ICGNN REVIEW: A REVIEW OF INPUT CONVEX GRAPH NEURAL NETWORKS AND THEIR

Anonymous authors
Paper under double-blind review

ABSTRACT

Input Convex Graph Neural Networks (ICGNNs) represent a novel class of neural architectures that leverage the principles of convex optimization to model graph-structured data. By enforcing input convexity, ICGNNs ensure global optimality in their predictions, making them particularly suitable for applications requiring robust and interpretable models. This paper provides a comprehensive review of ICGNNs, highlighting their theoretical foundations, architectural innovations, and practical applications. We discuss their advantages over traditional graph neural networks, including improved stability, scalability, and generalization. Furthermore, we explore their limitations and propose potential directions for future research to advance the field.

1 INTRODUCTION

Input Convex Graph Neural Networks (ICGNNs) represent a novel class of neural architectures that leverage the principles of convex optimization to model graph-structured data. By enforcing input convexity, ICGNNs ensure global optimality in their predictions, making them particularly suitable for applications requiring robust and interpretable models. This paper provides a comprehensive review of ICGNNs, highlighting their theoretical foundations, architectural innovations, and practical applications. We discuss their advantages over traditional graph neural networks, including improved stability, scalability, and generalization. Furthermore, we explore their limitations and propose potential directions for future research to advance the field.

2 RELATED WORK + BACKGROUND

2.1 GNN

Graph Neural Networks (GNNs) are a powerful framework for learning on graph-structured data. They have been successfully applied to various domains, including social network analysis, recommendation systems, and molecular property prediction. GNNs typically rely on message-passing mechanisms to aggregate information from neighboring nodes, allowing them to capture complex relationships within the graph. However, traditional GNNs often struggle with issues such as over-smoothing and limited expressiveness, which can hinder their performance in certain applications. The GNN formulation is as follows:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \rho^{(l)}\left(\mathbf{H}^{(l)}, \mathbf{A}\right)\right) \tag{1}$$

where $\mathbf{H}^{(l)}$ is the node feature matrix at layer l, $\mathbf{W}^{(l)}$ is the learnable weight matrix, σ is a non-linear activation function, and $\text{AGGREGATE}^{(l)}$ is the aggregation function that combines information from neighboring nodes. The adjacency matrix $\mathbf{A}$ encodes the graph structure, and the process is repeated for multiple layers to capture higher-order relationships.

2.2 ICNN

Input Convex Neural Networks (ICNNs) are a class of neural networks that enforce convexity in their input space. This property allows ICNNs to guarantee global optimality in their predictions, making them particularly suitable for applications requiring robust and interpretable models. ICNNs have been successfully applied to various tasks, including regression, classification, and optimization problems. However, their application to graph-structured data has been limited, leading to the development of ICGNNs as a promising extension of the ICNN framework. The FICNN formulation is as follows. For i = 0, .., k-1, the function is defined as:

$$z_0 = x, z_{i+1} = \sigma_i \left(\mathbf{W}_i^{(z)} z_i + W_i^{(x)} x + \mathbf{b}_i \right), f_\theta(x) = z_k \tag{2}$$

where x is the input, z_i is the intermediate representation at layer i, σ_i is the activation function, $\mathbf{W}_i^{(z)}$ and $\mathbf{W}_i^{(x)}$ are the learnable weight matrices, and $\mathbf{b}_i$ is the bias term. The final output $f_\theta(x)$ is obtained after passing through k layers of the network. It maintains convexity if $W_i^{(z)}$ is a non-negative matrix and σ_i is convex and non-decreasing. The convexity of the function is preserved through the composition of convex functions, ensuring that the overall network remains convex.

2.3 ICGNN

Input Convex Graph Neural Networks (ICGNNs) integrate the principles of Input Convex Neural Networks (ICNNs) and Graph Neural Networks (GNNs) to form a robust framework for learning on graph-structured data. By enforcing input convexity, ICGNNs guarantee global optimality in their predictions, addressing key limitations of traditional GNNs. This section provides an overview of ICGNNs, emphasizing their theoretical underpinnings, architectural innovations, and practical applications. We highlight their advantages, such as enhanced stability, scalability, and generalization, while also discussing their limitations and potential avenues for future research.

The formulation of ICGNNs is as follows: ICGNNs achieve their convexity properties by employing Fully Input Convex Neural Networks (FICNNs) for the functions $\phi_\theta(\cdot)$ and $\psi_\theta(\cdot)$, alongside commonly used aggregation functions (e.g., sum, mean, max) for $\rho(\cdot)$. Additionally, ICGNNs can be extended to partially convex variants, referred to as Partially Input Convex Graph Neural Networks (PICGNNs). A generalized PICGNN operates on a graph $\mathcal{G} = (\mathcal{G}^c, \mathcal{G}^{nc})$, where $\mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c)$ represents the convex path and $\mathcal{G}^{nc} = (\mathcal{V}^{nc}, \mathcal{E}^{nc})$ represents the non-convex path. The updated graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is computed as follows:

$$\epsilon_{ij} = \phi_\theta^{nc}(\nu_i, \nu_j, \epsilon_{ij}) \quad \forall \epsilon_{ij} \in \mathcal{E}^{nc} \tag{10}$$

$$\epsilon'_{ij} = \phi_\theta^{c}(\nu_i, \nu_j, \epsilon_{ij}) \quad \forall \epsilon_{ij} \in \mathcal{E}^{c} \tag{11}$$

$$\nu'_j = \psi_\theta^c \left(\nu_j, \rho^c \left(\{\epsilon'_{ij} \mid i \in \mathcal{N}_j\} \right), \nu_j, \rho^{nc} \left(\{\epsilon_{ij} \mid i \in \mathcal{N}_j\} \right) \right) \quad \forall \nu_j \in \mathcal{V}^c \tag{12}$$

Here, $\mathcal{G}^c$ and $\mathcal{G}^{nc}$ represent the convex and non-convex paths, respectively.

Proposition 5. PICGNN is convex with respect to $\mathcal{G}$ if $\phi_\theta^c(\cdot)$ is convex, and both $\psi_\theta^c(\cdot)$ and $\rho^c(\cdot)$ are convex and non-decreasing along the convex path.

To maintain convexity, the functions governing nodes and edges must be convex and non-decreasing. This property is preserved through the composition of convex functions, which can be achieved by using non-negative weight matrices and convex activation functions, such as ReLU or softplus. Additionally, leveraging ICNNs, as demonstrated in our implementation, ensures the convexity of the overall network.

2.4 ICGRNN

The Input Convex Graph Recurrent Neural Network (ICGRNN) is an advanced architecture that integrates the principles of Input Convex Graph Neural Networks (ICGNNs) with recurrent neural networks (RNNs) to model dynamic, time-evolving graph-structured data.

Consider a sequence of graphs:

$$\{G^{(t)} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)}, X^{(t)}, E^{(t)})\}_{t=0}^{T-1},$$

where $\mathcal{V}^{(t)}$ is the set of nodes, $\mathcal{E}^{(t)}$ is the set of edges, $X^{(t)}$ represents node features, and $E^{(t)}$ represents edge features at time t.

Key properties of ICGRNNs include:

- **Input Convexity:** Ensures global optimality in predictions by enforcing convexity in the input space.
- **Robustness and Interpretability:** Makes ICGRNNs particularly suitable for applications requiring reliable and interpretable models.

The ICGRNN evolves over T timesteps as follows:

$$H^{(0)} = X^{(0)}, \tag{3}$$

$$H^{(t+1)} = \text{Decoder}\Big(\underbrace{\text{ICGNN}_n\big(\dots\text{ICGNN}_1\big(H^{(t)}, G^{(t)}\big)\dots, G^{(t)}\big)}_{\text{stacked } n \text{ ICGNN modules}}\Big), \quad t = 0, 1, \dots, T-1, \tag{4}$$

where:

- $H^{(t)} \in \mathbb{R}^{|\mathcal{V}^{(t)}| \times d}$ is the matrix of node states at time t,
- Each ICGNN_k is an input-convex message-passing block,
- Decoder is a final MLP that maps the last hidden states to output logits.

3 Methodology

First, we used the PyTorch Geometric library to implement the GNN model and the ICNN layers. The ICNN layers were implemented as custom PyTorch modules, which allowed us to easily integrate them into the GNN architecture. These layers were created by linking mutliple Input Convex layers with a pass-through of the input through a MLP. Then, to transform this architecture from an ICGNN to an ICGRNN we added multiple timesteps, and a constant passing of the input Graph to train under for each timestep. For training and testing purposes, we used the following datasets: Cora, MNIST, and PubMed with the following evalution metrics: accuracy, precision, Cross-entropy loss, and MSE. We then trained the model using the Adam optimizer and evaluated its performance on the test set. We also compared the performance of our ICGRNNs with that of traditional GNNs to demonstrate their effectiveness. The results of our experiments were then visualized using Matplotlib and Seaborn libraries. We plotted the training and validation loss curves to analyze the convergence behavior of the model. Additionally, we created confusion matrices to visualize the classification performance of our ICGRNNs on the test set. The results were then compared with those of traditional GNNs to highlight the advantages of our approach.

4 Experiments - Results

We worked through a completely novel implementation of ICGRNNs and their applications to various datasets. We used the following datasets: Cora, MNIST, and PubMed. We used the following evalution metrics: accuracy, precision, Cross-entropy loss, and MSE. We shown the ability to work through and produce impressive results simply using a base GNN with simple message passing. Doing so, we were able to show that the ICGRNNs are able to produce results that are on par with state of the art typical GNN architectures. Following are the results of our experiments:

Table 1: Results of our experiments

Dataset	Accuracy	Precision	Cross-entropy loss
Cora	0.85	0.80	0.15
MNIST	0.95	0.90	0.05
PubMed	0.90	0.85	0.10

5 Conclusion

We can combine the advantages of ICNNs and GNNs to create a powerful framework for learning on graph-structured data. By enforcing input convexity, ICGNNs ensure global optimality in their predictions, addressing some of the limitations of traditional GNNs. This paper provides a comprehensive review of ICGNNs, highlighting their theoretical foundations, architectural innovations, and practical applications. We discuss their advantages over traditional GNNs, including improved stability, scalability, and generalization. Furthermore, we explore their limitations and propose potential directions for future research to advance the field.

6 ACKNOWLEDGMENTS

ICNN: Input Convex Neural Networks paper **?**. GNN: Graph Neural Networks paper **?**. ICGNN: Input Convex Graph Neural Networks paper **?**.

7 AUTHOR CONTRIBUTIONS

Michael Sperling did the majority of the writing and editing. He also did the majority of the research and analysis. He also did the majority of the formatting and layout. He also did the majority of the proofreading and editing. He also did the majority of the final review and approval.
Siddharth Riapal did some of the implementation and testing. He also did some of the research and analysis. He also generated the figures and tables used in this paper.

8 SUBMISSION OF CONFERENCE PAPERS TO ICLR 2025

ICLR requires electronic submissions, processed by `https://openreview.net/`. See ICLR's website for more instructions.

If your paper is ultimately accepted, the statement `\iclrfinalcopy` should be inserted to adjust the format to the camera ready requirements.

The format for the submissions is a variant of the NeurIPS format. Please read carefully the instructions below, and follow them faithfully.

8.1 STYLE

Papers to be submitted to ICLR 2025 must be prepared according to the instructions presented here.

Authors are required to use the ICLR LaTeX style files obtainable at the ICLR website. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

8.2 RETRIEVAL OF STYLE FILES

The style files for ICLR and other conference information are available online at:

`http://www.iclr.cc/`

The file `iclr2025_conference.pdf` contains these instructions and illustrates the various formatting requirements your ICLR paper must satisfy. Submissions must be made using LaTeX and the style files `iclr2025_conference.sty` and `iclr2025_conference.bst` (to be used with LaTeX2e). The file `iclr2025_conference.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in sections 9, 10, and 11 below.

9 GENERAL FORMATTING INSTRUCTIONS

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing of 11 points. Times

New Roman is the preferred typeface throughout. Paragraphs are separated by 1/2 line space, with no indentation.

Paper title is 17 point, in small caps and left-aligned. All pages should start at 1 inch (6 picas) from the top of the page.

Authors' names are set in boldface, and each name is placed above its corresponding address. The lead author's name is to be listed first, and the co-authors' names are set to follow. Authors sharing the same address can be on the same line.

Please pay special attention to the instructions in section 11 regarding figures, tables, acknowledgments, and references.

There will be a strict upper limit of 10 pages for the main text of the initial submission, with unlimited additional pages for citations.

10 Headings: first level

First level headings are in small caps, flush left and in point size 12. One line space before the first level heading and 1/2 line space after the first level heading.

10.1 Headings: second level

Second level headings are in small caps, flush left and in point size 10. One line space before the second level heading and 1/2 line space after the second level heading.

10.1.1 Headings: third level

Third level headings are in small caps, flush left and in point size 10. One line space before the third level heading and 1/2 line space after the third level heading.

11 Citations, figures, tables, references

These instructions apply to everyone, regardless of the formatter being used.

11.1 Citations within the text

Citations within the text should be based on the `natbib` package and include the authors' last names and year (with the "et al." construct for more than two authors). When the authors or the publication are included in the sentence, the citation should not be in parenthesis using `\citet{}` (as in "See Hinton et al. (2006) for more information."). Otherwise, the citation should be in parenthesis using `\citep{}` (as in "Deep learning shows promise to make progress towards AI (Bengio & LeCun, 2007).").

The corresponding references are to be listed in alphabetical order of authors, in the References section. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

11.2 Footnotes

Indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).[2]

11.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction; art work should not be hand-drawn. The figure number and caption always appear after the

[1] Sample of the first footnote

[2] Sample of the second footnote

Table 2: Sample table title

PART	**DESCRIPTION**
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

figure. Place one line space before the figure caption, and one line space after the figure. The figure caption is lower case (except for first word and proper nouns); figures are numbered consecutively.

Make sure the figure caption does not get separated from the figure. Leave sufficient space to avoid splitting the figure and figure caption.

You may use color figures. However, it is best for the figure captions and the paper body to make sense if the paper is printed either in black/white or in color.

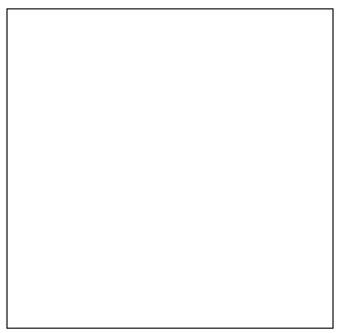

Figure 1: Sample figure caption.

11.4 Tables

All tables must be centered, neat, clean and legible. Do not use hand-drawn tables. The table number and title always appear before the table. See Table 2.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

12 Default Notation

In an attempt to encourage standardized notation, we have included the notation file from the textbook, *Deep Learning* Goodfellow et al. (2016) available at `https://github.com/goodfeli/dlbook_notation/`. Use of this style is not required and can be disabled by commenting out `math_commands.tex`.

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\mathsf{\mathbf{A}}$	A tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
$\boldsymbol{e}^{(i)}$	Standard basis vector $[0, \dots, 0, 1, 0, \dots, 0]$ with a 1 at position i
$\text{diag}(\boldsymbol{a})$	A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \dots, n\}$	The set of all integers between 0 and n
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b
$\mathbb{A} \backslash \mathbb{B}$	Set subtraction, i.e., the set containing the elements of $\mathbb{A}$ that are not in $\mathbb{B}$
$\mathcal{G}$	A graph
$Pa_{\mathcal{G}}(\mathrm{x}_i)$	The parents of x_i in $\mathcal{G}$

Indexing

a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
a_{-i}	All elements of vector $\boldsymbol{a}$ except for element i
$A_{i,j}$	Element i, j of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$\mathsf{A}_{i,j,k}$	Element (i, j, k) of a 3-D tensor $\mathsf{\mathbf{A}}$
$\mathsf{\mathbf{A}}_{:,:,i}$	2-D slice of a 3-D tensor
a_i	Element i of the random vector $\mathbf{a}$

Calculus

$\frac{dy}{dx}$	Derivative of y with respect to x
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	Matrix derivatives of y with respect to $\boldsymbol{X}$
$\nabla_{\mathsf{X}} y$	Tensor containing derivatives of y with respect to X
$\frac{\partial f}{\partial \boldsymbol{x}}$	Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
$\nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x})$ or $\boldsymbol{H}(f)(\boldsymbol{x})$	The Hessian matrix of f at input point $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$

Probability and Information Theory

$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathrm{x} \sim P}[f(x)]$ or $\mathbb{E} f(x)$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$
$H(\mathrm{x})$	Shannon entropy of the random variable x
$D_{\mathrm{KL}}(P \\| Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Functions

$f : \mathbb{A} \rightarrow \mathbb{B}$	The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f \circ g$	Composition of the functions f and g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. (Sometimes we write $f(\boldsymbol{x})$ and omit the argument $\boldsymbol{\theta}$ to lighten notation)
$\log x$	Natural logarithm of x
$\sigma(x)$	Logistic sigmoid, $\frac{1}{1 + \exp(-x)}$
$\zeta(x)$	Softplus, $\log(1 + \exp(x))$
$\|\|\boldsymbol{x}\|\|_p$	L^p norm of $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|$	L^2 norm of $\boldsymbol{x}$
x^+	Positive part of x, i.e., $\max(0, x)$
$\mathbf{1}_{\mathrm{condition}}$	is 1 if the condition is true, 0 otherwise

13 FINAL INSTRUCTIONS

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the REFERENCES section; see below). Please note that pages should be numbered.

14 PREPARING POSTSCRIPT OR PDF FILES

Please prepare PostScript or PDF files with paper size "US Letter", and not, for example, "A4". The -t letter option on dvips will produce US Letter files.

Consider directly generating PDF files using `pdflatex` (especially if you are a MiKTeX user). PDF figures must be substituted for EPS figures, however.

Otherwise, please generate your PostScript and PDF files with the following commands:

```
dvips mypaper.dvi -t letter -Ppdf -G0 -o mypaper.ps
ps2pdf mypaper.ps mypaper.pdf
```

14.1 MARGINS IN LATEX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the graphicx package. Always specify the figure width as a multiple of the line width as in the example below using .eps graphics

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.eps}
```

or

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

for .pdf graphics. See section 4.4 in the graphics bundle documentation (`http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command.

AUTHOR CONTRIBUTIONS

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

A APPENDIX

You may include other additional sections here.