Convolutional Neural Networks for Signals on Graphs

Vincent Gripon
joint work with Carlos Lassance, Bastien Pasdeloup
and Jean-Charles Vialatte

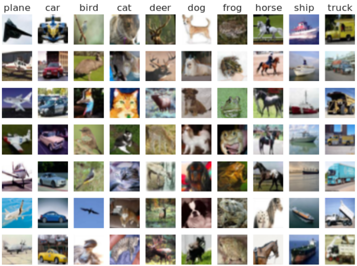
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Sept. 4th, 2018

Context

- Graph Convolutional Neural Networks may refer to:
  - Graph learning (graph embedding...),
  - Node classification (semi-supervised learning...),
  - **Signal on graphs processing (irregular domains...).**
- Motivation:

CIFAR-10 dataset



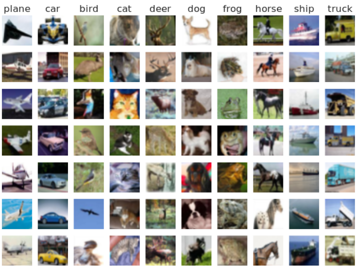
Error rate

- Without structure (MLP): 31%,
- With structure (CNN): 4%.

Context

- Graph Convolutional Neural Networks may refer to:
  - Graph learning (graph embedding. . .),
  - Node classification (semi-supervised learning. . .),
  - **Signal on graphs processing (irregular domains. . .).**
- Motivation:

CIFAR-10 dataset



Error rate

- Without structure (MLP): 31%,
- With structure (CNN): 4%.

Convolutional Neural Networks are defined using the underlying (often 2D) vector space. But how to extend to more complex domains with no explicit underlying vector space?

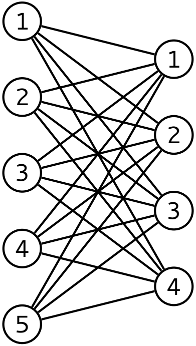
Fully connected layers

$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) \, ,$$

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \\ w_{51} & w_{52} & w_{53} & w_{54} \end{pmatrix}$$

Fully connected layers

$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) ,$$



$$\begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \\ w_{51} & w_{52} & w_{53} & w_{54} \end{pmatrix}$$

Convolutional layers

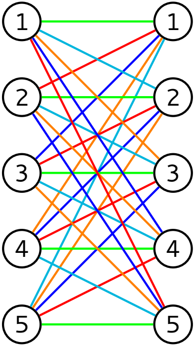
$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) ,$$

$$\begin{pmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \\ w_5 & w_1 & w_2 & w_3 & w_4 \\ w_4 & w_5 & w_1 & w_2 & w_3 \\ w_3 & w_4 & w_5 & w_1 & w_2 \\ w_2 & w_3 & w_4 & w_5 & w_1 \end{pmatrix}$$

Convolutional layers

$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) \ ,$$

Weight sharing,

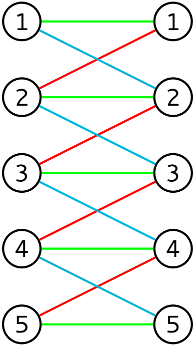


$$\begin{pmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \\ w_5 & w_1 & w_2 & w_3 & w_4 \\ w_4 & w_5 & w_1 & w_2 & w_3 \\ w_3 & w_4 & w_5 & w_1 & w_2 \\ w_2 & w_3 & w_4 & w_5 & w_1 \end{pmatrix}$$

Convolutional layers

$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) \, ,$$

Weight sharing, localization,

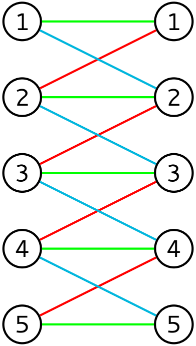


$$\begin{pmatrix} w_1 & w_2 & 0 & 0 & 0 \\ w_5 & w_1 & w_2 & 0 & 0 \\ 0 & w_5 & w_1 & w_2 & 0 \\ 0 & 0 & w_5 & w_1 & w_2 \\ 0 & 0 & 0 & w_5 & w_1 \end{pmatrix}$$

Convolutional layers

$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) ,$$

Weight sharing, localization, feature maps,

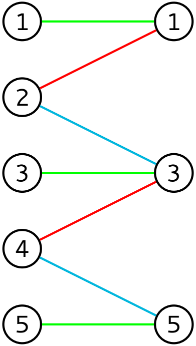


$$\begin{pmatrix} w_1 & w_2 & & & \\ w_5 & w_1 & w_2 & & \\ & w_5 & w_1 & w_2 & \\ & & w_5 & w_1 & w_2 \\ & & & w_5 & w_1 \end{pmatrix}$$

Convolutional layers

$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) \ ,$$

Weight sharing, localization, feature maps, pooling/strides,

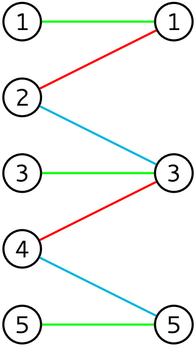


$$\begin{pmatrix} w_1 & w_2 & & & \\ & w_5 & w_1 & w_2 & \\ & & & w_5 & w_1 \end{pmatrix}$$

Convolutional layers

$$\mathbf{y} = f(W\mathbf{x} + \mathbf{b}) \ ,$$

Weight sharing, localization, feature maps, pooling/strides, data-augmentation.



$$\begin{pmatrix} w_1 & w_2 & & & \\ & w_5 & w_1 & w_2 & \\ & & & w_5 & w_1 \end{pmatrix}$$

Existing approaches

Spectral approaches

- Use Fourier transform on graphs (eigenspace of Laplacian of the graph),
- Convolution = point-wise multiplication in Fourier domain,
- Learn Fourier domain coefficients of convolutions.

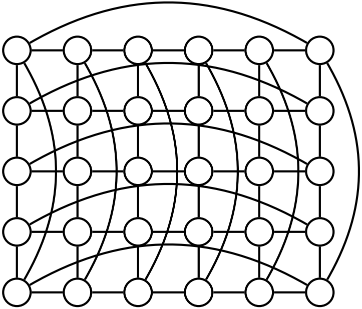
Vertex-domain approaches

- Design weight-sharing in the vertex domain,
- Use heuristics to map neighbors of vertices,
- **Design translations in the vertex domain.**

Question: can we generalize CNNs to signals on graphs?

1. **Sanity check:** should perform as well as CNNs on standard signals,
2. **Generalization:** should improve performance compared to MLP on irregular signals.

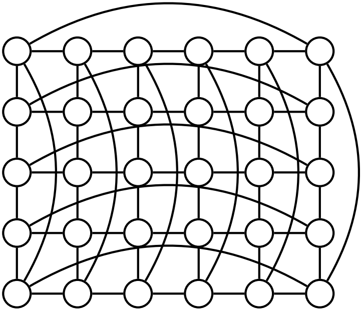
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

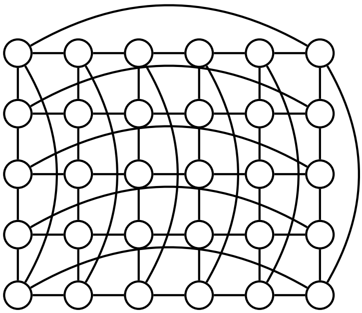
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

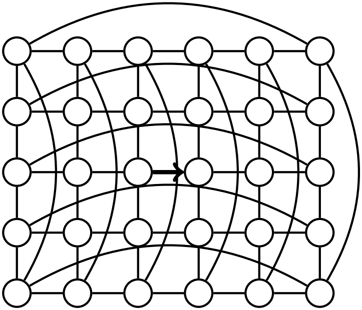
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

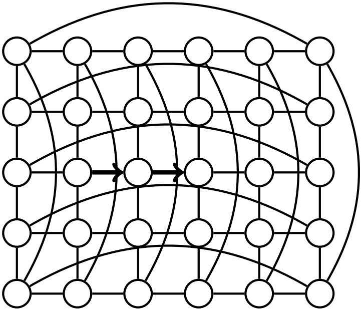
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

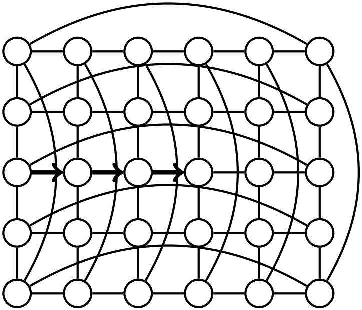
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

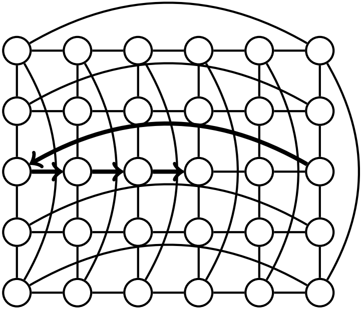
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

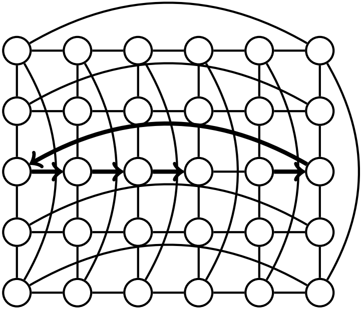
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

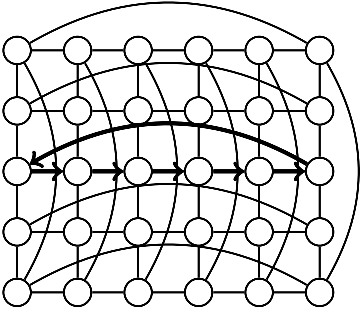
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

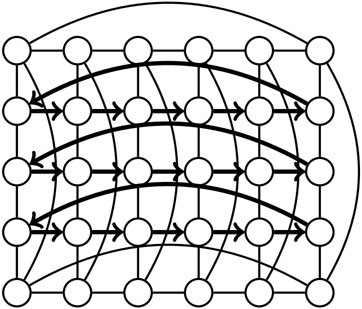
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

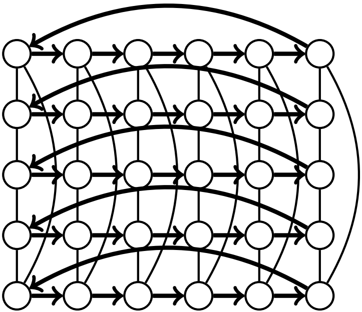
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

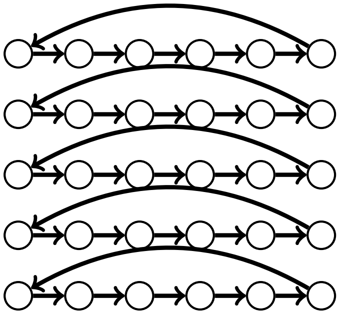
Defining translations in the vertex domain



Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

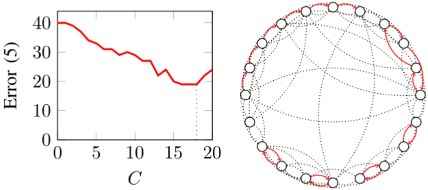
Defining translations in the vertex domain



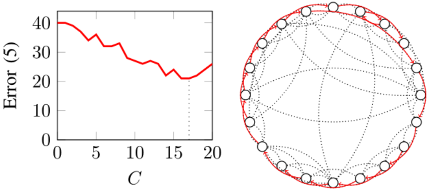
Define translations as functions that are such that:

- One-to-one on vertices,
- Neighbors are associated with neighbors,
- Image of a vertex is one of its neighbors.

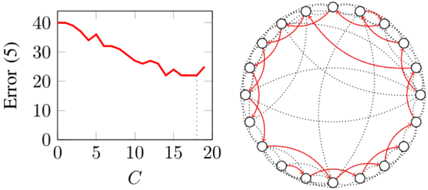
Experiments on small-world nets



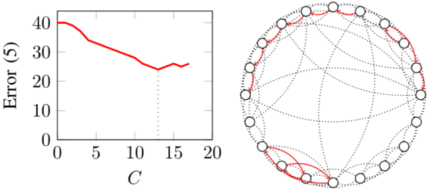
(a) $P = 0.1$, first pseudo-translation found for $C = 18$.



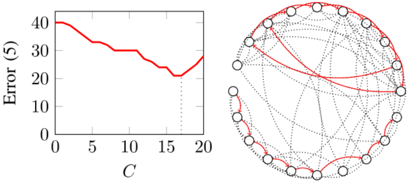
(b) $P = 0.1$, second pseudo-translation found for $C = 17$.



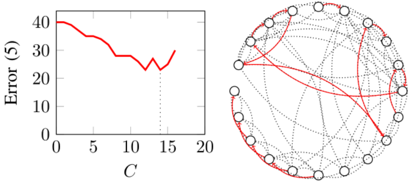
(c) $P = 0.1$, third pseudo-translation found for $C = 18$.



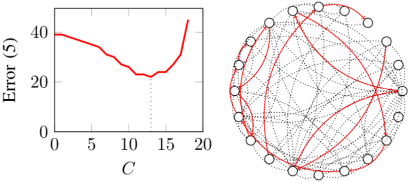
(d) $P = 0.1$, fourth pseudo-translation found for $C = 13$.



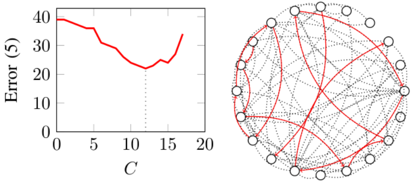
(a) $P = 0.3$, first pseudo-translation found for $C = 17$.



(b) $P = 0.3$, third pseudo-translation found for $C = 14$. The second pseudo-translation found was the inverse of the first.



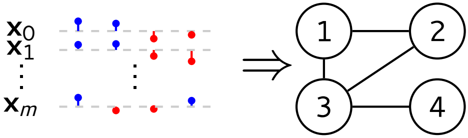
(c) $P = 0.5$, first pseudo-translation found for $C = 13$.



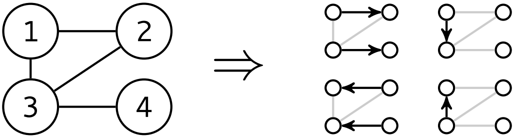
(d) $P = 0.5$, second pseudo-translation found for $C = 12$.

From translations to Conv. Nets

Step 0 : infer a graph

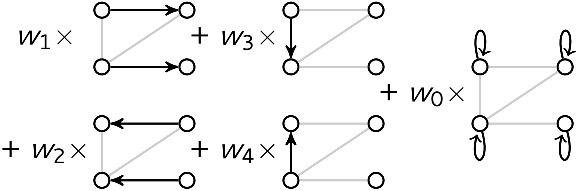


Step 1: infer translations

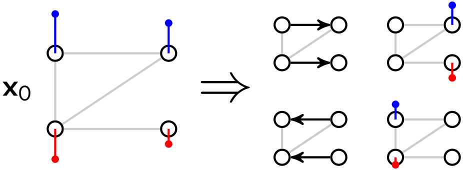


From translations to Conv. Nets

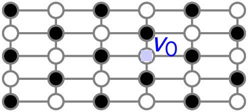
Step 2: design convolution weight-sharing

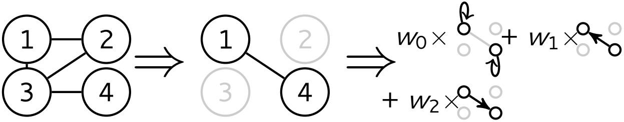


Step 3: design data-augmentation

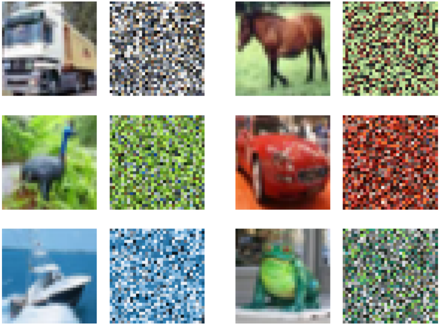


Step 4: design graph subsampling and convolution weight-sharing





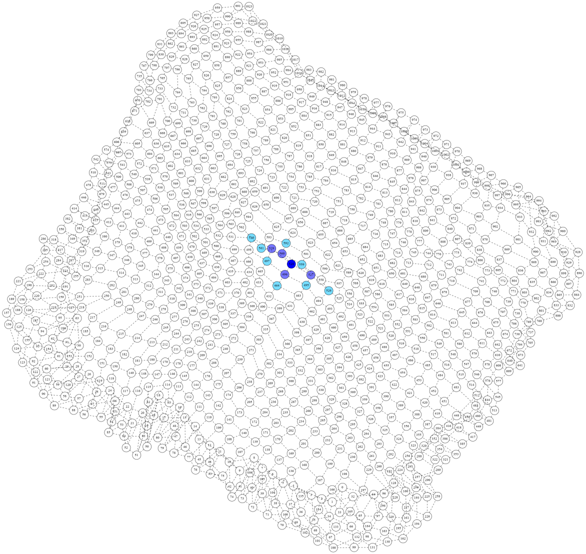
Sanity check: scrambled CIFAR-10 experiments



- 10 categories,
- 60’000 examples,
- 10’000 tests.

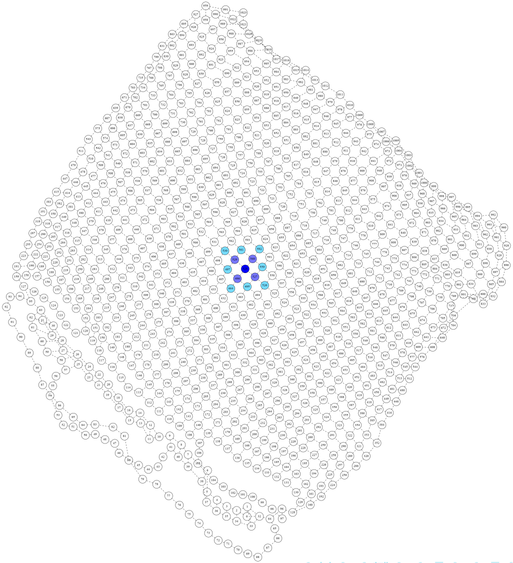
Graph inference

Scrambled CIFAR-10 thresholded empirical covariance matrix



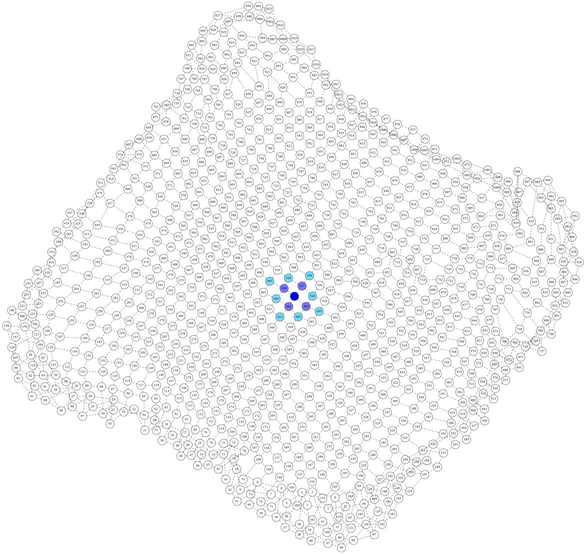
Graph inference

Scrambled CIFAR-10 smooth (Kalofolias et al.)



Graph inference

Scrambled CIFAR-10 smooth and stationary



Results

Scrambled CIFAR-10 (Toy architecture, no DA)

CNN	Cov.	Smooth	Stationary	U-CNN	MLP
56.74%	85.03%	85.33%	85.81%	85.98%	69%

Results

Scrambled CIFAR-10 (Toy architecture, no DA)

CNN	Cov.	Smooth	Stationary	U-CNN	MLP
56.74%	85.03%	85.33%	85.81%	85.98%	69%

Scrambled CIFAR-10 (ResNet)

Support	MLP	U-CNN	Grid Graph		Covariance Graph	
			Defferard et al, 2016	Proposed	Proposed	Pasdeloup et al, 2017
Full Data Augmentation	78.62%	**93.80%**	85.13%	93.94%	92.57%	—
Data Augmentation - Flip	—	92.73%	84.41%	92.94%	91.29%	—
Graph Data Augmentation	—	—	—	92.81%	**91.07%**	—
None	69.62%	87.78%	—	88.83%	85.88%	82.52%

Results

Scrambled CIFAR-10 (Toy architecture, no DA)

CNN	Cov.	Smooth	Stationary	U-CNN	MLP
56.74%	85.03%	85.33%	85.81%	85.98%	69%

Scrambled CIFAR-10 (ResNet)

Support	MLP	U-CNN	Grid Graph		Covariance Graph	
			Defferard et al, 2016	Proposed	Proposed	Pasdeloup et al, 2017
Full Data Augmentation	78.62%	**93.80%**	85.13%	93.94%	92.57%	—
Data Augmentation - Flip	—	92.73%	84.41%	92.94%	91.29%	—
Graph Data Augmentation	—	—	—	92.81%	**91.07%**	—
None	69.62%	87.78%	—	88.83%	85.88%	82.52%

Pines fMRI dataset

Graph	None		Neighborhood Graph	
Method	MLP	CNN (kernel 1x1)	Defferard et al, 2016	Proposed
Accuracy	82.62%	84.30%	82.80%	**85.08%**

Conclusion

Conclusion

- Extension of regular CNNs to irregular domains,
- Promising results on toy datasets,
- Comprehensible technique with lots of theorems.

Ongoing/future work

- Challenging datasets (including highly nonregular),
- Weighted graphs to infer translations,
- Computation bottleneck of finding translations.

https://github.com/brain-bzh/MCNN
"Characterization and inference of graph diffusion processes from observations of stationary signals"
"Translations on graphs with neighborhood preservation"
"Matching Convolutional Neural Networks without Priors about Data"

Conclusion

Conclusion

- Extension of regular CNNs to irregular domains,
- Promising results on toy datasets,
- Comprehensible technique with lots of theorems.

Ongoing/future work

- Challenging datasets (including highly nonregular),
- Weighted graphs to infer translations,
- Computation bottleneck of finding translations.

https://github.com/brain-bzh/MCNN

“Characterization and inference of graph diffusion processes from observations of stationary signals”

“Translations on graphs with neighborhood preservation”

“Matching Convolutional Neural Networks without Priors about Data”

Conclusion

Conclusion

- Extension of regular CNNs to irregular domains,
- Promising results on toy datasets,
- Comprehensible technique with lots of theorems.

Ongoing/future work

- Challenging datasets (including highly nonregular),
- Weighted graphs to infer translations,
- Computation bottleneck of finding translations.

https://github.com/brain-bzh/MCNN

"Characterization and inference of graph diffusion processes from observations of stationary signals"

"Translations on graphs with neighborhood preservation"

"Matching Convolutional Neural Networks without Priors about Data"

Conclusion

Conclusion

- Extension of regular CNNs to irregular domains,
- Promising results on toy datasets,
- Comprehensible technique with lots of theorems.

Ongoing/future work

- Challenging datasets (including highly nonregular),
- Weighted graphs to infer translations,
- Computation bottleneck of finding translations.

https://github.com/brain-bzh/MCNN

"Characterization and inference of graph diffusion processes from observations of stationary signals"

"Translations on graphs with neighborhood preservation"

"Matching Convolutional Neural Networks without Priors about Data"

Conclusion

Conclusion

- Extension of regular CNNs to irregular domains,
- Promising results on toy datasets,
- Comprehensible technique with lots of theorems.

Ongoing/future work

- Challenging datasets (including highly nonregular),
- Weighted graphs to infer translations,
- Computation bottleneck of finding translations.

https://github.com/brain-bzh/MCNN

"Characterization and inference of graph diffusion processes from observations of stationary signals"

"Translations on graphs with neighborhood preservation"

"Matching Convolutional Neural Networks without Priors about Data"

Conclusion

Conclusion

- Extension of regular CNNs to irregular domains,
- Promising results on toy datasets,
- Comprehensible technique with lots of theorems.

Ongoing/future work

- Challenging datasets (including highly nonregular),
- Weighted graphs to infer translations,
- Computation bottleneck of finding translations.

https://github.com/brain-bzh/MCNN

"Characterization and inference of graph diffusion processes from observations of stationary signals"

"Translations on graphs with neighborhood preservation"

"Matching Convolutional Neural Networks without Priors about Data"

Conclusion

Conclusion

- Extension of regular CNNs to irregular domains,
- Promising results on toy datasets,
- Comprehensible technique with lots of theorems.

Ongoing/future work

- Challenging datasets (including highly nonregular),
- Weighted graphs to infer translations,
- Computation bottleneck of finding translations.

https://github.com/brain-bzh/MCNN
"Characterization and inference of graph diffusion processes from observations of stationary signals"
"Translations on graphs with neighborhood preservation"
"Matching Convolutional Neural Networks without Priors about Data"