Matching Convolutional Neural Networks without Priors about Data

Carlos Eduardo Rosar Kos Lassance*, Jean-Charles Vialatte*† and Vincent Gripon*

*IMT Atlantique
{carlos.rosarkoslassance, jc.vialatte, vincent.gripon}@imt-atlantique.fr
†Cityzen Data

Abstract—We propose an extension of Convolutional Neural Networks (CNNs) to graph-structured data, including strided convolutions and data augmentation on graphs. Our method matches the accuracy of state-of-the-art CNNs when applied on images, without any prior about their 2D regular structure. On fMRI data, we obtain a significant gain in accuracy compared with existing graph-based alternatives.

I. INTRODUCTION

Convolutional Neural Networks (CNNs) [1] have been able to surpass traditional machine learning methods in various image based tasks [2, 3]. This is possible as they exploit the learning capabilities of deep neural networks while also taking advantage of the intrinsic regular 2D structure of the data. But when data lacks regular structure [4], there is no natural notion of convolutions, stride/pooling or data augmentation. Such irregularities occur in various domains covering social networks to neuroscience, internet of things, citation graphs, point cloud manifolds… The question of developing solutions that are counterparts of CNNs in irregular domains has recently been a very active field of research.

In this paper we introduce a method that extends CNNs to irregular domains. Contrary to many alternative works, we ensure that our proposed methodology matches the performance of CNNs when applied to regular domains, even without knowledge of the underlying structure. To that end, we infer a graph to represent the topology of the data. From this graph, we infer translations. The weight-sharing schemes of our proposed convolutional layers are then defined based on those translations, as well as data-augmentation and stride.

At the end of the process, the obtained architecture is very similar to a traditional CNN and can thus be trained using the same routines and libraries, and equivalent computational and memory footprints. We first perform experiments on the CIFAR-10 dataset without knowledge about the fact it is made of images. We show that our method is able to reach performance similar to state-of-art CNNs, thus implying that – at least for regular domains – it allows to completely leverage the underlying structure. Then, we perform experiments on an irregular neuroscience dataset and demonstrate a gain in performance compared with completely unstructured deep learning methods and alternative graph-based CNNs.

II. RELATED WORK

Deep learning on graphs can refer to three distinct problems: classification of graphs, of nodes in a graph, or of signals on graphs. In this paper, we are interested only in the latter task that is to leverage the graph structure of signals in deep learning models, by redefining the convolutional layer. Such methods have already been proposed in the literature. We distinguish two categories of solutions.

In the first category, convolution is defined as pointwise multiplication in the spectral domain of the graph, which is defined using the Laplace-Beltrami operator [5]. This method originated the first spectral graph CNNs [6, 7]. An approximation of the spectral graph convolution using Chebychev polynomials has been proposed [8], and has the advantage to be both faster and localized in the vertex domain. Another variant with Cayley polynomials [9] also localizes the convoluted filter in the spectral domain.

In the second category, convolution is defined directly in the vertex domain. These works were originally motivated by chemistry datasets [10, 11]. Convolution is defined as a function of the kernel weights and neighboring vertices (the receptive field), usually based on dot products. As such, it retains the property of being localized and of sharing weights. But there remains the need to specify how the shared weights are allocated in this receptive field [12]. This allocation can depend on an arbitrary order [13], on the number of hops [14, 15], on both vertices and their neighbors [16, 17], on another learned kernel [18], on an attention mechanism [19], on pattern identification [20], or on translation identification [21]. All these methods also differ in the function that maps the receptive fields and the weight kernel to the neuron’s outputs. But in the end, these definitions overlap. That is why some authors have proposed unified frameworks [22].

We tackle another point. Given a dataset with no structure between the features of the input vectors, our goal is to demonstrate that we can still define meaningful convolutional, stride/pooling layers. The first step to determine whether the results obtained on unstructured data is satisfactory or not is to stress it on regular data while disregarding its structure. We deal with this step on image datasets. Even though some previous works match the performance of CNNs on image datasets, ours is the first that can do it without structure prior.
III. Methodology

Our method is based on [21], where the authors have introduced a way to infer a graph from training signals, then translations from the obtained graph to design ad-hoc CNNs. We extend this approach and design strided convolutions along graph downscaling, data augmentation and convolutions on downscaled graphs. Figure 1 depicts the proposed method.

![Figure 1. Outline of the proposed method](image)

A. Background

Define a graph $G = (V, E)$ with $V$ the set of vertices, and $E \subseteq \binom{V}{2}$ the set of edges. We suppose the graph is connected, as conversely the process can be applied to each connected component of $G$. We denote by $d$ the max degree of the graph and $n = |V|$ the number of vertices.

The authors of [21] propose to inductively define translations as functions from vertices to vertices as follows:

**Definition 1: Candidate-translation**

A candidate-translation is a function $\phi : U \rightarrow V$, where $U \subset V$ and such that:

- $\phi$ is injective: $\forall v, v' \in U, \phi(v) = \phi(v') \Rightarrow v = v'$,
- $\phi$ is edge-constrained: $\forall v \in U, (v, \phi(v)) \in E$,
- $\phi$ is strongly neighborhood-preserving: $\forall v, v' \in U, (v, v') \in E \Leftrightarrow (\phi(v), \phi(v')) \in E$.

The cardinal $|V - U|$ is called the loss of $\phi$. Two candidate-translations $\phi$ and $\phi'$ are said to be aligned if $\exists v \in V, \phi(v) = \phi'(v)$. We define $N_i(v)$ as the set of vertices that are at most $i$-hop away from a vertex $v \in V$.

**Definition 2: Translation**

A translation in a graph $G$ is a candidate-translation such that there is no aligned translation with a strictly smaller loss, or is the identity function.

Note that if the graph is a 2D grid, obtained translations are exactly natural translations on images [23].

**Definition 3: Local translation**

A local translation of center $v \in V$ is a translation in the subgraph of $G$ induced by $N_2(v)$, that has $v$ in its definition domain.

As local translations can’t be used to design data augmentation and convolutions on downscaled graphs, we also design proxies to global translations.

**Definition 4: Proxy-translations**

A family of proxy-translations $(\psi_p)_{p=0,\ldots,k-1}$ initialized by $v_0 \in V$ is defined algorithmically as follows:

1) We place an indexing kernel on $N_1(v_0)$ i.e. $N_1(v_0) = \{v_0, v_1, \ldots, v_{k-1}\}$ with $\forall p, \psi_p(v_0) = v_p$.
2) We move this kernel using each local translation $\phi$ of center $v_0$: $\forall p, \psi_p(\phi(v_0)) = \phi(v_p)$.
3) We repeat 2) from each new center reached until saturation. If a center is being reached again, we keep the indexing that minimizes the sum of losses of the local translations that has lead to it.

B. Efficiently Finding Translations

Finding translations is an NP-complete problem [24], such that for large graphs the method is not suitable. In order to break down complexity, the authors of [21] propose to search for local translations. They also introduce approximate translations which we omit for the sake of simplicity, but the description would be similar. We describe in three steps how we efficiently find proxy-translations.

**First step: finding local translations**

For each vertex $v \in G$, we identify all local translations using a brute-force algorithm. This process requires finding
We stop when the kernel has been moved everywhere in an indexing of at most \( O(1) \). This process is depicted in Figure 4. Since it requires at most \( O(1) \) (\( v_0 \)) time, we place an indexing around each vertex. Each color corresponds to one translation. An illustration on a grid graph is given in Figure 5. The extended convolution layer \( \phi \) is defined as:

\[
y_v = h \left( \sum_{p=0}^{\kappa-1} w_p x_{\phi_p(v)} + b \right)
\]

where \( h \) is the activation function, \( b \) is the bias term, \( x_{\perp} = 0 \) and:

\[
\phi_p(v) = \psi_p(v) \quad \text{if } \psi_p \text{ is defined on } v
\]

\[
\phi_p(v) = \perp \notin V \quad \text{else}
\]

Note that we defined convolution layers using the formalism of proxy-translations, but they can also be defined using only the formalism of local translations [21].

D. Extended Data Augmentation

Once translations are obtained on \( G \), one can use them to move training vectors, artificially creating new ones. Note that this type of data-augmentation is poorer than for images since no flipping, scaling or rotations are used.
E. Extended Downscaling Layers

Downscaling is a tricky part of the process because it supposes one can somehow regularly sample vectors. As a matter of fact, a nonregular sampling is likely to produce a highly irregular downscaled graph, on which looking for translations irremediably leads to poor accuracy, as we noticed in our experiments.

We rather define the translations of the strided graph using the previously found proxy-translations on \( G \).

First step: extended convolution with stride \( r \)

Given an arbitrary initial vertex \( v_0 \in V \), the set of kept vertices \( V_{1r} \) is defined inductively as follows:

\[
\begin{align*}
V^0_{1r} &= \{v_0\}, \\
V^r_{1r+1} &= V^r_{1r} \cup \{v \in V, \forall v' \in V^r_{1r}, v \notin N_{r-1}(v') \land \exists v' \in V^r_{1r}, v \in N_r(v')\}.
\end{align*}
\]

This sequence is nondecreasing and bounded by \( V \), so it eventually becomes stationary and we obtain \( V_{1r} = \lim_n V^r_{1r} \). Figure 6 illustrate the first downscaling \( V_{12} \) on a grid graph.

The output neurons of the extended convolution layer with stride \( r \) are \( V_{1r} \).

Second step: convolutions for the strided graph

Using the proxy-translations on \( G \), we move a localized \( r \)-hop indexing kernel over \( G \). At each location, we associate the vertices of \( V_{1r} \) with indices of the kernel, thus obtaining what we define as induced \( r \)-translations on the set \( V_{1r} \). In other words, when the kernel is centered on \( v_0 \), if \( v_1 \in V_{1r} \) is associated with the index \( p_0 \), we obtain \( \phi_{p_0}^r(v_0) = v_1 \). Subsequent convolutions at lower scales are defined using these induced \( r \)-translations similarly to Subsection C.

IV. EXPERIMENTS

To validate our method we performed experiments with two different datasets, CIFAR-10 [25] and PINES fMRI dataset [26]. The code is available at github.com/brain-bzh/MCNN.

A. CIFAR-10

On the CIFAR-10 dataset, our models are based on a variant of a deep residual network, namely PreActResNet18 [2]. We tested different combinations of graph support and data augmentation. For the graph support, we use either a regular 2D grid or either an inferred graph obtained by keeping the four neighbours that covary the most. Table I summarizes out results. In particular, it is interesting to note that results obtained without any structure prior (91.07%) are only 2.7% away from the baseline using classical CNNs on images (93.80%). This gap is even smaller (less than 1%) when using the grid prior. Also, without priors our method significantly outperforms the others.

B. PINES fMRI

The PINES dataset consists of fMRI scans on 182 subjects, during an emotional picture rating task [26]. We fetched individual first-level statistical maps (beta images) for the minimal and maximal ratings from https://neurovault.org/collections/1964, to generate the dataset. Full brain data was masked on the MNI template and resampled to a 16mm cubic grid, in order to reduce dimensionality of the dataset while keeping a regular geometrical structure. Final volumes used for classification contain 369 signals for each subject and rating.

We used a shallow network. The results on Table II show that our method was able to improve over CNNs, MLPs and other graph-based extended convolutional neural networks.

V. CONCLUSION

We proposed a new methodology that extends classical convolutional neural networks to irregular domains represented by a graph. The methodology scales linearly well with the order of the graph. Moreover, training can be performed using existing libraries for deep learning.

We performed experiments and showed that our method is able to match performance of classical convolutional neural networks on images without explicit knowledge about the underlying regular 2D structure. It also significantly outperforms existing extended convolutional neural networks alternatives based on graphs. We also demonstrated the ability of the method to adapt to slightly irregular domains by performing experiments on a neuroimaging dataset.

However, the main limitation is that on highly irregular domains, the obtained translations aren’t very helpful to design meaningful convolutions, especially if the degree of the graph varies a lot. Hence this requires to add constraints to the graph inferring step to obtain an exploitable graph if it is not.

Future work includes extending to highly irregular domains, which might require to revisit the definitions of translations.
ACKNOWLEDGEMENTS
This work was funded in part with the support of Région Bretagne and computations were performed with the use of Nvidia GPUs, courtesy of Nvidia.

REFERENCES
[1] Y. LeCun, Y. Bengio et al., “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
[2] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
[4] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: going beyond euclidean data,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.
[5] F. R. K. Chung, *Spectral Graph Theory* (CBMS Regional Conference Series in Mathematics, No. 92). American Mathematical Society, 1996.
[6] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” *arXiv preprint arXiv:1312.6203*, 2013.
[7] M. Henaff, J. Bruna, and Y. LeCun, “Deep convolutional networks on graph-structured data,” *arXiv preprint arXiv:1506.05163*, 2015.
[8] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3837–3845.
[9] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, “Cayleygraphs: Graph convolutional neural networks with complex rational spectral filters,” *arXiv preprint arXiv:1705.07664*, 2017.
[10] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2215–2223.
[11] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, “Molecular graph convolutions: moving beyond fingerprints.” *Journal of computer-aided molecular design*, vol. 30, no. 8, pp. 595–608, 2016.
[12] J.-C. Vialatte, V. Gripon, and G. Mercier, “Generalizing the convolution operator to extend cnns to irregular domains,” *arXiv preprint arXiv:1606.01166*, 2016.
[13] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016, pp. 2014–2023.
[14] J. Atwood and D. Towsley, “Diffusion-convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 1993–2001.
[15] J. Du, S. Zhang, G. Wu, J. M. Moura, and S. Kar, “Topology adaptive graph convolutional networks,” *arXiv preprint arXiv:1710.10370*, 2017.
[16] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, “Geometric deep learning on graphs and manifolds using mixture model cnns,” *arXiv preprint arXiv:1611.08402*, 2016.
[17] M. Simonovsky and N. Komodakis, “Dynamic edge-conditioned filters in convolutional neural networks on graphs,” *arXiv preprint arXiv:1704.02901*, 2017.
[18] J.-C. Vialatte, V. Gripon, and G. Coppin, “Learning local receptive fields and their weight sharing scheme on graphs,” *arXiv preprint arXiv:1706.02684*, 2017.
[19] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *stat*, vol. 1050, p. 20, 2017.
[20] A. Sankar, X. Zhang, and K. C.-C. Chang, “Motif-based convolutional neural network on graphs,” *arXiv preprint arXiv:1711.05697*, 2017.
[21] B. Pasdeloup, V. Gripon, J.-C. Vialatte, and D. Pastor, “Convolutional neural networks on irregular domains through approximate translations on inferred graphs,” *arXiv preprint arXiv:1710.10035*, 2017.
[22] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” *arXiv preprint arXiv:1704.01212*, 2017.
[23] N. Grelier, B. Pasdeloup, J.-C. Vialatte, and V. Gripon, “Neighborhood-preserving translations on graphs,” in *Proceedings of IEEE GlobalSIP*, 2016, pp. 410–414.

[24] B. Pasdeloup, V. Gripon, N. Grelier, J.-C. Vialatte, and D. Pastor, “Translations on graphs with neighborhood preservation,” *arXiv preprint arXiv:1709.03859*, 2017.
[25] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.
[26] L. J. Chang, P. J. Gianaros, S. B. Manuck, A. Krishnan, and T. D. Wager, “A sensitive and specific neural signature for picture-induced negative affect,” *PLoS biology*, vol. 13, no. 6, p. e1002180, 2015.