

Supplementary Materials: FTF-ER: Feature-Topology Fusion-Based Experience Replay Method for Continual Graph Learning

Anonymous Authors

This supplementary material provides more details of the research background and experiment results that are omitted from the manuscript due to the page limit. As a supplement to the background, Section 1 introduces the basic architecture of Graph Neural Networks (GNNs) and describes three types of GNNs that are applied as backbones in our study. To strengthen the theoretical foundation of our proposed FTF-ER, Section 2 presents the description and generalization of the Hodge Decomposition Theorem, along with the proof of the graph Laplacian operator. Finally, Section 3 provides additional implementation details and presents a qualitative analysis of our experimental results.

1 GRAPH NEURAL NETWORKS

Graph neural networks (GNNs) are deep learning models defined on a graph \mathcal{G} . At each layer of GNNs, nodes in \mathcal{V} update their hidden representations by aggregating and transforming information from their neighborhoods. the process of aggregation and transformation is accomplished by an update function that takes into account the hidden representations of the node and its adjacent nodes.

Formally, for each node v in \mathcal{V} , its hidden representation at the l -th layer of the GNN, denoted as $h_v^{(l)}$, is computed as follows:

$$h_v^{(l)} = U(h_v^{(l-1)}, h_u^{(l-1)}), \forall u \in \mathcal{N}(v), \quad (1)$$

where $U(\cdot)$ is a differentiable function and $\mathcal{N}(v)$ represents the set of neighbors of node v in the graph. $U(\cdot)$ takes the hidden representation of the current node $h_v^{(l-1)}$ and its neighboring nodes $h_u^{(l-1)}$ as inputs.

The graph convolutional network (GCN) [1] designs $U(\cdot)$ based on a first-order approximation of the spectra of the graph, which fixes the adjacency matrix A . In the case of the attention-based GNN such as the graph attention network (GAT) [7], $U(\cdot)$ is designed based on pairwise attention. Furthermore, in [8], the authors highlight the performance limitations of GNNs and propose a new network called the graph isomorphism network (GIN). In our paper, the aforementioned three types of neural networks serve as the backbones of our FTF-ER.

2 THEOREM AND PROOF

2.1 Hodge Decomposition Theorem

Hodge decomposition theorem is a fundamental result in the theory of differential forms and Riemannian geometry. On a compact and oriented Riemannian manifold, the Hodge decomposition theorem states that any differential form can be uniquely decomposed into the sum of three components:

- **An exact form:** A differential form that is the exterior derivative of another form.
- **A co-exact form:** A differential form whose codifferential (adjoint of the exterior derivative) is zero.

- **A harmonic form:** A differential form that is both closed (its exterior derivative is zero) and co-closed (its codifferential is zero).

Let $\Omega^k(M)$ be a k -form on an n -dimensional smooth manifold M , d be the exterior derivative operator, and δ be the adjoint map of d . Then we can define the Hodge-Laplace operator:

Definition 2.1 (Hodge-Laplace operator).

$$\Delta \triangleq d\delta + \delta d : \Omega^k(M) \mapsto \Omega^k(M). \quad (2)$$

Given the Definition 2.1, we state the Hodge decomposition theorem as follows:

THEOREM 2.2 (HODGE DECOMPOSITION THEOREM). *For any $\alpha \in \Omega^{k-1}(M)$, $\beta \in \Omega^{k+1}(M)$ and $\Delta\gamma = 0$, we have*

$$\omega \in \Omega^k(M) \Rightarrow \omega = d\alpha + \delta\beta + \gamma, \quad (3)$$

where $d\alpha$ is an exact k -form, $\delta\beta$ is a co-exact k -form and γ satisfying $\Delta\gamma = 0$ is also referred to as a harmonic form.

This decomposition is unique and orthogonal with respect to the L^2 inner product on the space of differential forms. The Hodge decomposition theorem has significant implications in various areas of mathematics and physics, including the study of cohomology, the theory of partial differential equations, and the formulation of Maxwell's equations in differential form language. Besides, this theorem allows us to categorize differential forms into three distinct types, each with its own physical interpretation:

- **An exact form $d\alpha$:** This differential form can be expressed as the gradient of a scalar field, making it particularly useful for describing potential energies associated with various fields. In physics, an exact form is commonly employed to represent electric potential energy or magnetic potential energy, providing valuable insights into the behavior of these fields.
- **A co-exact form $\delta\beta$:** A differential form that can be written as the curl of a vector field is classified as a co-exact form. It plays a crucial role in describing circulation phenomena related to fields, such as magnetic flux in the context of magnetic fields. By utilizing a co-exact form, physicists can effectively model and analyze the circulatory aspects of these fields.
- **A harmonic form γ :** Unlike an exact or a co-exact form, a harmonic form is characterized by the absence of potential energies or circulations. It represents a state of equilibrium or scenarios devoid of sources or sinks. In the realm of physics, a harmonic form is frequently associated with fields that are free from sources, such as source-free electric fields or source-free magnetic fields. This form provides a framework for studying the behavior of fields in the absence of external influences.

2.2 Hodge Decomposition Theorem on Graphs

By defining several concepts on graphs, including d , Δ and δ when $k = 0$, we can generalize the Hodge decomposition theorem from its manifold version to graphs.

Definition 2.3 (Hodge Potential Score).

$$\Omega^0(\mathcal{G}) \triangleq \{s : \mathcal{V} \mapsto \mathbb{R}\}. \quad (4)$$

Definition 2.4 (Edge Flows).

$$\Omega^1(\mathcal{G}) \triangleq \{\mathcal{X} : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R} \mid \mathcal{X}(i, j) = -\mathcal{X}(j, i), (i, j) \in \mathcal{E}\}. \quad (5)$$

Definition 2.5 (Gradient Operator). Let \mathbf{grad} be the gradient operator, $s_i, s_j \in \Omega^0(\mathcal{G})$, we have

$$(d_0 s)(i, j) \triangleq (\mathbf{grad} s)(i, j) \triangleq s_j - s_i, (i, j) \in \mathcal{E}. \quad (6)$$

Definition 2.6 (Negative Divergence Operator). Let w be the weight of an element in $\Omega^k(\mathcal{G})$, $w_i \in \Omega^0(\mathcal{G})$, $w_{ij} \in \Omega^1(\mathcal{G})$ and \mathbf{div} be the divergence operator, we have

$$(\delta_0 \mathcal{X})(i) \triangleq (-\mathbf{div} \mathcal{X})(i) = -\sum_j \frac{w_{ij}}{w_i} \mathcal{X}(i, j). \quad (7)$$

Definition 2.7 (Graph Laplacian Operator).

$$\Delta_0 \triangleq \delta_0 d_0 \triangleq -\mathbf{div}(\mathbf{grad}). \quad (8)$$

We note that we denote Δ_0 as the graph Laplacian operator. The proof that Δ_0 corresponds to the usual graph Laplacian operator is given in Section 2.3.

Given the definitions above, we state the Hodge decomposition theorem on graphs for the case where $k = 0$ at first:

THEOREM 2.8 (HODGE DECOMPOSITION THEOREM ON GRAPHS FOR $k = 0$). Let \mathbf{Im} be the image set and \mathbf{ker} be the kernel set, we have

$$\Omega^0(\mathcal{G}) = \mathbf{Im} d_0 \oplus \mathbf{Im} \delta_0 \oplus \mathbf{ker} \Delta_0. \quad (9)$$

In our paper, we employ the gradient field decomposition provided by the $k = 0$ version of the Hodge decomposition theorem on graphs to compute the topological importance of nodes. For the sake of theoretical completeness, we then describe the theorem for the case where $k \in \mathbb{N}$. To generalize the Hodge decomposition theorem on graphs for $k = 0$ to the case where $k \in \mathbb{N}$, we first denote K_k as the set of k -cliques on the graph. Then we have

$$\begin{aligned} \Omega^k(\mathcal{G}) &\triangleq \{u : \mathcal{V}^{k+1} \mapsto \mathbb{R} \mid \\ &u(i_{\sigma(0)}, \dots, i_{\sigma(k)}) = \text{sign}(\sigma)u(i_0, \dots, i_k), \\ &(i_0, \dots, i_k) \in K_{k+1}\}. \end{aligned} \quad (10)$$

For $d_k : \Omega^k(\mathcal{G}) \mapsto \Omega^{k+1}(\mathcal{G})$ and $\delta_k : \Omega^k(\mathcal{G}) \mapsto \Omega^{k-1}(\mathcal{G})$, we define

$$(d_k u)(i_0, \dots, i_{k+1}) \triangleq \sum_{j=0}^{k+1} (-1)^{j+1} u(i_0, \dots, i_{j-1}, i_{j+1}, \dots, i_{k+1}), \quad (11)$$

and

$$(\delta_k u)(i_0, \dots, i_{k+1}) \triangleq \sum_{j=0}^{k+1} (-1)^j u(i_0, \dots, i_{j-1}, i_{j+1}, \dots, i_{k+1}). \quad (12)$$

Then we have

$$\Delta_k \triangleq \delta_k d_k + d_{k-1} \delta_{k-1}. \quad (13)$$

Given the definitions above, we state the Hodge decomposition theorem on graphs for $k \in \mathbb{N}$:

THEOREM 2.9 (HODGE DECOMPOSITION THEOREM ON GRAPHS). Let \mathbf{Im} be the image set and \mathbf{ker} be the kernel set, we have

$$\Omega^k(\mathcal{G}) = \mathbf{Im} d_k \oplus \mathbf{Im} \delta_k \oplus \mathbf{ker} \Delta_k. \quad (14)$$

Theorem 2.9 reveals that a graph signal can be decomposed into three orthogonal components:

- **Gradient component:** The gradient component represents the conservative or curl-free part of the graph signal. It can be expressed as the gradient of a potential function defined on the nodes of the graph. This means that the flow along any closed path in the graph sums up to zero. In other words, the gradient component captures the portion of the signal that exhibits no rotational behavior. It is analogous to the irrotational component in the continuous Hodge decomposition.
- **Curl component:** The curl component represents the rotational or divergence-free part of the graph signal. It can be expressed as the curl of a potential function defined on the edges of the graph. This means that the net flow out of any node in the graph is zero. The curl component captures the portion of the signal that exhibits rotational behavior but has no divergence.
- **Harmonic component:** The harmonic component is the part of the graph signal that is both gradient-free and divergence-free. It represents the signal's behavior that is not captured by either the gradient or the curl components. In other words, it is the part of the signal that is constant on connected components of the graph and has zero gradient and zero divergence.

These three components (i.e., gradient, curl, and harmonic) form a complete and orthogonal decomposition of the graph signal. They provide a way to analyze and understand the different aspects of the signal's behavior on the graph. The gradient component captures the conservative part, the curl component captures the rotational part, and the harmonic component captures the part that is neither conservative nor rotational. This decomposition is particularly useful in applications involving signal processing, data analysis, and machine learning on graph-structured data.

2.3 Graph Laplacian Operator

In this section, we give the proof that Δ_0 in our paper corresponds to the usual graph Laplacian operator for completeness, which may be found in [3].

PROOF. Let $s \in \Omega^1(\mathcal{G})$, we have

$$(\mathbf{grad} s)(i, j) = \begin{cases} s(j) - s(i), & \text{if } (i, j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

Let a_{ij} be an element of the adjacency matrix \mathbf{A} , the gradient may be written as $(\mathbf{grad} s)(i, j) = a_{ij}(s(j) - s(i))$ and then

$$\begin{aligned} (\Delta_0 s)(i) &= -(\mathbf{div}(\mathbf{grad} s))(i) = -(\mathbf{div} a_{ij}(s(j) - s(i)))(i) \\ &= -\sum_{j=1}^n a_{ij}(s(j) - s(i)) = \xi_i s(i) - \sum_{j=1}^n a_{ij} s(j), \end{aligned} \quad (16)$$

where for any node v_i ($i = 1, \dots, n$), we define its degree as

$$\xi_i = \deg(i) = \sum_{j=1}^n a_{ij}. \quad (17)$$

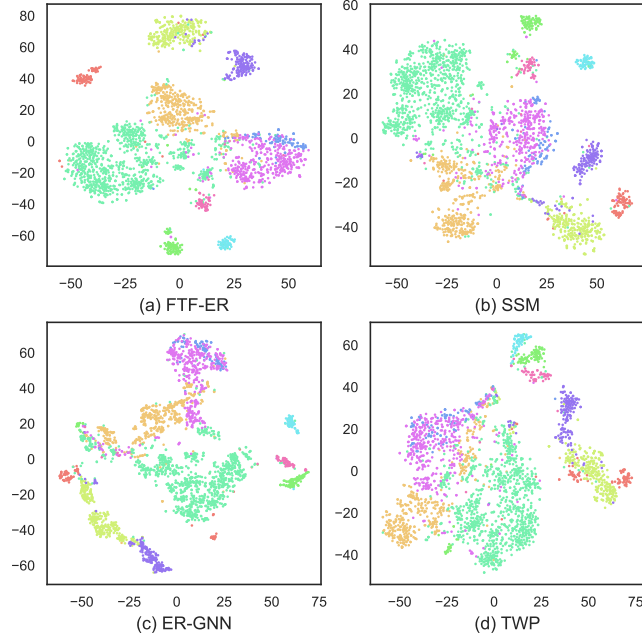


Figure 1: 2-D t-SNE projections of embeddings in four models. The nodes with different labels are represented by dots in different colors.

If we regard a function $s \in \Omega^1(\mathcal{G})$ as a vector $(s_1, \dots, s_n) \in \mathbb{R}^n$ where $s(i) = s_i$ and set $\mathbf{D} = \text{diag}(\xi_1, \dots, \xi_n) \in \mathbb{R}^{n \times n}$, then Eq. (16) becomes

$$\Delta_0 s = \begin{bmatrix} \xi_1 - a_{11} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & \xi_2 - a_{22} & \cdots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & \xi_n - a_{nn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = (\mathbf{D} - \mathbf{A})s. \quad (18)$$

So Δ_0 may be regarded as $\mathbf{D} - \mathbf{A}$, the usual definition of a graph Laplacian. \square

3 ADDITIONAL EXPERIMENTAL DETAILS

3.1 Implementation Details

We use Adam optimizer to optimize the models, setting the initial learning rate to 0.005 and the number of training epochs to 200 on all datasets. The regularizer hyper-parameter for EWC [2], MAS and TWP is always set to 10,000. And β for TWP [4] is set to 0.01. For those experience replay baselines, i.e., GEM [5], ER-GNN [10] and SSM [9], we set the buffer size for each class to be 60, 60, 400, 100 for Amazon Computers, Corafull, OGB-Arxiv, and Reddit, respectively. For our method, we choose a buffer size that is the same as that of other methods and select a suitable β from [0.0, 0.25, 0.5, 0.75, 1.0] for different datasets. Additionally, we set the structure of all backbones as a 2-layer network with a hidden layer dimension of 256 for fairness. Finally, due to the abundance of experimental results for each method across all the three backbones, we only present the best results of each method on each dataset.

3.2 Qualitative Analysis

In order to demonstrate the effectiveness of the node representations learned by FTF-ER, we conduct a qualitative analysis on the Amazon Computers dataset. For this purpose, we generate a series of standard t-SNE [6] 2D projected plots of node representations to reinforce this analysis. We select the complete test data set containing 5 tasks and 10 classes for demonstration, to analyze the overall performance of each model after undergoing the complete continual learning process. Given FTF-ER's ability to differentiate between the importance of nodes at the feature and topological levels, we anticipate that nodes sharing the same labels will be positioned closely in the projection space, indicating similar representation vectors. **Figure 1** visualizes the hidden layer representations of four CGL methods, namely FTF-ER, SSM [9], ER-GNN [10], and TWP [4]. Experimental results show that FTF-ER exhibits a clearer separation of nodes from distinct communities compared to alternative methods. The nodes with different labels are represented by dots in different colors. This showcases the capability of our FTF-ER in capturing distinctions among nodes within diverse communities through the gathered node information.

REFERENCES

- [1] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [2] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* (2017), 3521–3526.
- [3] Lek-Heng Lim. 2020. Hodge Laplacians on graphs. *Siam Review* (2020), 685–715.
- [4] Huihui Liu, Yiding Yang, and Xinchao Wang. 2021. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 8653–8661.
- [5] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems* (2017).
- [6] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 11 (2008).
- [7] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [8] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).
- [9] Xikun Zhang, Dongjin Song, and Dacheng Tao. 2022. Sparsified subgraph memory for continual graph representation learning. In *IEEE International Conference on Data Mining*. 1335–1340.
- [10] Fan Zhou and Chengtai Cao. 2021. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4714–4722.