From vectors to graphs: architectures for set and graph generation

Based on *Top-N: Equivariant set and graph generation without exchangeability*
C.V., Pascal Frossard   (ICLR 2022)

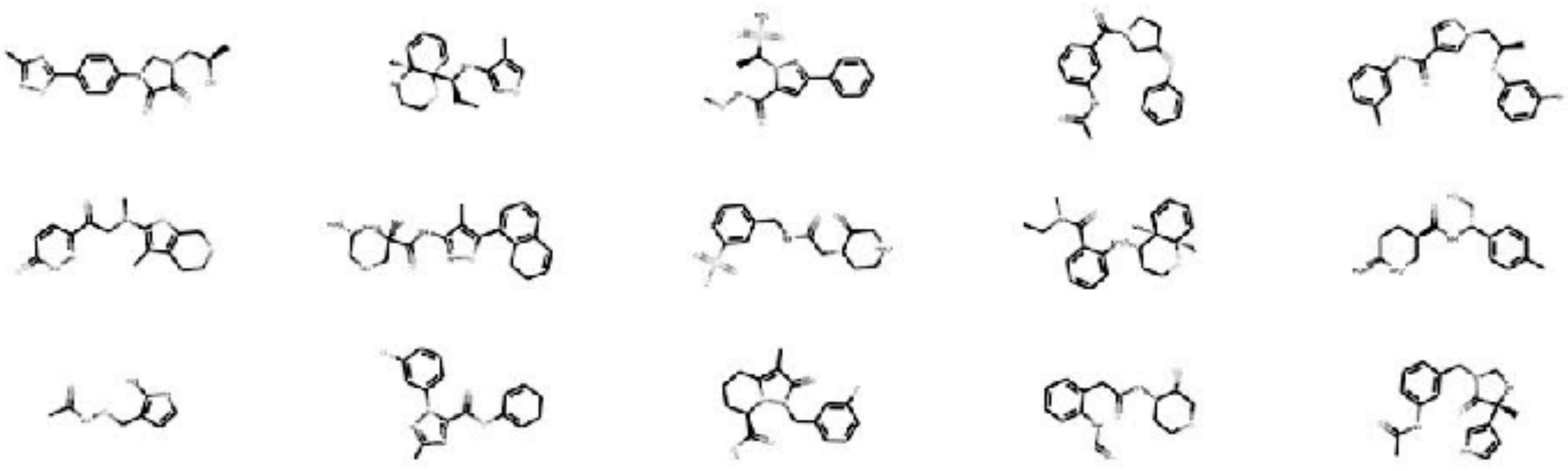
Clément Vignac

March 8, 2022

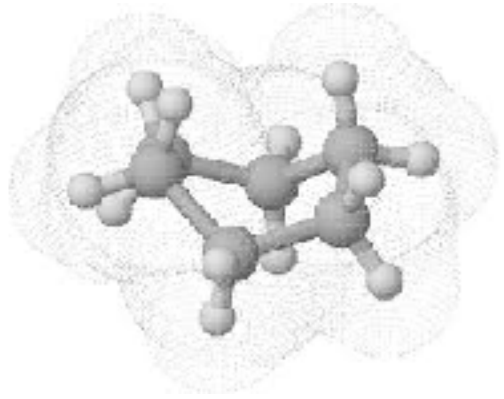
Signal Processing Laboratory
(LTS4)

Motivation: molecule generation for drug discovery

Predict new molecular graphs



Predict molecular conformers (=3d location of each atom) → set generation



How to build models for one-shot set and graph generation?

Settings

A generative model for sets or graphs can take as input:

- a latent vector   - a latent set   - a latent graph

focus of this talk

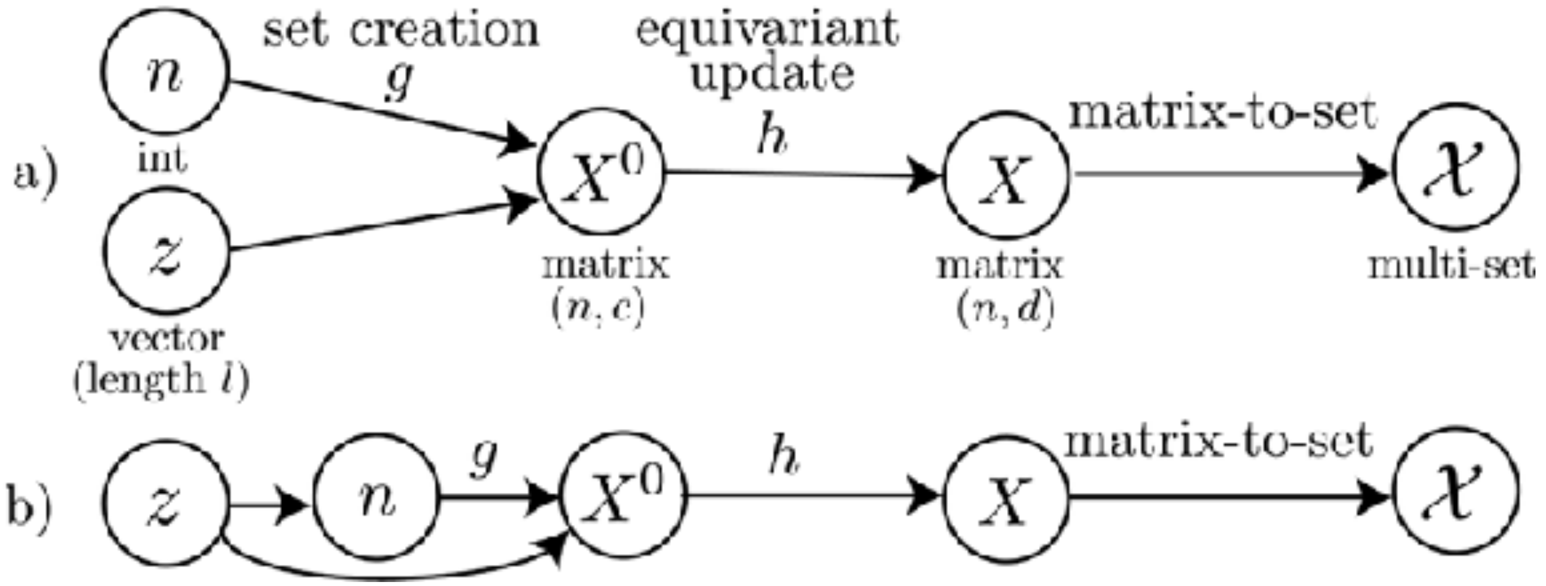
The latent vector is needed to condition on molecule-level properties (e.g., solubility)

Outline:

1. Problem modelling   2. An equivariance perspective   3. Proposition: Top-n creation

One-shot generation

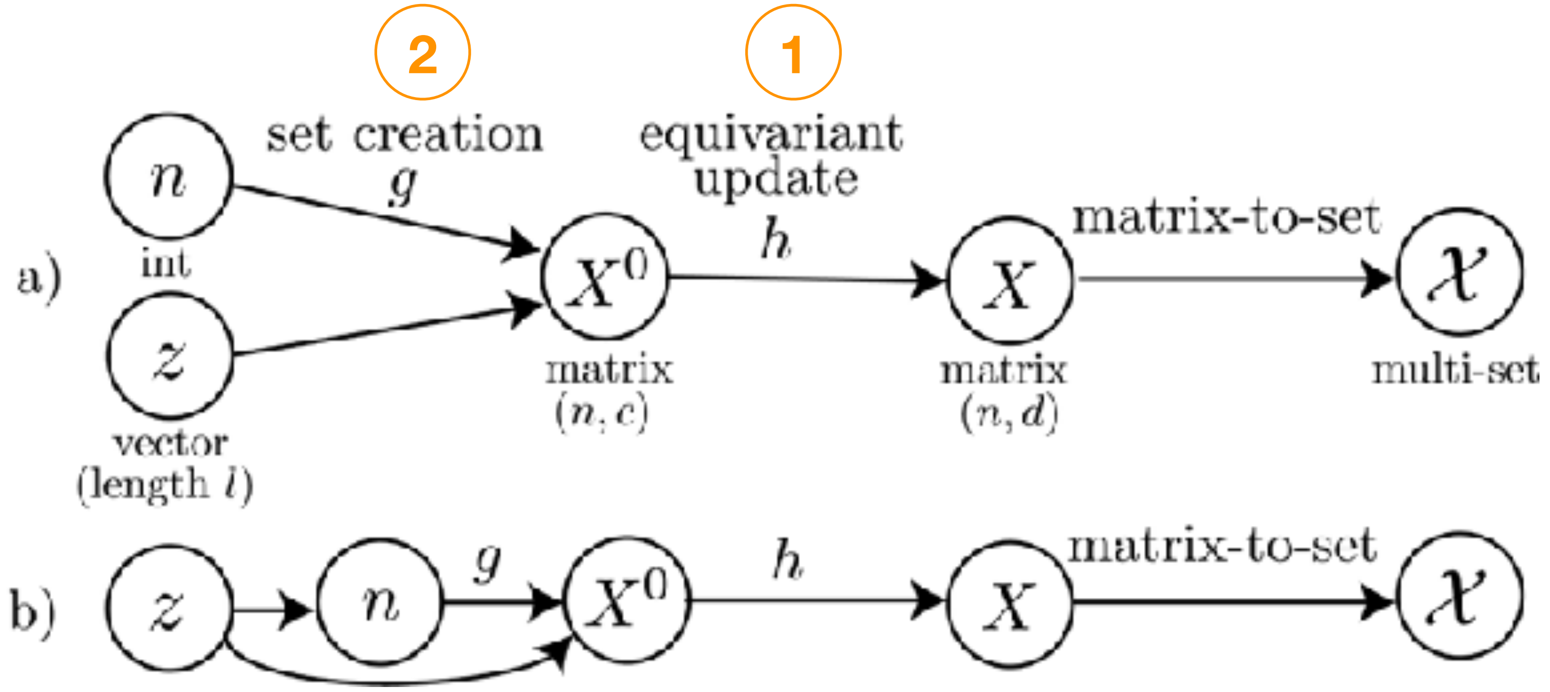
- Existing methods follow one of these two graphical models



- For graphs, edge weights $\boldsymbol{A}^0$ and $\boldsymbol{A}$ are predicted as well

One-shot generation

- Existing methods follow one of these two graphical models



- For graphs, edge weights $\boldsymbol{A}^0$ and $\boldsymbol{A}$ are predicted as well

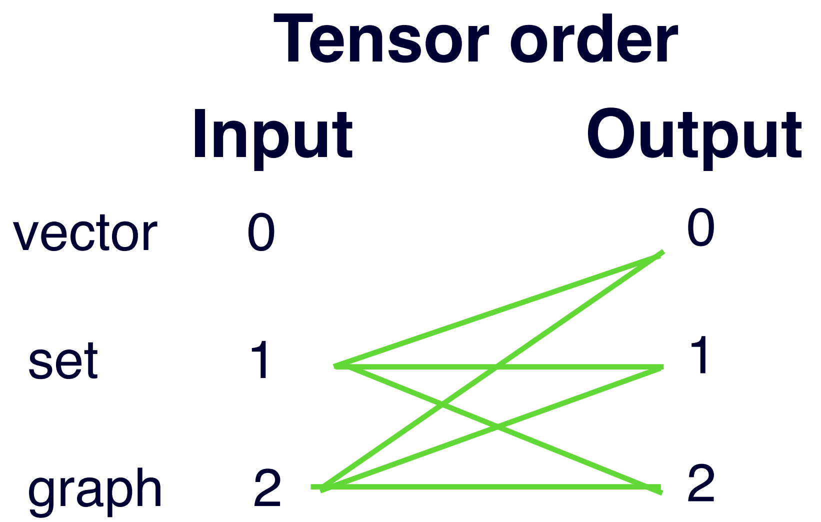
Existing methods for equivariant update

To map a set / graph to another set / graph, any equivariant function can be used. The choice depends on the tensor order of the input and output.

vector: d → order 0

set: $n \times d$ → order 1

edge feat: $n \times n \times d$ → order 2



2 → 0: graph neural network, global pooling

2 → 1: graph neural network, extract node features

1 → 0: Deep sets

2 → 2: graph neural network, extract messages

1 → 2: Set2Graph

1 → 1: Point Nets or Transformers (1 → 2 → 1)

i → j : *Maron et al. 2019 + Keriven and Peyré 2020*

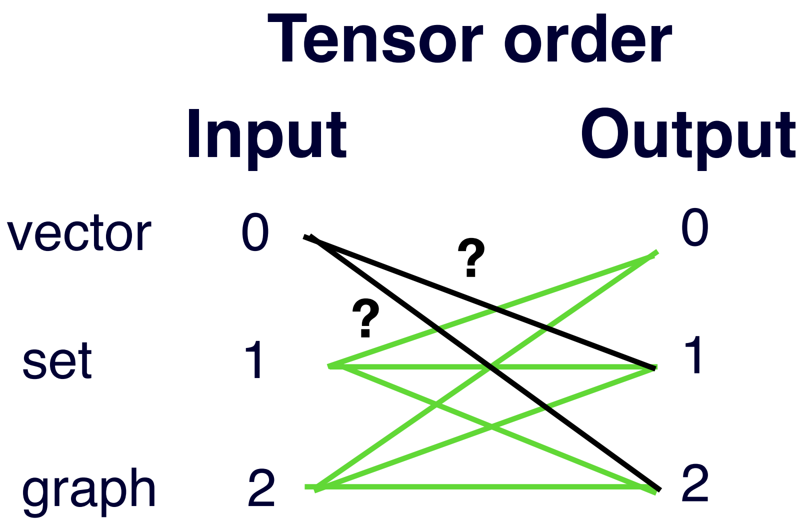
Existing methods for equivariant update

To map a set / graph to another set / graph, any equivariant function can be used. The choice depends on the tensor order of the input and output.

vector: d → order 0

set: $n \times d$ → order 1

edge feat: $n \times n \times d$ → order 2



2 → 0: graph neural network, global pooling

2 → 1: graph neural network, extract node features

1 → 0: Deep sets

2 → 2: graph neural network, extract messages

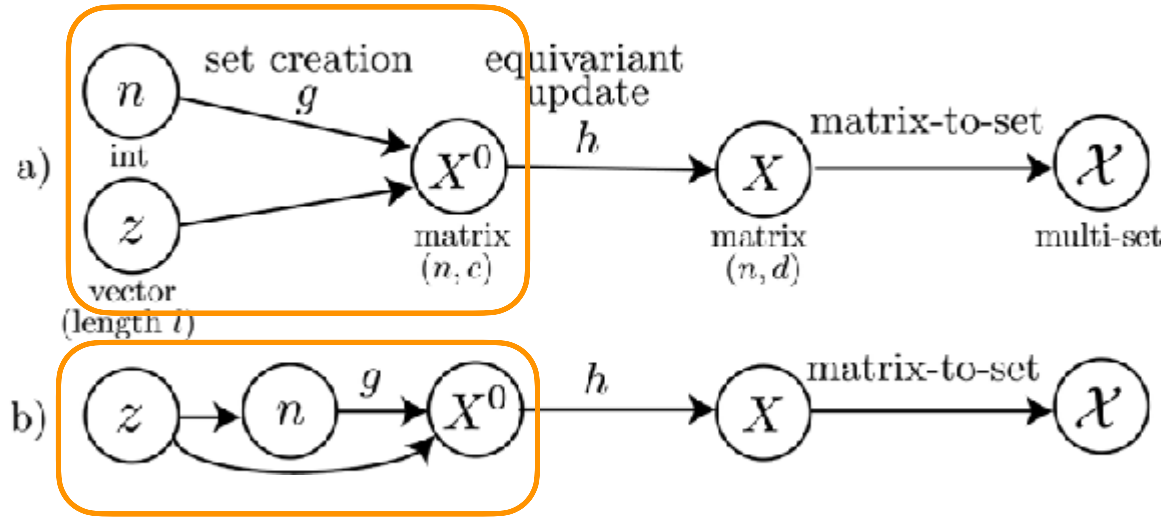
1 → 2: Set2Graph

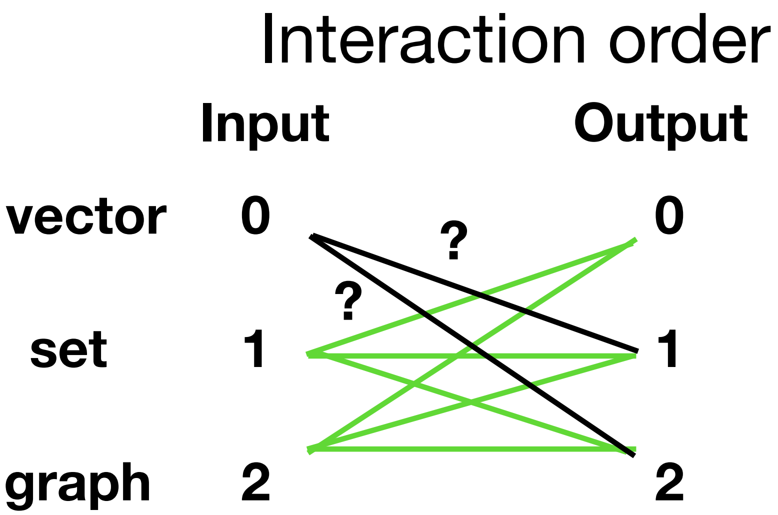
1 → 1: Point Nets or Transformers (1 → 2 → 1)

i → j : *Maron et al. 2019 + Keriven and Peyré 2020*

The challenges of creation:

How to create a set or graph from a vector?





Interaction order

	Input	Output
vector	**0**	**0**
set	**1**	**1**
graph	**2**	**2**

?

?

- We can focus on set creation only
- Requirements:
  - Be able to generate varying numbers of points
  - Some notion of equivariance

What is the right notion of equivariance?

- For discriminative tasks:
  permutation invariance or equivariance $f(\pi.\boldsymbol{X}) = \pi.f(\boldsymbol{X})$

- For generative tasks:

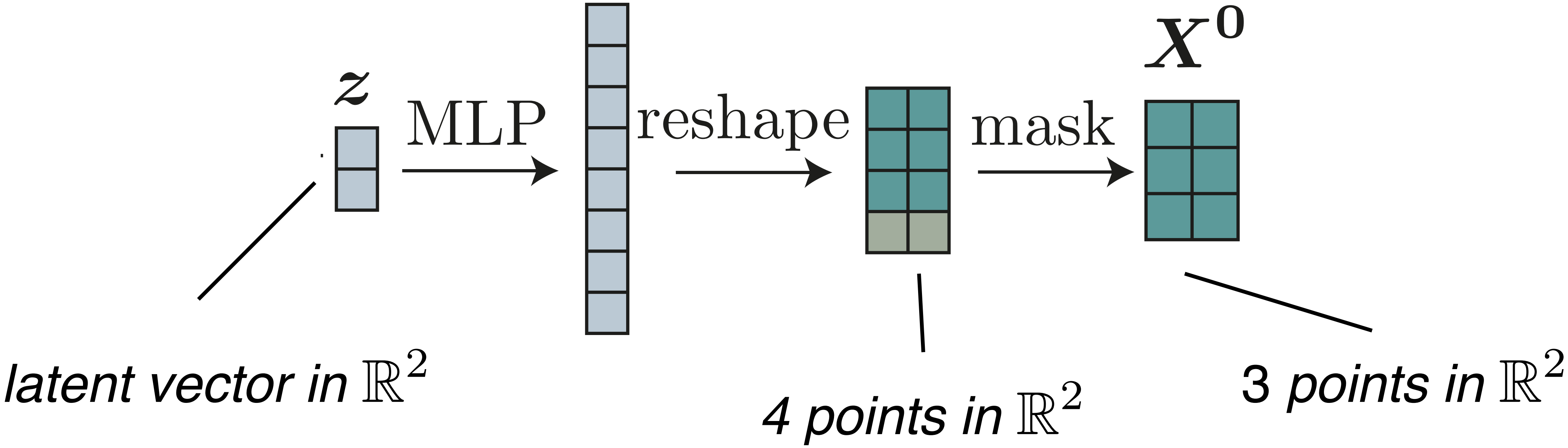
  - $f(\pi.\boldsymbol{z}) = f(\boldsymbol{z}) = \pi.f(\boldsymbol{z}) \implies$ all rows are equal

  - Exchangeability: $\forall \pi,\ \mathbb{P}(f(\boldsymbol{z})) = \mathbb{P}(\pi.f(\boldsymbol{z}))$
    all permutations of the generated sets are equally likely

Is it the right notion?

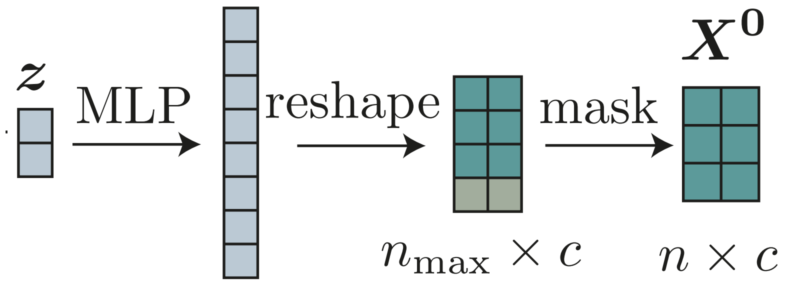
Existing methods for set creation

MLP based creation


$\boldsymbol{z}$
MLP
reshape
mask
$\boldsymbol{X}^0$
latent vector in $\mathbb{R}^2$
4 points in $\mathbb{R}^2$
3 points in $\mathbb{R}^2$

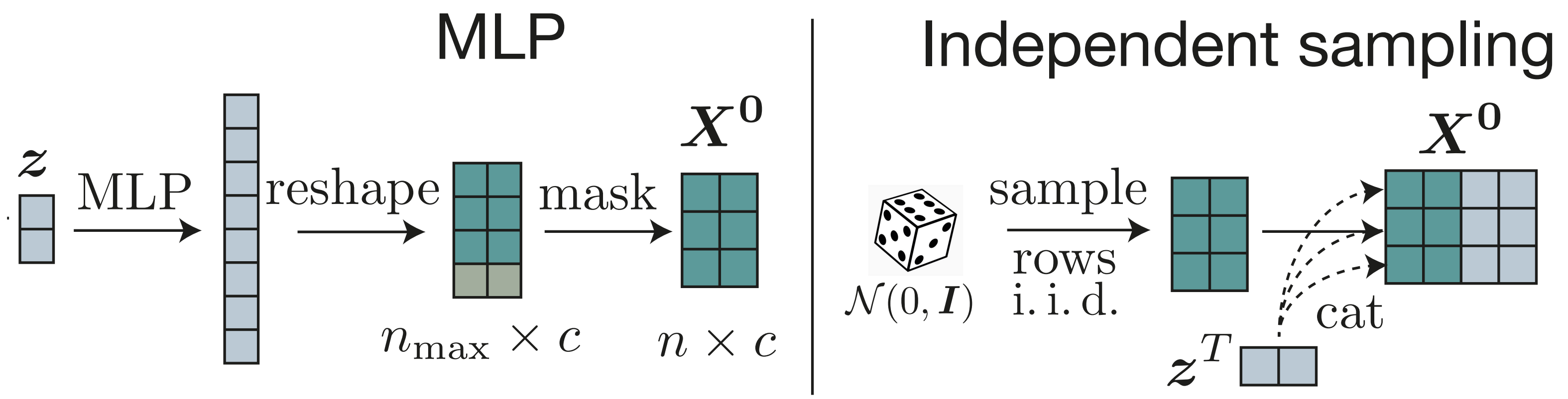
Existing methods for set creation

MLP



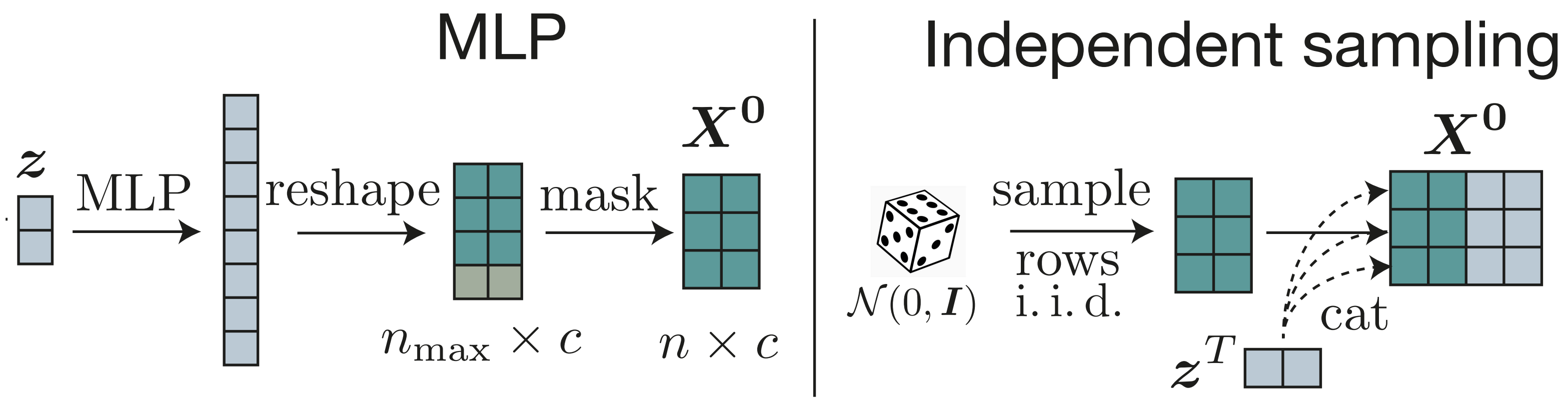
	MLP
Deterministic	✓
Extrapolation ability	✗
No arbitrary masking	✗
Performance	~
Exchangeability	✗

Existing methods for set creation

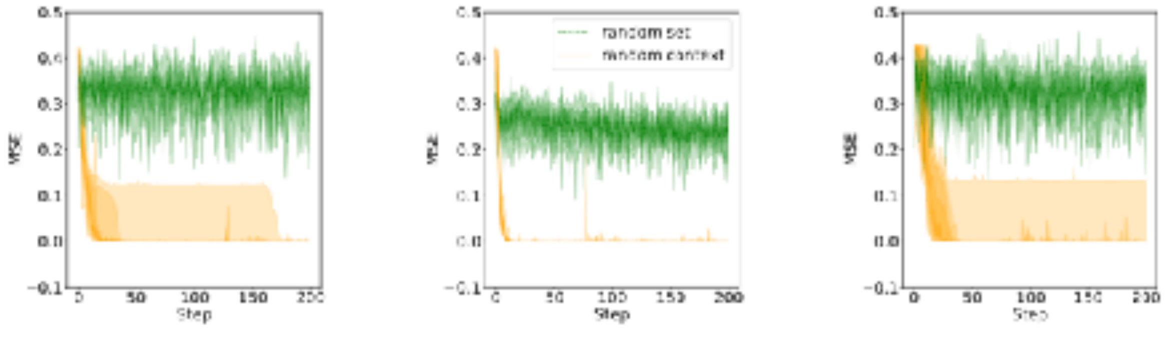


	MLP	**Random i.i.d.**
Deterministic	✓	✗
Extrapolation ability	✗	✓
No arbitrary masking	✗	✓
Performance	~	✗
Exchangeability	✗	✓

Existing methods for set creation



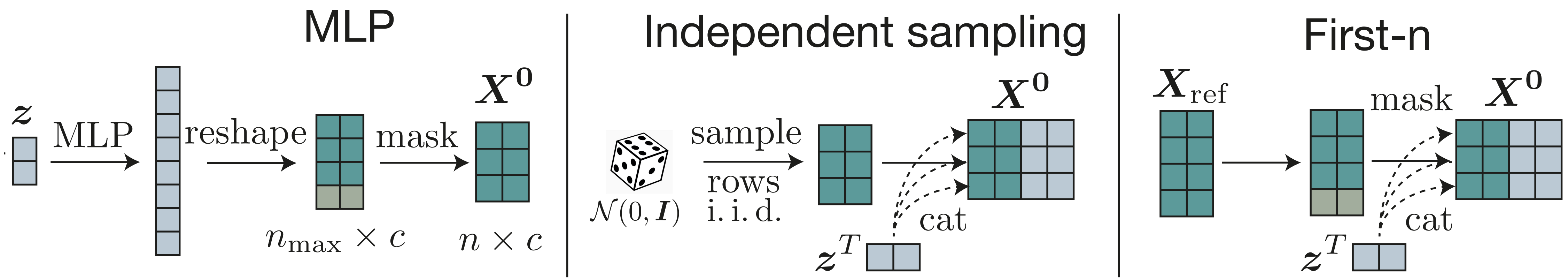
	MLP	Random i.i.d.
Deterministic	✓	✗
Extrapolation ability	✗	✓
No arbitrary masking	✗	✓
Performance	~	✗
Exchangeability	✗	✓



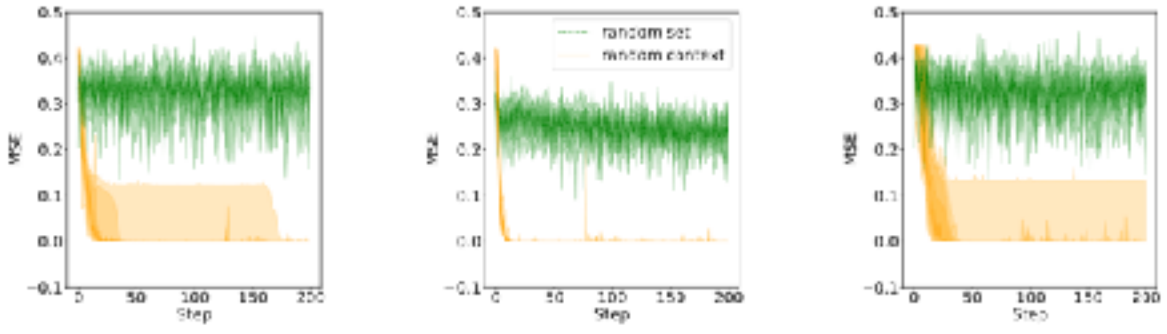
(a) PointMLP (b) PointNetST (c) Self-Attention

GG-GAN, Krawczuk et al. 21

Existing methods for set creation



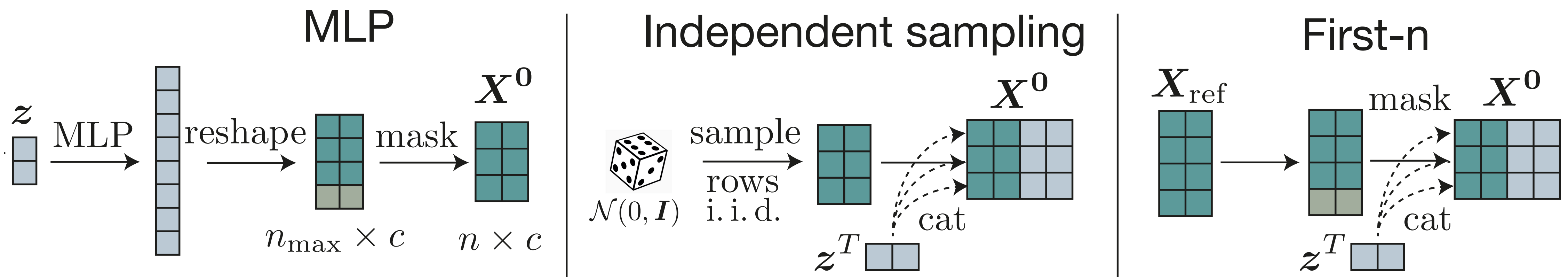
	MLP	Random i.i.d.	First-n
Deterministic	✓	✗	✓
Extrapolation ability	✗	✓	✗
No arbitrary masking	✗	✓	✗
Performance	~	✗	✓
Exchangeability	✗	✓	✗



(a) PointMLP (b) PointNetST (c) Self-Attention

GG-GAN, Krawczuk et al. 21

Existing methods for set creation



	MLP	Random i.i.d.	First-n
Deterministic	✓	✗	✓
Extrapolation ability	✗	✓	✗
No arbitrary masking	✗	✓	✗
Performance	~	✗	✓
Exchangeability	✗	✓	✗

intuitively still seems to be equivariant

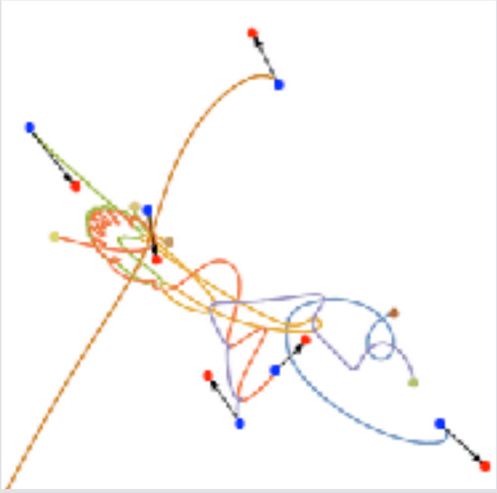
Conclusion of the problem modelling

- No existing method is very satisfying
- The common definition of equivariance is not suited to generative models
- Exchangeability does not seem to correlate with performance

A novel perspective on permutation equivariance

Observation 1

In discriminative tasks, equivariant functions are used with invariant loss functions



example: N-body problem → rotation equivariance

L2 loss is suited, not L1

Observation 2

Equivariant model + invariant loss $\Longrightarrow$ the training dynamics do not depend on the group elements used to represent train data

Use this observation as a definition of equivariance

(F, l)-equivariance

F_Θ : hypothesis class parametrized by $\theta \in \Theta$

l : loss function

$\mathbb{G}$: symmetry group of the problem

Definition

(F_Θ, l) is equivariant to the action of $\mathbb{G}$ if the dynamics of $\theta \in \Theta$ trained with gradient descent on l do not depend on the group elements used to represent the training data

In practice: write one SGD step, check that the parameter updates do not depend on the group elements

Sufficient conditions for (F, l)-equivariance

Discriminative tasks

Equivariant model + invariant loss function $\Longrightarrow$ *equivariance*

Generative tasks

GANs: invariant discriminator + standard GAN loss
$\Longrightarrow$ equivariance

No constraint on the generator

VAEs: invariant encoder +
the loss satisfies
$\Longrightarrow$ equivariance.

$$\forall g \in \mathbb{G}, l(g.\boldsymbol{X}, \hat{\boldsymbol{X}}) = l(\boldsymbol{X}, \hat{\boldsymbol{X}})$$

No constraint on the decoder

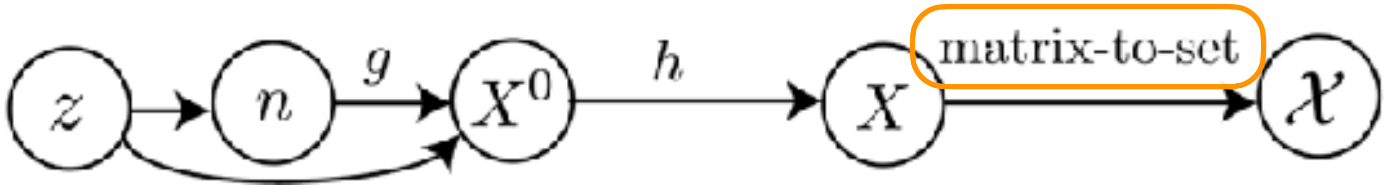
satisfied by common loss functions for sets

(F, l)-equivariance captures common practice in both settings

Why exchangeability is not needed in GANs and VAEs

- $\forall \pi, \ \mathbb{P}(f(\boldsymbol{z})) = \mathbb{P}(\pi . f(\boldsymbol{z}))$ does not appear in the sufficient conditions for equivariance

- Intuition: all permutations of *X* result in the same set



- Explains the good performance of MLP and First-n generation

Back to set creation

	MLP	random i.i.d.	First-n
Deterministic	✓	✗	✓
Extrapolation ability	✗	✓	✗
No arbitrary masking	✗	✓	✗
Performance	~	✗	✓
~~Exchangeability~~	~~✗~~	~~✓~~	~~✗~~

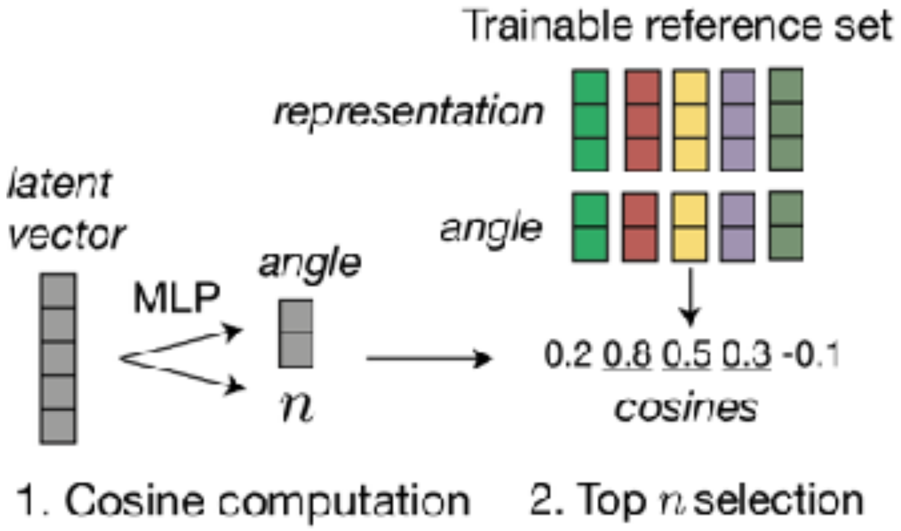
Can we replace the masking of the first rows by a learnable mechanism?

Outline

1. Problem modelling 2. An equivariance perspective **3. Proposition: Top-n creation**

Top-n creation

Idea: select the *n* points that align best with the latent vector

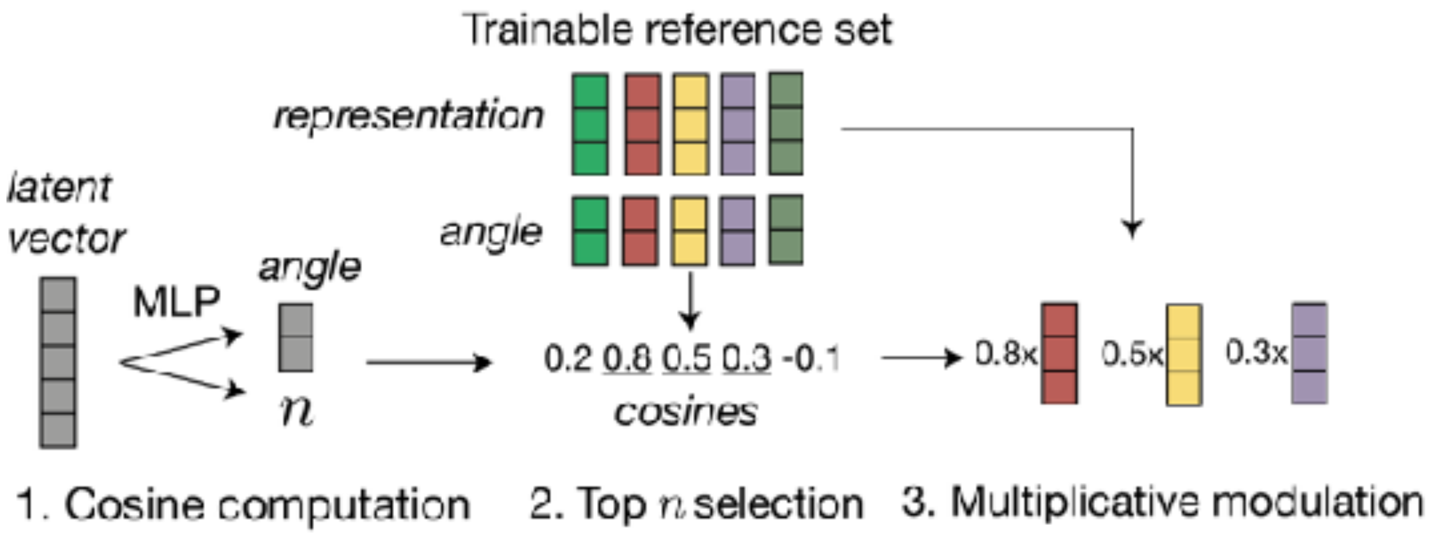


$$\boldsymbol{a} = \mathrm{MLP}_1(\boldsymbol{z})$$
$$\boldsymbol{c} = \boldsymbol{\Phi}\ \boldsymbol{a}\ /\ \mathrm{vec}((||\boldsymbol{\phi}_i||_2)_{1 \leq i \leq n_0})$$
$$\boldsymbol{s} = \mathrm{argsort}_{\downarrow}(\boldsymbol{c})[:\boldsymbol{n}]$$
$$\boldsymbol{X}^0 = \boldsymbol{R}[s]$$

Issue: Top-n selection is not differentiable

Top-n creation

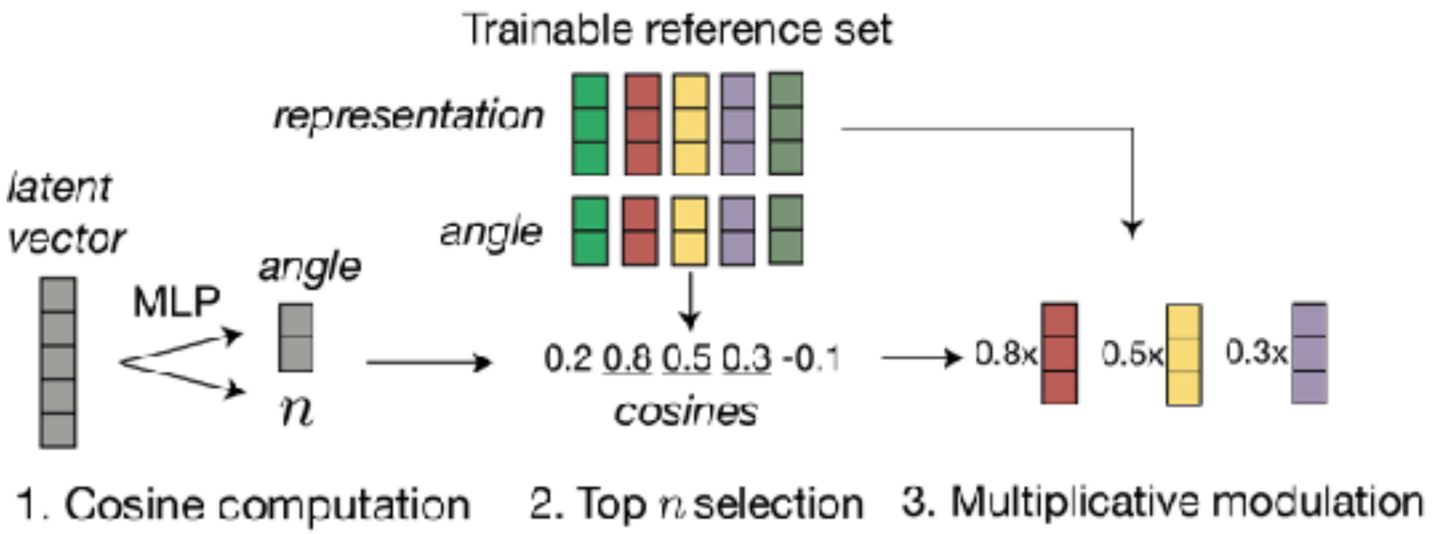
Idea: select the *n* points that align best with the latent vector



$$
\begin{aligned}
\boldsymbol{a} &= \mathrm{MLP}_1(\boldsymbol{z}) \\
\boldsymbol{c} &= \boldsymbol{\Phi}\ \boldsymbol{a}\ /\ \mathrm{vec}((||\boldsymbol{\phi}_i||_2)_{1 \leq i \leq n_0}) \\
\boldsymbol{s} &= \mathrm{argsort}_{\downarrow}(\boldsymbol{c})[:n] \\
\tilde{\boldsymbol{c}} &= \mathrm{softmax}(\boldsymbol{c}[\boldsymbol{s}]) \\
\boldsymbol{X}^0 &= \boldsymbol{R}[\boldsymbol{s}] \odot \tilde{\boldsymbol{c}}\, \boldsymbol{W}_1 + \tilde{\boldsymbol{c}}\, \boldsymbol{W}_2 \\
\boldsymbol{X}^0 &= \boldsymbol{X}^0 \odot \boldsymbol{1}_n\ \boldsymbol{z}^T\ \boldsymbol{W}_3 + \boldsymbol{1}_n\ \boldsymbol{z}^T\ \boldsymbol{W}_4
\end{aligned}
$$

Top-n creation

Idea: select the *n* points that align best with the latent vector



$$
\begin{aligned}
\boldsymbol{a} &= \mathrm{MLP}_1(\boldsymbol{z}) \\
\boldsymbol{c} &= \boldsymbol{\Phi}\ \boldsymbol{a} \;/\; \mathrm{vec}((||\boldsymbol{\phi}_i||_2)_{1 \le i \le n_0}) \\
\boldsymbol{s} &= \mathrm{argsort}_{\downarrow}(\boldsymbol{c})[:n] \\
\tilde{\boldsymbol{c}} &= \mathrm{softmax}(\boldsymbol{c}[\boldsymbol{s}]) \\
\boldsymbol{X}^0 &= \boldsymbol{R}[\boldsymbol{s}] \odot \tilde{\boldsymbol{c}}\, \boldsymbol{W}_1 + \tilde{\boldsymbol{c}}\, \boldsymbol{W}_2 \\
\boldsymbol{X}^0 &= \boldsymbol{X}^0 \odot \boldsymbol{1}_n\ \boldsymbol{z}^T\ \boldsymbol{W}_3 + \boldsymbol{1}_n\ \boldsymbol{z}^T\ \boldsymbol{W}_4
\end{aligned}
$$

- Modulation provides a path in the computational graph for gradients to flow

$$\frac{\partial \boldsymbol{X}^0}{\partial \boldsymbol{a}} \neq 0$$

- How to choose the number of reference points?
  Tradeoff: more points = slower training, often better generalization

Analysis

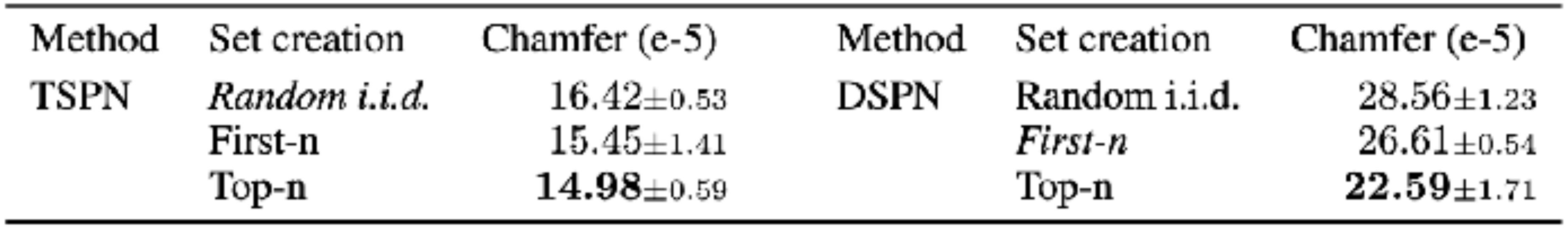
	MLP	random i.i.d.	First-n	Top-n
Deterministic	✓	✗	✓	✓
Extrapolation ability	✗	✓	✗	~
No arbitrary masking	✗	✓	✗	✓
Performance	~	✗	✓	✓

up to n_{ref} points

- *Does the reference set restrict expressivity?*
  No: any set can be approximated by First-n or Top-n + a 2-layer pointwise MLP

Experiments

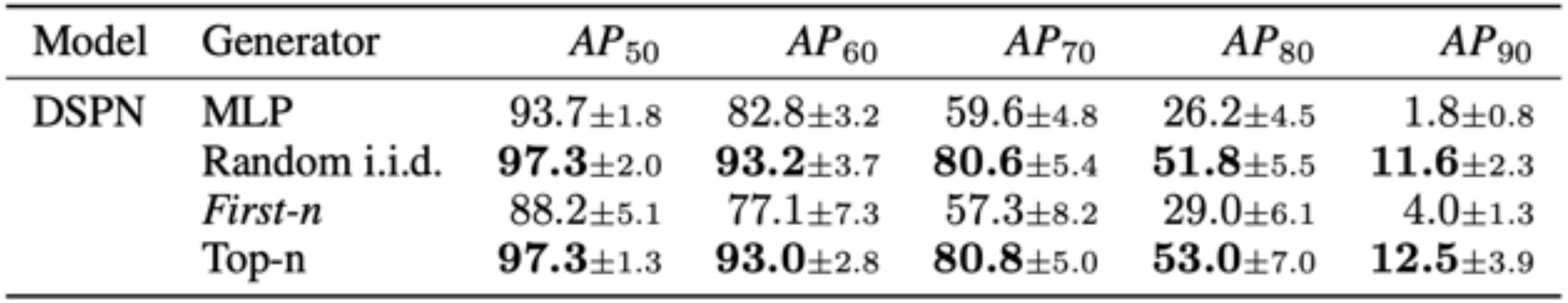
- SetMNIST reconstruction:



Method	Set creation	Chamfer (e-5)	Method	Set creation	Chamfer (e-5)
TSPN	*Random i.i.d.*	16.42±0.53	DSPN	Random i.i.d.	28.56±1.23
	First-n	15.45±1.41		*First-n*	26.61±0.54
	Top-n	**14.98**±0.59		Top-n	**22.59**±1.71

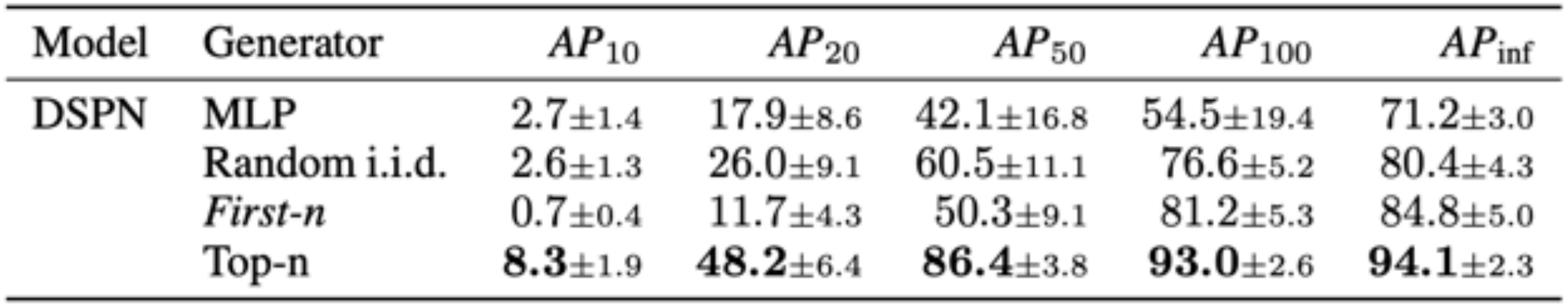
- Object detection on CLEVR:

Bounding box prediction



Model	Generator	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
DSPN	MLP	93.7±1.8	82.8±3.2	59.6±4.8	26.2±4.5	1.8±0.8
	Random i.i.d.	**97.3**±2.0	**93.2**±3.7	**80.6**±5.4	**51.8**±5.5	**11.6**±2.3
	First-n	88.2±5.1	77.1±7.3	57.3±8.2	29.0±6.1	4.0±1.3
	Top-n	**97.3**±1.3	**93.0**±2.8	**80.8**±5.0	**53.0**±7.0	**12.5**±3.9

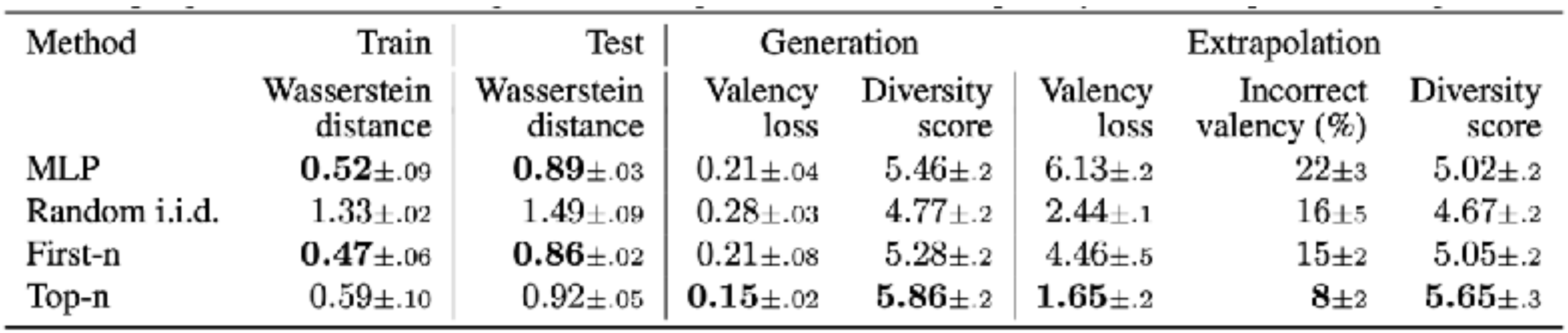
Full state prediction



Model	Generator	AP_{10}	AP_{20}	AP_{50}	AP_{100}	AP_{inf}
DSPN	MLP	2.7±1.4	17.9±8.6	42.1±16.8	54.5±19.4	71.2±3.0
	Random i.i.d.	2.6±1.3	26.0±9.1	60.5±11.1	76.6±5.2	80.4±4.3
	First-n	0.7±0.4	11.7±4.3	50.3±9.1	81.2±5.3	84.8±5.0
	Top-n	**8.3**±1.9	**48.2**±6.4	**86.4**±3.8	**93.0**±2.6	**94.1**±2.3

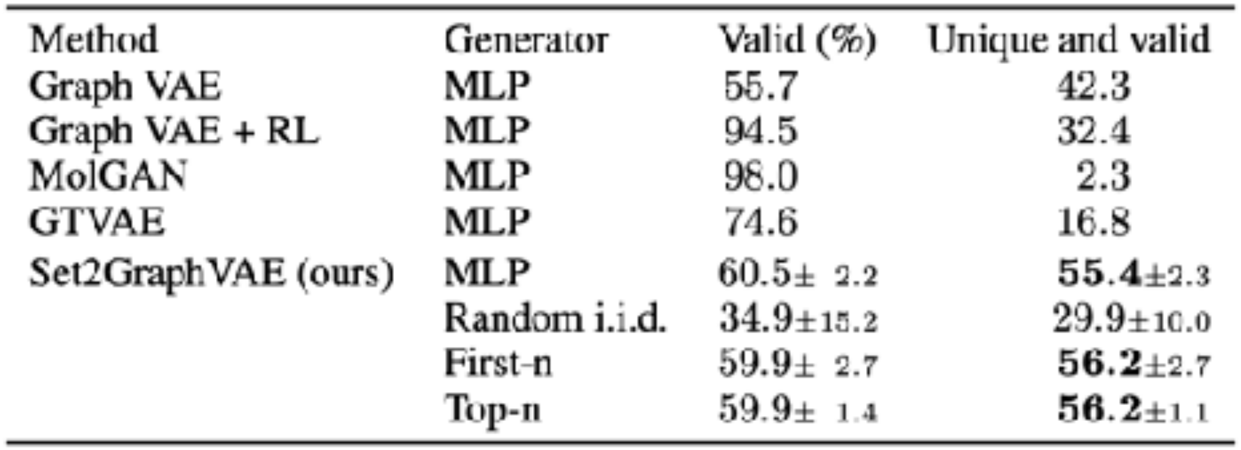
Experiments (2)

- Synthetic molecule-like dataset in 3d



Method	Train	Test	Generation		Extrapolation		
	Wasserstein distance	Wasserstein distance	Valency loss	Diversity score	Valency loss	Incorrect valency (%)	Diversity score
MLP	**0.52**±.09	**0.89**±.03	0.21±.04	5.46±.2	6.13±.2	22±3	5.02±.2
Random i.i.d.	1.33±.02	1.49±.09	0.28±.03	4.77±.2	2.44±.1	16±5	4.67±.2
First-n	**0.47**±.06	**0.86**±.02	0.21±.08	5.28±.2	4.46±.5	15±2	5.05±.2
Top-n	0.59±.10	0.92±.05	**0.15**±.02	**5.86**±.2	**1.65**±.2	**8**±2	**5.65**±.3

- Molecule generation on QM9



Method	Generator	Valid (%)	Unique and valid
Graph VAE	MLP	55.7	42.3
Graph VAE + RL	MLP	94.5	32.4
MolGAN	MLP	98.0	2.3
GTVAE	MLP	74.6	16.8
Set2GraphVAE (ours)	MLP	60.5± 2.2	**55.4**±2.3
	Random i.i.d.	34.9±15.2	29.9±10.0
	First-n	59.9± 2.7	**56.2**±2.7
	Top-n	59.9± 1.4	**56.2**±1.1

Conclusion

- (F, I)-equivariance captures in a single notion common practice both for discriminative and generative tasks
- Exchangeability does not seem to be useful for equivariance in GANs and VAEs
- Top-n creation is a simple and effective tool to generate sets from a latent vector

References

- Maron et al., Invariant and equivariant graph networks, *ICLR 2019*
- Keriven and Peyré, Universal invariant and equivariant graph networks, *Neurips 2020*
- Krawczuk et al., GG-GAN: A geometric graph generative adversarial network, *Under review*
- Gao and Ji, Graph U-Nets, *ICML 2019*

Questions?

Clement Vignac

`clement.vignac@epfl.ch`
Signal Processing Laboratory
(LTS4)

Vignac and Frossard, Top-N: Equivariant set and graph generation without exchangeability
ICLR 2022

Relationship with Blondel et al. 2020

- Top-n is based on a hard selection process $\boldsymbol{s} = \mathrm{argsort}_{\downarrow}(\boldsymbol{c})[:n]$

- Soft alternatives should be possible
→ each selected point would be a linear combination of references points

- Output of soft rank: an approximation of the rank of each point

Reference point:	**x_1**	**x_2**	**x_3**
Soft rank:	1.2	2.7	2.1

Soft-rank alternatives to Top-n should be possible, but would not be a straightforward application of Blondel et al. 2020

Blondel et al., Fast differentiable sorting and ranking, *ICML 2020*

Why does i.i.d. generation perform poorly?

- Intuition: two sources of randomness ⇒ reconstruction is more difficult
- VAEs maximize the evidence lower bound (ELBO)

$$\mathcal{L}(\boldsymbol{X}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{X})}[\log p_\theta(\boldsymbol{X}, \boldsymbol{z}) - \log q_\phi(\boldsymbol{z}|\boldsymbol{X})] \leq p_\theta(\boldsymbol{X})$$

- The decoder is stochastic $\Longrightarrow$ $\log p_\theta(\boldsymbol{X}, \boldsymbol{z})$ cannot be computed in close form
- Jensen: $\mathcal{L}(\boldsymbol{X}) \geq E_{\boldsymbol{X}^0, \boldsymbol{z}} \left[\log p_\theta(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{X}^0) - \log q_\phi(\boldsymbol{z}|\boldsymbol{X})\right]$
- “Random i.i.d. generation maximises the ELBO of the ELBO”

Common loss functions for sets

- Chamfer's distance

$$d_{\text{Cham}} = \sum_{1\leq i\leq n} \min_{j\leq n'} ||\boldsymbol{x}_i - \boldsymbol{x}'_j||_2^2 + \sum_{1\leq j\leq n'} \min_{i\leq n} ||\boldsymbol{x}_i - \boldsymbol{x}'_j||_2^2$$

- Wasserstein distance

$$d_{\mathcal{W}_2} = \inf_{u\in\{\Gamma(\boldsymbol{X},\boldsymbol{X}')\}} \sum_{\substack{1\leq i\leq n\\ 1\leq j\leq n'}} u(\boldsymbol{x}_i, \boldsymbol{x}'_j)\; \|\boldsymbol{x}_i - \boldsymbol{x}'_j\|_2^2$$