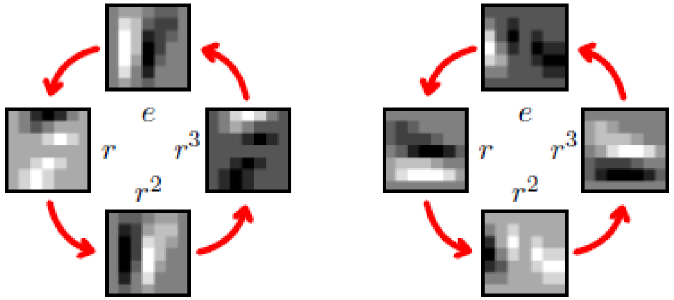


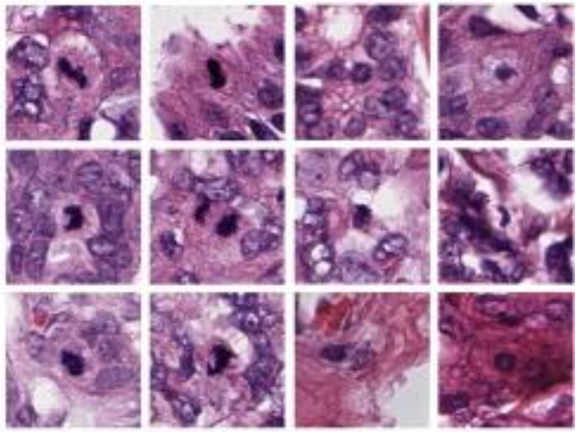
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun 1998

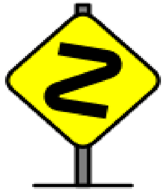


- More general groups
  - Group CNN: SO(2), SE(2), SO(3), etc.
    (Cohen & Welling ICML 2016; Cohen & Welling ICLR 2017; Weiler&Cesa NeurIPS 2019; Weiler et al NeurIPS 2018)
  - Competitive results in applications
    e.g. Medical images
    (Weiler, Hamprecht, Storath. CVPR 2018; Lafarge et al, *Medical Image Analysis* 2021)





Our study:
Equivariant Representation Learning

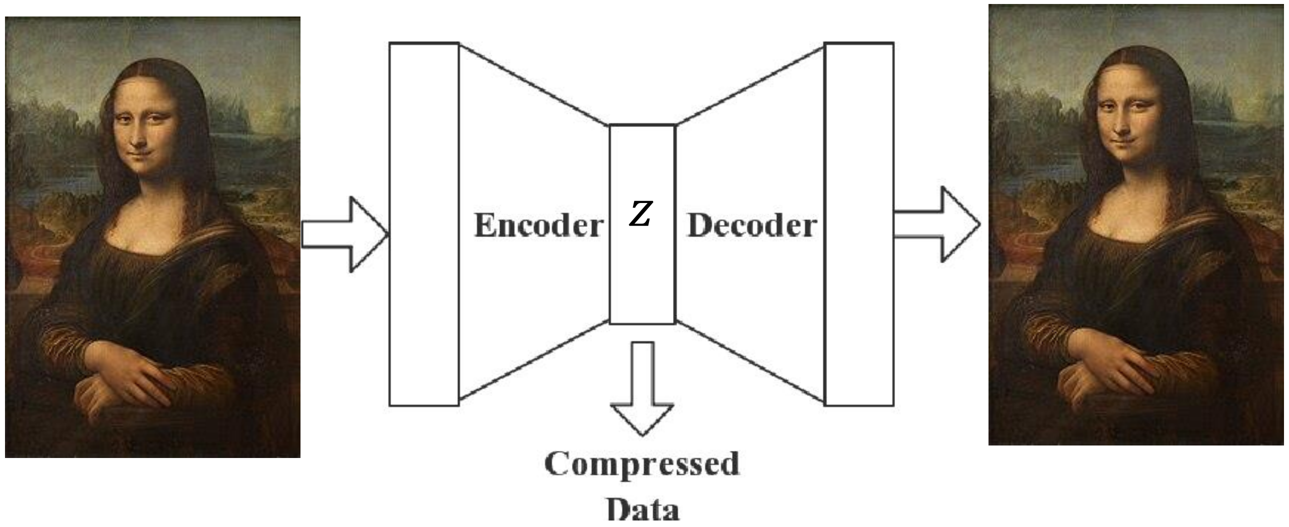


In ML, “representation” refers to a useful expression of data, which is often learned as a mapping of data.

In Math, “representation” refers to a homomorphism between the relations of two different (algebraic) objects, such as representation of a group.

Autoencoder: a typical model for representation learning

$$x \approx \Psi\big(\Phi(x)\big)$$



Latent representation

- Model for data compression

$$\min_{w,\theta} \sum_x \left\| x - \Psi_w\big(\Phi_\theta(x)\big) \right\|^2$$

$z = \Phi_\theta(x)$: encoder (Neural)
$\tilde{x} = \Psi_w(z)$: decoder (Neural)

- Representation learning

The latent variable z is expected to encode meaningful expression/representation of data.

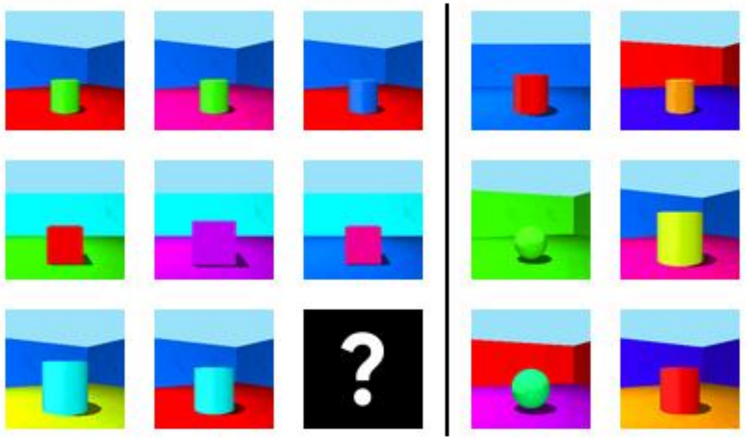
Factorized representation

- Factorized/disentangled representations

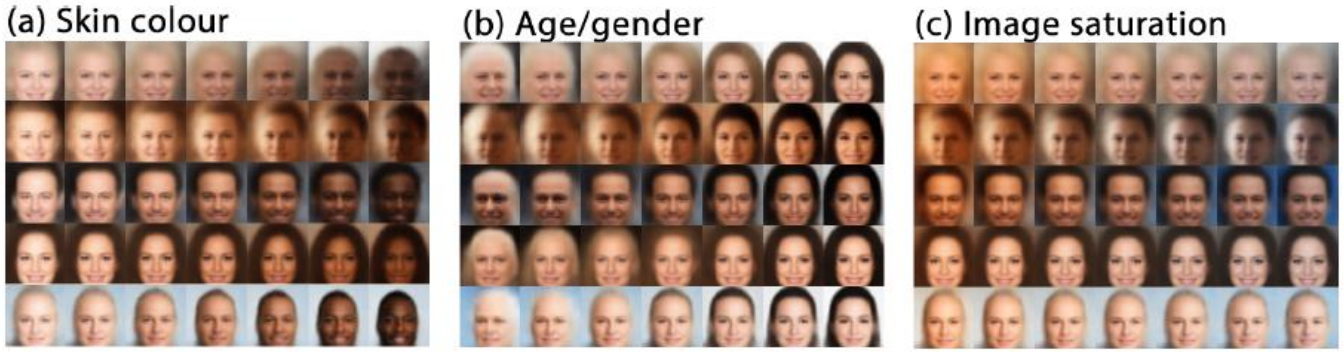
  $z = (z_1, \dots, z_d)$: each z_i has a specific role.

  - Easier interpretation
  - Control of each factor
  - Imposing factorization

    β-VAE (Higgins et al 2017),
    FactorVAE (Duan et al 2022) etc



Locatello et al ICML2020

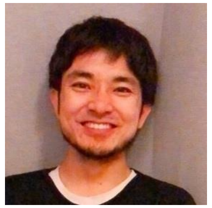


β-VAE (Higgins et al 2017)

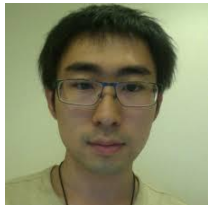
- We use "group representation"
  for factorized/disentangled representation learning.

Neural Fourier Transform:
Learning group representation from data

Miyato, Koyama, Fukumizu. NeurIPS 2022;
Koyama, Fukumizu., Hayashi, Miyato. ICLR 2024



Takeru Miyato
(U Tubingen/
Preferred Networks)



Masanori Koyama
(Preferred Elements)

Kohei Hayashi
(Preferred Elements)

Review: Fourier Transform

- Group representation

  $\rho: G \to GL(V)$, group homomorphism, V: vector space.

- LCA (locally compact Abelian) group G

  $$\Phi: L^2(G) \to L^2(\hat{G}), \quad f \mapsto \hat{f}(\rho) := \int_G f(x)\overline{\rho(x)}dx.$$ <u>Isometry.</u>

  $\hat{G}$: the set of continuous characters (1-dim irreducible representations).

- Compact group

  $[\rho] \in \hat{G} := \{$equivalence classes of irreducible unitary representations of $G\}$

  Unitary map $\Phi: L^2(G) \to \mathcal{B}^2(\hat{G}) := \ell^2\left(\bigoplus_{[\rho]\in\hat{G}} \mathcal{B}_2(H_\rho)\right)$,

  $$f \mapsto \hat{f}(\rho) := \int_G f(g)\rho(g)^* dg = \int_G f(g)\rho(g^{-1})dg$$

  (ρ, H_ρ); unitary repr. $\mathcal{B}_2(H_\rho)$: Hilbert space of Hilbert-Schmidt operators on H_ρ.

Equivariance of Fourier Transform

$f \in \mathcal{F}(G)$ function on G (compact or LCA group)
L_g: shift operation on $\mathcal{F}(G)$. $(L_g f)(h) \coloneqq f(g^{-1}h)$.
$\rho: G \to GL(V)$ representation of G.

$$\begin{tikzcd} \mathcal{F}(G) \arrow[r, "\text{FT } \Phi"] \arrow[d, "L_g"'] \arrow[dr, phantom, "\circlearrowleft"] & \mathcal{B}^2(\hat{G})) \arrow[d, "A_g"] \\ \mathcal{F}(G) \arrow[r, "\text{FT } \Phi"'] & \mathcal{B}^2(\hat{G}) \end{tikzcd}$$

<u>Prop</u>. (Equivariance of Fourier transform)

$$\Phi \circ L_g = A_g \circ \Phi,$$

where A_g acts on $\ell^2\left(\bigoplus_{[\rho]} \mathcal{B}_2(H_\rho)\right)$ by $A_g\left(B_\rho\right)_{\rho\in\hat{G}} = \left(B_\rho \rho(g^{-1})\right)_{\rho\in\hat{G}}$.

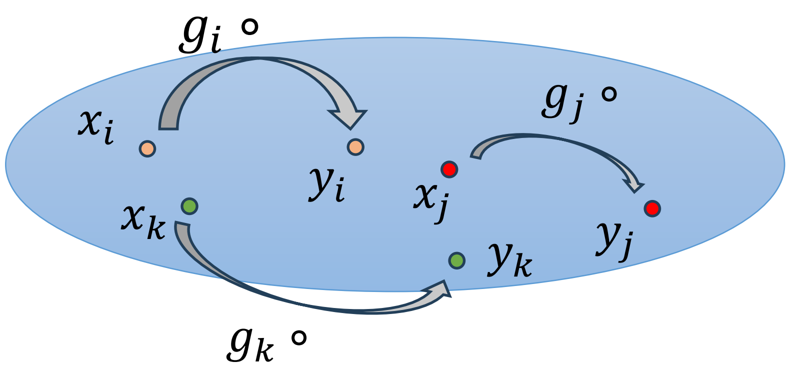
$$\because) \int_G f(g^{-1}h)\rho(h^{-1})dh = \int_G f(\tilde{h})\rho(\tilde{h}^{-1}g^{-1})d\tilde{h} = \int_G f(\tilde{h})\rho(\tilde{h}^{-1})\rho(g^{-1})d\tilde{h} = \Phi(f)(\rho)\rho\left(g^{-1}\right)$$

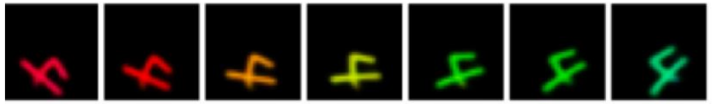
We use the objective function of NN training to approximately realize this equivariance relation.

Equivariant Representation Learning

(Miyato, Koyama, F. NeurIPS 2022; Koyama, F., Hayashi, Miyato ICLR 2024)

- General setting of data
  - Some group G acts on data space $\mathcal{X}$.
  - Data: many examples of group action
    - Paired data: $(x_i, g_i \circ x_i)$ $\quad x_i \in \mathcal{X},\ g_i \in G$
    - Sequences: $\left(x_i, g_i \circ x_i, g_i^2 \circ x_i, g_i^3 \circ x_i, \dots\right)$
    - Triplet: $\left(x_i, g_i \circ x_i, g_i^2 \circ x_i\right)$





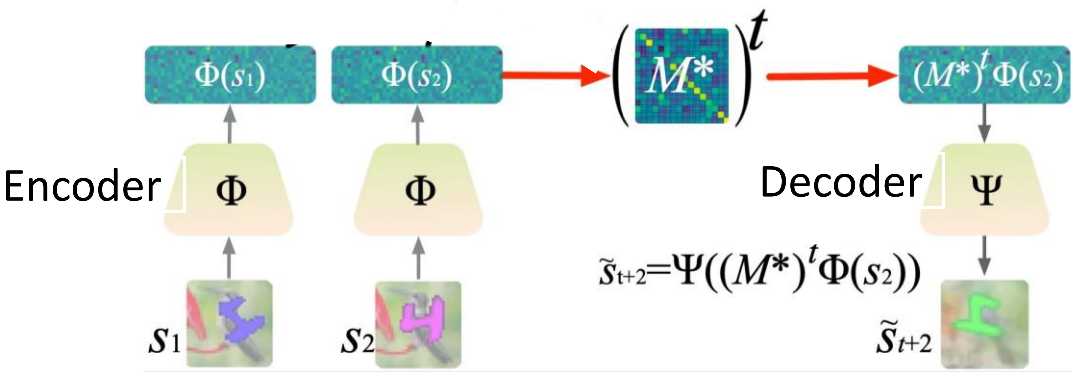
- Equivariant Autoencoder

  Encoder: $\Phi: \mathcal{X} \to \mathbb{R}^{m\times a}$,

  Decoder: $\Psi: \mathbb{R}^{m\times a} \to \mathcal{X}$

  M: matrix applied to the latent space

$$\boxed{\Psi\left(M_g^t \Phi(s)\right) \approx g^t \circ s}$$



Existing approaches to equivariant learning

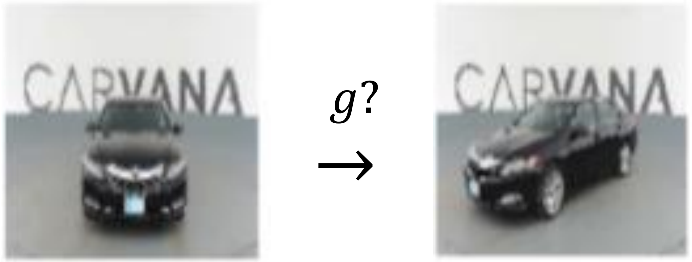
- Group G and its action are explicitly known
  - CNN: Built-in architecture to a specific group
  - Data augmentation： augmentation using the group action

This work

- Group action does exist, but may not be known explicitly
  - Not acting on the data space
  - May be observed with unknown nonlinearity

- Approach:

  Learn the group representation from data by equivariance constraint.

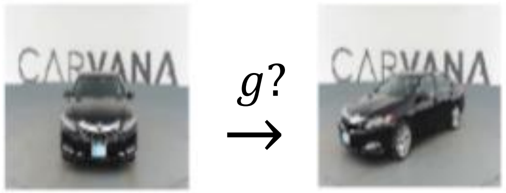


Rotation in the *latent* space

Nonlinear observation（fisheye lens）

- Various scenarios

	G	element g	Action	M_g
U-NFT	Unknown	Unknown	Unknown	Learning
G-NFT	Known	Unknown	Unknown	Harmonic functions
g-NFT	Known	Known	unknown	Harmonic functions Learn only Φ, Ψ



Hardest setting: Unsupervised Learning of Equivariant Structure from Sequences (Miyato et al. NeurIPS 2022)

- U-NFT: Neither G or $g \in G$ is known

- Data: many sequences $\{\mathbf{s}^{(i)}\}_{i=1}^{N}$

  $\mathbf{s}^{(i)} = (s_0^{(i)}, s_1^{(i)}, s_2^{(i)}, \dots, s_T^{(i)})$. $s_t^{(i)} \in \mathcal{X}$

- Sequence is driven by group action

  G: group (unknown) acting on $\mathcal{X}$.

  $\exists g^{(i)} \in G$ s.t. $s_t^{(i)} = (g^{(i)})^t \circ s_0^{(i)}$.

  - Each seq $\mathbf{s}^{(i)}$ has its own $g^{(i)} \in G$, but unknown.
  - Stationarity: $g^{(i)} \in G$ is the same in a sequence.

$\mathbf{s}^{(1)} = \mathbf{s}\left(g^{(1)}, s_0^{(1)}\right)$ e.g. $g^{(1)} = (15^\circ$ rot, (-2,1)shift, ΔRGB)

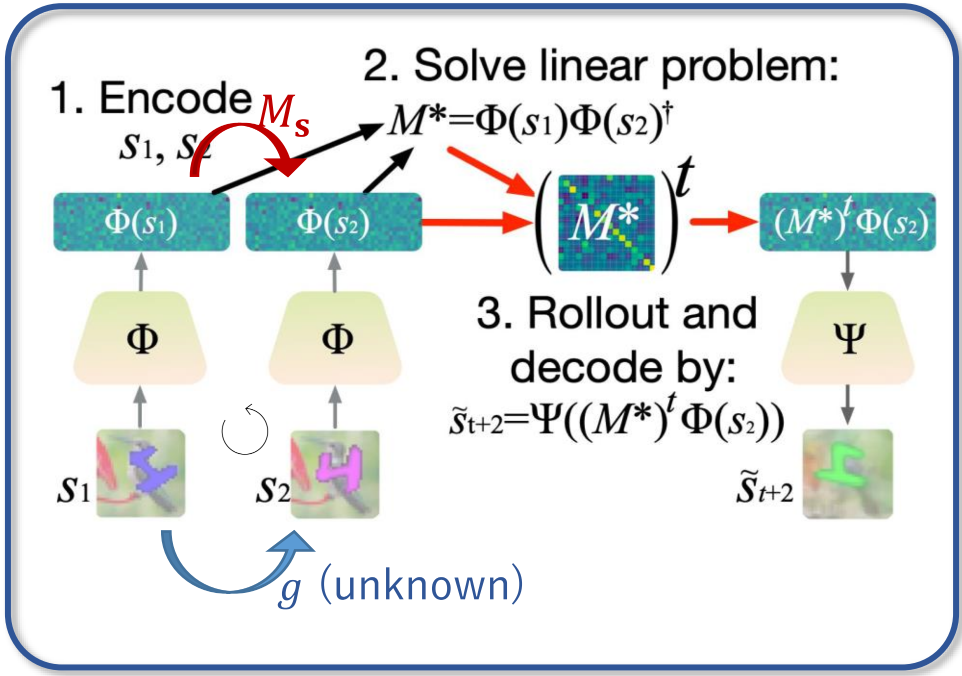
$\mathbf{s}^{(2)} = \mathbf{s}\left(g^{(2)}, s_0^{(2)}\right)$

Rotation, Shifts, Color rotation

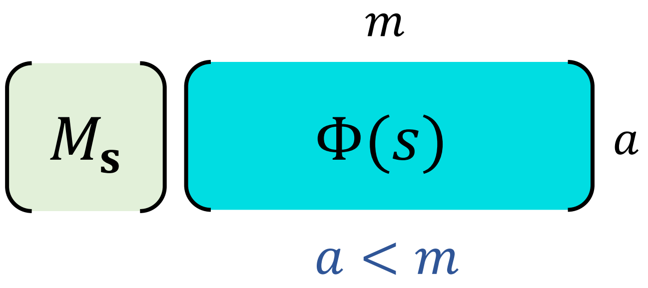
A sequence $\boldsymbol{s}$ is generated given an initial image s_0 and a group element $g \in G$.

Model: Linear latent transition with AE

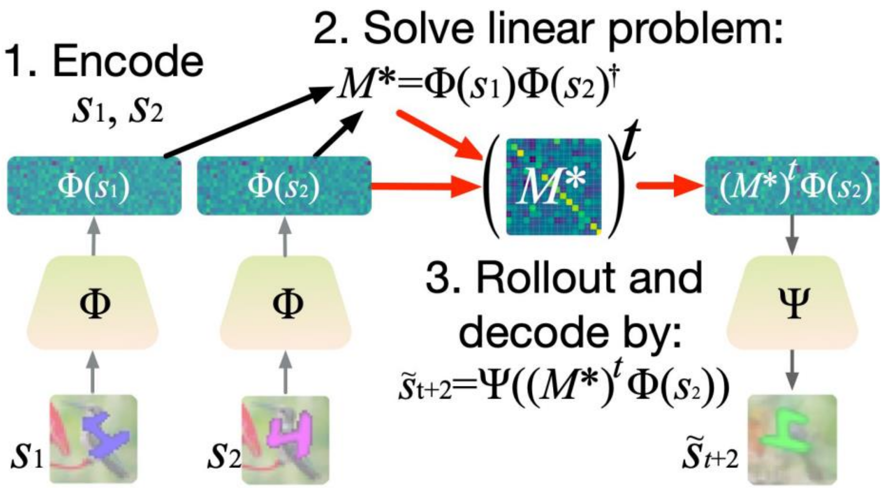
- Autoencoder.
  Encoder $\Phi: \mathcal{X} \to \mathbb{R}^{a \times m}$, Decoder $\Psi: \mathbb{R}^{a \times m} \to \mathcal{X}$
- Linear transform for latent:
  $M_{\mathbf{s}} \in GL(\mathbb{R}^a)$ depends on sequence $\mathbf{s}$.
  $\Phi(s_{t+1}) = M_{\mathbf{s}} \Phi(s_t)$
- Matrix latent: $\Phi(s) \in \mathbb{R}^{a \times m}$
  Can have a small matrix $M_{\mathbf{s}}$.
  Can incorporate multiplicities of representation.



- Objective function
  $\min E\|M_{\mathbf{s}}\Phi(s_t) - \Phi(s_{t+1})\|^2$ [Equivariance]
  $\min E\left\|\Psi\left(M_{\mathbf{s}}^{\ell}\Phi(s_t)\right) - s_{t+\ell}\right\|^2$ [Pred./Reconst.]



Estimation of $M_{\mathbf{s}}$



- <u>End-to-end algorithm</u>:

  $\mathbf{s} = (s_0, \dots, s_L, s_{L+1}, \dots, s_{L+P})$

  $\mathbf{s}_0^L \coloneqq (s_0, \dots, s_L)$ for estimating M_s , $(s_{L+1}, \dots, s_{L+P})$ for prediction

  - LS estimator M : $\widehat{M}(\mathbf{S}_0^L|\Phi) \coloneqq \arg\min_M \sum_{t=0}^{L-1} \|M\Phi(s_t) - \Phi(s_{t+1})\|^2 = H_{+1}H_{+0}^\dagger$

    $H_{+1} \coloneqq (\Phi(s_1); \cdots; \Phi(s_L)), H_{+0} \coloneqq (\Phi(s_0); \cdots; \Phi(s_{L-1})),$

    $\widehat{M_s}$ depends on sequence $\mathbf{s}$.

  - Learning of Φ, Ψ: $\min_{\Phi,\Psi} \sum_{\mathbf{s}} \sum_{p=1}^{P} \left\| \Psi\left(\widehat{M}(\mathbf{S}_0^L|\Phi)^p \Phi(s_L)\right) - s_{L+p} \right\|^2$

    Plug-in and auto-grad!

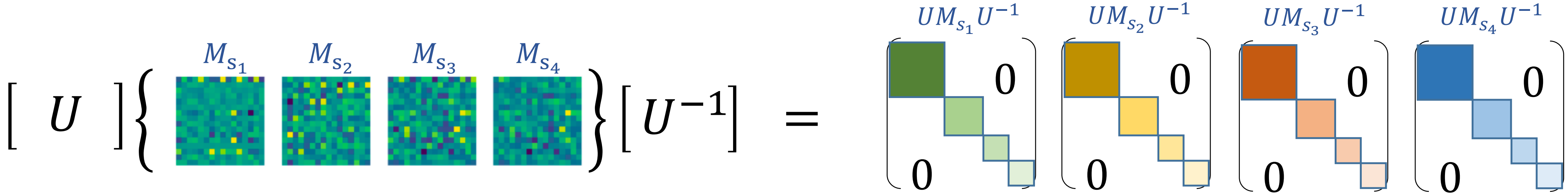
- 3-time steps are sufficient:

  We use $L = 1, P = 1$. $\mathbf{s} = (s_0, s_1, s_2)$ (3 points) $\widehat{M_s} = \Phi(s_1)\Phi(s_0)^\dagger$

Irreducible decomposition of M_s

- Assume $\widehat{M}_s$ depend only on g. Then, $g \mapsto \widehat{M}(g)$ is a group representation. (See next slide)
- Try irreducible decomposition. It will work if the representation is completely reducible.
- Algorithmically, simultaneous block-diagonalization is applicable.

  A common change of basis over all the sequences.



U is common.

Disentangled representation by irreducible decomposition

Theory about $M_{\mathbf{s}}$

Denote $M(g, s_0) := M_{\mathbf{s}}$ ($g \in G, s_0$: initial point of $\mathbf{s}$.)

Prop. Suppose $\Phi(s)$ is of full-rank, and the linear map $M(g, s_0)$ satisfies
the equivariant condition: $M(g, s_0)\Phi(s) = \Phi(g \circ s_0)$ $(\forall s_0 \in \mathcal{X}, \forall g \in G)$.
If $M(g, s_0) = M(g)$,
then, M: $g \mapsto M(g)$ is a group representation of G.

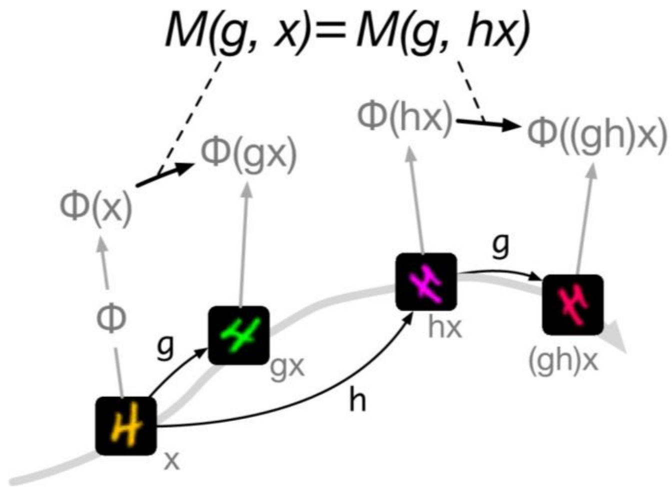
Proof) Easy from the equivariant condition.

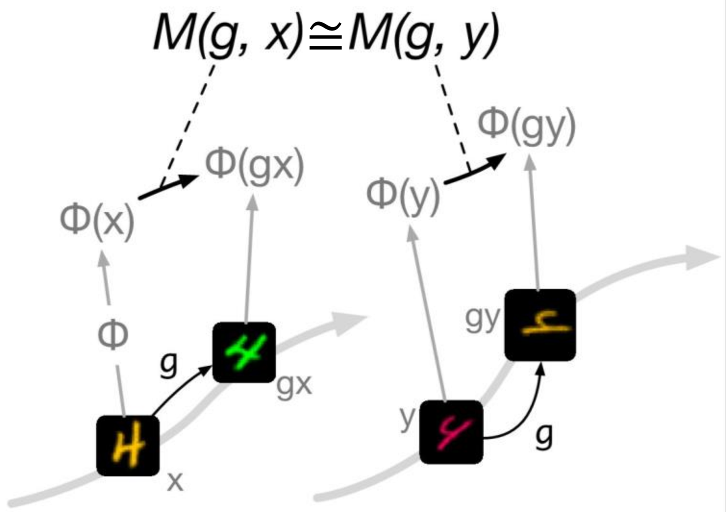
- For the moment, there is no theoretical guarantee that the proposed algorithm realizes the condition $M(g, s_0) = M(g)$.
- We have some partial results on this condition.

- Partial theoretical results on “$M(g, s_0) = M(g)$?”.

Prop. 1. (Intra-orbits.)
G: commutative compact Lie group,
$M(g, h \circ s) = M(g, s) \quad (\forall h \in G,\ s \in \mathcal{X})$.

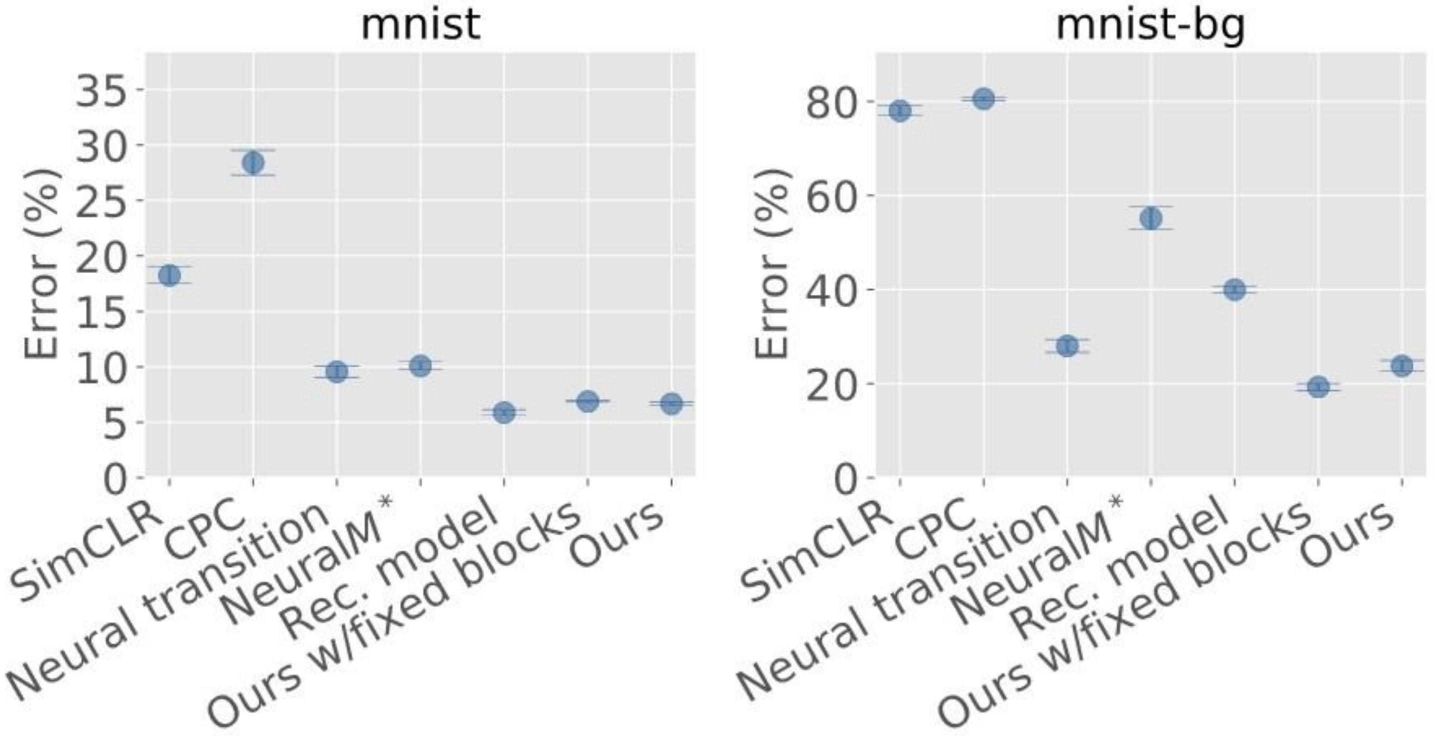
Prop. 2. (inter-orbits) Existence is guaranteed
G: connected Lie group.
We can find matrices P_s such that
$\tilde{\Phi}(s) := P_s \Phi(s),\ \ \tilde{M}(g, s) := P_s M(g, x)$
give $\tilde{M}(g, s) = \tilde{M}(g)$ (does not depend on s)





Experiment 1: Effective representation

- Linear classification with learned latent variable $\Phi(x)$
  - MSP model is trained with only “4”
  - $G = \mathrm{SE}(2) \times$ {Color change}
  - Using the features $\Phi(x)$, we made 10 class linear classifier for “0”,⋯,”9”.

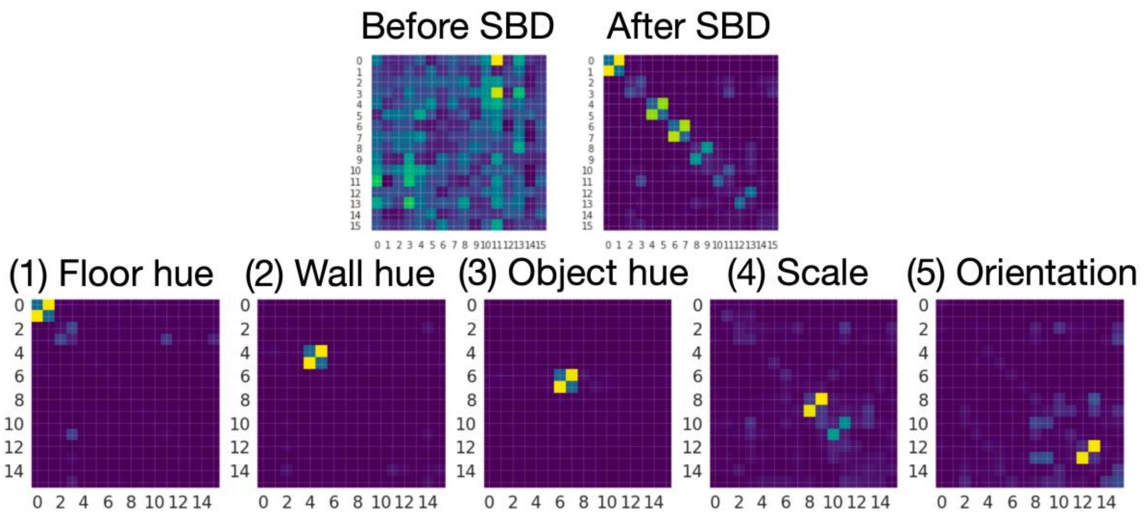


* SimCLR, CPC: standard methods of self-supervised learning

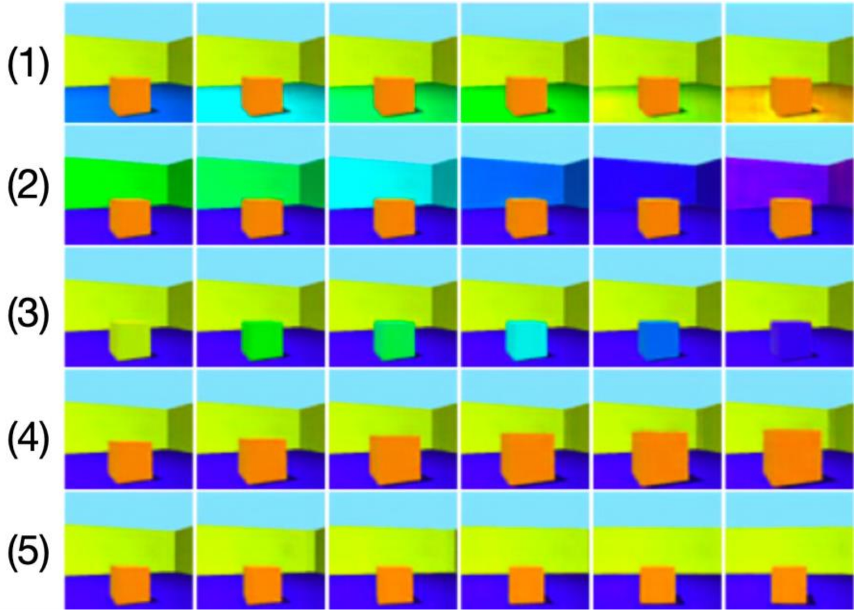
Experiment 2: disentanglement

Rendered image sequences:

G = product group of 5 types of changes. $g \in G$ generates a sequence.



Blocks obtained by simultaneous block diagonalization



Reconstruction by each block

$$\hat{s}_t := \Psi\left(M_b^t \Phi(s_0)\right)$$

Neural Fourier Transform (Koyama et al ICLR 2024)

- Proposed method: a nonlinear generalization of Fourier transform
  - Learning by the equivariance constraint.

$$L_g x \approx \Psi \circ M(g) \circ \Phi(x) = \underbrace{\Psi \circ P^{-1}}_{\text{Inv. Fourier Transform}} \circ \underbrace{\begin{pmatrix} \rho_1(g) & & O \\ & \rho_2(g) & \\ O & & \ddots \end{pmatrix}}_{\text{Irreducible representations}} \circ \underbrace{P \circ \Phi(x)}_{\text{Fourier Transform}}.$$

- *c.f.* Classical Fourier transform on function $\left\{0, \frac{1}{N}, \dots, \frac{N-1}{N}\right\} \subset \mathbb{S}^1$.

$$\hat{f}_n = \Phi(f)(n) = \sum_{k=0}^{N-1} e^{-i2\pi \frac{k}{N}} f\left(\frac{k}{N}\right)$$

Equivariance: $\Phi\big(f(\cdot - a/N)\big) = e^{i2\pi \frac{a}{N} n} \hat{f}_n$

or $L_{a/N} f = \Phi^{-1} \circ \begin{pmatrix} 1 & \cdots & O \\ \vdots & \ddots & \vdots \\ O & \cdots & e^{i2\pi a \frac{N-1}{N}} \end{pmatrix} \circ \Phi(f)$

Properties of Neural Fourier Transform

- NFT works for “data”, while the standard FT works for functions on a group.

- Group action may not be known:

  NFT learns the transforms though data (examples of actions). It may not know the group or group actions.

- Training-based FT

  It uses only necessary frequencies to express the data.

Abstract construction

$$\begin{tikzcd} \mathcal{H} \arrow[r, "M_g"] & \mathcal{H} \\ \mathcal{X} \arrow[u, "\Phi"] \arrow[r, "g\,\circ"'] \arrow[ur, phantom, "\circlearrowleft"] & \mathcal{X} \arrow[u, "\Phi"'] \end{tikzcd}$$

$\mathcal{E} \coloneqq \{\Phi: \mathcal{X} \to V \mid V$: vector space, G-equivariant
with unitary repr. $\rho: G \to \mathrm{GL}(V)$, $V = \mathrm{Cl\,Span}\{\Phi(x) \mid x \in \mathcal{X}\}\}$

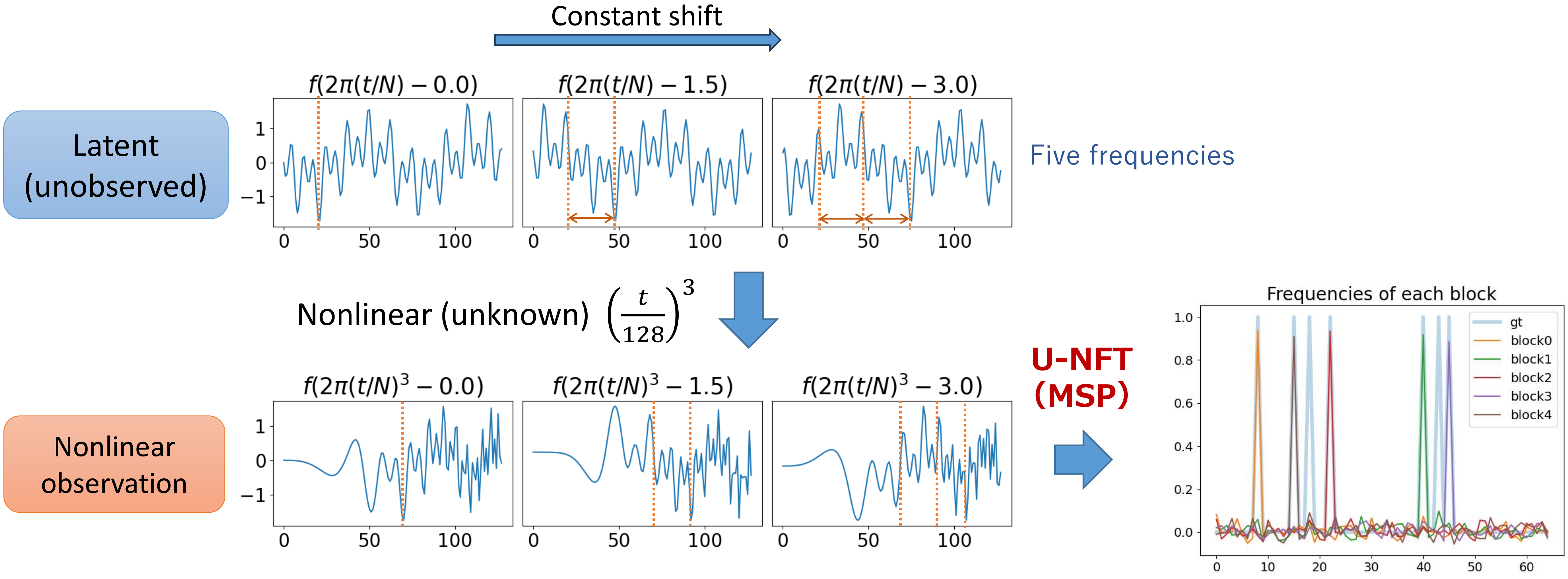
$\mathcal{K} \coloneqq \{k$: G-invariant positive definite kernel on $\mathcal{X}\}$ $\quad k(g \circ x, g \circ y) = k(x, y)$

<u>Theorem</u> $\mathcal{E}$ has one-to-one correspondence with $\mathcal{K}$ up to G-isomorphism.

- If $\Phi: \mathcal{X} \to V$ is in $\mathcal{E}$, then $k(x, y) \coloneqq \langle \Phi(x), \Phi(y) \rangle$ gives an invariant pd kernel.
- Conversely, if K is an invariant pd kernel, the feature map $\Phi: \mathcal{X} \to \mathcal{H}_K$, $x \mapsto K(\cdot, x)$, is equivariant w.r.t. the left regular repr. $L_g f = f(g^{-1} \cdot)$ on $\mathcal{H}_K$.
- An invariant pd kernel is obtained by $k_G(x, y) = \int_G k(gx, gy) d\mu(g)$, for example.
- In general, $\mathcal{H}_K$ is infinite dimensional. We need to approximate it with a finite dimensional latent space.

Experiments

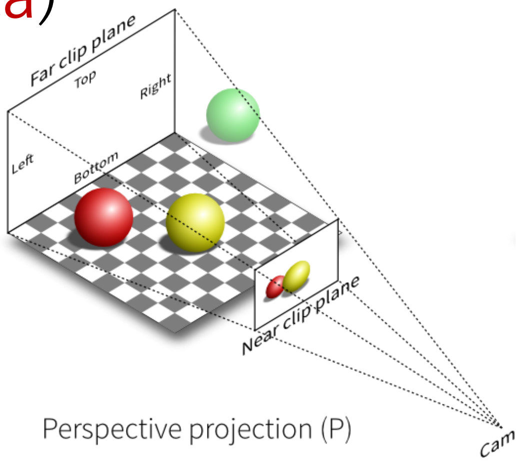
1) Nonlinear observation of 1 dim signal (G unknown, U-NFT)



Obtain correct frequencies.

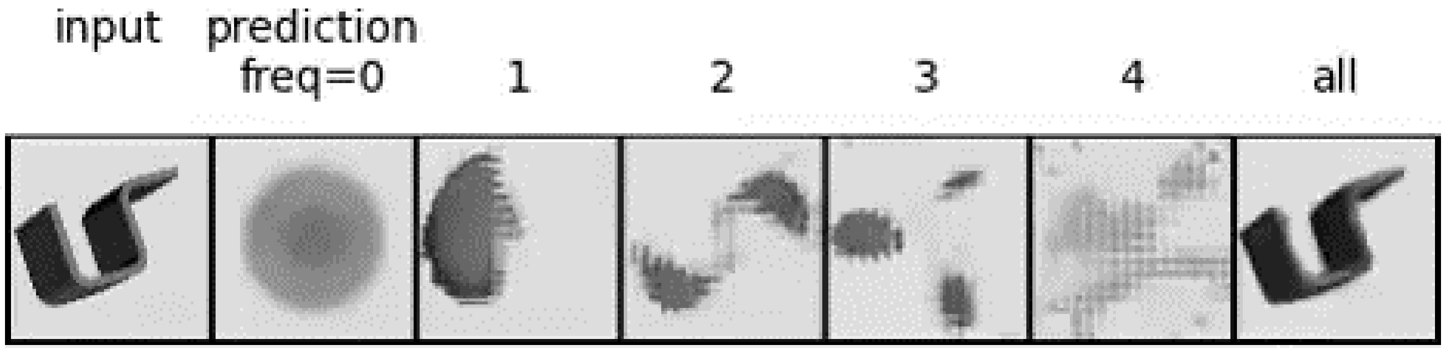
2) g-NFT：Novel view synthesis from 2D training images

- Training with 2D images, which are projected from rendered 3D images.
- Data: Paired 2D images and 3D rotation (S, S', R).
  $R \in SO(3)$, S, S': 2D images, $S' = R \circ S$
- $M(g)$: fixed by spherical harmonics.
  Only encoder Φ and decoder Ψ are trained.
- Testing: Provide a new 2D image X_0 (not in the training data)
  and apply arbitrary 3D rotation g by $\hat{\Psi}\left(M(g)\hat{\Phi}(\mathrm{X}_0)\right)$.



Perspective projection (P)

Arbitrary 3D rotation can be applied in the latent space for “unseen” 2D image (not within training data).



Conclusions

- Group actions are useful in machine learning.
  - Conventional methods:
    - Data augmentation
    - Built-in architecture: (Group) convolutional neural networks

- A new method: equivariant representation learning.
  - Group representation is learned in the latent space.
  - Group action may not be known.
  - Neural Fourier Transform
    - Training-based Fourier transform.
    - Useful in representation learning
      - Disentanglement, adaptivity of frequencies, etc.