Deep Energy-Based Generative Modeling and Learning

Yifei Xu

Ph. D. Defense

Advisor: Ying Nian Wu
Department of Statistics
University of California, Los Angeles

March 3rd , 2022

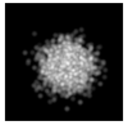
Self-introduction

徐 亦飞 Yifei Xu

- 2003 – 2013 : Shanghai Experimental School (K-12)
  - Primary, Mid, High School (K-12)
- 2013 – 2017 : Shanghai Jiao Tong University
  - Bachelor of Engineering --- **Computer Science**
  - Zhiyuan College ACM Honored Class
  - Advisor: Liqing Zhang
- 2017 – Now : University of California, Los Angeles
  - Doctor of Philosophy --- **Statistics**
  - Advisor: Ying Nian Wu

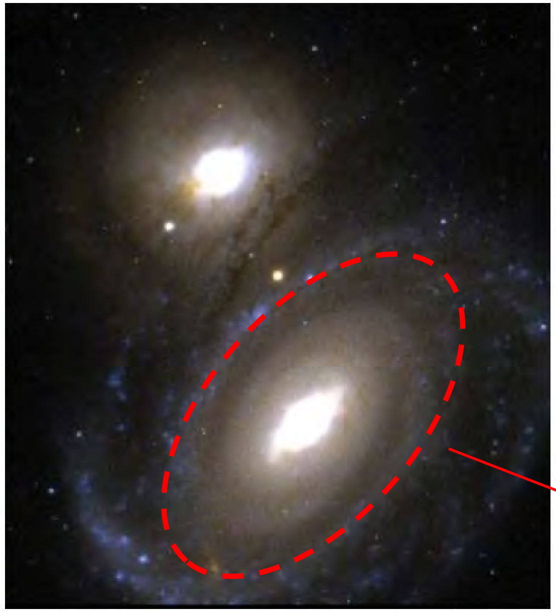
Knowledge Representation: Concepts and Models

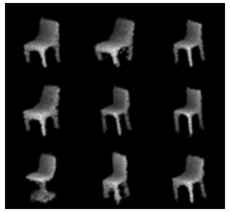
Concept ⟷ Set ⟷ Model



Random object

An object



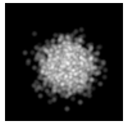


A concept, e.g., chair
(a set of 3Dchair object)

Energy-based Model

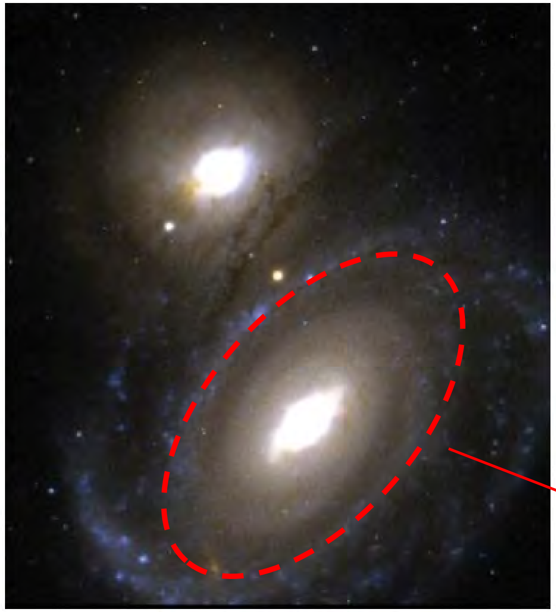
Concept ⟷ Set ⟷ Model

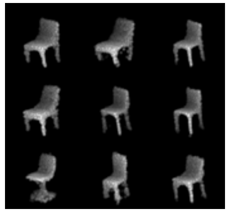
$$p_\theta(X) = \frac{1}{Z(\theta)} \exp f_\theta(X)\, p_0(X)$$



Random object

An object





A concept, e.g., chair
(a set of 3Dchair object)

Energy-based Model

Directly model the probability:

$$\log p_\theta(X) \propto f_\theta(X)$$

$$f_\theta(X): \boldsymbol{R}^D \to \boldsymbol{R}$$

Any differentiable function
e.g. weight sum of a heuristic rule,
Gabor filter on image, or neural network.

$$p_\theta(X) = \frac{1}{Z(\theta)} \exp f_\theta(X)\, p_0(X)$$

$$Z(\theta) = \int \exp f_\theta(X)\, dX$$

The normalization constant to ensure overall probability sum up to 1.

$$p_0(X) \sim N(0, I_D)$$

The white noise prior distribution.

Discriminative, generative and descriptive

Discriminative Task

$$p_\theta(C|X)$$

Generative Task

$$p_\theta(\mathrm{X}|\mathrm{z})$$

Discriminative, generative and descriptive

Descriptive Model

$$p_\theta(X)$$

Discriminative Model

$$p_\theta(C|X)$$

Generative Model

$$p_\theta(\mathrm{X}|\mathrm{z})$$

Discriminative, generative and descriptive

Descriptive Model

$$p_\theta(X)$$

Discrimination

$$p_\theta(k|X) = \frac{\exp f_{\theta_k}(X)}{\sum_{l=1}^{K} \exp f_{\theta_l}(X)}$$

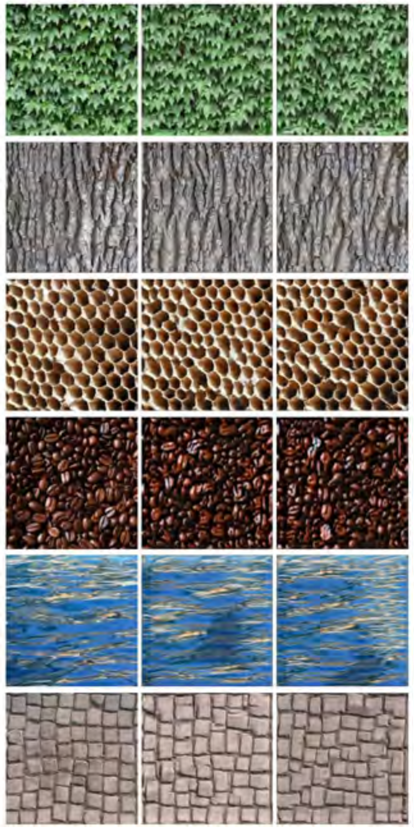
Generation

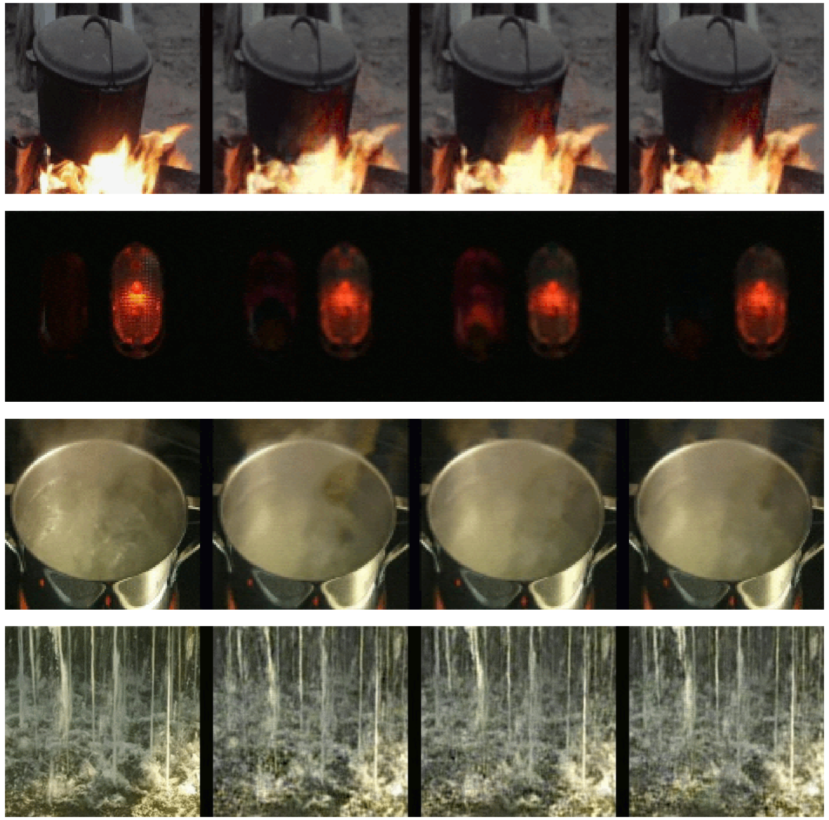
$$X \sim p_\theta(X)$$

Advantage of EBM

Simplicity	Directly model the probability.
Stability	No need assisting network to ensure balance.
Flexibility	Any bottom-up function can act as energy.
Adaptivity	Avoid mode collapse and avoiding spurious modes from out-of-distribution samples.
Compositionality	Models to be combined through product of experts or other hierarchical techniques.

Everything to generate





Learning

Representing

Controlling

Learning

How to model a set?
Generation? Reconstruction?
Semi-supervised representation learning?

Representing

How to represent a 3D shape?
Voxel? Point Cloud? Mesh?
A function itself can be a form of data representation?
How EBM works with VAE?

Controlling

What is inverse optimal control?
How to control a vehicle driving on the road?
How to control if we do not even know what good is?
How to do control efficiently and accurately?

Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

Representing

Energy-based Implicit Function for 3D Shape Representation

Controlling

Energy-based Continuous Inverse Optimal Control

0. Fundamental

- **Train an EBM using MLE**
- **Sample-based Approximation**

Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

Representing

Energy-based Implicit Function for 3D shape representation

Controlling

Energy-based Continuous Inverse Optimal Control

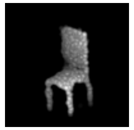
Energy-based Model --- Training

- **Maximum Likelihood Estimation:**

$$l(\theta) = E_{q_{data}}[\log p_\theta(X)] \approx \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(X_i)$$

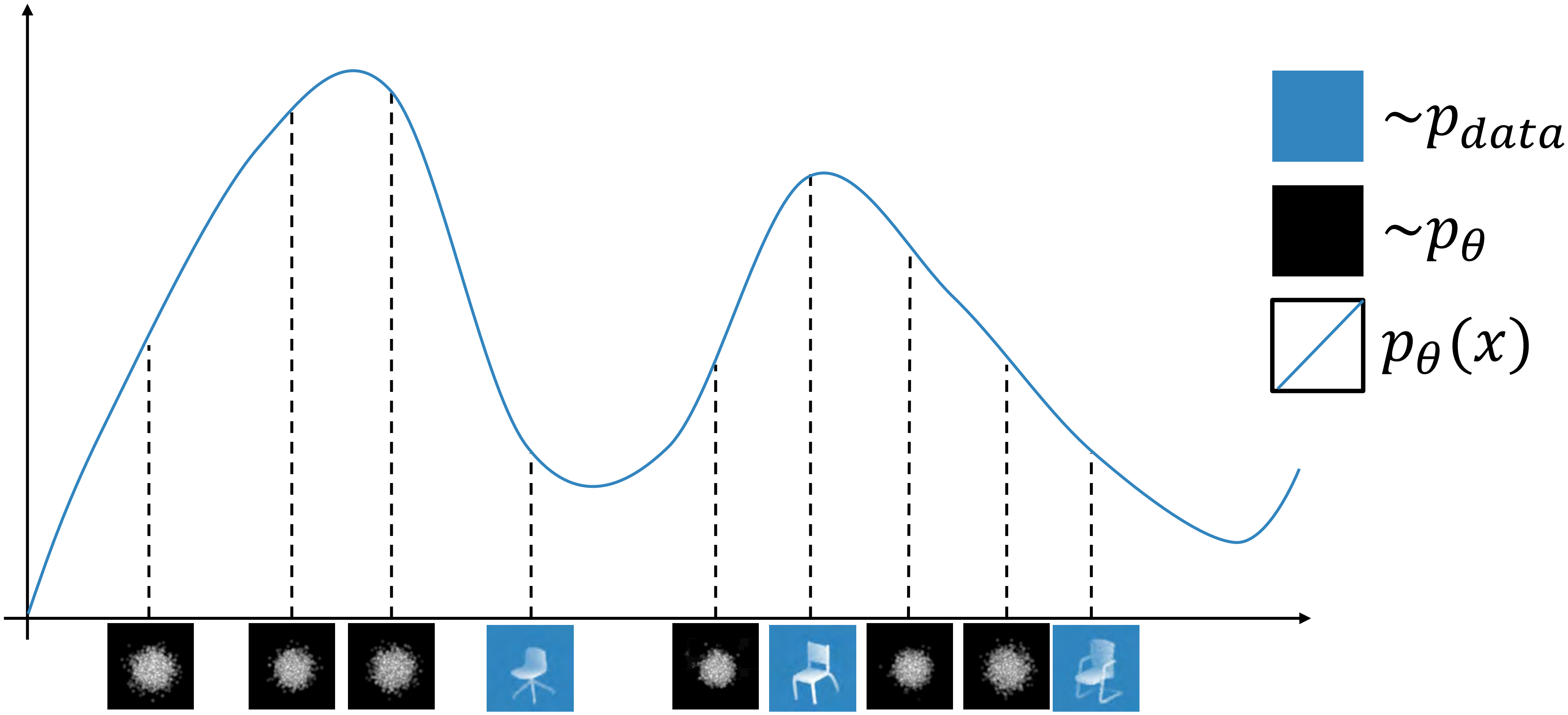
- **Train model by gradient descent:**

$$\frac{\partial}{\partial\theta} l(\theta) = E_{q_{data}}\left[\frac{\partial}{\partial\theta} f_\theta(X)\right] - E_{p_\theta}\left[\frac{\partial}{\partial\theta} f_\theta(X)\right] \approx \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial\theta} f_\theta(X_i) - \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial\theta} f_\theta(\tilde{X}_i)$$

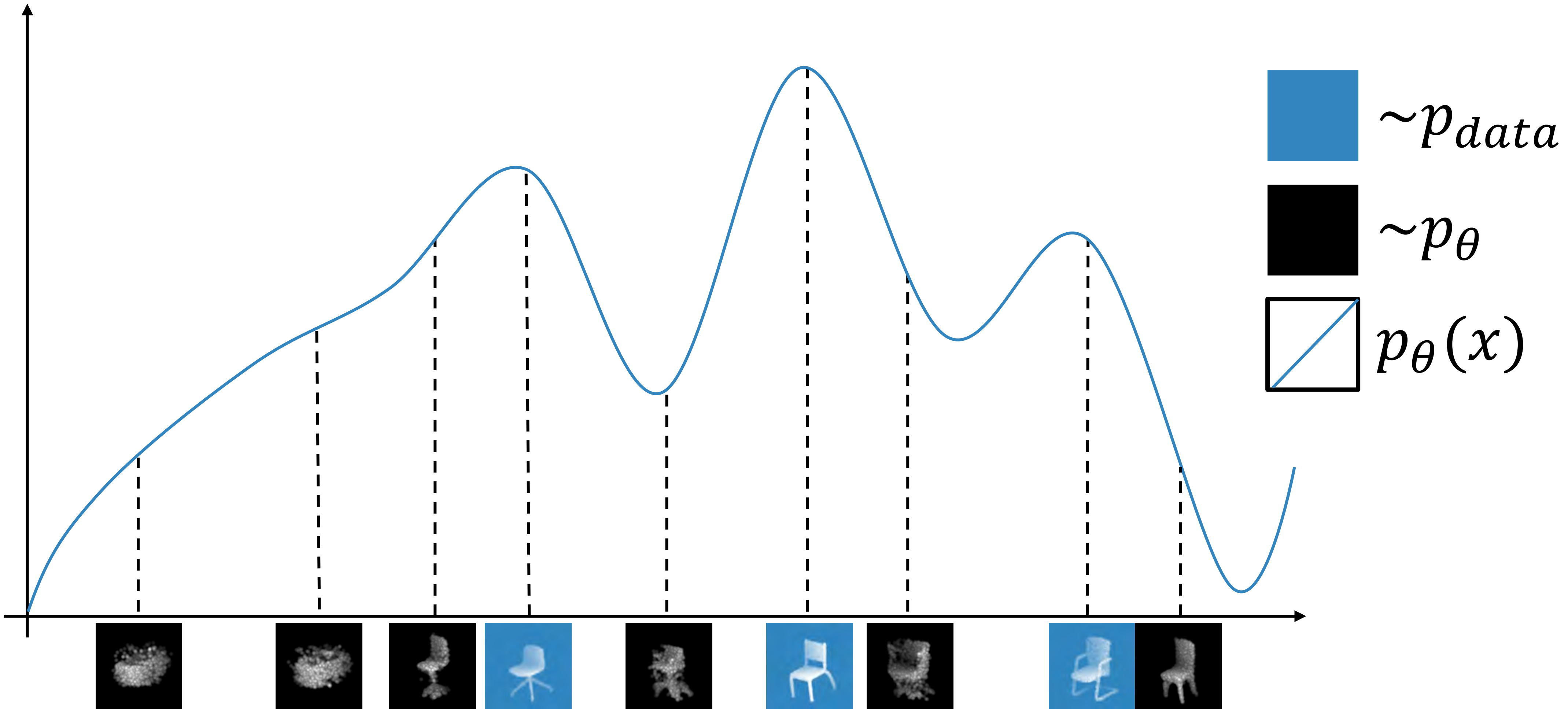
 $\sim p_{data}$  $\sim p_\theta$

Use MCMC sampling

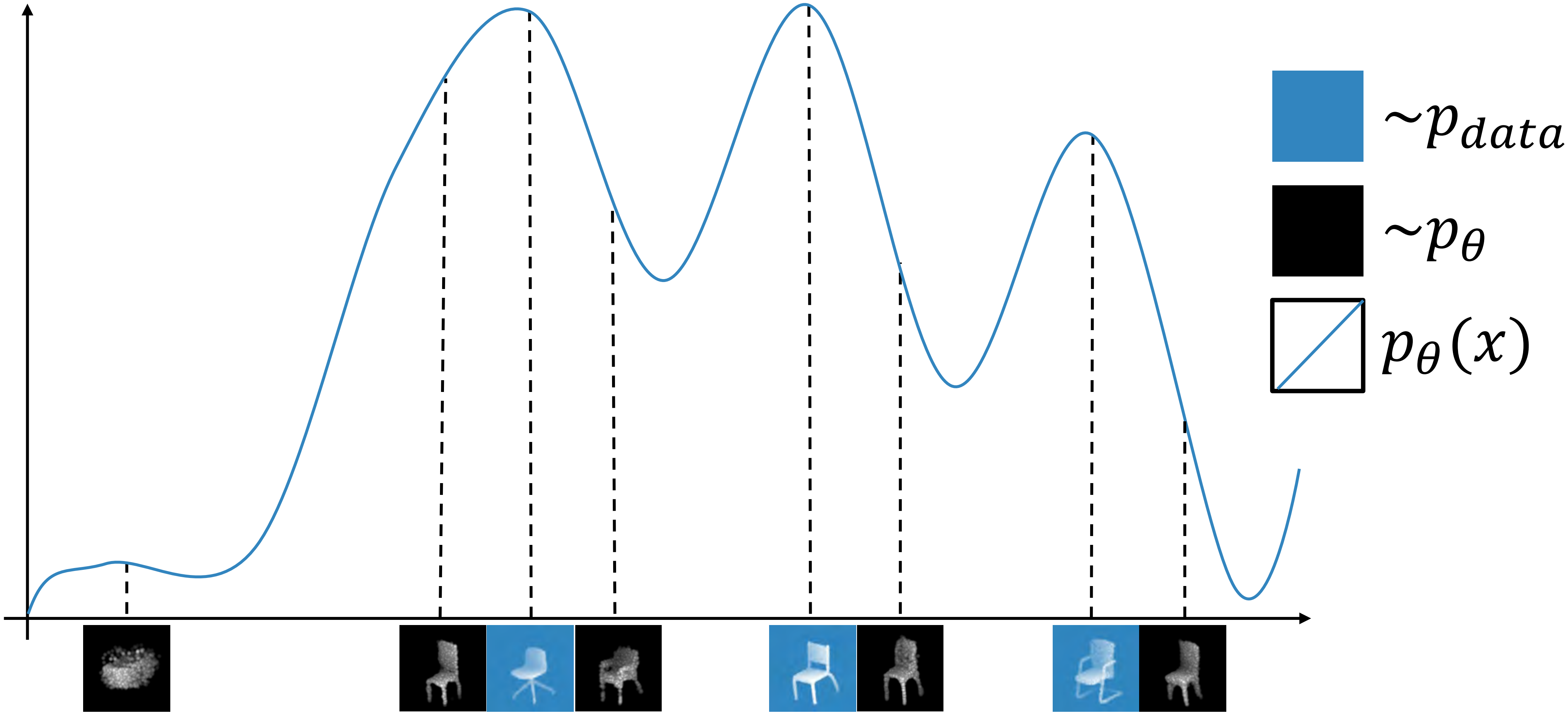
Energy-Based Model --- Training



Energy-Based Model --- Training



Energy-Based Model --- Training



1. Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

2. Representing

Energy-based Implicit Function for 3D shape representation

3. Controlling

Energy-based Continuous Inverse Optimal Control

Current challenges?

Why EBM helps?

How to model and sample?

One more thing…

1. Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

$$p_\theta = \frac{1}{Z_\theta} \exp f_\theta$$

Energy-Based Model

on

2. Representing

Energy-based Implicit Function for 3D shape representation

3. Controlling

Energy-based Continuous Inverse Optimal Control

Point Clouds

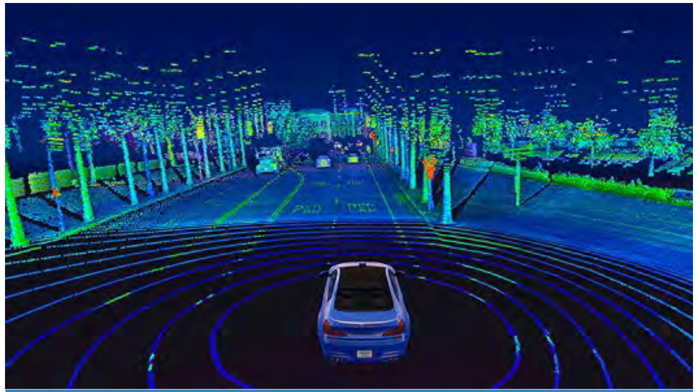
Point Cloud

Why Special?
Unordered set

$$\{x, y, z\} = \{y, z, x\}$$

Scene Reconstruction

Face ID



Lidar in AV

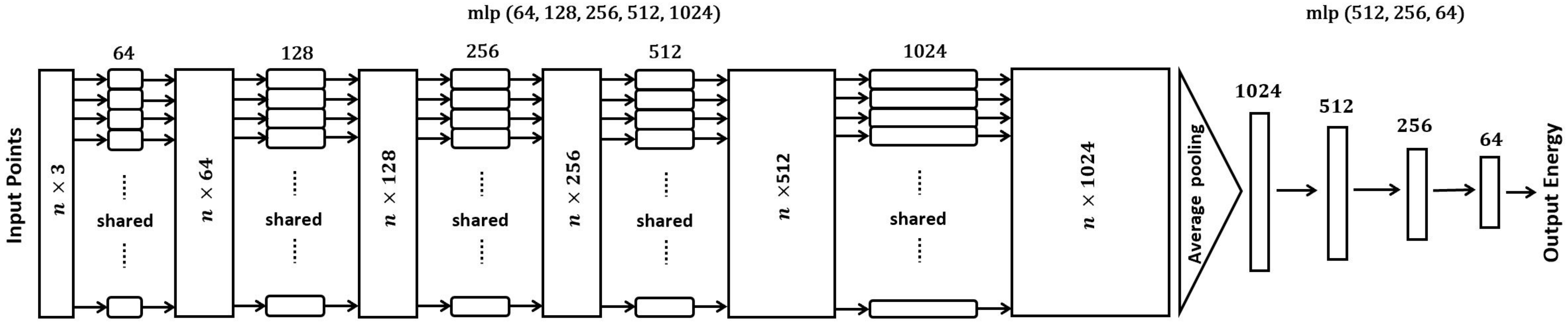
Input-permutation-invariant Score Function

- **Energy-Based Model on point cloud:**

$$p_\theta(X) = \frac{1}{Z(\theta)} exp\, f_\theta(X)\, p_0(X)$$

$Z(\theta)$: Normalizing Constant; $p_0(X)$: prior distribution

$f_\theta(X)$ is parameterized by a bottom-up input-permutation-invariant neural network.



$$f_\theta(\{x_1, \ldots, x_M\}) = g(\{h(x_1), \ldots, h(x_m)\})$$

Energy-based Model --- Sampling

- **Langevin Dynamics MCMC sampling:**

$$X_0 = N(0, \sigma^2)$$

$$X_{\tau+1} = X_\tau + \underbrace{\frac{\delta^2}{2} \frac{\partial}{\partial X} f_\theta(X_\tau)}_{\textit{Transformation}} + \underbrace{\delta U_\tau}_{\textit{Noise}}$$

where $U_\tau \sim N(0,1)$;

- **(K-step) Short-run MCMC generator:**

Short-run MCMC procedure $\xrightarrow{\text{regard as}}$ K-layer generator model

$$\tilde{X} = M_\theta(Z, \xi)\,, \qquad Z \sim p_0(Z)$$

Generation Results

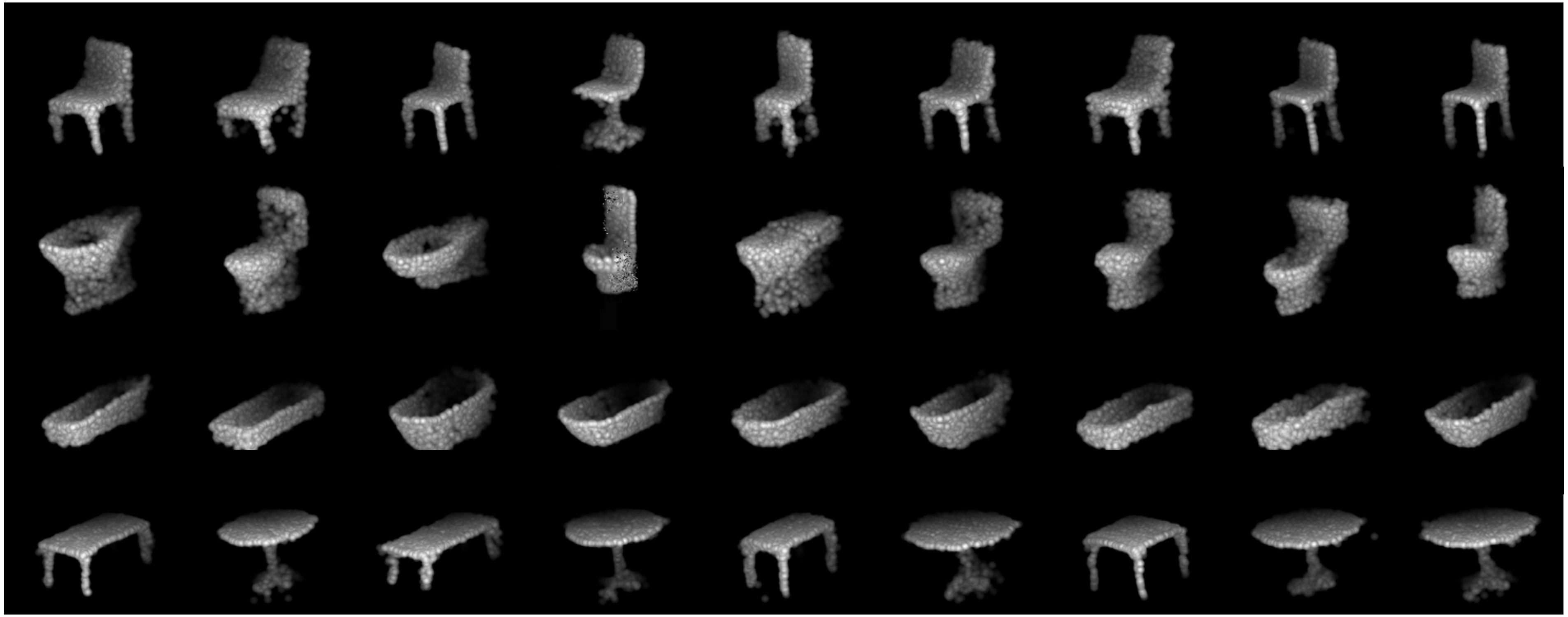
- We synthesize 3D point clouds by short-run MCMC sampling from the learned model.

Chair

Toilet

Bathtub

Table



Lowest quantitively score in **8 / 10** category

Reconstruction Results

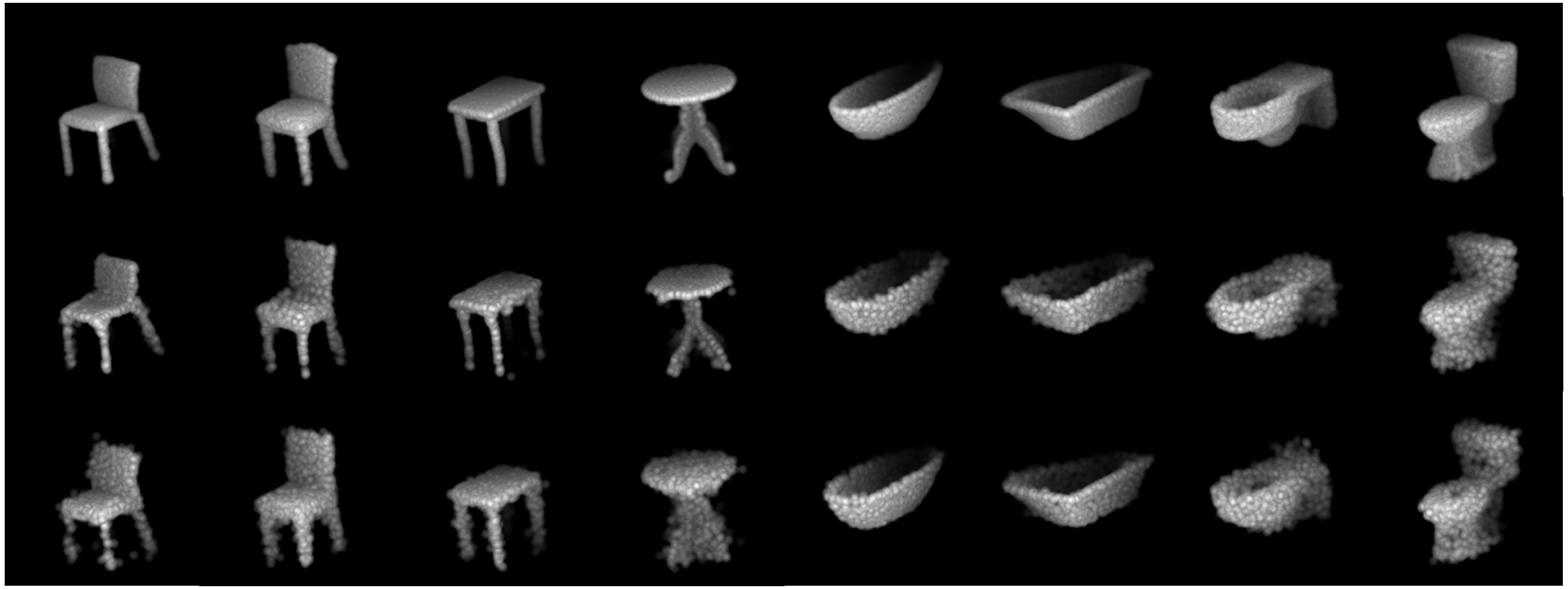
- Short-run MCMC procedure $\xrightarrow{\text{regard as}}$ K-layer generator model $M_\theta(Z, \xi)$

$$Z = \arg\min_{z} L(Z) = \|X - M_\theta(Z)\|^2$$

Ground Truth

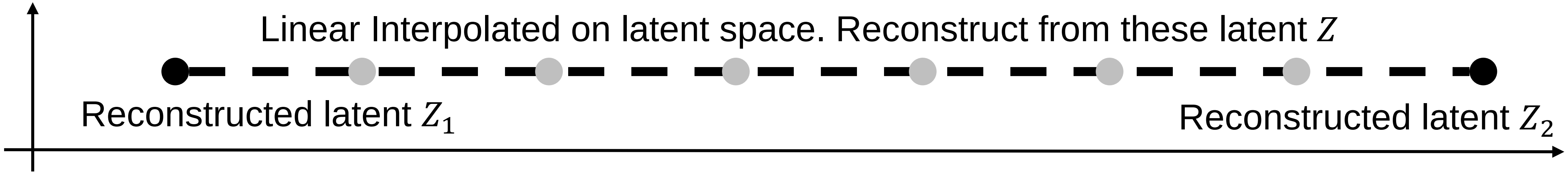
GPointNet (Ours)

PointFlow



Lowest reconstruction loss in **ALL** category

Interpolation Results

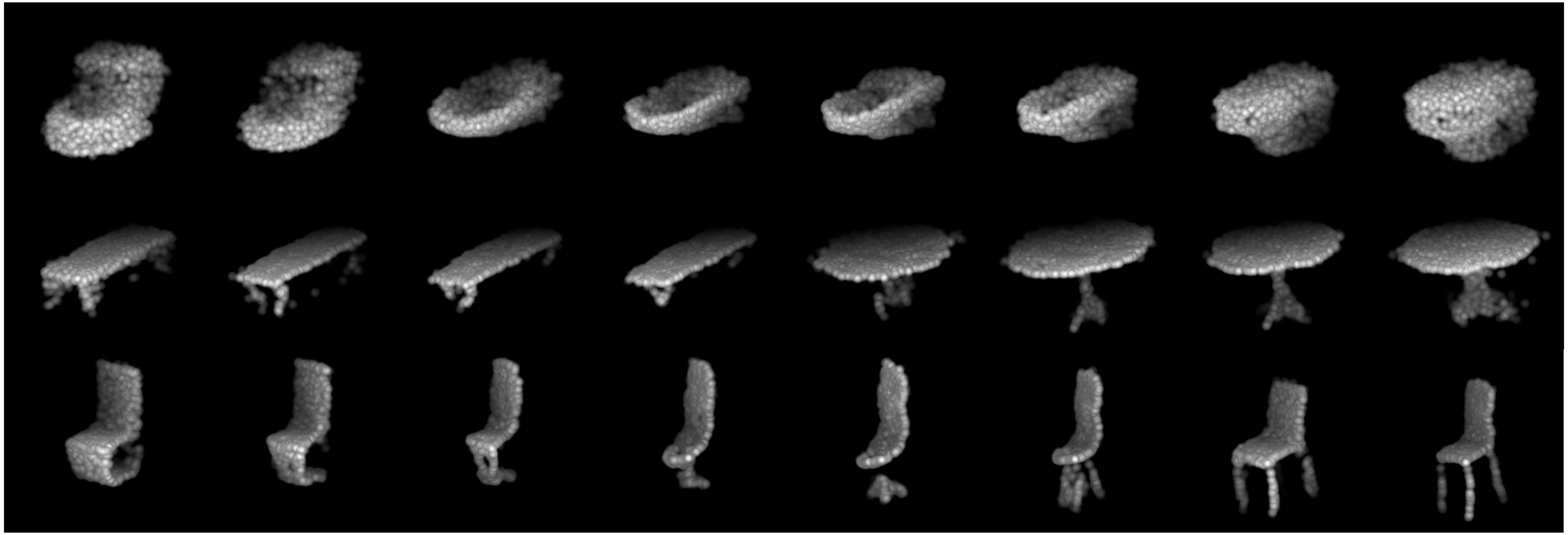


$$Z_\rho = (1-\rho)Z_1 + \rho Z_2$$

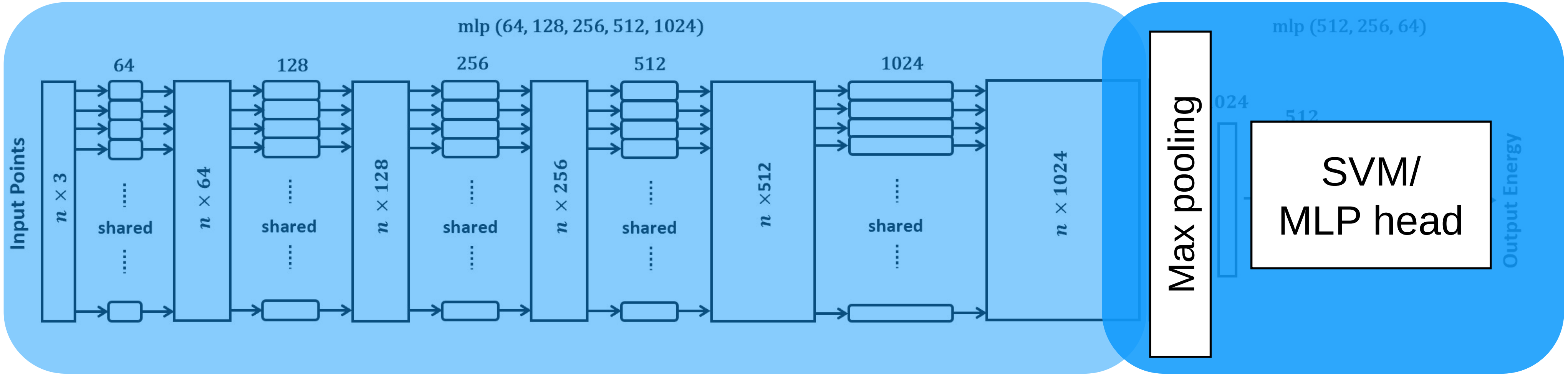
Toilet

Table

Chair



Representation Learning



Unsupervised Learning
EBM Generative Feature Learning

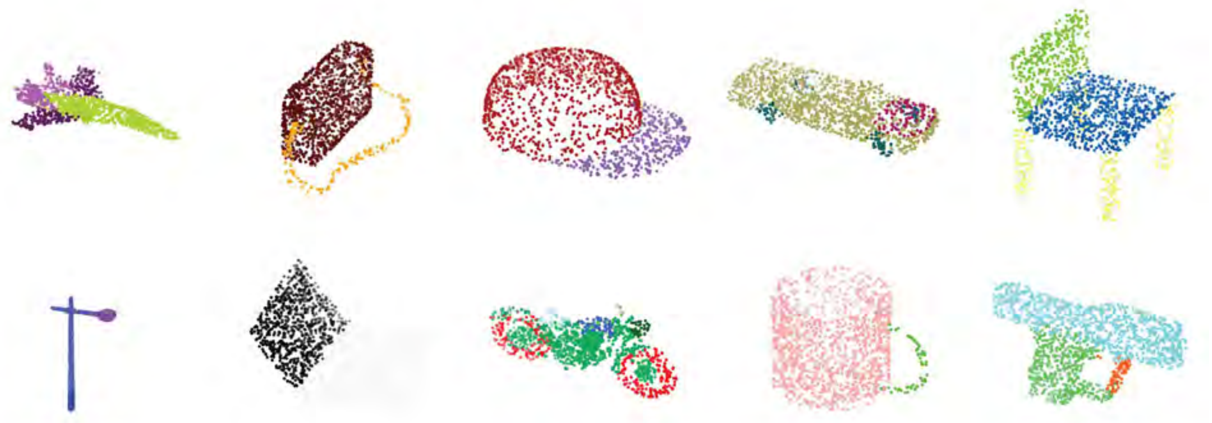
Supervised Learning
Downstream Task Learning

Representation Learning

Supervised Learning
Downstream Task Learning

Method	Accuracy
SPH [18]	79.8%
LFD [4]	79.9%
PANORAMA-NN [33]	91.1%
VConv-DAE [34]	80.5%
3D-GAN [38]	91.0%
3D-WINN [16]	91.9%
3D-DescriptorNet [44]	92.4%
Primitive GAN [19]	92.2%
FoldingNet [51]	94.4%
l-GAN [1]	95.4%
PointFlow [50]	93.7%
Ours	93.7%

Classification



Segmentation

1. Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

2. Representing

Energy-based Implicit Function for 3D shape representation

3. Controlling

Energy-based Continuous Inverse Optimal Control

Current challenge?

Unordered point set is non-trivial to deal with;
No good generative model for point cloud.

Why EBM helps?

No assisting network needed;
Derived from PointNet

How to model and sample?

Short-run MCMC by Langevin Dynamic
Regarded as k-layer generator

One more thing…

Representation learning
on Classification and Segmentation

1. Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

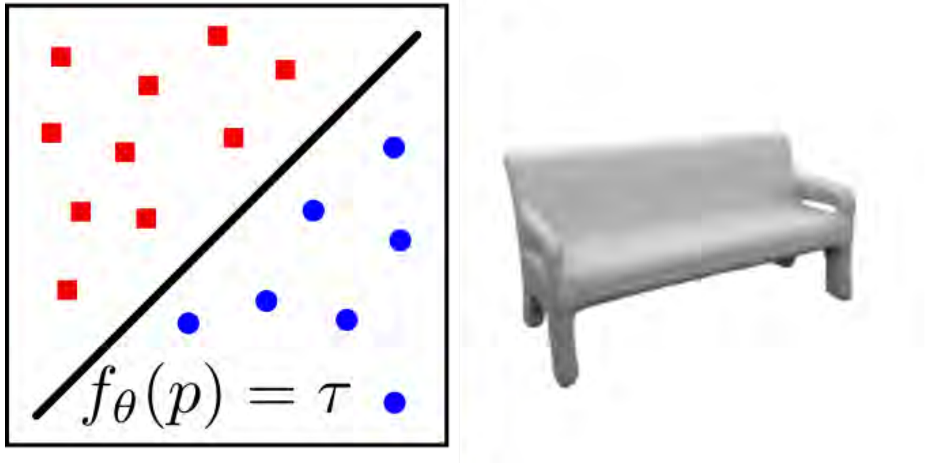
$$p_\theta = \frac{1}{Z_\theta} \exp f_\theta$$

Energy-Based Model

on

2. Representing

Energy-based Implicit Function for 3D shape representation

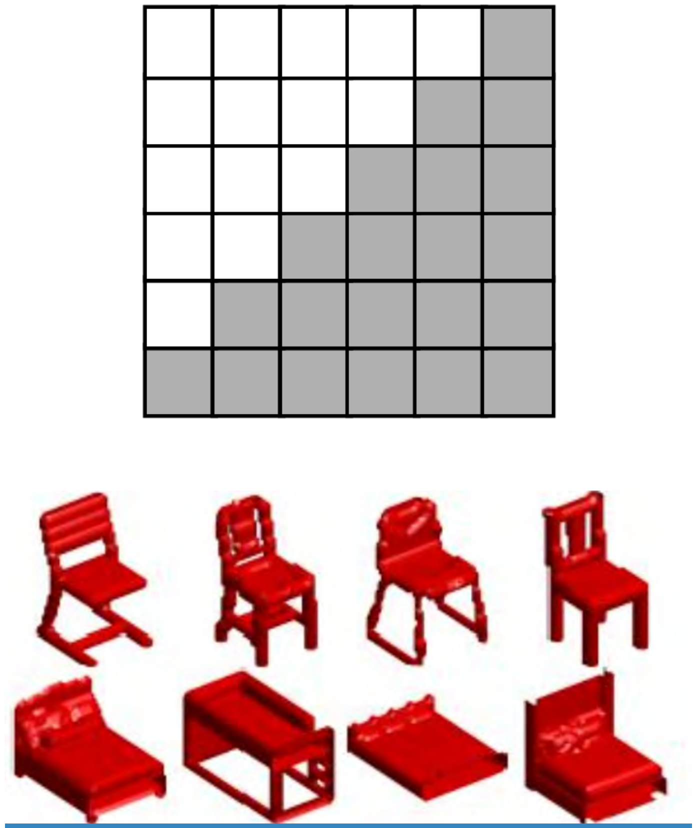


Implicit Representation

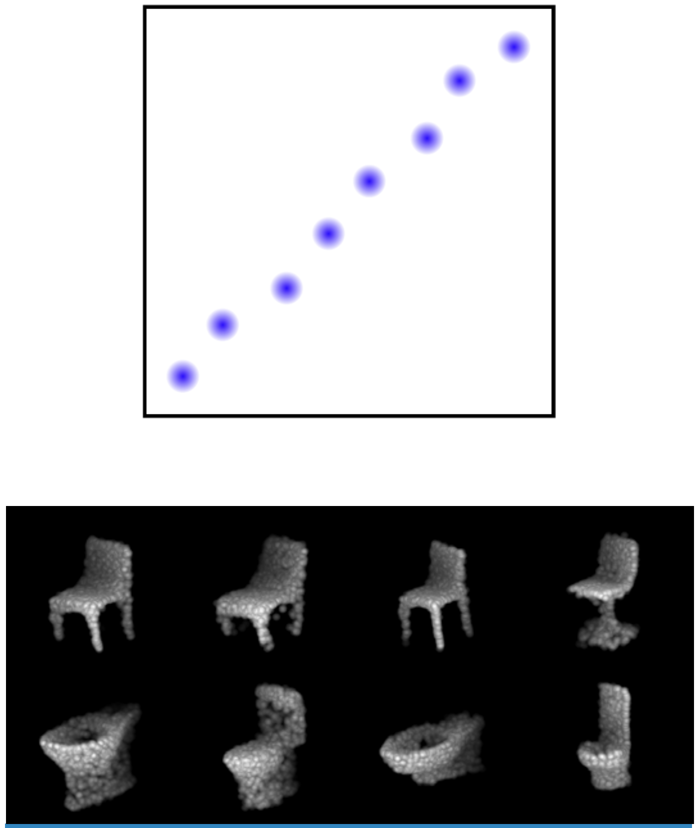
3. Controlling

Energy-based Continuous Inverse Optimal Control

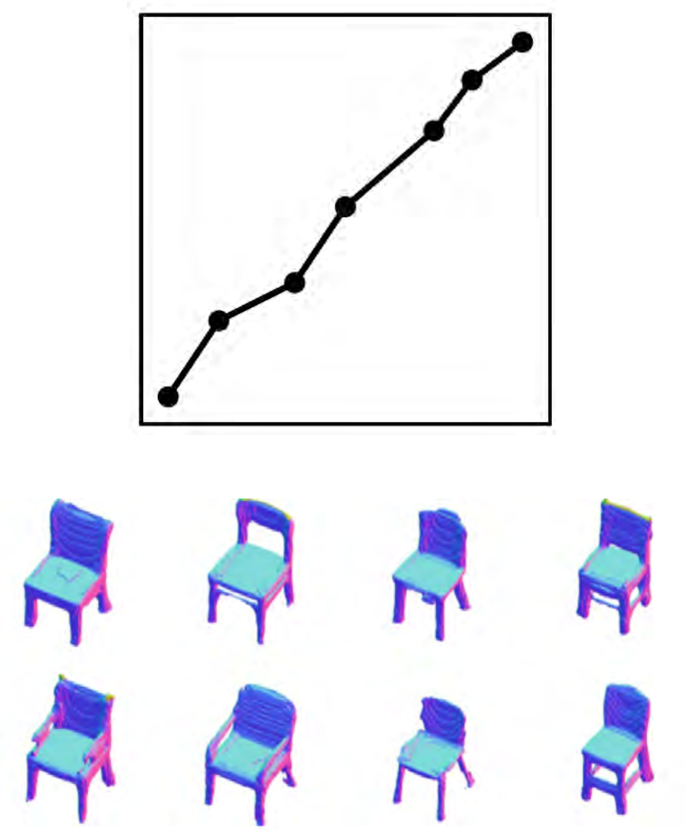
Represent a 3D shape



Voxel

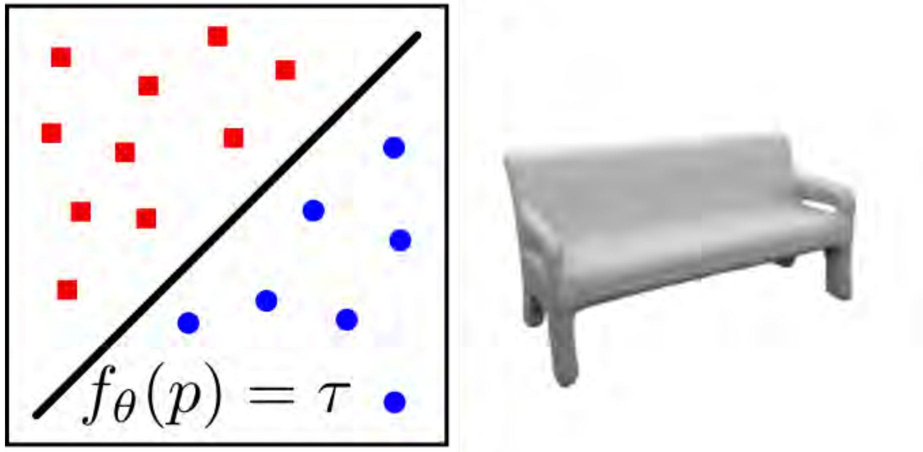


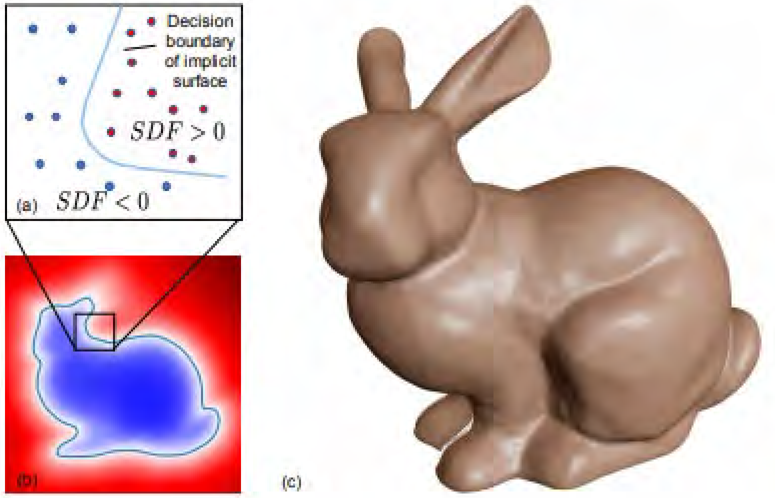
Point Cloud



Mesh

Related Work



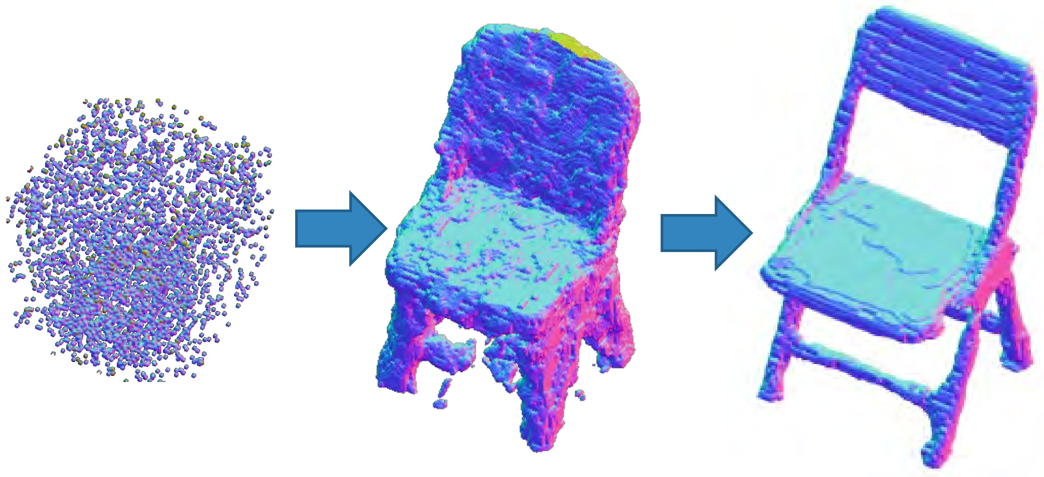


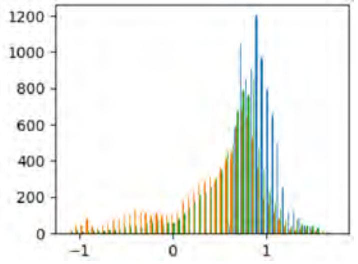
- **Occupancy Network (1 = on the surface; 0 = off the surface)**
  - Need sample negative points (point off the surface)
  - Train as a classifier.
- **Signed distance function (distance to the surface)**
  - Only work on watertight object.
  - Must have explicit definition of “in” and “out”
  - Need to calculate SDF over all training point.

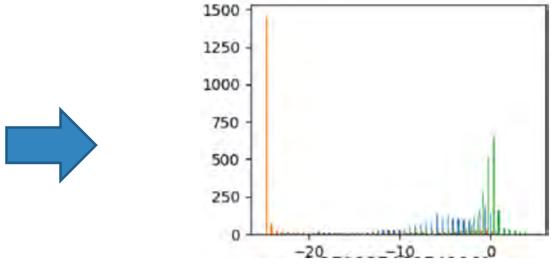
Energy-Based Implicit Function

- Defined on point

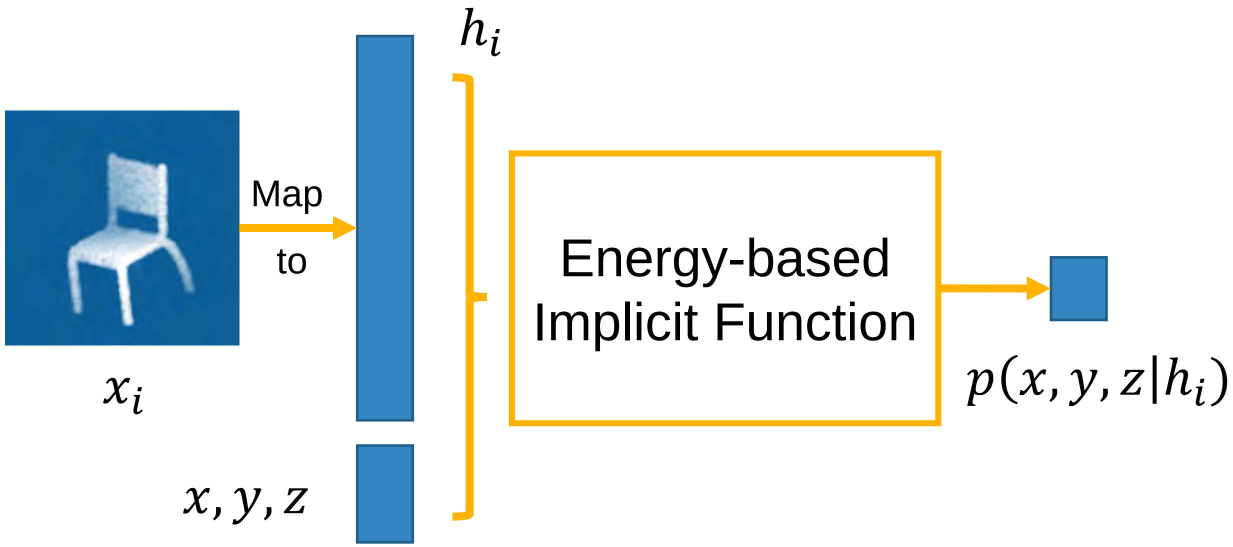
$$p_\theta(x, y, z) = \frac{1}{Z(\theta)} \exp f_\theta(x, y, z)$$







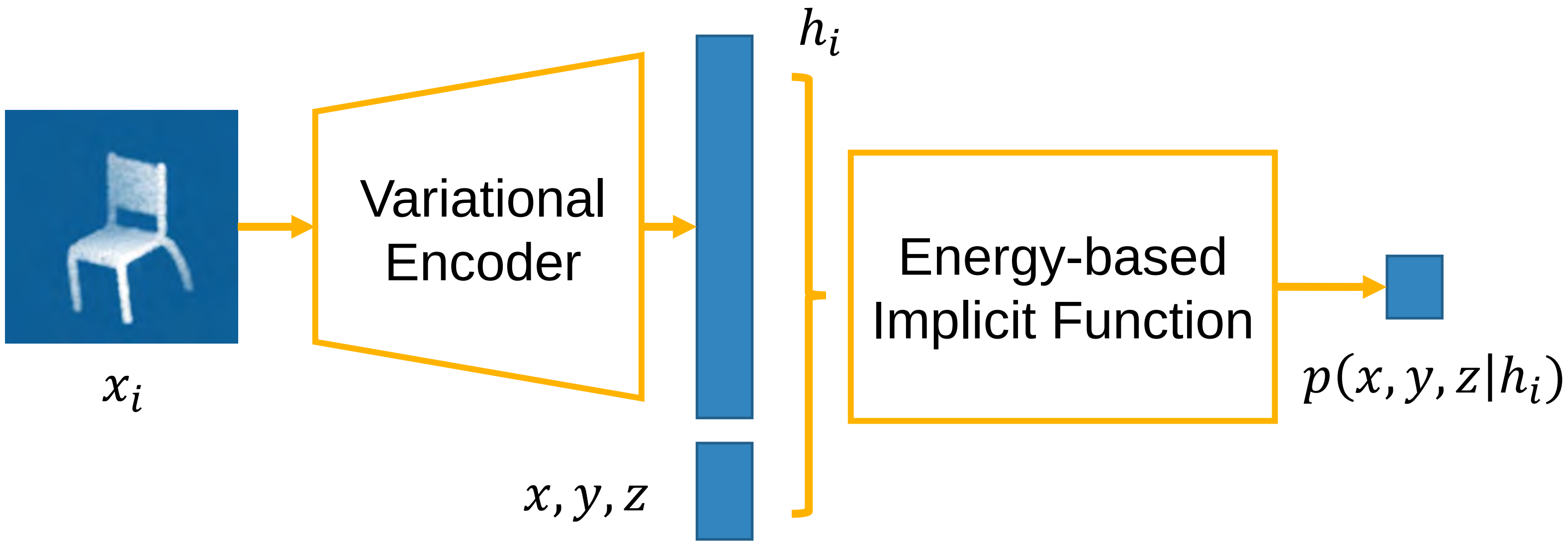
Conditional EBIF meets VAE



MLE Loss

Loss: $L(\phi|X_i) = \boxed{E_{q_{\phi(z|x_i)}} \log p_{\theta(x|z)}}$

Conditional EBIF meets VAE



MLE Loss KL Loss

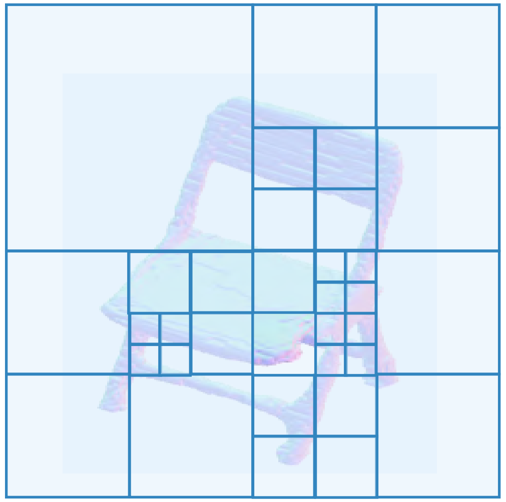
Variational Loss: $L(\phi|X_i) = \underbrace{E_{q_{\phi(Z|x_i)}} \log p_{\theta(x|Z)}}_{\text{MLE Loss}} - \underbrace{KL\left(q_{\phi(z)} \parallel p_{0(z)}\right)}_{\text{KL Loss}}$

Reparametrized trick: $\sum_{\{x,y,z\}\in X_i} \log p_\theta(x, y, z|\mu_i + \epsilon\sigma_i)$ $\frac{1}{2}\sum_{j=1}^{J}(1 + 2\log\sigma - \mu_j^2 - \sigma_j^2)$

Importance Sampling

Maximum Likelihood Estimation use gradient descent:

$$\frac{\partial}{\partial\theta} l(\theta) = E_{q_{data}}\left[\frac{\partial}{\partial\theta} f_\theta(X)\right] - E_{p_\theta}\left[\frac{\partial}{\partial\theta} f_\theta(X)\right]$$



Importance Sampling

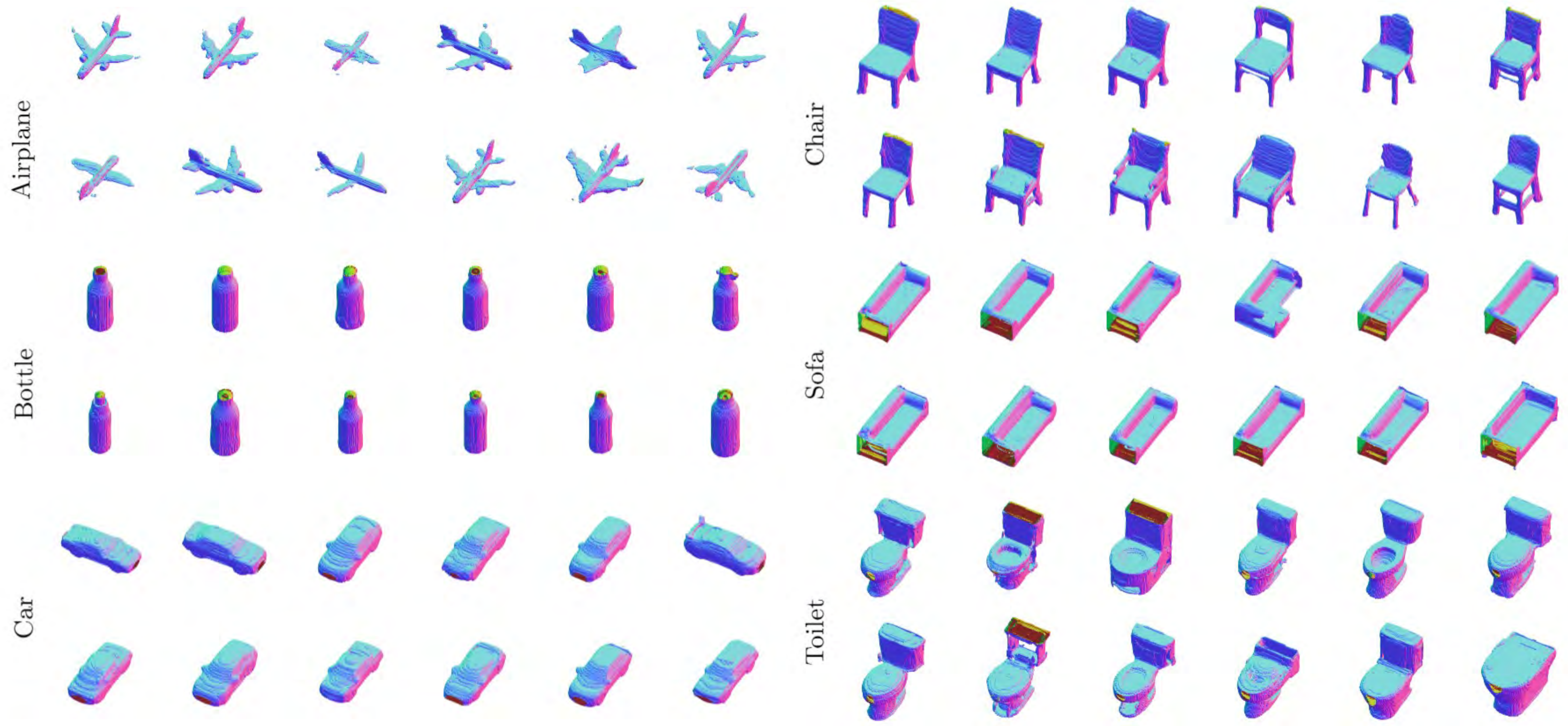
$$E_p[h(x)] = \int h(x)p(x)dx = \int \frac{p(x)}{q(x)} h(x)q(x)dx = E_q\left[\frac{p(x)}{q(x)} h(x)\right]$$

To make a better approximation:

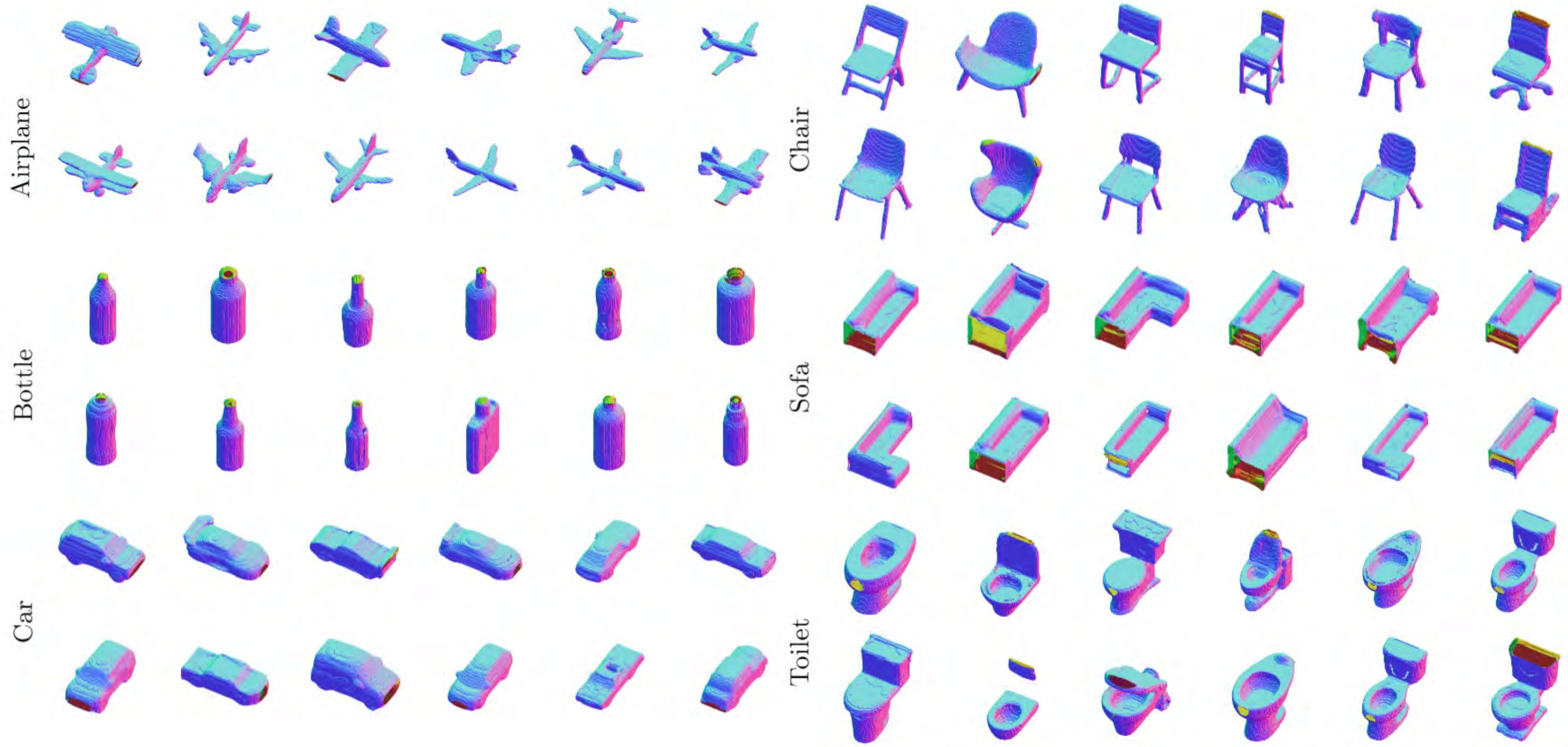
1. The reference should be easy to sample. --- use uniform distribution
2. The reference should be not too far away from the target distribution --- Piecewise Uniform

For point x in a specific cube G: $q(\boldsymbol{x}) = \frac{1}{Z_q} \exp f_\theta(\overline{\boldsymbol{x}})$, Where $\overline{\boldsymbol{x}}$ is the center point of the cube G.

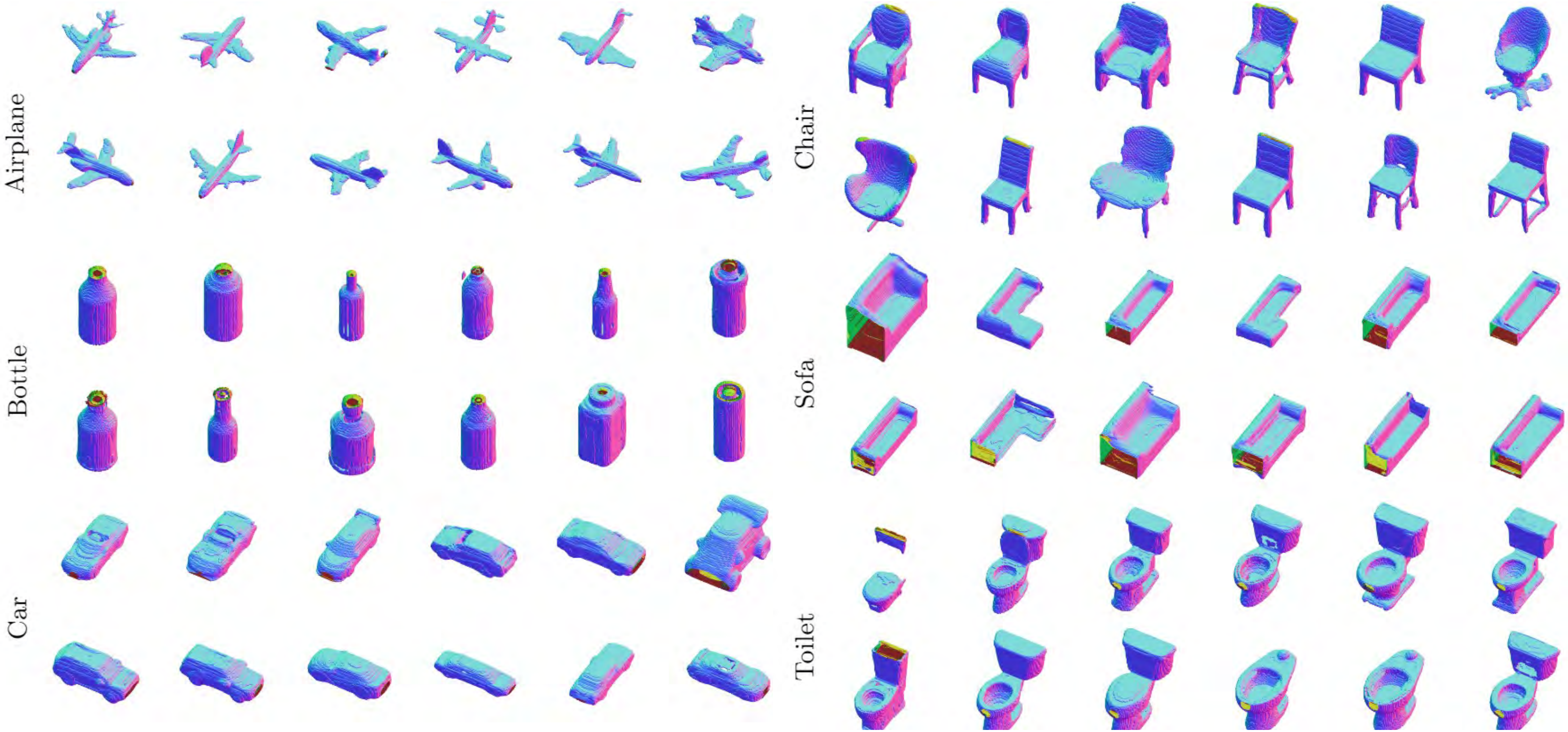
Generation Results



Reconstruction Results



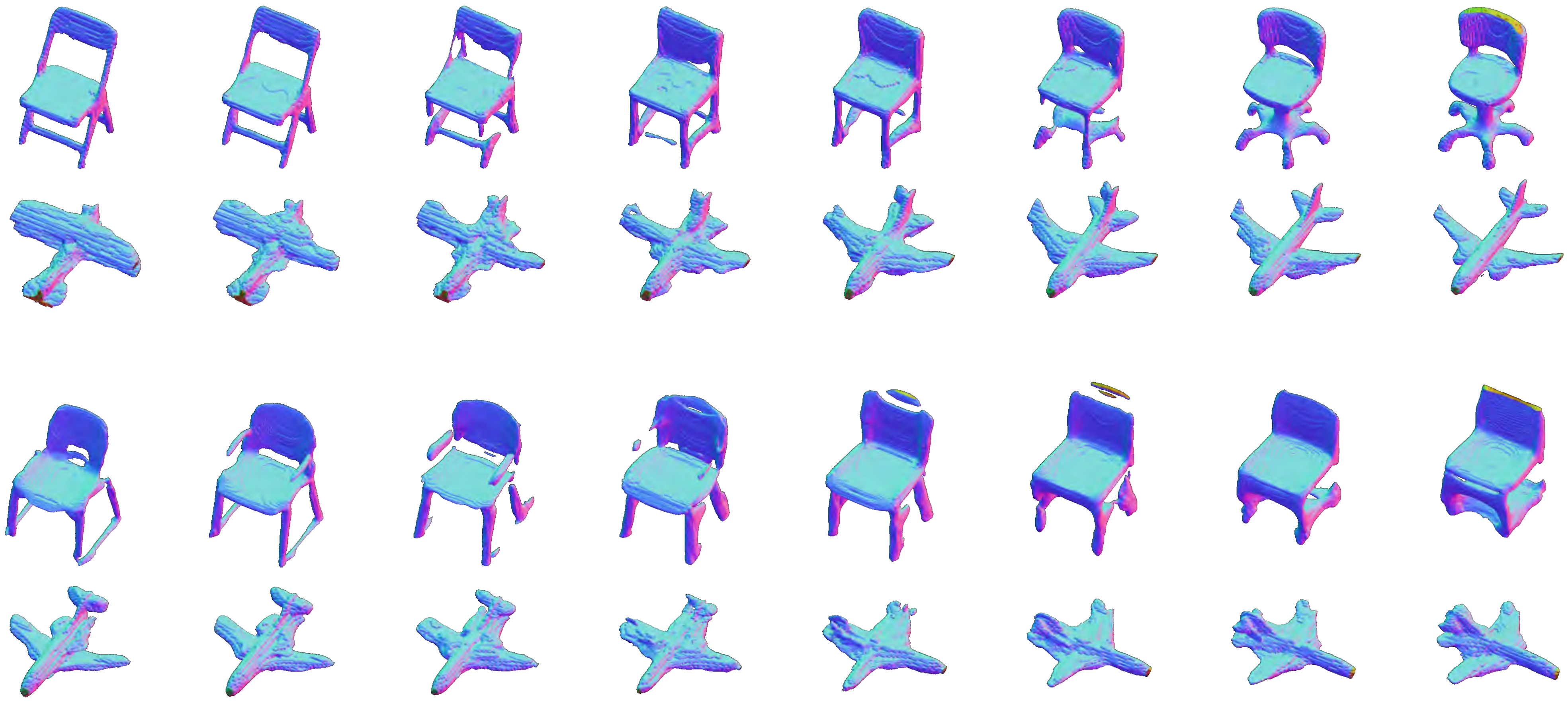
Testing reconstruction



Interpolation Results

Training

Testing



1. Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

2. Representing

Energy-based Implicit Function for 3D shape representation

3. Controlling

Energy-based Continuous Inverse Optimal Control

Current challenge?

Cannot deal with non-watertight object;
Need human-defined function.

Why EBM helps?

A natural representation --- $p(x, y, z)$
No need to sample negative points

How to model and sample?

Importance Sampling
Multi-grid / volume-adaptive piecewise uniform

One more thing…

Cooperate with VAE
Good generation results

1. Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

2. Representing

Energy-based Implicit Function for 3D shape representation

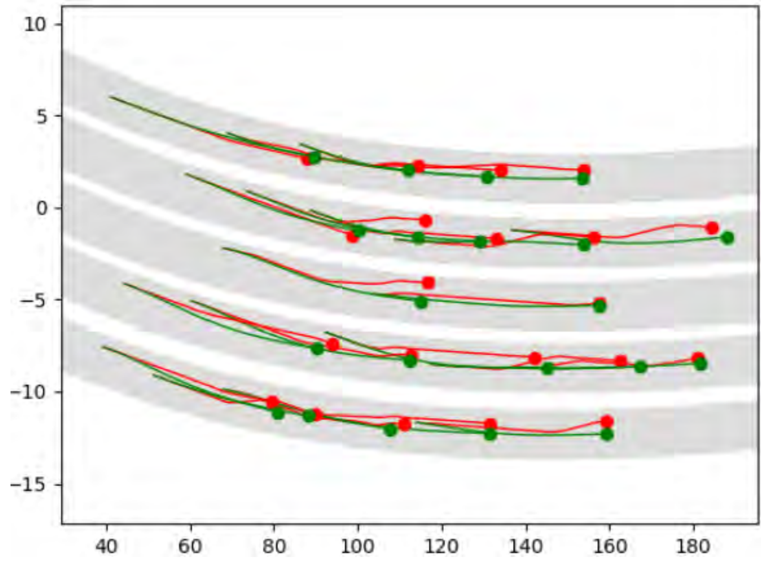
3. Controlling

Energy-based Continuous Inverse Optimal Control

$$p_\theta = \frac{1}{Z_\theta} \exp f_\theta$$

Energy-Based Model

on



Inverse Optimal Control

Continuous Inverse Optimal Control

The MDP formulation for self-driving

State: x = {longitude, latitude, speed, heading angle, acceleration, steering angle};

x_t: State

change of acceleration, change of steering angle

u_t: Control

Dynamic : The transition function. f is bicycle model. $x_{t+1} = f(x_t, u_t)$

f: Dynamic Function

Continuous Inverse Optimal Control

The MDP formulation for self-driving

x_t: State

u_t: Control

f: Dynamic Function

C_θ: Cost Function

$$U = \arg\min_{U} C_\theta(X, U)$$

Continuous Inverse Optimal Control

The MDP formulation for self-driving

x_t: State

u_t: Control

f: Dynamic Function

C_θ: Cost Function

Learn from

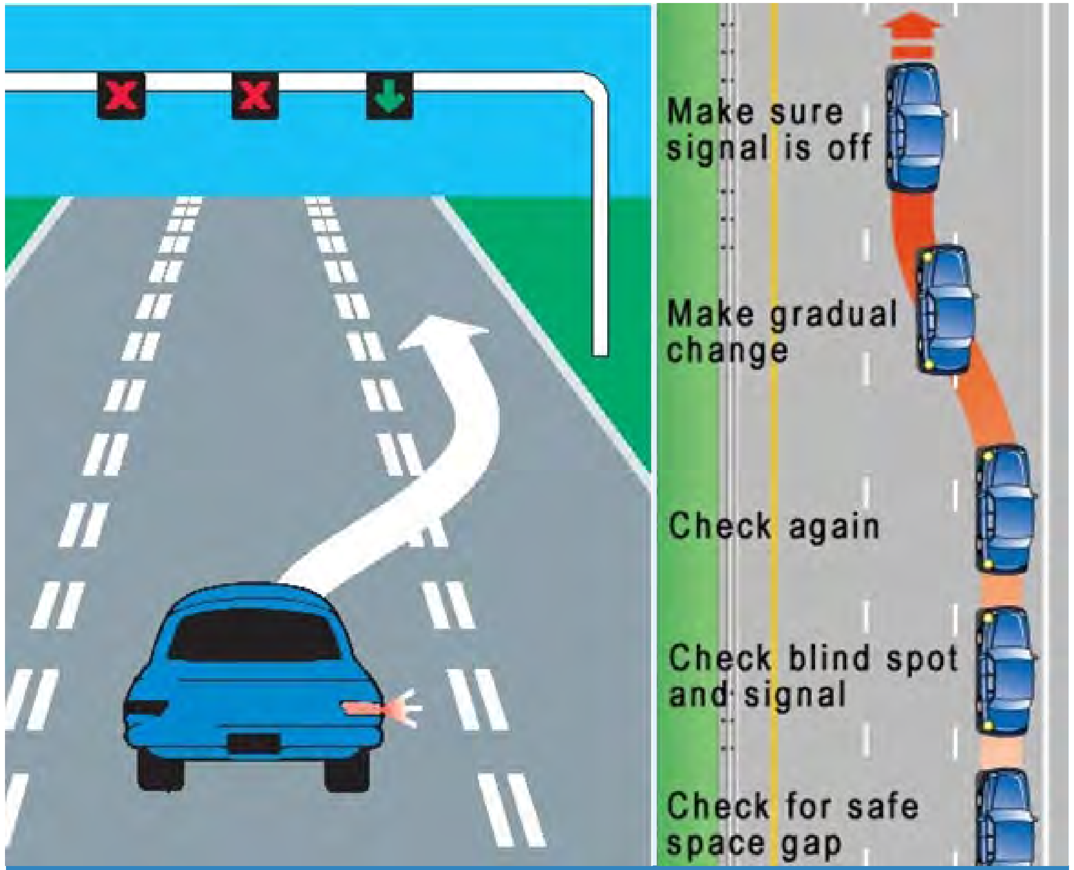
Expert Demonstration: $\{\tau_i\}$

A sequence of state and control pair (X, U)

$$\theta = \arg\min_\theta C_\theta(\tau_i)$$

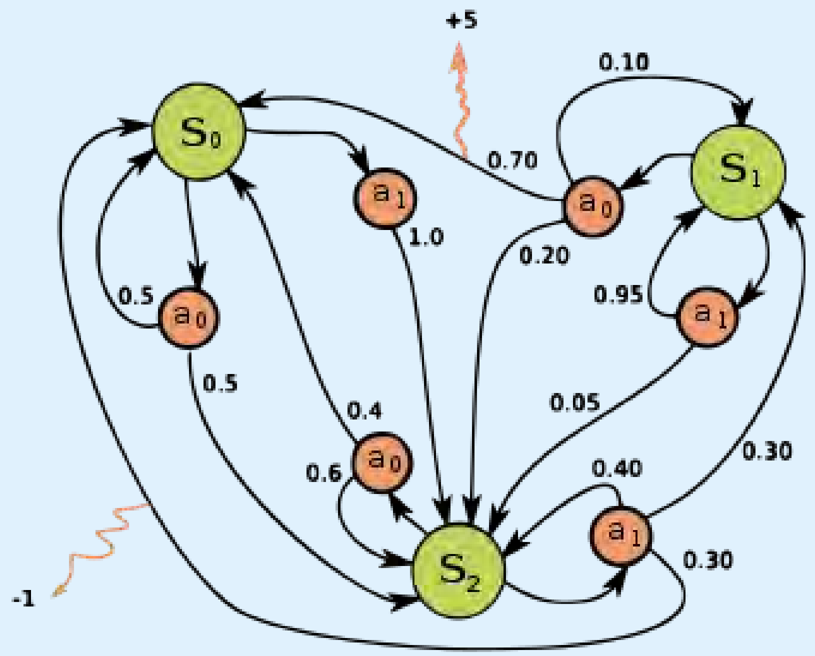
How human drive?

Instinct

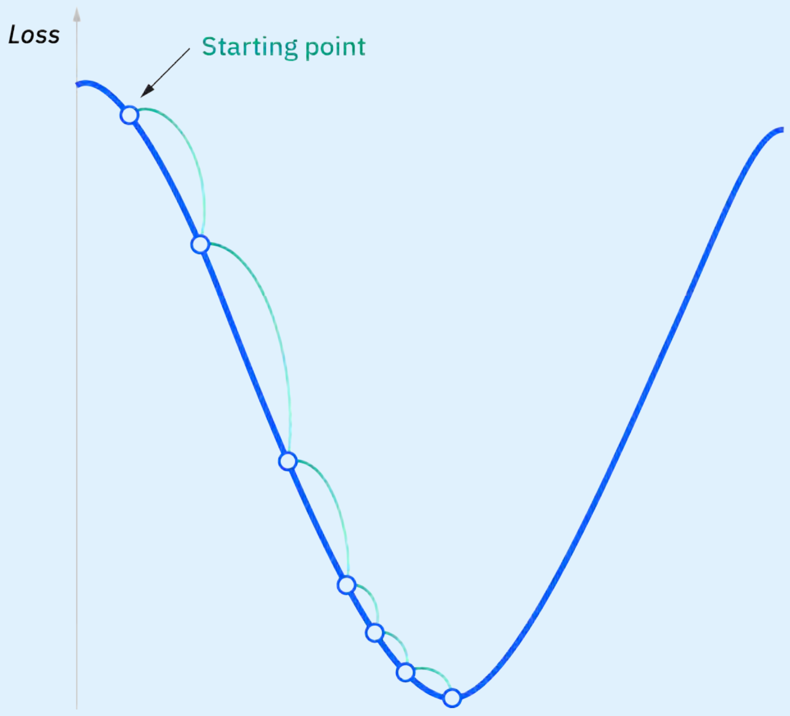


Computation

How an agent drives?



Fast thinking: Policy Method



Slow thinking: Optimize Method

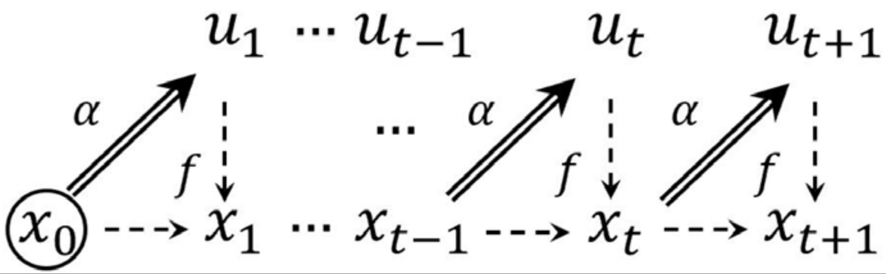
Two type of model for optimal control

$$u = \arg\min_{u} \boxed{Q_\theta(x, u)}$$

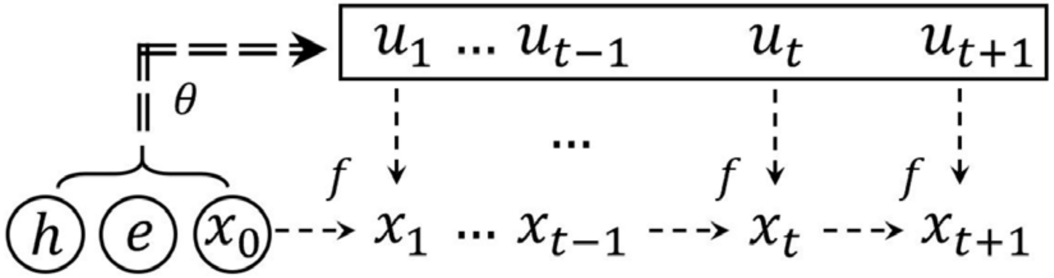
Q_θ: The expected future cost

$$U = \arg\min_{U} \boxed{C_\theta(X, U)}$$

U: A sequence of control u C_θ: A sequence of cost c_θ



Fast thinking: Policy Method



Slow thinking: Optimize Method

Conditional EBM

Fast thinking: Policy Method	Slow thinking: Optimize Method

- Conditional Energy-based Model:

$$p_\theta(\boldsymbol{\tau}|e,h) = p_\theta(\mathbf{u}|e,h) = \frac{1}{Z_\theta(e,h)}\exp[-C_\theta(\mathbf{x},\mathbf{u},e,h)]$$

Where $Z_\theta(e,h)$ is the normalizing constant. $e :=$ environment; $h :=$ history.

- Previous work: Use Laplace approximation to approximate Z_θ

$$P(\mathbf{u}|\mathbf{x}_0) \approx e^{r(\mathbf{u})}\left[\int e^{r(\mathbf{u})+(\tilde{\mathbf{u}}-\mathbf{u})^{\mathrm{T}}\mathbf{g}+\frac{1}{2}(\tilde{\mathbf{u}}-\mathbf{u})^{\mathrm{T}}\mathbf{H}(\tilde{\mathbf{u}}-\mathbf{u})}d\tilde{\mathbf{u}}\right]^{-1} = \left[\int e^{-\frac{1}{2}\mathbf{g}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{g}+\frac{1}{2}(\mathbf{H}(\tilde{\mathbf{u}}-\mathbf{u})+\mathbf{g})^{\mathrm{T}}\mathbf{H}^{-1}(\mathbf{H}(\tilde{\mathbf{u}}-\mathbf{u})+\mathbf{g})}d\tilde{\mathbf{u}}\right]^{-1} = e^{\frac{1}{2}\mathbf{g}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{g}}\left|-\mathbf{H}\right|^{\frac{1}{2}}(2\pi)^{-\frac{d_{\mathbf{u}}}{2}},$$

Conditional EBM

Fast thinking: Policy Method

Slow thinking: Optimize Method

$$p_\theta(\boldsymbol{\tau}|e,h) = p_\theta(\mathbf{u}|e,h) = \frac{1}{Z_\theta(e,h)} \exp[-C_\theta(\mathbf{x},\mathbf{u},e,h)]$$

- Our method: Sampling-based Approach / Optimization-based Approach:

$$\frac{\partial}{\partial\theta} l(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\partial}{\partial\theta} C_\theta(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i, e, h) - \frac{\partial}{\partial\theta} C_\theta(\boldsymbol{x}_i, \boldsymbol{u}_i, e, h)\right]$$

Sampled through Langevin dynamics

or

Predicted through optimization method

Conditional EBM

Fast thinking: Policy Method | Slow thinking: Optimize Method

$$p_\theta(\boldsymbol{\tau}|e,h) = p_\theta(\mathbf{u}|e,h) = \frac{1}{Z_\theta(e,h)}\exp[-C_\theta(\mathbf{x},\mathbf{u},e,h)]$$

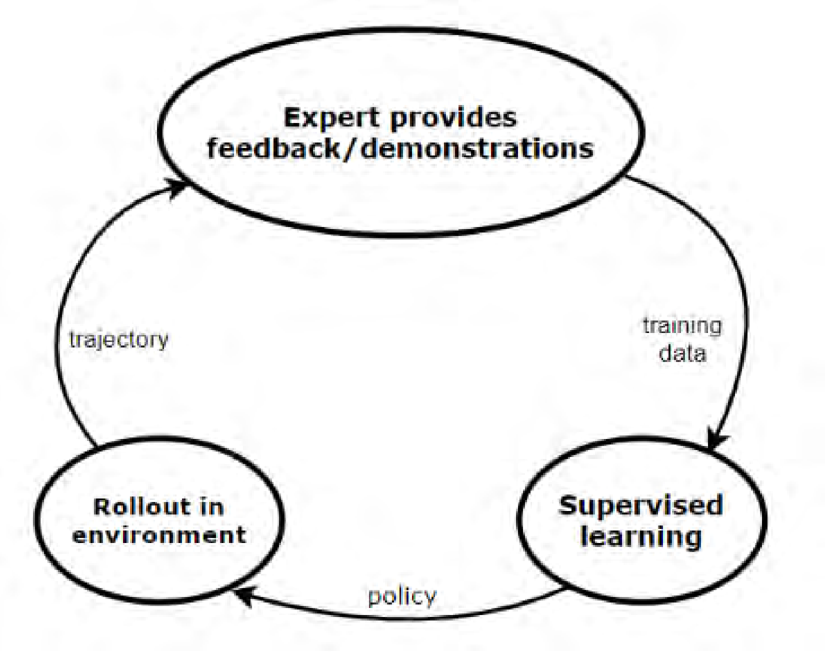
- Sample-based approach:
  - More exploration
  - Finding corner case
  - Better generalization

- Support more complex cost function:
  - 1D CNN
  - LSTM
  - MLP

Two type of model for optimal control

Fast thinking: Policy Method | Slow thinking: Optimize Method

- Previous work: Imitate policy learnt from expert demonstration. Result used directly.



Two type of model for optimal control

Fast thinking: Policy Method | Slow thinking: Optimize Method

- Our method: Imitate policy learn from the optimized result (slow thinking result)
- A generator is used as a fast initializer of the sampling (or the optimization):

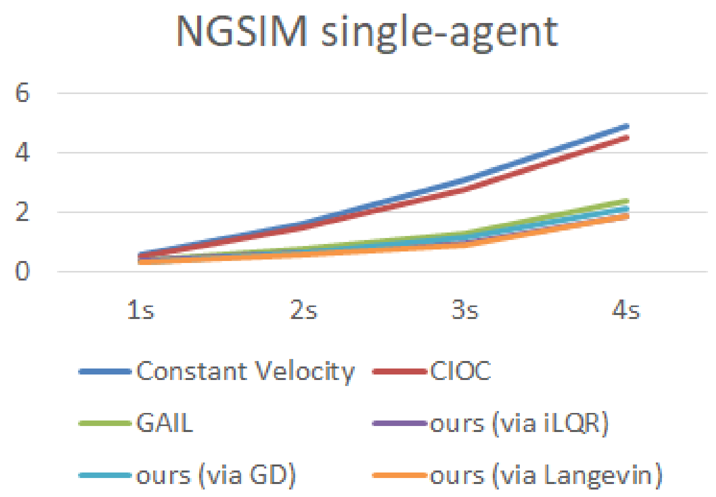
$$q_\alpha(u, \xi | e, h) = q_\alpha(u | \xi, e, h) p(\xi)$$

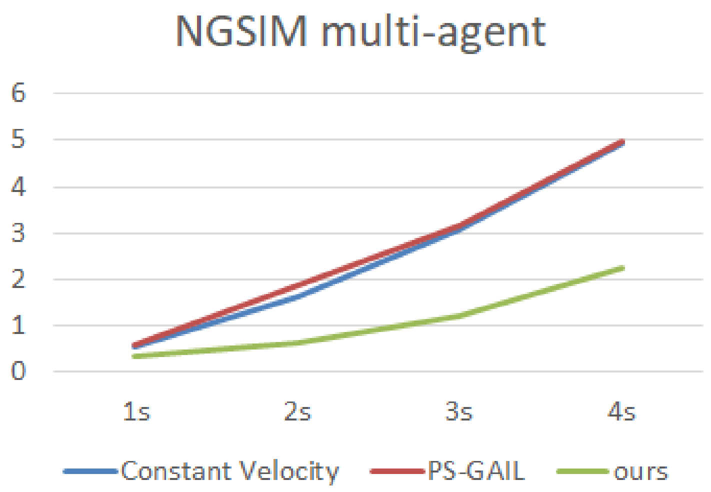
- Loss for encoder:

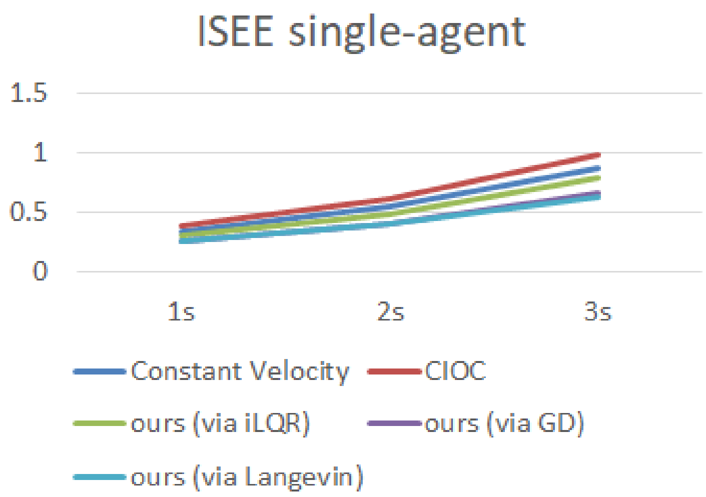
$$L_g(\alpha) = \frac{1}{n} \sum_{i=1}^{n} \| \tilde{u}_i - q_\alpha(\xi, e, h) \|^2$$

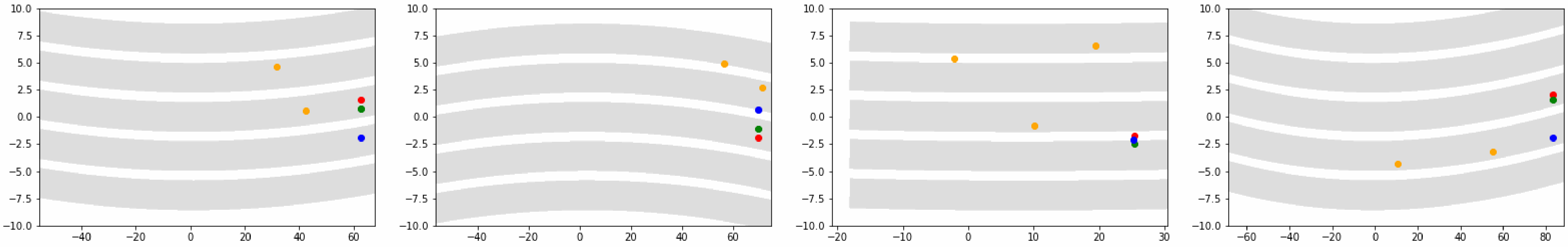
- Result is used as the initialization of the sampling / optimization

EBIOC: Predicted trajectories



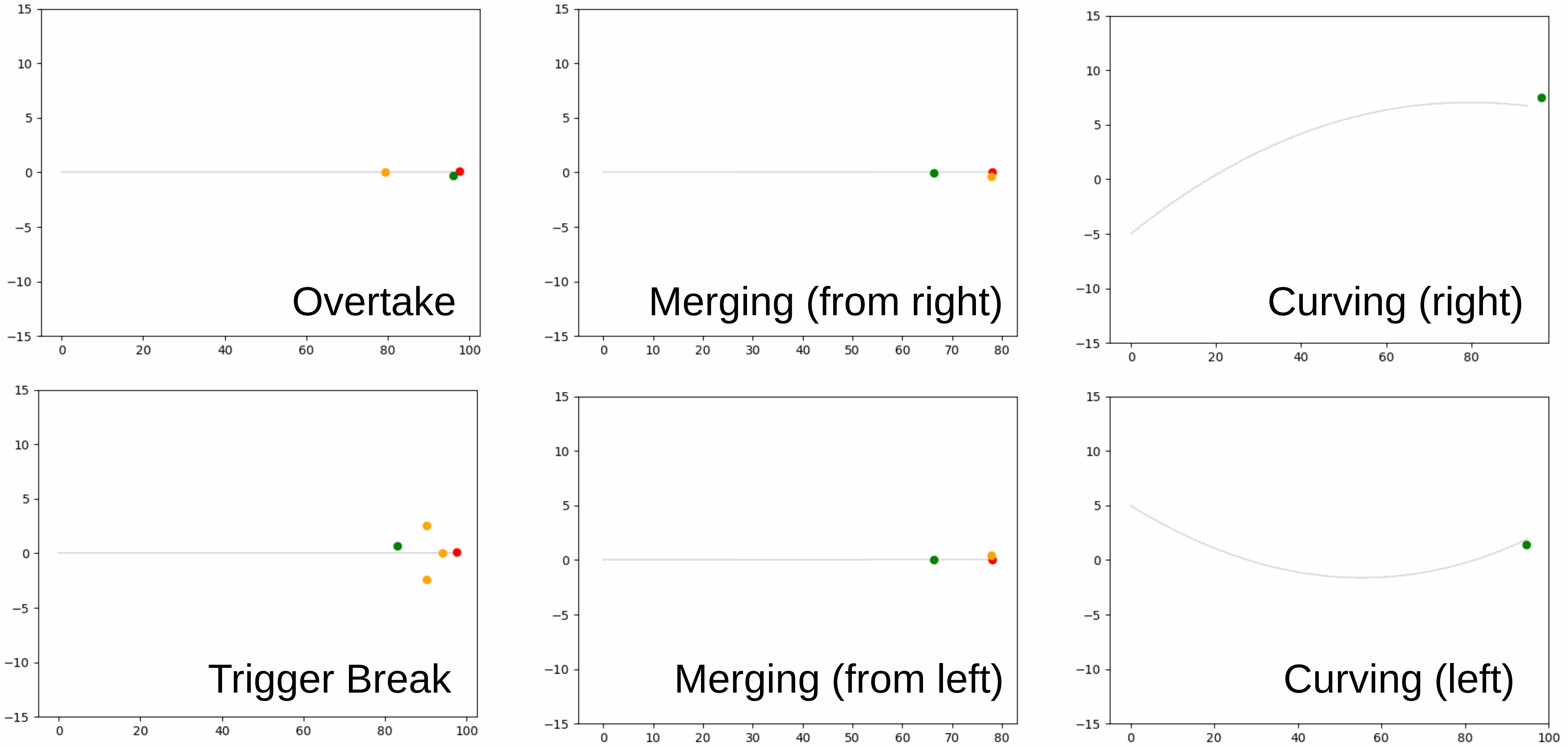






■ Ground Truth; ■ Ours; ■ GAIL; ■ Other Vehicle; ■ Lane.

EBIOC: Toy examples

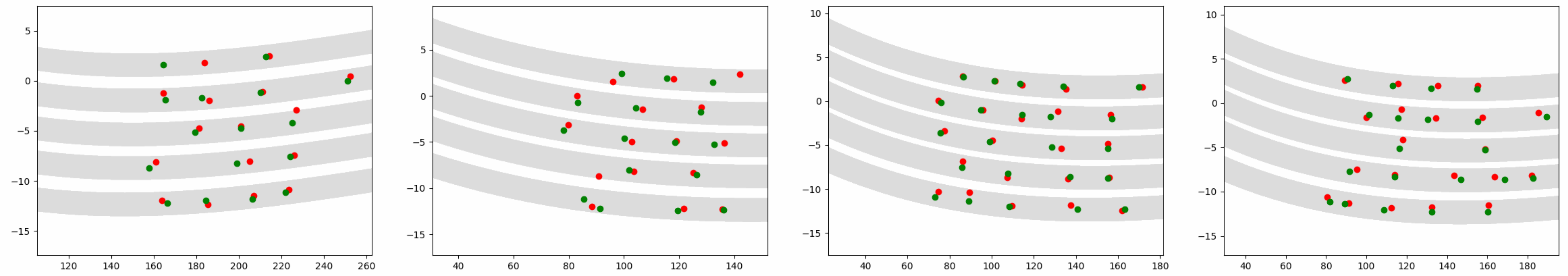


■ Green: predicted trajectories; ■ Orange: other vehicles; ■ Red: Rollout if keep constant control.

EBIOC: Multi-agent Prediction Result

- Meta-cost: sum of each cost = Product of probability = A joint probability distribution
- Assumption: Fully cooperate + Share information

$$p_\theta(\boldsymbol{U}|\boldsymbol{e},\boldsymbol{h}) = \prod_{k=1}^{K} p_\theta(\boldsymbol{u^k}|e,h^k) = \frac{1}{Z_\theta(\boldsymbol{e},\boldsymbol{h})}\exp\left[-\sum_{k=1}^{K} c_\theta(\boldsymbol{x^k},\boldsymbol{u^k},e,h^k)\right]$$



Multi-agent prediction on NGSIM US101 dataset (■ Lane; ■ Ground Truth; ■ Ours)

1. Learning

Generative PointNet: Energy-Based Learning on Unordered Point Sets

2. Representing

Energy-based Implicit Function for 3D shape representation

3. Controlling

Energy-based Continuous Inverse Optimal Control

Current challenge?

Trade off between exploration and exploitation
Trade off between efficiency and accuracy

Why EBM helps?

Sample-based: Better exploration
Combine policy method and optimization

How to model and sample?

Sample-base: Langevin Dynamic
Optimized-based: iLQR

One more thing…

Multi-agent setting
Joint EBM cooperative learning

Publications

2022.03 Jianwen Xie, Yaxuan Zhu, **Yifei Xu**, Dingcheng Li, Ping Li " Generative Learning with Latent Space Flow-based Prior Model" *In review*

2022.02 **Yifei Xu**†, Jingqiao Zhang†, Ru He†, Liangzhu Ge†, Chao Yang, Cheng Yang, Ying Nian Wu " SAS: Self-Augmented Strategy for Language Model Pre-training" *In Proc. 36th AAAI Conference on Artificial Intelligence (AAAI) 2022*

2021.06 **Yifei Xu**†, Jianwen Xie†, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu " Generative PointNet : Deep Energy-Based Learning on Point Sets for 3D Generation and Reconstruction" *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2021.*

2020.02 **Yifei Xu**, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, Ying Nian Wu " Energe-based Continous Inverse Optimal Control" *Journal version submitted to TNNLS; NeurIPS workshop on Machine Learning for Autonomous Driving, 2020*

2018.11 Tianyang Zhao, **Yifei Xu**, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, Ying Nian Wu " Convolutional Spatial Fusion for Multi-Agent Trajectory Prediction" *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019.*

2017.03 Jianwen Xie, **Yifei Xu**, Erik Nijkamp, Ying Nian Wu, Song-Chun Zhu "Generative Hierarchical Structure Learning of Sparse FRAME Models" IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017.

Reference

[1] E. Todorov, "Optimal control theory," *Bayesian brain: probabilistic approaches to neural coding*, pp. 269–298, 2006.
[2] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *ICINCO (1)*, 2004, pp. 222–229.
[3] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
[4] J. Xie, Y. Lu, S.-C. Zhu, and Y. Wu, "A theory of generative convnet," in *International Conference on Machine Learning*, 2016, pp. 2635–2644.
[5] R. M. Neal *et al.*, "Mcmc using hamiltonian dynamics," *Handbook of markov chain monte carlo*, vol. 2, no. 11, p. 2, 2011.
[6] P. J. Bickel and K. A. Doksum, *Mathematical statistics: basic ideas and selected topics, volumes I-II package*. CRC Press, 2015.
[7] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
[8] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *Journal of Machine Learning Research*, vol. 6, pp. 695–709, 2005.
[9] J. Xie, Y. Lu, R. Gao, S. Zhu, and Y. Wu, "Cooperative learning of energy-based model and latent variable model via mcmc teaching." AAAI, 2018.
[10] J. Xie, Y. Lu, R. Gao, S.-C. Zhu, and Y. N. Wu, "Cooperative training of descriptor and generator networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 27–45, 2018.
[11] J. Xie, Z. Zheng, X. Fang, S.-C. Zhu, and Y. N. Wu, "Cooperative training of fast thinking initializer and slow thinking solver for multi-modal conditional learning," *arXiv preprint arXiv:1902.02812*, 2019.
[12] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *AAAI*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.
[13] S. C. Zhu, Y. Wu, and D. Mumford, "Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 107–126, 1998.
[14] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
[15] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," *arXiv preprint arXiv:1507.04888*, 2015.
[16] J. Xie, S.-C. Zhu, and Y. N. Wu, "Learning energy-based spatial-temporal generative convnets for dynamic patterns," *IEEE transactions on pattern analysis and machine intelligence*, 2019.
[17] J. Xie, W. Hu, S.-C. Zhu, and Y. N. Wu, "Learning sparse frame models for natural image patterns," *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 91–112, 2015.
[27] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 4565–4573.
[28] Y. Li, J. Song, and S. Ermon, "Infogail: Interpretable imitation learning from visual demonstrations," in *Advances in Neural Information Processing Systems*, 2017, pp. 3812–3822.
[29] M. Monfort, A. Liu, and B. D. Ziebart, "Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, pp. 3672–3678. [Online]. Available: http://dl.acm.org/citation.cfm?id=2888116.2888226
[30] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *arXiv preprint arXiv:1206.4617*, 2012.
[31] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.
[32] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2018.
[33] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *Proceedings of the International Conference on Robotics and Automation (ICRA) 2018*, May 2018.
[34] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 336–345.
[35] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018.
[36] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
[37] J. Xie, Z. Zheng, X. Fang, S. Zhu, and Y. N. Wu, "Learning cycle-consistent cooperative networks via alternating MCMC teaching for unsupervised cross-domain translation," in *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021, pp. 10 430–10 440.
[38] J. Xie, Z. Zheng, and P. Li, "Learning energy-based model with variational auto-encoder as amortized sampler," in *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021, pp. 10 441–10 451.
[39] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.
[40] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.
[41] T. M. Cover and J. A. Thomas, *Elements of information theory, Second Edition*. Wiley, 2006.
[42] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.
[43] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," in *International Conference on Machine Learning (ICML)*, vol. 32, 2014, pp. 1683–1691.
[44] E. Nijkamp, M. Hill, T. Han, S. Zhu, and Y. N. Wu, "On the anatomy of mcmc-based maximum likelihood learning of energy-based models," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 5272–5280.
[45] J. Xie, R. Gao, Z. Zheng, S.-C. Zhu, and Y. N. Wu, "Learning dynamic generator model by alternating back-propagation through time," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5498–5507.
[46] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 812–818.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[48] J. Colyar and J. Halkias, "Us highway dataset," vol. Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030, 2007.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.
[50] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 204–211.
[51] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, 2015, pp. 1889–1897.
[52] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[53] R. P. Bhattacharyya, D. J. Phillips, B. Wulfe, J. Morton, A. Kuefler, and M. J. Kochenderfer, "Multi-agent imitation learning for driving simulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1534–1539.
[54] R. P. Bhattacharyya, D. J. Phillips, C. Liu, J. K. Gupta, K. Driggs-Campbell, and M. J. Kochenderfer, "Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, May 2019.

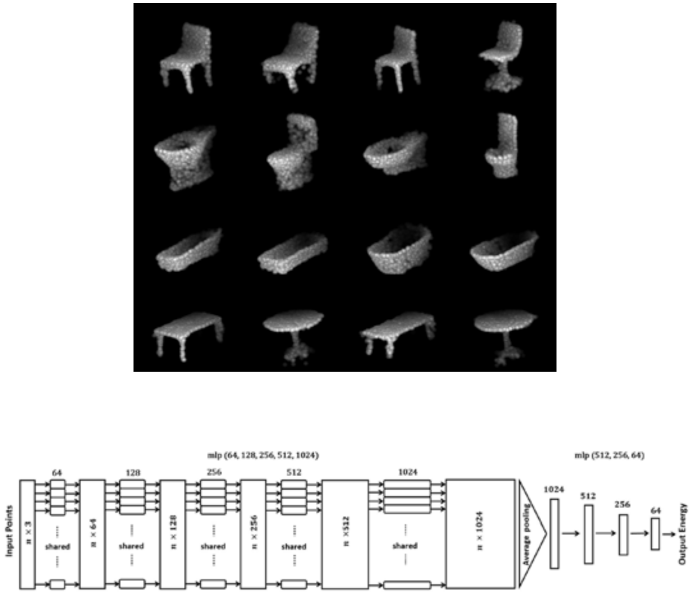
UCLA

Thanks

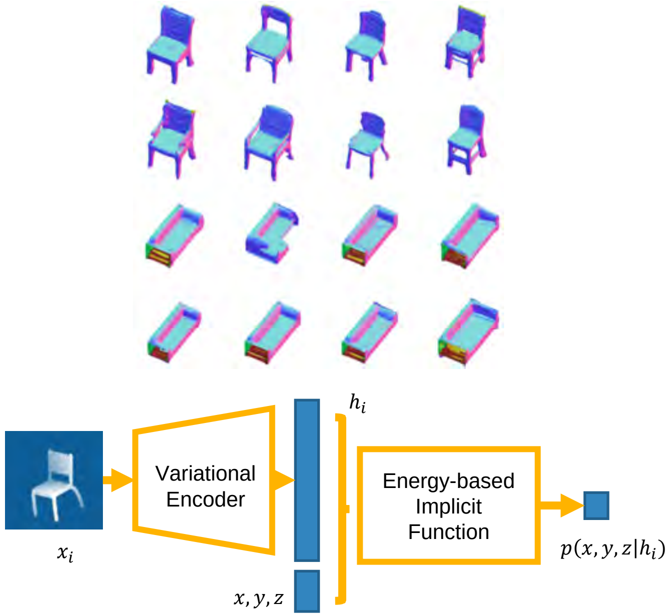
Yifei Xu

Q&A

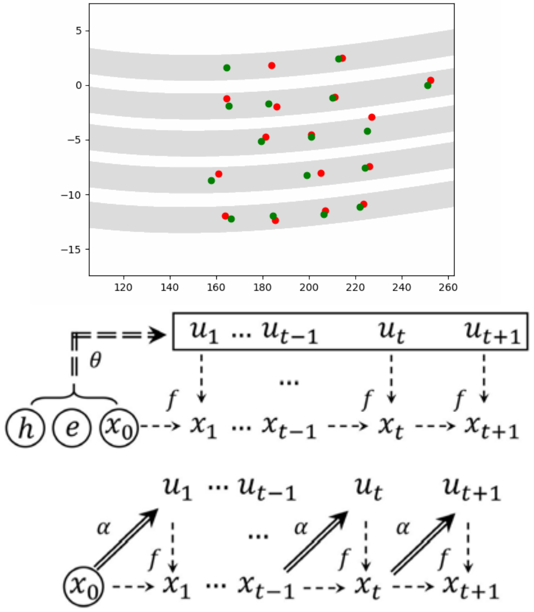
Energy-based Model: $p(X) = \frac{1}{Z} \exp f(X)$



Point Cloud



Implicit Function



Control