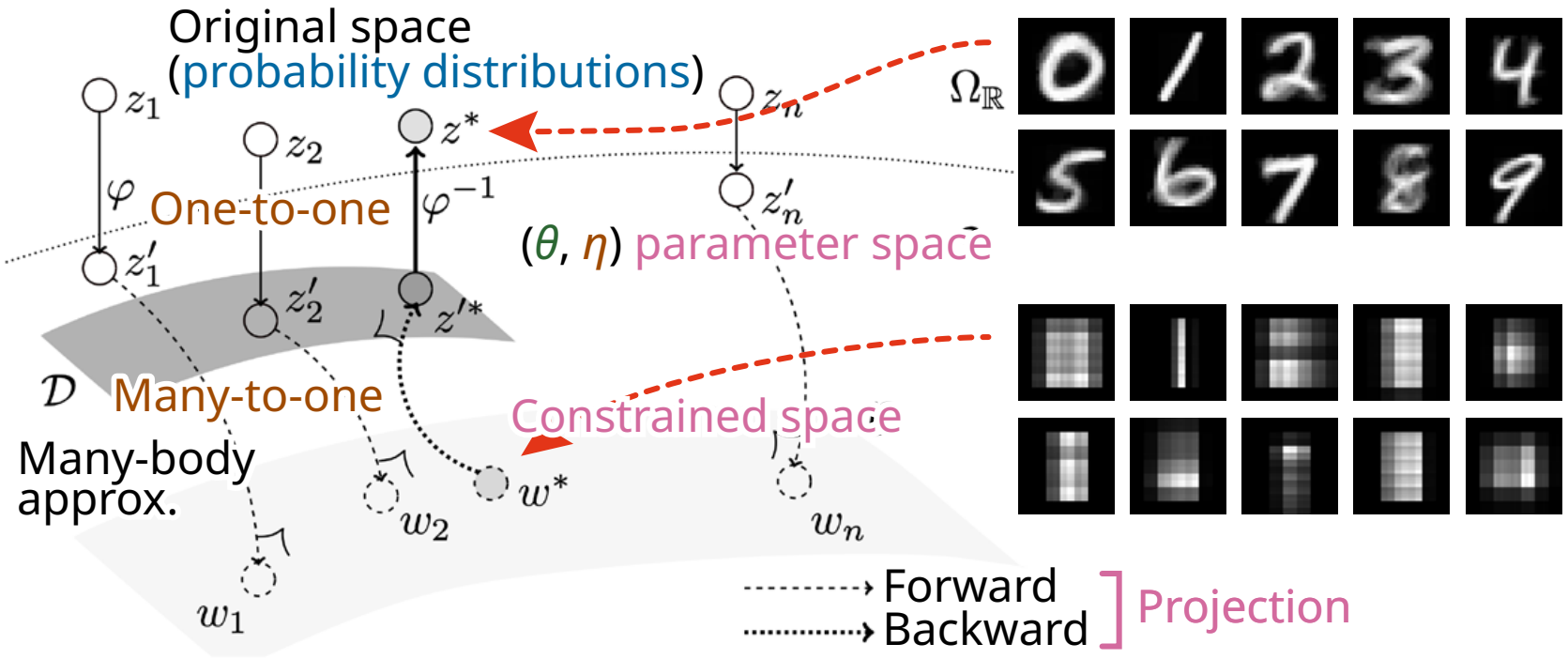
Mar. 4, 2026
BAQ2026

Inter-University Research Institute Corporation /
Research Organization of Information and Systems
National Institute of Informatics

Information-Geometric Modeling and Learning

Mahito Sugiyama (NII) https://mahito.nii.ac.jp/

Data Augmentation [Hu & Sugiyama, ICLR2026]



Formulation

1. **Data:** Points (probability distributions) on a statistical manifold
2. **Model:** Parameterization via (θ, η)-coordinates
   - We use the log-linear model on posets [Sugiyama et al. ICML2017]
     - Explicitly model any-order interactions between variables
     - Generalization of **higher-order Boltzmann machines** [Sejnowski,1986] and its **information geometry** [Amari, 2001]
3. **Learning:** (Forward) Projection onto constrained space
   - We use many-body approximation [Ghalamkari et al. NeurIPS2023]
4. **Correctness:** Convex optimization

Formulation

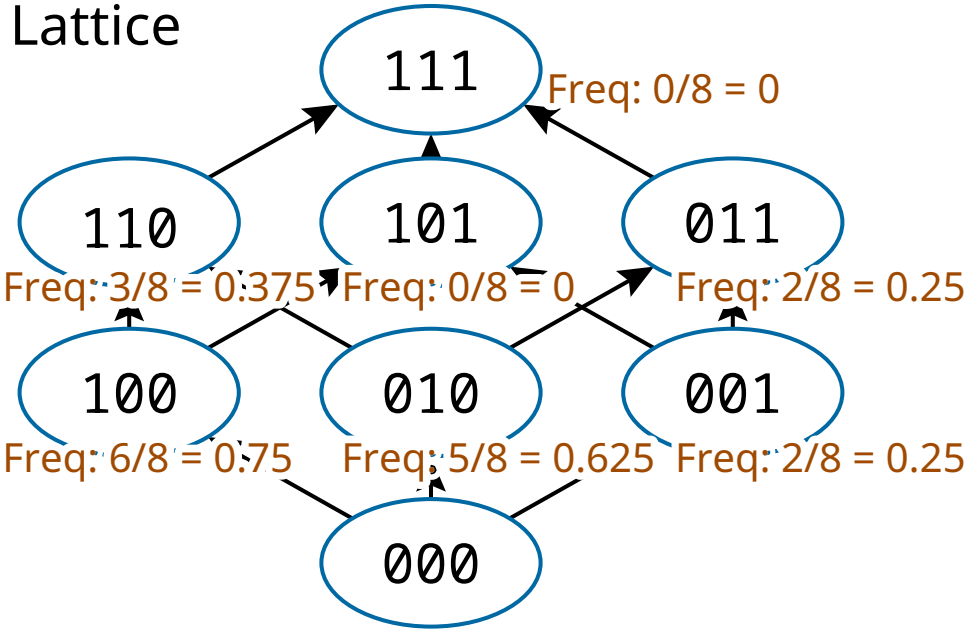
1. **Data:** Points (probability distributions) on a statistical manifold
2. **Model:** Parameterization via (θ, η)-coordinates
   - We use the log-linear model on posets [Sugiyama et al. ICML2017]
     - Explicitly model any-order interactions between variables
     - Generalization of **higher-order Boltzmann machines** [Sejnowski,1986] and its **information geometry** [Amari, 2001]
3. **Learning:** (Forward) Projection onto constrained space
   - We use many-body approximation [Ghalamkari et al. NeurIPS2023]
4. **Correctness:** Convex optimization

Example: Hierarchical Distribution (1/2)

Dataset

	Bread	Milk	Apple
ID 1	1	0	0
ID 2	1	1	0
ID 3	1	0	0
ID 4	0	1	1
ID 5	0	1	1
ID 6	1	1	0
ID 7	1	0	0
ID 8	1	1	0

Lattice

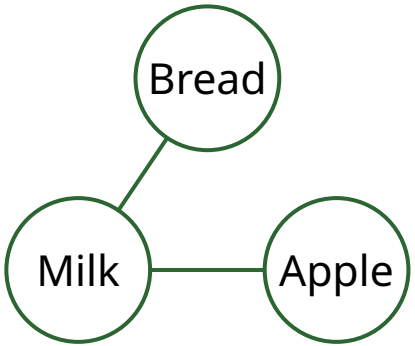


Example: Hierarchical Distribution (2/2)

log(prob.) = –10.41 + 9.43[Bread] + 8.52[Milk] – 9.84[Apple]
– 9.03[Bread&Milk] + 9.43[Milk&Apple]

MLE

Boltzmann machine



Bread	Milk	Apple	Probability from data	Learned probability
0	0	0	?	0.0000300109
1	0	0	0.375	0.3749599867
0	1	0	?	0.1499903954
0	0	1	?	0.0000000016
1	1	0	0.375	0.2250096042
1	0	1	?	0.0000200043
0	1	1	0.25	0.0999895960
1	1	1	?	0.1500004008

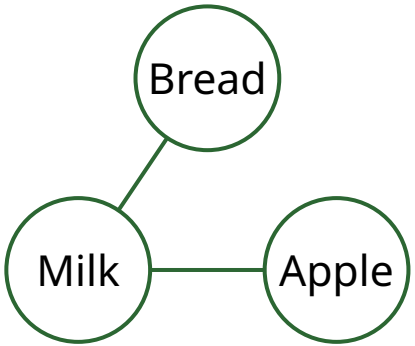
Example: Hierarchical Distribution (2/2)

1st order interaction + 9.43[Bread] + 8.52[Milk] – 9.84[Apple]
– 9.03[Bread&Milk] + 9.43[Milk&Apple]

2nd order interaction

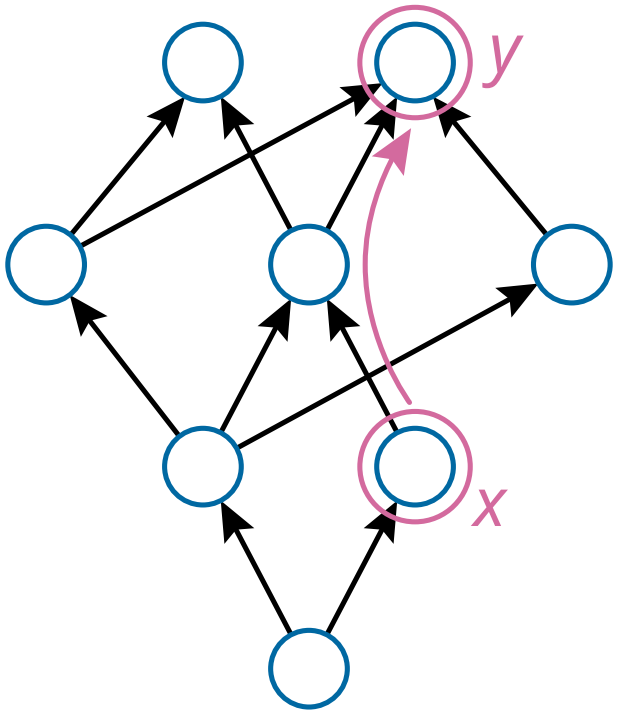
MLE

Boltzmann machine



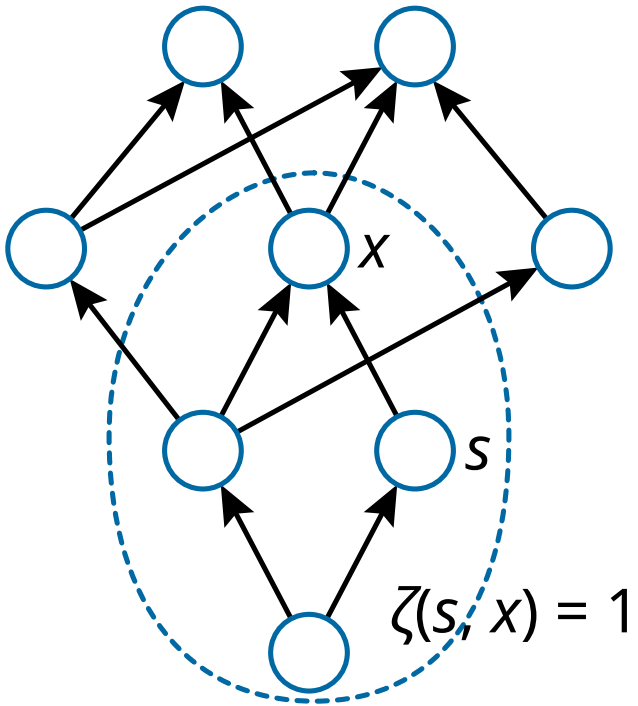
Bread	Milk	Apple	Probability from data	Learned probability
0	0	0	?	0.0000300109
1	0	0	0.375	0.3749599867
0	1	0	?	0.1499903954
0	0	1	?	0.0000000016
1	1	0	0.375	0.2250096042
1	0	1	?	0.0000200043
0	1	1	0.25	0.0999895960
1	1	1	?	0.1500004008

Generalization to Posets



- Partially ordered set (poset) $(S, \leq)$
  - (i) $x \leq x$ (reflexivity)
  - (ii) $x \leq y, y \leq x \Rightarrow x = y$ (antisymmetry)
  - (iii) $x \leq y, y \leq z \Rightarrow x \leq z$ (transitivity)
  - We assume that S is finite and $\bot \in S$
- Equivalent to a DAG
  - $x \leq y \iff y$ is reachable from x
- Variable interaction hierarchy is a poset
  - "$\leq$" is "$\subseteq$" between variable sets

Zeta and Möbius Functions

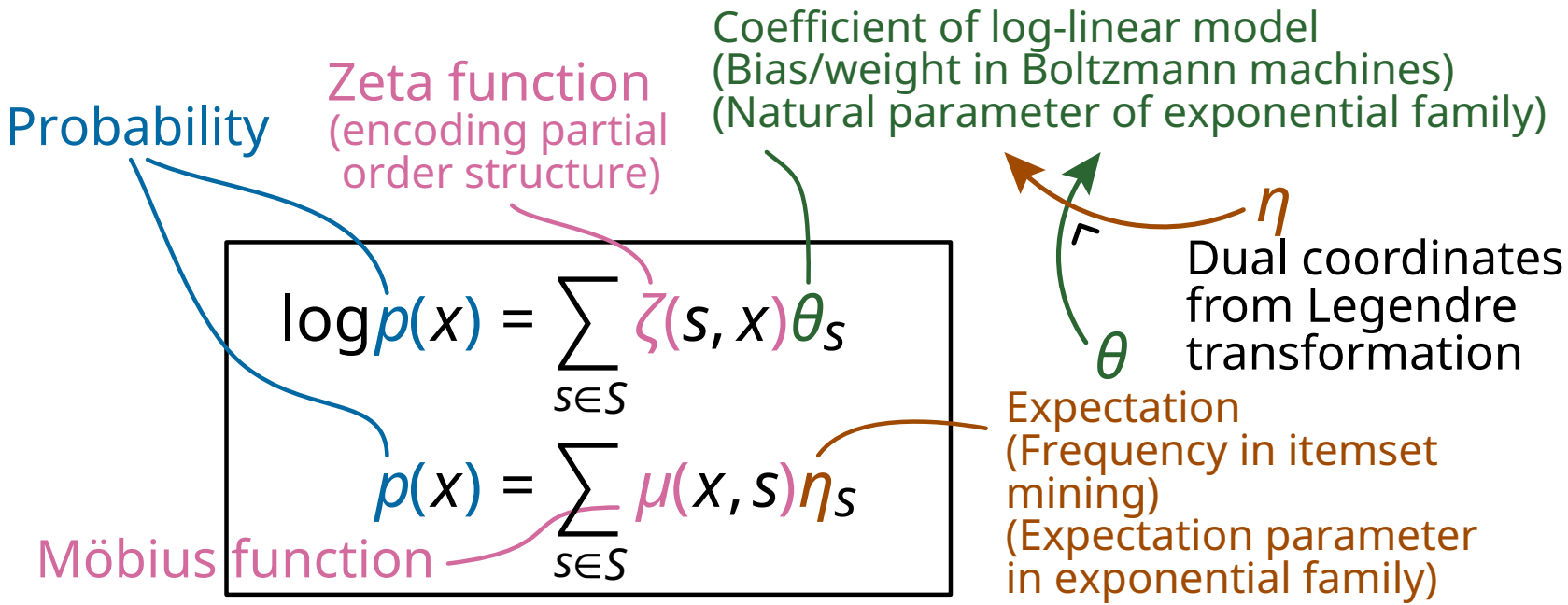


- Zeta function $\zeta : S \times S \to \{0, 1\}$

  $$\zeta(s, x) = \begin{cases} 1 & \text{if } s \leq x, \\ 0 & \text{otherwise.} \end{cases}$$

  - (integral)
- Möbius function $\mu = \zeta^{-1}$
  - $\zeta\mu = \delta$, where
    $\delta_{xy} = 1$ if $x = y$ and $\delta_{xy} = 0$ otherwise
  - (differential)
- Incidence algebra is induced

The Log-Linear Model on Posets



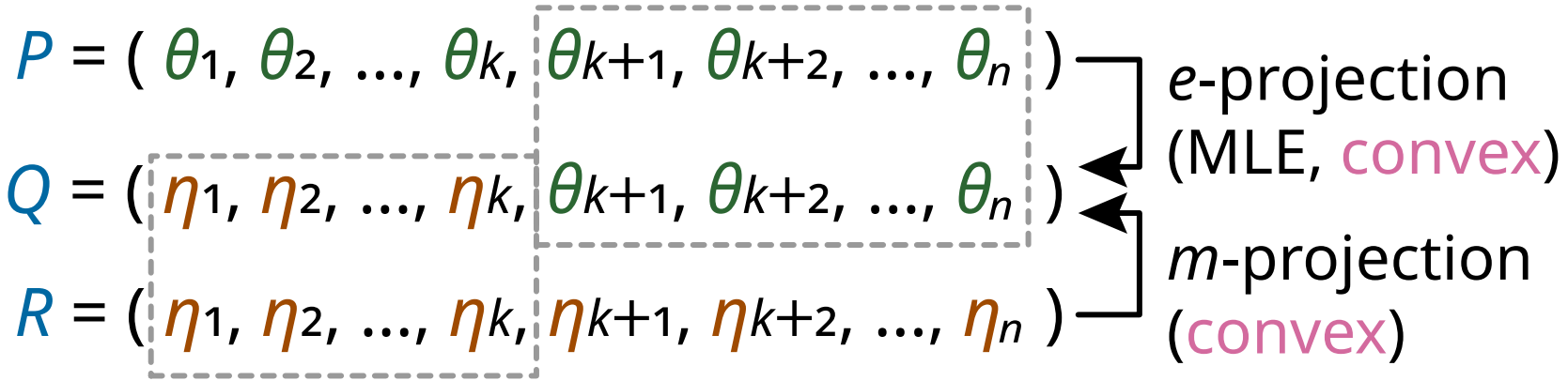
Sugiyama, Nakahara & Tsuda, **Tensor Balancing on Statistical Manifold**, ICML2017

Formulation

1. **Data:** Points (probability distributions) on a statistical manifold
2. **Model:** Parameterization via (θ, η)-coordinates
   - We use the log-linear model on posets [Sugiyama et al. ICML2017]
     - Explicitly model any-order interactions between variables
     - Generalization of **higher-order Boltzmann machines** [Sejnowski,1986] and its **information geometry** [Amari, 2001]
3. **Learning:** (Forward) Projection onto constrained space
   - We use many-body approximation [Ghalamkari et al. NeurIPS2023]
4. **Correctness:** Convex optimization

Mixed Coordinate System

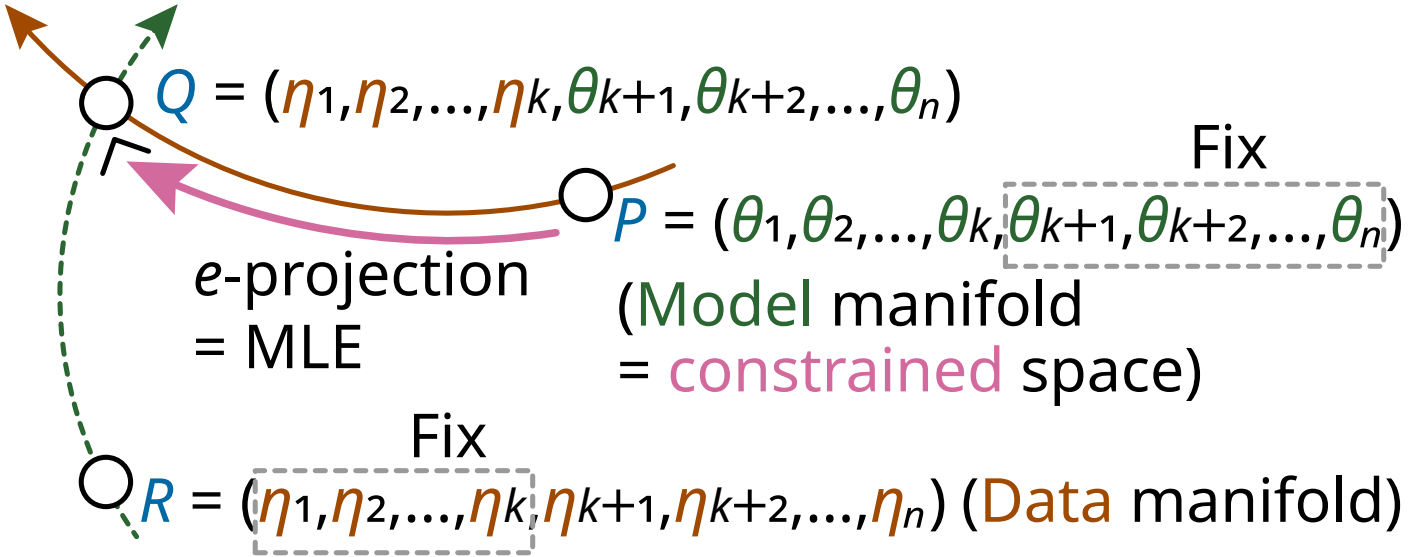
- Many problems are formulated as coordinate mixing



Pythagorean theorem: (Q always uniquely exists)

$\mathrm{KL}(P, R) = \mathrm{KL}(P, Q) + \mathrm{KL}(Q, R)$

Two Submanifolds



$\theta_{\text{next}} \leftarrow \theta - \varepsilon(\eta - \hat{\eta}_{\text{target}})$ (gradient descent) or

$\boldsymbol{\theta}_{\text{next}} \leftarrow \boldsymbol{\theta} - G^{-1}(\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}_{\text{target}})$ (natural gradient, G: FIM)

Mixed Coordinate System (Example)

- Many problems are formulated as coordinate mixing

$P = (\ 0\ ,\ 0\ ,\ \ldots,\ 0\ ,\ 0,\ \ 0\ \ ,\ \ldots,\ 0\ )$ → Uniform dist.

$Q = (\ \hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_k, 0,\ \ 0\ \ ,\ \ldots,\ 0\ )$

$R = (\ \hat{\eta}_1, \hat{\eta}_2, \ldots, \hat{\eta}_k, \hat{\eta}_{k+1}, \hat{\eta}_{k+2}, \ldots, \hat{\eta}_n\ )$ → Empirical dist.

Pythagorean theorem: (Q always uniquely exists)

$\mathrm{KL}(P, R) = \mathrm{KL}(P, Q) + \mathrm{KL}(Q, R)$

Mixed Coordinate System (Example)

- Many problems are formulated as coordinate mixing

$$P_1 = (\ \theta_1^{(1)},\ \theta_2^{(1)},\ \ldots,\ \theta_k^{(1)},\ \boxed{0,\ 0,\ \ldots,\ 0}\ )$$ Initial state

$$\vdots \qquad\qquad \vdots$$

$$P_t = (\ \theta_1^{(t)},\ \theta_2^{(t)},\ \ldots,\ \theta_k^{(t)},\ 0,\ 0,\ \ldots,\ 0\ )$$ Converged

$$= (\ \eta_1^{(t)},\ \eta_2^{(t)},\ \ldots,\ \eta_k^{(t)},\ 0,\ 0,\ \ldots,\ 0\ )$$

$$\| \qquad \| \qquad\qquad \|$$

$$\hat{\eta}_1 \quad \hat{\eta}_2 \qquad \hat{\eta}_k$$

Fixed

Many-Body Approximation (MBA)

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$$= H_0 + H_i^{(1)} + \cdots + H_i^{(4)} + H_{ij}^{(12)} + \cdots + H_{kl}^{(34)} + H_{ijk}^{(123)} + \cdots + H_{jkl}^{(234)} + H_{ijkl}^{(1234)}$$

One-Body Approximation

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$= H_0 + H_i^{(1)} + \cdots + H_i^{(4)} + 0 + \cdots + 0 + 0 + \cdots + 0 + 0$

$\sum \theta_{111l'}$

One body

- **Fix the unused parameters at zero**

Two-Body Approximation

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$= H_0 + H_i^{(1)} + \cdots + H_i^{(4)} + H_{ij}^{(12)} + \cdots + H_{kl}^{(34)} + 0 + \cdots + 0 + 0$

$\sum\sum \theta_{11k'l'}$

Two body

- **Fix the unused parameters at zero**

Three-Body Approximation

- We explicitly model associations between features/modes using the log-linear model on posets

$\log \mathcal{P}_{ijkl}$ ($\mathcal{P}$: 4th-order tensor)

$$= H_0 + H_i^{(1)} + \cdots + H_i^{(4)} + H_{ij}^{(12)} + \cdots + H_{kl}^{(34)} + H_{ijk}^{(123)} + \cdots + H_{jkl}^{(234)} + \quad 0$$

$$\sum\sum\sum \theta_{1j'k'l'}$$

Three body

- **Fix the unused parameters at zero**

Graphical Understanding of MBA

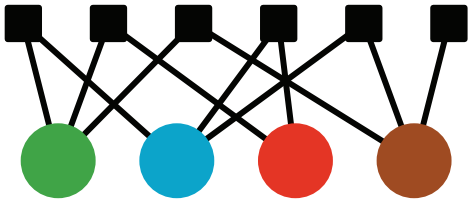
One-body approximation

$\mathcal{P}_{ijkl} = p_i^{(1)} p_j^{(2)} p_k^{(3)} p_l^{(4)}$

= Rank-1 approximation
(mean-field approximation)

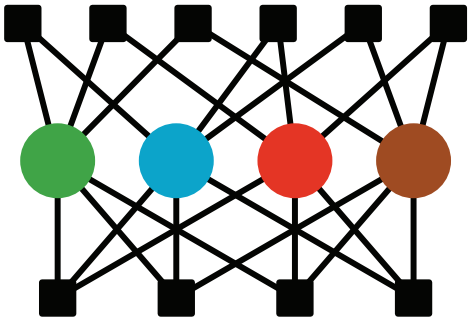
Two-body approximation

$\mathcal{P}_{ijkl} = \chi_{ij}^{(12)} \chi_{ik}^{(13)} \cdots \chi_{kl}^{(34)}$



Three-body approximation

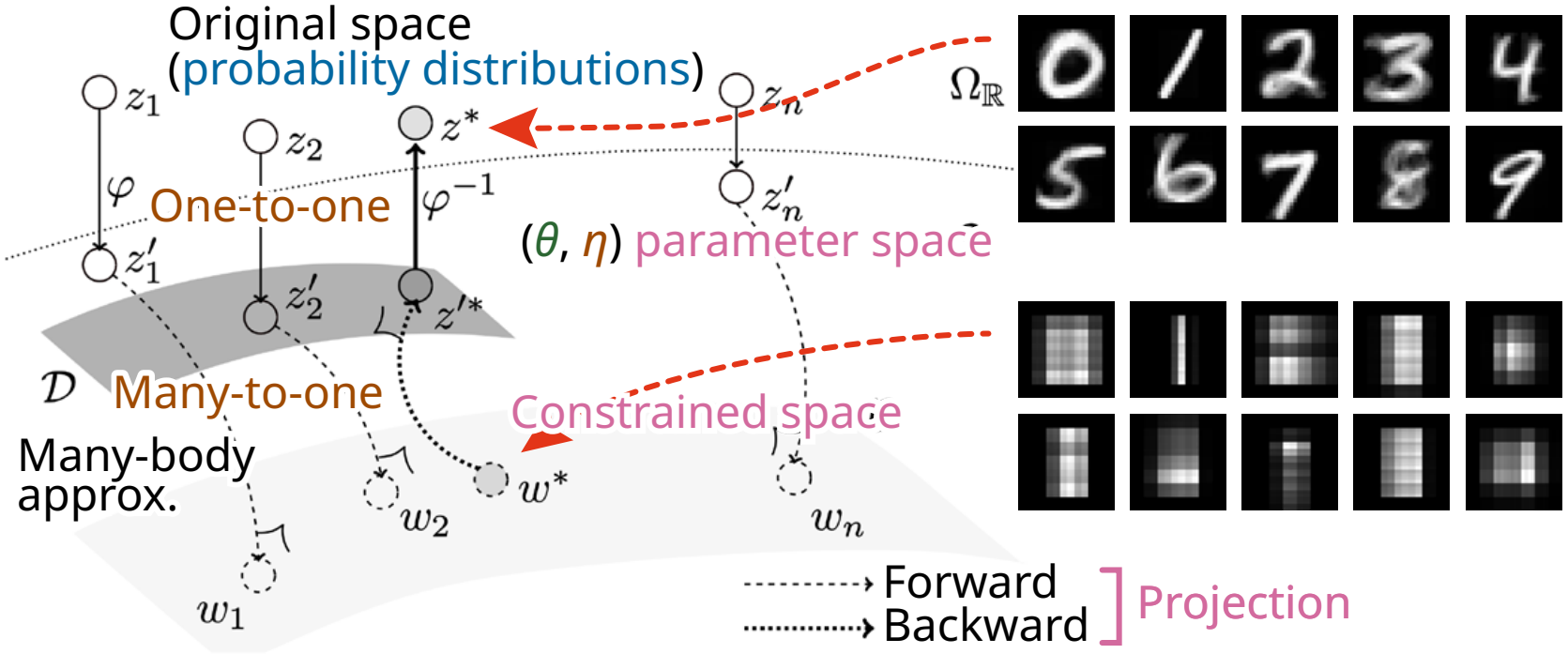
$\mathcal{P}_{ijkl} = \chi_{ijk}^{(123)} \chi_{ijl}^{(124)} \chi_{ikl}^{(134)} \chi_{jkl}^{(234)}$



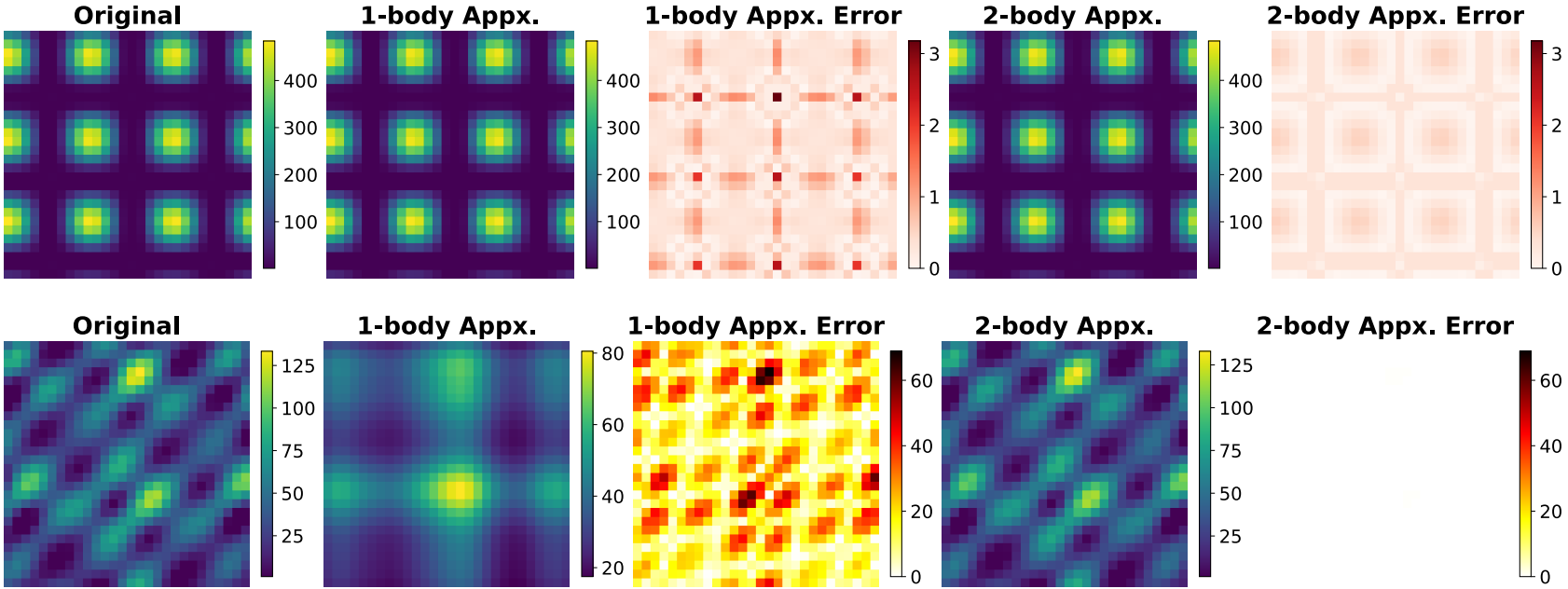
Larger capability

Ghalamkari, Sugiyama, & Kawahara, **Many-body Approximation for Non-negative Tensors**, NeurIPS2023 [Code (Julia)] [Code (Python)]

Data Augmentation (Revisited)



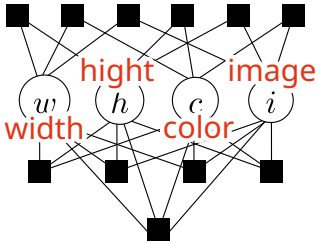
MBA on Synthetic Data (Images)



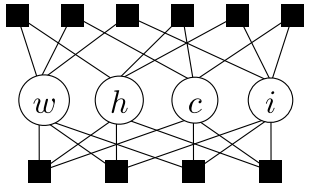
MBA on Real Data (Images)

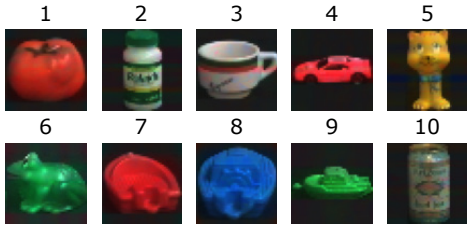
More flexible modeling by many-body approximation

a. Up to four-body

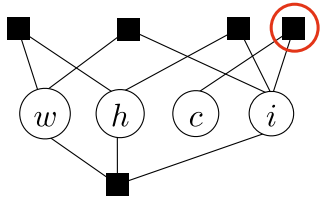


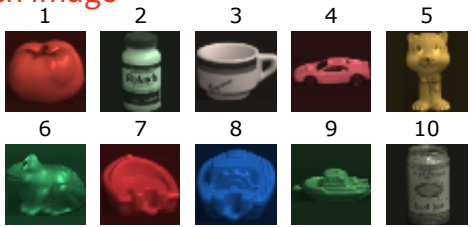
b. Up to three-body



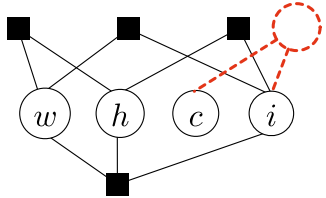


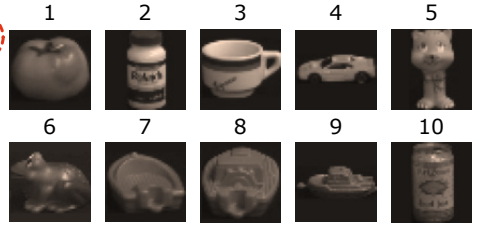
c. Color changes for each image



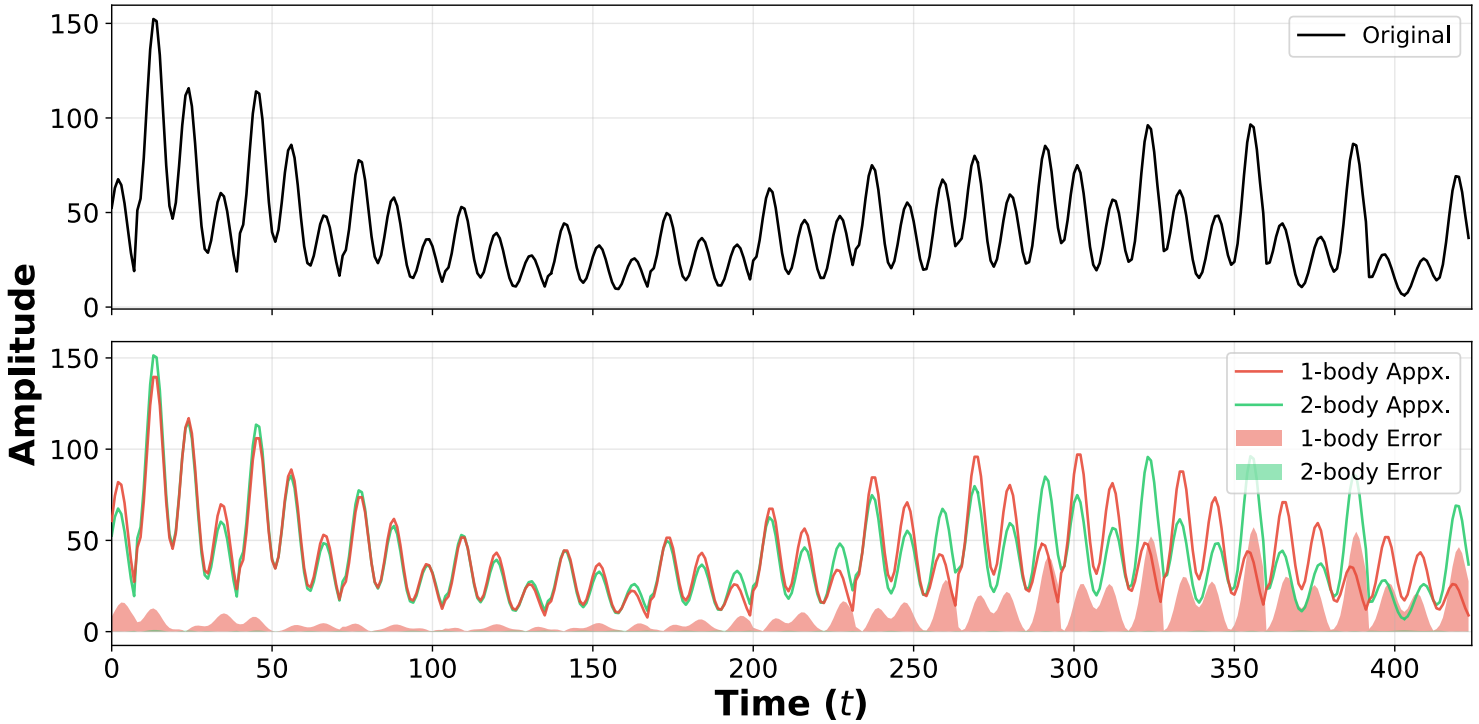


d. When the interaction between colors and image indices is removed...

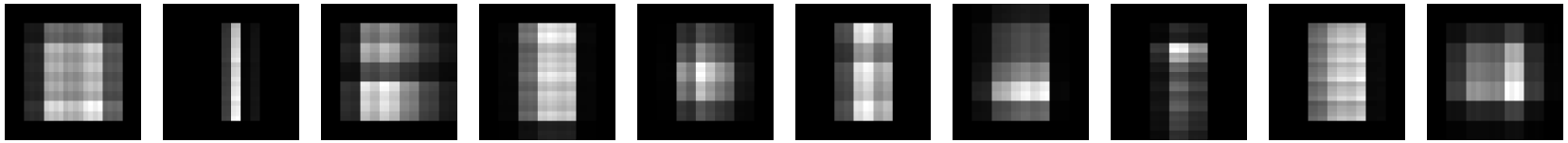




MBA on Synthetic Data (Times Series)

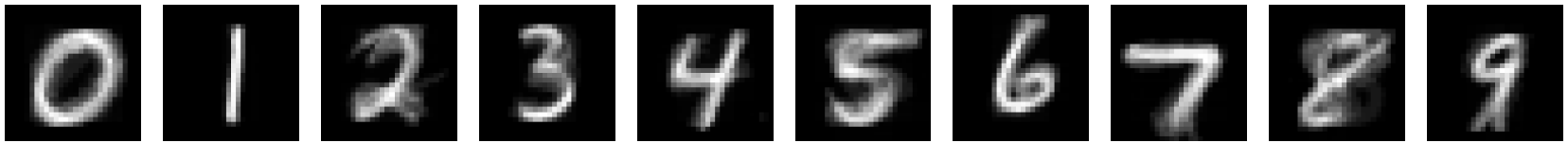


Data Augmentation on MNIST



Forward projection (MBA) using 17 θ-parameters

$$(\theta_0, \underbrace{\theta_1, \dots, \theta_{17}}_{\text{17 parameters are available}}, \underbrace{0, 0, 0, \dots, 0, 0}_{\text{767 parameters are fixed}})$$



Backward projection using 767 θ-parameters

$$(\theta_0, \underbrace{\bar{\theta}_1, \dots, \bar{\theta}_{17}}_{\text{17 parameters are fixed to mean of neighbor } \theta}, \underbrace{\theta_{18}, \theta_{19}, \theta_{20}, \dots, \theta_{783}, \theta_{784}}_{\text{767 parameters are available}})$$

Data Augmentation on CIFAR-10

Forward projection (MBA) using 1,410 θ-parameters

$$(\theta_0, \underbrace{\theta_1, \dots, \theta_{1410}}_{\text{1410 parameters are available}}, \underbrace{0, \dots, 0}_{\text{1662 parameters are fixed}})$$

Backward projection using 2,334 θ-parameters

$$(\theta_0, \underbrace{\bar{\theta}_1, \dots, \bar{\theta}_{738}}_{\text{738 parameters are fixed to mean of neighbor } \theta}, \underbrace{\theta_{739}, \dots, \theta_{3072}}_{\text{2334 parameters are available}})$$

Test Accuracy on Real Data (1/2)

Training Set	Dataset		
	MNIST	CIFAR-10	Speech Commands
OG	$97.98 \pm 0.19\%$	$88.57 \pm 0.57\%$	$84.48 \pm 0.50\%$
OG$^{\text{STD}}$	$97.98 \pm 0.24\%$	$89.89 \pm 0.44\%$	$82.98 \pm 0.50\%$
OG$^{\text{AE}}$	$97.97 \pm 0.25\%$	$88.36 \pm 0.46\%$	$83.13 \pm 0.32\%$
OG$^{\text{MU}}$	$96.45 \pm 0.23\%$	$86.60 \pm 0.49\%$	$81.85 \pm 0.61\%$
OG$^{\text{MMU}}$	$97.52 \pm 0.30\%$	$88.02 \pm 0.39\%$	$83.06 \pm 0.54\%$
OG$^{\text{PROPOSAL}}$	$97.91 \pm 0.21\%$	$88.07 \pm 0.46\%$	$84.35 \pm 0.37\%$

STD: Standard techniques, **AE**: AutoEncoder, **MU**: MixUp,
MMU: Manifold MixUp, **PROPOSAL**: Our proposal

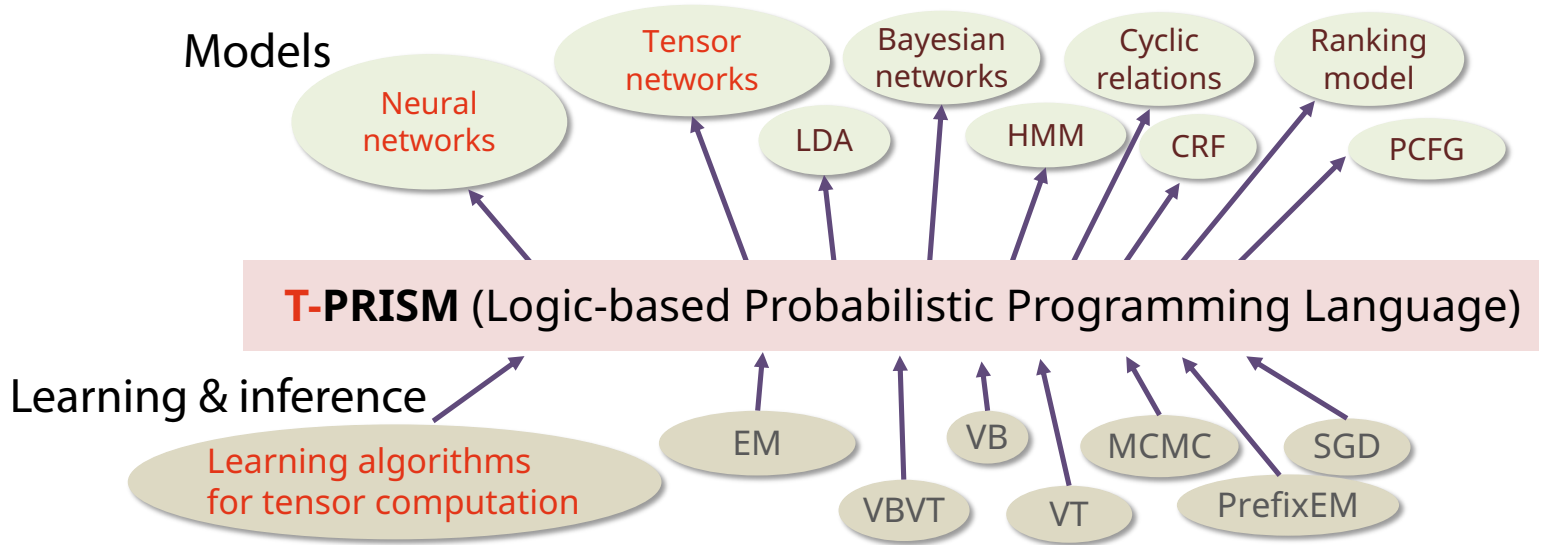
Test Accuracy on Real Data (2/2)

Training Set	Dataset		
	Connect. Bench	Taiwanese Bank.	Wine Quality
OG	$88.10 \pm 8.58\%$	$96.54 \pm 0.56\%$	$55.00 \pm 1.69\%$
OG$^{\text{STD}}$	$85.24 \pm 7.66\%$	$96.17 \pm 0.57\%$	$57.85 \pm 1.81\%$
OG$^{\text{AE}}$	$82.86 \pm 7.59\%$	$95.92 \pm 0.62\%$	$57.23 \pm 1.67\%$
OG$^{\text{MU}}$	$89.29 \pm 4.97\%$	$96.55 \pm 0.68\%$	$57.76 \pm 1.67\%$
OG$^{\text{MMU}}$	$91.19 \pm 5.06\%$	$96.44 \pm 0.53\%$	$58.70 \pm 1.74\%$
OG$^{\text{PROPOSAL}}$	$93.81 \pm 4.54\%$	$96.53 \pm 0.47\%$	$59.03 \pm 1.74\%$

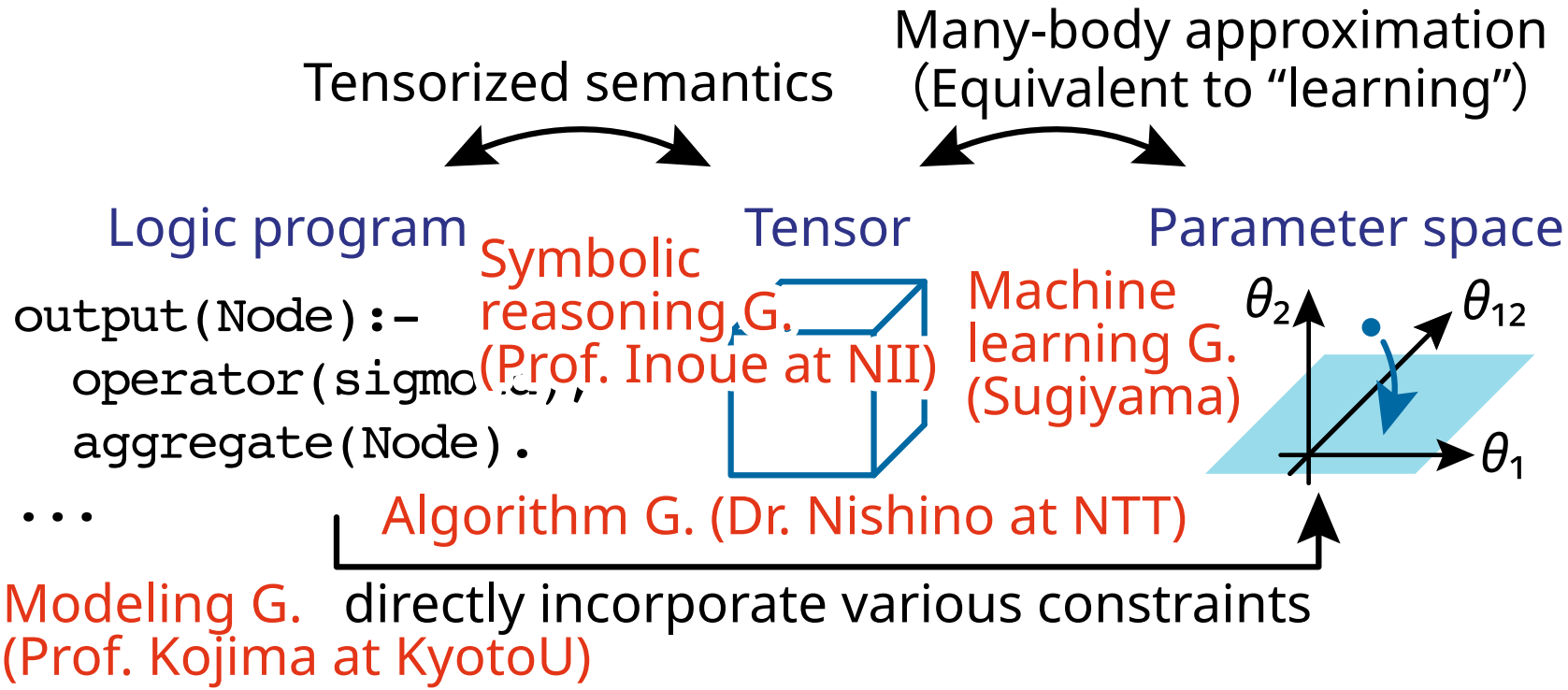
STD: Standard techniques, **AE**: AutoEncoder, **MU**: MixUp, **MMU**: Manifold MixUp, **PROPOSAL**: Our proposal

Implementation on T-PRISM

- Describe tensor contraction process by logical formulae
  - https://github.com/kojima-r/pyLegendreDecomposition



Overview of Our CREST Project



Summary

- Information-geometric modeling of data and learning via log-linear model on posets
  - Modeling by (θ, η)-coordinates (energy-based model)
  - Learning by many-body approximation
    $\in$ Projection onto constrained space by convex optimization
- If you are interested in this topic more...
  - Tensor balancing (with IG formulation) [Sugiyama et al. ICML2017]
  - Model selection and extension [Zhou & Sugiyama, UAI2025, Info.Geo.]
  - Deformed decomposition [Ghalamkari et al. AISTATS2026]
- **Slide:** `https://mahito.nii.ac.jp/pdf/BAQ2026.pdf`