TANGO WITH DRIFT: A COUNTERACTIVE FIELD FOR COMPACT GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph representation learning has become a cornerstone of modern machine learning. Yet, inhibiting task-irrelevant and misleading signals on graphs remains highly challenging. Existing methods mainly focus on the static outcome level, *i.e.*, by regularizing either graph structures or final representations after detrimental signals have already diffused and become entangled with predictive cues. This not only makes post-hoc removal intrinsically difficult, but also risks collateral damage to task-relevant structures. In this paper, we propose CounterDiff, which casts compact graph learning as a propagation-refinement problem under a graph stochastic partial differential equation (SPDE) view. CounterDiff characterizes the spectral signature of detrimental propagation and enforces coherent counteractions while preserving the predictive flow. We extensively evaluate CounterDiff on diverse graph learning tasks, and the consistent gains highlight the importance of propagation-level intervention for compact yet predictive graph learning.

1 INTRODUCTION

Graphs provide a natural abstraction for relational structures across diverse domains. The prevalence of graph-structured data has driven rapid advances in graph representation learning, which has emerged as a fundamental paradigm for capturing complex dependencies beyond Euclidean data, with growing impact on applications such as neuroscience Behrouz et al. (2024), social networks Fan et al. (2019), and financial modeling Giudici & Raffinetti (2023).

Despite the advances, existing studies mainly focus on developing more expressive networks to handle the rapidly growing scale and complexity of graph data Ding et al. (2024b); Deng et al. (2024); Tönshoff et al. (2023); Chen et al. (2023). Yet they neglect a fundamental challenge that real-world graphs inevitably contain redundant or even misleading signals. Once involved in message passing, these detrimental components can diffuse across the graph, obscuring critical information and relationships, thus ultimately leading to untrustworthy results.

To address this, a common strategy is to perform structure editing or spatial compression. Typical approaches include: (i) statistical methods that aim to identify representative substructures or densely connected subgraphs according to predefined statistical criteria Fang et al. (2019a); Gionis & Tsourakakis (2015); Yan & Han (2002); and (ii) sparsification-based methods that simplify graph topology through pruning, pooling, or condensation Zheng et al. (2020); Chen et al. (2018); Wang et al. (2024b). However, these methods either lack task-awareness or rely on heuristic designs, failing to restrain the propagation of detrimental components while risking the loss of predictive information.

Another line of works follow the information bottleneck (IB) paradigm Tishby et al. (2000), which adapts the mutual information objective to graphs, and formulates the tradeoff between including task-relevant information and removing less important structures Wu et al. (2020); Yu et al. (2021); Hoang & Lee (2025); Ding et al. (2024a). Despite the theoretical elegance, applying IB to graphs remains challenging, as prevailing mutual information (MI) estimators are inherently established on independence assumptions, thus their variational approximations often introduce a substantial estimation gap, making it difficult to maintain theoretical consistency with the IB principle.

From the perspective of information propagation, the above approaches essentially perform static modeling at the outcome level. However, information on graphs is inherently dynamic, *i.e.*, both

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

task-relevant and task-irrelevant signals diffuse, mix, and even amplify through rounds of message passing. By the time these methods intervene, detrimental components may have already dispersed across the graph in a spatially heterogeneous manner and become deeply entangled with useful cues. Such entanglement not only makes post-hoc removal fundamentally difficult, but also risks causing collateral damage to task-relevant structures.

In this paper, we argue that the key to inhibiting detrimental components on graphs lies in the propagation process. This motivates a simple yet fundamentally different strategy from the perspective of stochastic dynamical systems. Specifically, we propose CounterDiff, which conceptually decomposes the deterministic drift into predictive flow and task-irrelevant diffusion, and reinterprets the stochastic term as a Graph Matérn Counteractive Field that produces topology-aware counteractive signals against detrimental propagation. This formulation shifts compact graph learning from post-hoc compression of representations and structures to direct intervention within the propagation process, counteracting task-irrelevant signals before they are broadcast across the graph. In particular, such refinement is progressively correctable rather than one-shot: each counteraction step need not be exact, since the next step inherits the chance to compensate for what the previous left undone.

CounterDiff comprises two complementary objectives: counteractive refinement and sufficiency alignment. Counteractive refinement characterizes the spectral signature of task-irrelevant propagation and enforces a coherent counteraction via a parsimony objective. Sufficiency alignment safeguards the predictive flow by leveraging a natural Markov relation from the deterministic diffusion to its counteracted outcome, yielding a principled and tractable objective that guides counteraction toward task-irrelevant propagation while preserving critical flows.

Our main contributions are summarized as follows: (1)We propose CounterDiff, which recasts compact graph learning as learning to counteract task-irrelevant propagation within a graph SPDE formulation. (2) We derive two theoretically grounded objectives to guide the counteractive field: a parsimony objective to coherently mitigate task-irrelevant propagation, and a variational sufficiency objective to preserve predictive flows. (3) We conduct extensive experiments on diverse graph learning tasks, and the consistent improvements over the state-of-the-arts models validate both the motivation and formulation of CounterDiff.

2 METHOD

In this section, we formulate compact graph representation learning from a stochastic diffusion perspective and introduce CounterDiff to structure a counteractive field against task-irrelevant propagation. Specifically, we construct a Graph Matérn Counteractive Field and regulate its intervention through parsimony and sufficiency objectives, followed by an efficient implementation with GNN parameterization and Chebyshev approximation.

2.1 BACKGROUND

Compact Graph Learning aims to preserve task-relevant information while removing unnecessary graph content. Existing approaches primarily achieve this through structural compression, such as graph sparsification, pooling, and condensation, or through the information bottleneck (IB) principle, which balances representation compactness and predictive sufficiency. However, these formulations share a fundamental limitation: they refine structural or representational outcomes rather than the propagation that produces them. Consequently, detrimental signals may already have propagated and become entangled with predictive cues before refinement takes effect, making their removal prone to collateral damage.

Graph Diffusion and Stochastic Propagation reformulates graph message passing as a continuous diffusion process and further generalizes it to stochastic propagation

$$d\mathbf{H}(t) = \underbrace{f(\mathbf{H}(t))dt}_{\textit{Drift term}} + \underbrace{g(\mathbf{H}(t))d\mathbf{W}(t)}_{\textit{Stochastic term}}, \tag{1}$$

where $\mathbf{H}(t)$ denotes the graph diffusion state, with the drift and stochastic terms capturing deterministic propagation and structured perturbations, respectively. We build on this formulation and reinterpret the stochastic term as a structured counteractive field against task-irrelevant propagation.

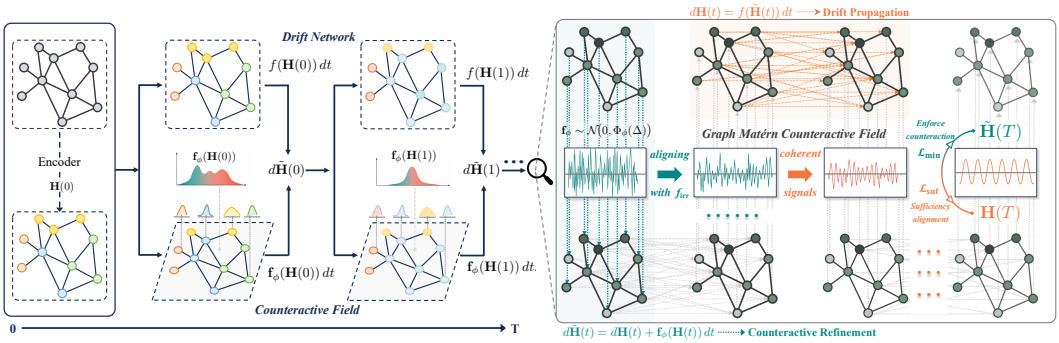


Figure 1: Overview of the proposed formulation.

2.2 Graph Matérn Counteractive Field

Our primary goal is to learn compact yet predictive graph representations. To this end, we aim to mitigate the preservation and propagation of the task-irrelevant nuisances, while maintaining the flow of predictive information intact. That is, compared with the regular diffusion process shown in Eq. (1), we seek a more parsimonious yet predictive one $\tilde{\mathbf{H}}$ such that:

$$\mathcal{C}(\tilde{\mathbf{H}}) < \mathcal{C}(\mathbf{H}) \quad \text{s.t.} \quad H(\mathbf{Y} \mid \tilde{\mathbf{H}}) \leq H(\mathbf{Y} \mid \mathbf{H}), \tag{2}$$

where $\mathcal{C}$ serves to measure the degree of task-irrelevant diffusion, and Eq. (2) aims to refine the propagation process *w.r.t.* the target $\mathbf{Y}$. To achieve this, we consider Eq. (1), where the drift term models graph propagation dynamics and thus naturally entangles the predictive component $f_{\text{pre}}(\mathbf{H}(t))$ with task-irrelevant nuisances $f_{\text{irr}}(\mathbf{H}(t))$:

$$f(\mathbf{H}(t)) = f_{\text{pre}}(\mathbf{H}(t)) + f_{\text{irr}}(\mathbf{H}(t)). \tag{3}$$

Conceptually, this indicates achieving Eq. (2) requires suppressing the task-irrelevant component $f_{\text{irr}}(\mathbf{H}(t))$. This naturally aligns with the stochastic term, which provides a channel for structured intervention and, ideally, allows the refined diffusion to be driven primarily by the predictive part:

$$d\tilde{\mathbf{H}}(t) \approx f_{\text{pre}}(\mathbf{H}(t))\, dt. \tag{4}$$

This calls for a principled formulation to model the distributional pattern of $f_{\text{irr}}(\mathbf{H}(t))$, instead of stochastic perturbation. To this end, we present the following definition:

Definition 1 (Graph Gaussian Random Field). *Given an undirected graph $G = (V, E)$ with graph Laplacian $\Delta = U\Lambda U^\top$, a graph Gaussian random field on G is a multivariate Gaussian random variable with positive definite covariance matrix $\mathbf{C} = U\Phi(\Lambda)U^\top$, where Φ denotes a matrix-valued spectral function that applies a scalar function to the diagonal entries of Λ.*

While the above definition provides a topology-aware covariance prior through Laplacian spectrum, it cannot explicitly counteract or even capture the propagation of task-irrelevant components on graphs. To address this, we reformulate Φ with a Matérn spectral construction and define the Graph Matérn Counteractive Field as follows:

Definition 2 (Graph Matérn Counteractive Field). *Given an undirected graph $G = (V, E)$ with graph Laplacian $\Delta = U\Lambda U^\top$, a counteractive field on the graph is defined as a Gaussian field $\mathbf{f}_\phi \sim \mathcal{N}(0, \Phi_\phi(\Delta))$, where $\Phi_\phi(\Delta)$ denotes a covariance operator in the graph spectral domain:*

$$\Phi_\phi(\Delta) = U\,\mathrm{diag}\big(h_{\phi,1}(\lambda), \ldots, h_{\phi,n}(\lambda)\big)\,U^\top, \quad h_{\phi,r}(\lambda) = m_{\phi,r}(\lambda)\left(\frac{2\nu}{\kappa^2} + \lambda\right)^{-\nu}, \tag{5}$$

where $\nu, \kappa > 0$ are scale parameters and $m_{\phi,r}(\lambda)$ is a learnable spectral modulator.

The proposed field aims to characterize the diffusion pattern of the task-irrelevant component and thereby yields a principled formulation to mitigate $f_{\text{irr}}(\mathbf{H}(t))$. It comprises two terms: the Matérn base term encodes graph-aware smoothness and spatial correlation, while the learnable modulator shapes the spectral response toward task-irrelevant propagation.

More specifically, we regularize the refined diffusion signal $\tilde{\mathbf{H}}(t)$ on its dispersion, which encourages $h_{\phi,r}(\lambda)$ to provide counteraction against $f_{\text{irr}}(\mathbf{H}(t))$, thereby driving the diffusion signal toward a more compact and predictive representation. Formally, we define parsimony objective over $\tilde{\mathbf{H}}(t)$

$$\mathcal{L}_{\min} = \mathbb{E}_{\tilde{\mathbf{H}}(t)}\left[\log\det\big(\mathrm{Cov}(\tilde{\mathbf{H}}(t)) + \epsilon I\big)\right], \tag{6}$$

where $\tilde{\mathbf{H}}(t)$ denotes the refined diffusion signal at step t, $\mathrm{Cov}(\tilde{\mathbf{H}}(t))$ is its covariance matrix, $\det(\cdot)$ denotes the determinant, I is the identity matrix, and $\epsilon > 0$ is a small constant for numerical stability. This objective serves as a tractable surrogate for the information volume of $\tilde{\mathbf{H}}(t)$[1] and thus aligns with the parsimonious objective in Eq. (2). However, parsimony is not our only objective; here, we need to maintain the predictive flow intact. This calls for an additional constraint from the sufficiency side to preserve predictive information while further excluding the task-irrelevant component.

Discussion. Conceptually, Eq. (2) is aligned with the minimal-sufficiency objective of the information bottleneck principle, but distinguishes itself by regularizing the propagation process, rather than the resulting representation alone. Relative to Lin et al. (2024); Xu & Markovich (2025); Borovitskiy et al. (2021), our formulation inherits the stochastic diffusion view and extends it to characterize the structured pattern of task-irrelevant propagation, thereby providing a process-level perspective on compact graph learning. Further, it also differs from directly applying spectral filters: instead of performing static frequency selection, our construction models how task-irrelevant components propagate and correlate over the graph, and accordingly induces a structured counteractive response.

2.3 Sufficiency Constraint for Counteractive Diffusion

To prevent the predictive flows from being compromised, we first characterize the role of the counteractive field and the information-theoretic bounds it induces. Consider the Markov chain $X \to \mathbf{H}(t) \to \tilde{\mathbf{H}}(t)$, where X denotes the input graph signal, $\mathbf{H}(t)$ is the drift propagation (*i.e.*, $d\mathbf{H}(t) = f(\mathbf{H}(t))dt$), and $\tilde{\mathbf{H}}(t)$ is the refined diffusion signal obtained with the counteractive field. By the data processing inequality, post-processing cannot increase predictive information, and thus $I(Y;\mathbf{H}(t)) \geq I(Y;\tilde{\mathbf{H}}(t))$. Therefore, preserving predictive flow amounts to minimizing the predictive information loss induced by counteraction, which leads to the following sufficiency objective:

$$\min I(\mathbf{Y};\mathbf{H}(t)) - I(\mathbf{Y};\tilde{\mathbf{H}}(t)). \tag{7}$$

Notably, this does not aim to reduce $I(\mathbf{Y};\mathbf{H}(t))$, instead, it calls for (i) maximizing $I(\mathbf{Y};\mathbf{H}(t))$, (ii) while simultaneously approximating $I(\mathbf{Y};\tilde{\mathbf{H}}(t))$ to $I(\mathbf{Y};\mathbf{H}(t))$. As (i) is consistent with the downstream task, we focus (ii). To make this objective tractable, we introduce the following theorem:

Theorem 1. *Given the Markov chain $X \to \mathbf{H}(t) \to \tilde{\mathbf{H}}(t)$, the following objective is equivalent to Eq. (7). Therefore, optimizing it drives the counteractive field to suppress task-irrelevant propagation while retaining predictive information.*

$$\min H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)) - H(\mathbf{Y} \mid \mathbf{H}(t)). \tag{8}$$

where $H(\mathbf{Y} \mid \mathbf{H}(t))$ and $H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))$ denote the conditional uncertainty of the target given the original and refined diffusion signals, respectively.

On this basis, sufficiency of $\tilde{\mathbf{H}}(t)$ can be characterized by the following proposition.

Proposition 1. *Let $\tilde{\mathbf{H}}(t)$ denote the refined diffusion induced from $\mathbf{H}(t)$ by the counteractive field. Then $\tilde{\mathbf{H}}(t)$ is sufficient for $\mathbf{Y}$ if:*

$$\mathbb{E}_{\mathbf{H}(t)\sim P(\mathbf{H}(t))}\Big[D_{\mathrm{KL}}\big(P(\mathbf{Y} \mid \mathbf{H}(t)) \,\|\, P(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))\big)\Big] \to 0, \tag{9}$$

where D_{KL} denotes KL-divergence between the predictive distributions conditioned on $\mathbf{H}(t)$ and $\tilde{\mathbf{H}}(t)$, $P(\mathbf{H}(t))$ is the distribution of $\mathbf{H}(t)$, from which $\tilde{\mathbf{H}}(t)$ is induced by the counteractive field.

[1] From the Gaussian-process perspective of the counteractive field, covariance functionals naturally characterize the spread of the refined diffusion signal Li et al. (2025), and under such a Gaussian approximation, the log-determinant of its covariance further serves as a natural proxy for entropy up to an additive constant Williams & Rasmussen (2006).

We provide the technical proof for the theorem and proposition in the appendixes. Building on this foundation, the sufficiency objective is formulated as:

$$\mathcal{L}_{\text{suf}} = \mathbb{E}_{\mathbf{H}(t)\sim P(\mathbf{H}(t))}\Big[D_{\text{KL}}\big(P(\mathbf{Y} \mid \mathbf{H}(t)) \,\|\, P(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))\big)\Big]. \tag{10}$$

Complementary to Eq. (6), $\mathcal{L}_{\text{suf}}$ constrains the refined diffusion to preserve predictive sufficiency *w.r.t* the downstream task. Together, they encourage the counteractive field to suppress the task-irrelevant component $f_{\text{irr}}(\tilde{\mathbf{H}}(t))$ while retaining predictive information, yielding a tractable objective for minimal sufficient representations.

2.4 Practical Implementation

We implement the proposed formulation from the stochastic diffusion perspective. As GNN layers amount to discrete approximations of graph diffusion dynamics, their continuous-depth limit induces a graph ODE/SPDE formulation, providing a natural parameterization of the drift term Lin et al. (2024); Xu & Markovich (2025). On this basis, $g(\tilde{\mathbf{H}}(t))$ can be implemented as a graph-correlated signal injected at each diffusion step through an intervention forcing network to mitigate task-irrelevant propagation. Formally, we adopt:

$$\tilde{\mathbf{H}}(t) = \text{ODESolve}\big(\text{GNN}_\theta(\mathbf{H}(0)),\, \Psi_{\theta,\phi},\, [0,t]\big), \tag{11}$$

which represents $\tilde{\mathbf{H}}(t)$ as the solution obtained by integrating the dynamics $\Psi_{\theta,\phi}$ from the initial state $\text{GNN}_\theta(\mathbf{H}(0))$ over $[0,t]$. Here, $\Psi_{\theta,\phi}$ denotes the refined diffusion operator jointly parameterized by the backbone θ and the counteractive field ϕ, and takes the form:

$$\Psi_{\theta,\phi}(\tilde{\mathbf{H}}(t)) = \text{GNN}_\theta\big(F_\theta(\tilde{\mathbf{H}}(t)) + G_\theta(\tilde{\mathbf{H}}(t))\,\mathbf{f}_\phi\big). \tag{12}$$

Here, F_θ denotes deterministic transformation, while G_θ is a state-dependent injector for the counteractive signal $\mathbf{f}_\phi$. Further, $\mathbf{f}_\phi$ can be obtained as:

$$\mathbf{f}_\phi \sim \mathcal{N}\big(0, \Phi_\phi(\Delta)\big), \qquad \Phi_\phi(\Delta) = U\, h_{\phi,r}(\Lambda)\, U^\top. \tag{13}$$

To avoid the eigendecomposition required by $h_{\phi,r}(\Lambda)$, which is computationally prohibitive ($\mathcal{O}(|V|^3)$), we follow the pioneers Borovitskiy et al. (2021); Defferrard et al. (2016); Xu & Markovich (2025) and approximate the Matérn-based kernel with Chebyshev polynomials, yielding the following expansion of the corresponding kernel matrix:

$$h_{\phi,r}(\Delta) \approx \sum_{k=0}^{K} c_{k,r}(\phi)\, T_k(\tilde{\Delta}), \tag{14}$$

Here, $T_k(\tilde{\Delta})$ is the k-th degree Chebyshev polynomial on the rescaled graph Laplacian $\tilde{\Delta}$, and $c_{k,r}(\phi)$ are Chebyshev coefficients induced by $h_{\phi,r}(\lambda) = m_{\phi,r}(\lambda)\left(\frac{2\nu}{\kappa^2} + \lambda\right)^{-\nu}$. This reduces the computational complexity to $\mathcal{O}(K|E|)$, with the approximation error bounded as follows:

Theorem 2: *Given undirected graph $G = \{V, E\}$, the Chebyshev approximation in Eq. (14) admits the following error bound:*

$$\mathcal{O}\left(\left(\frac{\kappa^2 d_{\max}}{16\nu + \kappa^2 d_{\max}}\right)^K\right), \tag{15}$$

where K is the degree of the Chebyshev polynomial and $d_{\max}$ is the maximum degree of the graph.

The proof is provided in the supplementary materials. In practice, to avoid the trivial solution $I(\mathbf{Y};\mathbf{H}(t)) = I(\mathbf{Y};\tilde{\mathbf{H}}(t))$ caused by the degeneration of $I(\mathbf{Y};\mathbf{H}(t))$, we detach $P(\mathbf{Y} \mid \mathbf{H}(t))$ when computing Eq. (10). Finally, we adopt the downstream task objective in addition to the parsimony and sufficiency objectives:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\min} + \beta \cdot \mathcal{L}_{\text{suf}} + \mathcal{L}_{\text{task}}. \tag{16}$$

The overall framework is illustrated in Fig. 1. The left panel depicts step-wise refinement, where the top half illustrates the deterministic diffusion from $t = 0$ to $t = T$ with the drift network, and the bottom half depicts the counteractive signals injected at each step. On the other hand, the right panel

Table 1: **Performance comparison** on the Benchmarking GNN Suite and Long-Range Graph Benchmarks. Best results in bold, second-best underlined.

Datasets	MNIST	CIFAR10	PATTERN	CLUSTER	VOC-SP	COCO-SP	Pep-F	Pep-S
Num. # graph	70K	60K	14K	12K	11.4K	123.3K	15.5K	15.5K
AVG. # nodes	70.6	117.6	118.9	117.2	479.4	476.9	150.9	150.9
AVG. # edges	564.5	941.1	3039.3	2150.9	2710.5	2693.7	307.3	307.3
$\mathcal{L}_{\text{task}}$	CE	CE	WCE	WCE	WCE	WCE	MCE	L1
Metric	ACC ↑	ACC ↑	ACC ↑	ACC ↑	F1 ↑	F1 ↑	AP ↑	MAE ↓
GCN	90.70$_{\pm 0.22}$	55.71$_{\pm 0.38}$	71.89$_{\pm 0.33}$	68.50$_{\pm 0.98}$	0.208$_{\pm 0.003}$	0.134$_{\pm 0.001}$	68.60$_{\pm 0.5}$	0.246$_{\pm 0.001}$
GIN	96.48$_{\pm 0.25}$	55.25$_{\pm 1.53}$	85.39$_{\pm 0.14}$	64.72$_{\pm 1.55}$	0.272$_{\pm 0.005}$	0.213$_{\pm 0.001}$	66.21$_{\pm 0.7}$	0.247$_{\pm 0.002}$
GatedGCN	97.34$_{\pm 0.14}$	67.31$_{\pm 0.31}$	85.57$_{\pm 0.09}$	73.84$_{\pm 0.33}$	0.388$_{\pm 0.004}$	0.292$_{\pm 0.002}$	67.65$_{\pm 0.5}$	0.248$_{\pm 0.001}$
GPS	98.05$_{\pm 0.13}$	72.30$_{\pm 0.36}$	86.68$_{\pm 0.06}$	78.02$_{\pm 0.18}$	0.375$_{\pm 0.011}$	0.341$_{\pm 0.004}$	65.35$_{\pm 0.4}$	0.250$_{\pm 0.001}$
GRIT	98.11$_{\pm 0.11}$	76.47$_{\pm 0.88}$	87.19$_{\pm 0.08}$	80.03$_{\pm 0.28}$	–	–	69.88$_{\pm 0.8}$	0.246$_{\pm 0.001}$
GRED	98.83$_{\pm 0.01}$	76.85$_{\pm 0.19}$	86.76$_{\pm 0.02}$	78.49$_{\pm 0.10}$	–	–	71.33$_{\pm 0.1}$	0.246$_{\pm 0.001}$
Exphormer	98.55$_{\pm 0.03}$	74.69$_{\pm 0.13}$	86.70$_{\pm 0.03}$	78.07$_{\pm 0.04}$	0.398$_{\pm 0.004}$	0.346$_{\pm 0.001}$	65.27$_{\pm 0.4}$	0.248$_{\pm 0.001}$
GMN	98.39$_{\pm 0.21}$	75.76$_{\pm 0.42}$	87.14$_{\pm 0.12}$	–	0.439$_{\pm 0.011}$	0.397$_{\pm 0.010}$	70.71$_{\pm 0.8}$	0.247$_{\pm 0.003}$
CRaWL	97.94$_{\pm 0.05}$	69.01$_{\pm 0.26}$	–	–	–	–	70.74$_{\pm 0.3}$	0.251$_{\pm 0.002}$
NWalker	98.69$_{\pm 0.08}$	76.90$_{\pm 0.46}$	86.97$_{\pm 0.01}$	78.19$_{\pm 0.19}$	0.491$_{\pm 0.004}$	0.440$_{\pm 0.003}$	70.96$_{\pm 0.8}$	0.246$_{\pm 0.001}$
Ours	**99.56**$_{\pm 0.03}$	**80.79**$_{\pm 0.41}$	**88.63**$_{\pm 0.02}$	**80.81**$_{\pm 0.22}$	**0.528**$_{\pm 0.005}$	**0.464**$_{\pm 0.002}$	**73.53**$_{\pm 0.7}$	**0.232**$_{\pm 0.001}$

Table 2: **Performance** on OGB. Best results in bold, second-best underlined.

Dataset	MOLPCBA	PPA	ODE2
Num. # graph	437.9K	158.1K	452.7K
AVG. # nodes	26.0	243.4	125.2
AVG. # edges	28.1	2266.1	124.2
$\mathcal{L}_{\text{task}}$	MCE	CE	CE
Metric	AP ↑	ACC ↑	F1 ↑
GraphTrans	27.61$_{\pm 0.29}$	–	0.183$_{\pm 0.0024}$
SAT	–	75.22$_{\pm 0.56}$	0.194$_{\pm 0.0028}$
GIB†	27.60$_{\pm 0.33}$	71.94$_{\pm 0.77}$	0.169$_{\pm 0.0029}$
GRDL†	28.81$_{\pm 0.30}$	72.78$_{\pm 0.71}$	0.187$_{\pm 0.0032}$
GPS	29.07$_{\pm 0.28}$	80.15$_{\pm 0.33}$	0.189$_{\pm 0.0024}$
CRaWL	29.86$_{\pm 0.25}$	–	–
NeuralWalker	30.86$_{\pm 0.31}$	78.88$_{\pm 0.59}$	0.196$_{\pm 0.0025}$
Ours	**33.31**$_{\pm 0.14}$	**80.89**$_{\pm 0.48}$	**0.216**$_{\pm 0.0024}$

† Implemented under a unified training and evaluation pipeline for fair comparison.

Table 3: **Performance** on large graphs. Best results in bold, second-best underlined.

Dataset	RE	MS	TK
# nodes	22,662	10,000	11,758
# edges	32,927	39,402	519,000
$\mathcal{L}_{\text{task}}$	NNL	NNL	NNL
Metric	ACC ↑	RA ↑	RA ↑
GAT(-sep)	88.75$_{\pm 0.41}$	93.91$_{\pm 0.35}$	83.78$_{\pm 0.43}$
GPS	82.00$_{\pm 0.61}$	90.63$_{\pm 0.67}$	83.71$_{\pm 0.48}$
NAGphormer	74.34$_{\pm 0.77}$	84.19$_{\pm 0.66}$	78.32$_{\pm 0.95}$
Exphormer	89.03$_{\pm 0.37}$	90.74$_{\pm 0.53}$	83.77$_{\pm 0.78}$
Polynormer	92.55$_{\pm 0.37}$	97.46$_{\pm 0.36}$	85.91$_{\pm 0.74}$
GNSD*	88.69$_{\pm 0.91}$	91.69$_{\pm 0.88}$	80.60$_{\pm 1.56}$
SISPDE*	89.33$_{\pm 1.35}$	92.05$_{\pm 1.04}$	82.31$_{\pm 1.26}$
NeuralWalker	92.92$_{\pm 0.36}$	97.82$_{\pm 0.40}$	85.56$_{\pm 0.74}$
Ours	**93.68**$_{\pm 0.40}$	**99.19**$_{\pm 0.38}$	**88.24**$_{\pm 0.68}$

* Re-implemented to the best of our effort under a unified training and evaluation pipeline.

provides a closer view of the refinement process: each $\mathbf{H}(t)$ is induced from $\tilde{\mathbf{H}}(t-1)$ and further refined into $\tilde{\mathbf{H}}(t)$ via $\mathbf{f}_\phi$. Building on this foundation, $\mathcal{L}_{\text{min}}$ enforces parsimonious refinement, while $\mathcal{L}_{\text{suf}}$ steers counteraction against $f_{\text{irr}}(\mathbf{H}(t))$.

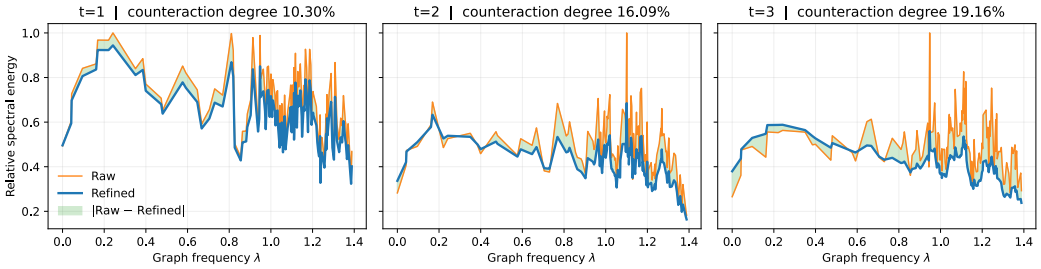
We term the proposed framework CounterDiff, as it learns a compact yet predictive solution by counteracting task-irrelevant propagation within a diffusion-based formulation. In the next section, we empirically validate its effectiveness across diverse tasks.

3 Experiments

3.1 Experimental Setup

Benchmarks and Evaluation Protocols. CounterDiff is evaluated on 14 benchmark datasets: (i) MNIST, CIFAR10, PATTERN, and CLUSTER from the Benchmarking GNN suite Dwivedi et al. (2023); (ii) PascalVOC-SP, COCO-SP, Peptides-func, and Peptides-struct from the Long-Range Graph Benchmarks (LRGB) citeLRGB; (iii) OGBG-Molpcba, OGBG-PPA, and OGBG-Code2 from the Open Graph Benchmarks (OGB) Hu et al. (2020); (iv) Roman Empire (RE), MineSweeper (MS)

Figure 2: Visualization of counteraction behavior. The horizontal axis denotes graph frequency λ, given by the normalized graph-Laplacian eigenvalue, and the vertical axis denotes the relative spectral energy associated with each frequency.



and Tolokers (TK) from the large graph benchmarks Platonov et al. (2023). Dataset details and evaluation protocols are summarized in Tab. 13. We report accuracy (ACC), F1 score (F1), average precision (AP), and ROC AUC (RA) as the respective evaluation metrics.

Implementation Details. We instantiate CounterDiff with Transformer-based Deng et al. (2024) and SSM-based Chen et al. (2025) backbones, while a broader backbone ablation including GCN Defferrard et al. (2016) is presented in the ablation study. For the main comparison, we report the best-performing variant among them. The weights of $\mathcal{L}_{\text{suf}}$ and $\mathcal{L}_{\min}$ (*e.g.,* α and β in Eq. (16)) are set to 0.45 and 1.5 by default. Key hyperparameters are searched over: latent dimension {128, 256, 512}, learning rate [1e^{-4}, 5e^{-3}], and Chebyshev polynomial order {8, 16, 32}. Downstream task objectives $\mathcal{L}_{\text{task}}$ follow the benchmark protocols, where B-/W-/M-CE denote binary/weighted/multi-label cross-entropy and NLL denotes negative log-likelihood. For tasks trained with L1 regression loss, we replace the classification-based sufficiency constraint with a prediction-consistency regularization between the deterministic and refined diffusions. Unless otherwise stated, all results are averaged over three runs, and the remaining settings generally follow Lin et al. (2024); Xu & Markovich (2025).

3.2 Comparison with State-of-the-Arts

We compare CounterDiff with state-of-the-art methods across four categories: (i) graph transformers: Polynormer Deng et al. (2024), NAGphormer Chen et al. (2023), Exphormer Shirzad et al. (2023), SAT Chen et al. (2022), GPS Rampásek et al. (2022), GRIT Ma et al. (2023), and GRED Ding et al. (2024b); (ii) state-space models: GMN Behrouz & Hashemi (2024), Graph-Mamba Wang et al. (2024a), CRaWL Tönshoff et al. (2023), and NeuralWalker Chen et al. (2025); (iii) graph information bottlenecks: including GIB Wu et al. (2020) and GRDL Wang & Fan (2024); and (iv) graph diffusion/SPDE-based models: GNSD Lin et al. (2024) and SISPDE Xu & Markovich (2025). Based on Tab. 1-3, we have the following observations:

(1) CounterDiff yields consistent gains beyond backbone scaling, demonstrating the benefit of learning more parsimonious yet predictive representations.
(2) CounterDiff substantially outperforms GIB-based methods and exhibits a more stable advantage across datasets and tasks, highlighting the benefit of mitigating task-irrelevant propagation at the diffusion-process level rather than relying solely on MI estimation over the final representation.
(3) CounterDiff also substantially outperforms GNSD and SISPDE, which employ stochastic diffusion primarily for uncertainty estimation, demonstrating the benefit of recasting the stochastic term as a counteractive field against task-irrelevant propagation. This further suggests that stochastic diffusion provides a principled perspective for learning compact yet predictive graph representations.

3.3 Ablation Study

Tab. 4 presents the ablation results of CounterDiff. "Basics" refers to the backbone trained only with $\mathcal{L}_{\text{task}}$, which employs the same plain backbone as the corresponding full model on each dataset. The subsequent rows progressively add the counteractive field, $\mathcal{L}_{\min}$, and $\mathcal{L}_{\text{suf}}$.

(1) Employing the counteractive field alone consistently degrades performance (1st-2nd rows), which is expected because an unconstrained intervention may suppress both task-relevant and ir-

Table 4: **Ablation study** of CounterDiff. Results are obtained by averaging five runs.

Setting				CIFAR	Pep-F	OGB-M	RE
Basics	Counter	$\mathcal{L}_{\min}$	$\mathcal{L}_{\text{suf}}$	ACC ↑	AP ↑	AP ↑	ACC ↑
✓	–	–	–	$76.12_{\pm 0.47}$	$70.39_{\pm 0.77}$	$30.27_{\pm 0.30}$	$91.60_{\pm 0.39}$
✓	✓	–	–	$75.82_{\pm 1.08}$	$68.13_{\pm 1.69}$	$27.95_{\pm 1.40}$	$88.81_{\pm 1.21}$
✓	✓	✓	–	$76.05_{\pm 2.29}$	$68.56_{\pm 2.11}$	$27.40_{\pm 1.35}$	$89.31_{\pm 1.92}$
✓	✓	✓	✓	**80.79**$_{\pm 0.41}$	**73.53**$_{\pm 0.7}$	**33.41**$_{\pm 0.14}$	**93.69**$_{\pm 0.40}$

Table 5: **Ablation study** on the backbone choice of CounterDiff.

Setting	CIFAR	Pep-F	OGB-M	RE
Setting	ACC ↑	AP ↑	AP ↑	ACC ↑
GCN	55.71	68.60	24.24	73.69
GCN (ours)	59.03	70.10	25.83	76.34
GT	76.12	70.39	29.05	91.66
GT (ours)	**80.79**	**73.53**	32.07	92.89
SSM	76.29	70.33	30.27	91.60
SSM (ours)	79.89	72.55	**33.41**	**93.69**

relevant components, behaving more like a perturbation than a targeted counteraction.
(2) Employing $\mathcal{L}_{\min}$ partially alleviates this issue but remains insufficient, as a parsimonious objective alone cannot ensure that compression is properly guided (2nd-3rd rows). The larger variance of this setting further confirms this: without sufficiency guidance, the compression may indiscriminately compromise either task-irrelevant or predictive propagation.
(3) Introducing $\mathcal{L}_{\text{suf}}$ yields stable and substantial improvements across all datasets, indicating that sufficiency effectively guides the counteractive field away from predictive information and toward task-irrelevant propagation.

Analysis on Backbone Choice. Further, Tab. 5 shows that CounterDiff consistently improves GCN, graph transformer, and SSM-based backbones across all benchmarks, suggesting that the gains are backbone-agnostic and attributable to the counteractive mechanism itself.

3.4 Understanding CounterDiff Behavior

Visualization. Fig. 2 visualizes the progressive counteraction from $\mathbf{H}(t)$ to $\tilde{\mathbf{H}}(t)$ over three propagation steps on CIFAR. Clearly, counteraction progressively adapts to the refined states inherited from previous steps, with conservative early refinement and stronger later intervention, selectively inhibiting certain propagation patterns while preserving or even enhancing others.

Counteraction of Task-irrelevant Propagation. Motivated by Fig. 2, we further conduct controlled nuisance injection experiments to verify whether such counteraction specifically targets task-irrelevant components while preserving predictive information. We first construct a label-independent nuisance $N = U_{\text{high}}Z$, $Z \sim \mathcal{N}(0, I)$, where U_{high} spans the highest 25% graph-Laplacian frequencies. The nuisance is injected into the raw diffusion at propagation step t with $T = 3$ via $H_{\text{raw}}^{(t,+)} = H_{\text{raw}}^{(t)} + \delta N$, where the scaling factor $\delta = \sqrt{qV(H_{\text{raw}}^{(t)})/V(N)}$ is controlled by information-volume ratio $q = V(N)/V(H_{\text{raw}}^{(t)})$ with $V(H) = \text{tr}(\text{Cov}(H))$. On this basis, we define counteraction degree $\mathcal{C}$ and intervened frequencies counteraction ratio $\tilde{\mathcal{C}}$:

$$\mathcal{C}(t) = 1 - \frac{\sum_\lambda E_{\text{refined}}(\lambda, t)}{\sum_\lambda E_{\text{raw}}(\lambda, t)}, \qquad \tilde{\mathcal{C}} = \frac{\sum_{i \in \mathcal{I}_{\text{high}}} \max\left([E_{\text{raw}}(\lambda_i, t) - E_{\text{refined}}(\lambda_i, t)], 0\right)}{\sum_i \max\left([E_{\text{raw}}(\lambda_i, t) - E_{\text{refined}}(\lambda_i, t)], 0\right)}, \quad (17)$$

where E_{raw} and E_{refined} denote the spectral energies of raw and refined diffusion states. $\tilde{\mathcal{C}}$ quantifies how much of the removed spectral energy is concentrated in the known nuisance subspace. Results are summarized in Tab. 6 and we have the following observations:

(1) Nuisance injection substantially increases $\mathcal{C}$ from 16.09% to 28.02% at $t = 2$, and to 45.04% when further injected at $t = 3$. This adaptive response indicates that **CounterDiff actively counteracts task-irrelevant propagation** rather than employing fixed suppression.
(2) $\tilde{\mathcal{C}}$ increases from 32.4% without nuisance to 46.1% after high-frequency injection, and further to 67.7% under accumulated perturbations. This implies CounterDiff **preferentially concentrates counteraction on the corrupted patterns** rather than apply uniform counteraction.
(3) Despite the repeated intervention and substantially in-

Table 6: Counteraction response under nuisance injection on CIFAR.

	$t=1$	$t=2$		$t=3$		
Injection	$\mathcal{C}$	$\mathcal{C}$	$\tilde{\mathcal{C}}$	$\mathcal{C}$	$\tilde{\mathcal{C}}$	Acc.
None	10.30	16.09	32.4	19.16	28.7	80.79
$t=2$	10.30	28.02	46.1	33.10	35.9	80.53
$t=2,3$	10.30	28.02	46.1	45.04	67.7	79.88

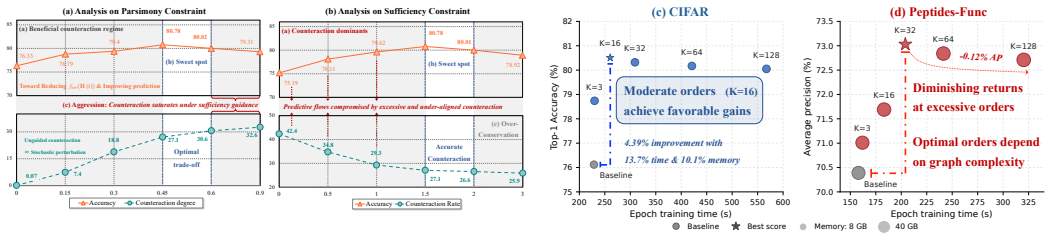


Figure 3: Analysis of CounterDiff regulation and efficiency. Left (a & b): effect of the sufficiency and parsimony objectives on counteraction behavior and predictive performance. Right (c & d): effect of Chebyshev order K on the performance efficiency trade-off.

creased counteraction, accuracy exhibits only slight variations. Together with the increased $\mathcal{C}$ and $\tilde{\mathcal{C}}$, this indicates that **CounterDiff effectively counteracts task-irrelevant propagation while preserving predictive sufficiency.**

How Do $\mathcal{L}_{\text{min}}$ and $\mathcal{L}_{\text{suf}}$ Shape Counteraction? Building on Tab. 6, we further investigate how the sufficiency and parsimony objectives jointly regulate the counteraction behavior. As shown in Fig. 3 (a) & (b), the top and bottom panels report prediction accuracy and counteraction degree, respectively. We fix one coefficient ($\beta = 1.5$ or $\alpha = 0.6$) and vary the other accordingly.

A higher counteraction degree indicates stronger contraction of propagated content, serving as an empirical measure of how CounterDiff balances nuisance mitigation and predictive preservation. As shown in Fig. 6 (a), increasing the parsimony weight α monotonically increases the counteraction degree, while accuracy first improves and then saturates. We identify three regimes:

(1) Beneficial counteraction regime ($\alpha \in [0, 0.45]$). Counteraction increases together with accuracy, suggesting that under the sufficiency constraint, additional counteraction primarily removes detrimental propagation while preserving predictive information.
(2) Sweet spot ($\alpha \in [0.45, 0.6]$). CounterDiff achieves the optimal balance between compactness and predictive sufficiency, obtaining substantial refinement with near-optimal performance.
(3) Aggressive regime ($\alpha \in [0.6, 0.9]$). Further counteraction provides diminishing returns and starts to degrade accuracy, revealing an empirical counteraction capacity under the sufficiency constraint, beyond which additional refinement may compromise predictive information.

Based on the above discussion, Fig. 3 (b) further reveals how the sufficiency objective regulates counteraction behavior:

(1) Without sufficiency constraint ($\beta = 0$). CounterDiff achieves the highest counteraction degree but performs worse than the unrefined baseline. This is expected, as the absence of $\mathcal{L}_{\text{suf}}$ favors stronger counteraction, which may remove predictive structures.
(2) Increasing β improves performance while slightly decreases the degree, suggesting $\mathcal{L}_{\text{suf}}$ calibrates counteraction by preventing unconstrained removal of useful information.
(3) The accuracy peaks at $\beta \in [1.5, 2.0]$, confirming the sweet spot identified in Fig. 3 (b). Excessively increasing β makes the model overly conservative and limits effective refinement.

Analysis of Chebyshev Order K. As K controls the spectral resolution of the counteractive field, we examine whether larger orders consistently improve CounterDiff. As shown in Fig. 3 (c) & (d), increasing K initially improves performance, but the gains quickly saturate while runtime and memory continue to grow. We conjecture excessive orders introduce redundant spectral responses and unnecessary variations, making moderate orders favorable for balancing performance and efficiency.

4 Conclusion

In this work, we revisit compact graph learning from the perspective of information propagation, and argue that compactness should be enforced not merely from compressing graph structures or representations, but from inhibiting task-irrelevant diffusion during message passing. To this end, we propose CounterDiff, a propagation-refinement framework grounded in a graph SPDE formulation. By decomposing the deterministic drift into predictive flow and task-irrelevant diffusion, and reinterpreting the stochastic term as a Graph Matérn Counteractive Field, CounterDiff shifts compact graph learning from post-hoc compression toward direct intervention in the propagation process.

5 AI USE STATEMETN

Generative AI tools were used solely to improve the readability of the manuscript and identify typographical errors. We have not used generative AI tools for research, ideation, methodology, or experiments. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 119–130, 2024.

Ali Behrouz, Parsa Delavari, and Farnoosh Hashemi. Unsupervised representation learning of brain activity via bridging voxel activity and functional connectivity. In *Proceedings of the International Conference on Machine Learning*, 2024.

Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. Matérn gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pp. 2593–2601, 2021.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International conference on machine learning*, pp. 1407–1418. PMLR, 2021.

Dexiong Chen, Leslie O'Bray, and Karsten M. Borgwardt. Structure-aware transformer for graph representation learning. In *Proceedings of the International Conference on Machine Learning*, volume 162, pp. 3469–3489, 2022.

Dexiong Chen, Till Hendrik Schulz, and Karsten M. Borgwardt. Learning long range dependencies on graphs via random walks. In *Proceedings of the International Conference on Learning Representations*, 2025.

Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *Proceedings of the International Conference on Learning Representations*, 2018.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. Nagphormer: A tokenized graph transformer for node classification in large graphs. In *Proceedings of the International Conference on Learning Representations*, 2023.

John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical proceedings of the Cambridge philosophical society*, volume 43, pp. 50–67, 1947.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Laurent Demanet and Lexing Ying. On chebyshev interpolation of analytic functions. *preprint*, 2010.

Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In *Proceedings of the International Conference on Learning Representations*, 2024.

Shifei Ding, Wei Du, Ling Ding, Lili Guo, and Jian Zhang. Learning efficient and robust multi-agent communication via graph information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17346–17353, 2024a.

Yuhui Ding, Antonio Orvieto, Bobby He, and Thomas Hofmann. Recurrent distance filtering for graph representation learning. In *Proceedings of the International Conference on Machine Learning*, 2024b.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *J. Mach. Learn. Res.*, 24:43:1–43:48, 2023.

Moshe Eliasof, Eldad Haber, and Eran Treister. Pde-gcn: Novel architectures for graph neural networks motivated by partial differential equations. *Advances in neural information processing systems*, 34:3836–3849, 2021.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *Proceedings of the World Wide Web Conference*, pp. 417–426, 2019.

Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks V. S. Lakshmanan, and Xuemin Lin. Efficient algorithms for densest subgraph discovery. *Proc. VLDB Endow.*, 12(11):1719–1732, 2019a.

Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks VS Lakshmanan, and Xuemin Lin. Efficient algorithms for densest subgraph discovery. *arXiv preprint arXiv:1906.00341*, 2019b.

Aristides Gionis and Charalampos E. Tsourakakis. Dense subgraph discovery: KDD 2015 tutorial. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 2313–2314, 2015.

Paolo Giudici and Emanuela Raffinetti. Safe artificial intelligence in finance. *Finance Research Letters*, 56: 104088, 2023.

Van Thuy Hoang and O-Joun Lee. Pre-training graph neural networks on molecules by using subgraph-conditioned graph information bottleneck. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17204–17213, 2025.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.

Nikhil S Ketkar, Lawrence B Holder, and Diane J Cook. Subdue: Compression-based frequent pattern discovery in graph data. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pp. 71–76, 2005.

Shigui Li, Wei Chen, and Delu Zeng. EVODiff: Entropy-aware variance optimized diffusion inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Katharina Limbeck, Lydia Mezrag, Guy Wolf, and Bastian Rieck. Geometry-aware edge pooling for graph neural networks. In *Advances in Neural Information Processing Systems*, 2025.

Xixun Lin, Wenxiao Zhang, Fengzhao Shi, Chuan Zhou, Lixin Zou, Xiangyu Zhao, Dawei Yin, Shirui Pan, and Yanan Cao. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 30457–30478, 2024.

Jiayi Luo, Qingyun Sun, Beining Yang, Haonan Yuan, Xingcheng Fu, Yanbiao Ma, Jianxin Li, and Philip S. Yu. Robust graph condensation via classification complexity mitigation. In *Advances in Neural Information Processing Systems*, 2025.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip H. S. Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *Proceedings of the International Conference on Machine Learning*, volume 202, pp. 23321–23337, 2023.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *Proceedings of the International Conference on Learning Representations*, 2023.

Ladislav Rampásek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, 2022.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *Proceedings of the International Conference on Machine Learning*, volume 202, pp. 31613–31632, 2023.

Eitan Tadmor. The exponential accuracy of fourier and chebyshev differencing methods. *SIAM Journal on Numerical Analysis*, 23(1):1–10, 1986.

Naftali Tishby, Fernando C. N. Pereira, and William Bialek. The information bottleneck method. *CoRR*, physics/0004057, 2000.

Jan Tönshoff, Martin Ritzert, Hinrikus Wolf, and Martin Grohe. Walking out of the weisfeiler leman hierarchy: Graph learning beyond message passing. *Trans. Mach. Learn. Res.*, 2023, 2023.

Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, Philadelphia, PA, 2019.

Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. volume abs/2402.00789, 2024a.

Yan Wang, Wenju Hou, Nan Sheng, Ziqi Zhao, Jialin Liu, Lan Huang, and Juexin Wang. Graph pooling in graph neural networks: methods and their applications in omics studies. *Artificial Intelligence Review*, 57 (11):294, 2024b.

Zixiao Wang and Jicong Fan. Graph classification via reference distribution learning: Theory and practice. In *Advances in Neural Information Processing Systems*, 2024.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. In *Advances in Neural Information Processing Systems*, 2020.

Fred Xu and Thomas Markovich. Uncertainty estimation on graphs with structure informed stochastic partial differential equations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the International Conference on Data Mining*, pp. 721–724, 2002.

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition. In *Proceedings of the International Conference on Learning Representations*, 2021.

Guibin Zhang, Xiangguo Sun, Yanwei Yue, Chonghe Jiang, Kun Wang, Tianlong Chen, and Shirui Pan. Graph sparsification via mixture of graphs. In *International Conference on Learning Representations*, 2025.

Xiao-Dong Zhang. The laplacian eigenvalues of graphs: a survey. gerald d. ling (ed.), linear algerbra research advances, 2007.

Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. Robust graph representation learning via neural sparsification. In *Proceedings of the International Conference on Machine Learning*, volume 119, pp. 11458–11468, 2020.

6 SUFFICIENCY CHARACTERIZATION OF COUNTERACTIVE REFINEMENT

In this section, we provide the theoretical justification for the sufficiency alignment in CounterDiff. We first prove Theorem 1 by showing that the mutual-information objective is equivalent to a conditional-entropy formulation. Based on this result, we then establish Proposition 1, which characterizes sufficiency through a KL-based predictive consistency criterion.

Theorem 1. *Given the Markov chain $X \to \mathbf{H}(t) \to \tilde{\mathbf{H}}(t)$, the following objective is equivalent to Eq. (7). Therefore, optimizing it drives the counteractive field to suppress task-irrelevant propagation while retaining predictive information.*

$$\min H(\mathbf{Y} \mid \mathbf{H}(t)) - H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)). \tag{18}$$

where $H(\mathbf{Y} \mid \mathbf{H}(t))$ and $H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))$ denote the conditional uncertainty of the target given the original and refined diffusion signals, respectively.

Proof. Recall the mutual-information objective in Eq. (7):

$$\min\ I(\mathbf{Y}; \mathbf{H}(t)) - I(\mathbf{Y}; \tilde{\mathbf{H}}(t)). \tag{19}$$

Since mutual information is symmetric, we have

$$I(\mathbf{Y}; \mathbf{H}(t)) = I(\mathbf{H}(t); \mathbf{Y}), \quad I(\mathbf{Y}; \tilde{\mathbf{H}}(t)) = I(\tilde{\mathbf{H}}(t); \mathbf{Y}). \tag{20}$$

Using the standard identity, we have:

$$I(\mathbf{Y}; \mathbf{H}(t)) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{H}(t)), \quad I(\mathbf{Y}; \tilde{\mathbf{H}}(t)) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)). \tag{21}$$

Combining with Eq. (19), we show:

$$\begin{aligned} I(\mathbf{Y}; \mathbf{H}(t)) - I(\mathbf{Y}; \tilde{\mathbf{H}}(t)) &= \big(H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{H}(t))\big) - \big(H(\mathbf{Y}) - H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))\big) \\ &= H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)) - H(\mathbf{Y} \mid \mathbf{H}(t)). \end{aligned}$$

where the marginal entropy $H(\mathbf{Y})$ cancels out, yielding

$$I(\mathbf{Y}; \mathbf{H}(t)) - I(\mathbf{Y}; \tilde{\mathbf{H}}(t)) = H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)) - H(\mathbf{Y} \mid \mathbf{H}(t)). \tag{22}$$

Therefore, minimizing the mutual-information gap is equivalent to minimizing the corresponding conditional-entropy gap, namely

$$\min\ I(\mathbf{Y}; \mathbf{H}(t)) - I(\mathbf{Y}; \tilde{\mathbf{H}}(t)) \quad \Longleftrightarrow \quad \min\ H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)) - H(\mathbf{Y} \mid \mathbf{H}(t)), \tag{23}$$

which is the same as Eq. (8) up to a trivial reordering of terms. This proves Theorem 1. Moreover, consider the Markov chain $X \to \mathbf{H}(t) \to \tilde{\mathbf{H}}(t)$, the data processing inequality implies $I(\mathbf{Y}; \mathbf{H}(t)) \geq I(\mathbf{Y}; \tilde{\mathbf{H}}(t))$. Therefore, Eq. (8) is equivalent to making the refined diffusion retain as much predictive information as possible from the original diffusion.

Proposition 1. *Let $\tilde{\mathbf{H}}(t)$ denote the refined diffusion induced from $\mathbf{H}(t)$ by the counteractive field. Then $\tilde{\mathbf{H}}(t)$ is sufficient for $\mathbf{Y}$ if:*

$$\mathbb{E}_{\mathbf{H}(t) \sim P(\mathbf{H}(t))} \Big[D_{\mathrm{KL}}\big(P(\mathbf{Y} \mid \mathbf{H}(t)) \,\|\, P(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))\big) \Big] \to 0, \tag{24}$$

where D_{KL} denotes KL-divergence between the predictive distributions conditioned on $\mathbf{H}(t)$ and $\tilde{\mathbf{H}}(t)$, $P(\mathbf{H}(t))$ is the distribution of $\mathbf{H}(t)$, from which $\tilde{\mathbf{H}}(t)$ is induced by the counteractive field.

Proof. Since $\tilde{\mathbf{H}}(t)$ is induced from $\mathbf{H}(t)$, *i.e.,* $\mathbf{H}(t) \to \tilde{\mathbf{H}}(t)$, we have $\mathbf{Y} \perp \tilde{\mathbf{H}}(t) \mid \mathbf{H}(t)$; for brevity, the expectation over $\tilde{\mathbf{H}}(t) \mid \mathbf{H}(t)$ is implicit below. We consider:

$$\mathbb{E}_{\mathbf{H}(t) \sim P(\mathbf{H}(t))} \Big[D_{\mathrm{KL}}\big(P(\mathbf{Y} \mid \mathbf{H}(t)) \,\|\, P(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))\big) \Big]. \tag{25}$$

Expanding the KL-divergence gives

$$\begin{aligned} &\mathbb{E}_{\mathbf{H}(t),\, \tilde{\mathbf{H}}(t)} \Big[D_{\mathrm{KL}}\big(P(\mathbf{Y} \mid \mathbf{H}(t)) \,\|\, P(\mathbf{Y} \mid \tilde{\mathbf{H}}(t))\big) \Big] \\ &= \mathbb{E}_{\mathbf{H}(t),\, \tilde{\mathbf{H}}(t),\, \mathbf{Y}} \Big[\log P(\mathbf{Y} \mid \mathbf{H}(t)) - \log P(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)) \Big]. \end{aligned} \tag{26}$$

By separating the two terms, we obtain

$$\begin{aligned} &\mathbb{E}_{\mathbf{H}(t),\, \tilde{\mathbf{H}}(t),\, \mathbf{Y}} [\log P(\mathbf{Y} \mid \mathbf{H}(t))] - \mathbb{E}_{\mathbf{H}(t),\, \tilde{\mathbf{H}}(t),\, \mathbf{Y}} \Big[\log P(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)) \Big] \\ &= - H(\mathbf{Y} \mid \mathbf{H}(t)) + H(\mathbf{Y} \mid \tilde{\mathbf{H}}(t)). \end{aligned} \tag{27}$$

Hence,

$$\mathbb{E}_{\mathbf{H}(t)\sim P(\mathbf{H}(t))}\Big[D_{\mathrm{KL}}\big(P(\mathbf{Y}\mid\mathbf{H}(t))\,\|\,P(\mathbf{Y}\mid\tilde{\mathbf{H}}(t))\big)\Big] = H(\mathbf{Y}\mid\tilde{\mathbf{H}}(t)) - H(\mathbf{Y}\mid\mathbf{H}(t)). \tag{28}$$

By Theorem 1, the right-hand side is exactly the conditional-entropy gap corresponding to the mutual-information objective $I(\mathbf{Y};\mathbf{H}(t)) - I(\mathbf{Y};\tilde{\mathbf{H}}(t))$. Consider the data processing inequality, $I(\mathbf{Y};\mathbf{H}(t)) \geq I(\mathbf{Y};\tilde{\mathbf{H}}(t))$, or equivalently, $H(\mathbf{Y}\mid\tilde{\mathbf{H}}(t)) \geq H(\mathbf{Y}\mid\mathbf{H}(t))$. Therefore, if

$$\mathbb{E}_{\mathbf{H}(t)\sim P(\mathbf{H}(t))}\Big[D_{\mathrm{KL}}\big(P(\mathbf{Y}\mid\mathbf{H}(t))\,\|\,P(\mathbf{Y}\mid\tilde{\mathbf{H}}(t))\big)\Big] \to 0, \tag{29}$$

then $H(\mathbf{Y}\mid\tilde{\mathbf{H}}(t)) - H(\mathbf{Y}\mid\mathbf{H}(t)) \to 0$, which, by Theorem 1, is equivalent to

$$I(\mathbf{Y};\mathbf{H}(t)) - I(\mathbf{Y};\tilde{\mathbf{H}}(t)) \to 0. \tag{30}$$

This implies that the refined diffusion retains the same predictive information about $\mathbf{Y}$ as the original diffusion. Hence, $\tilde{\mathbf{H}}(t)$ is sufficient for $\mathbf{Y}$. This proves Proposition 1.

7 Chebyshev Approximation of the Counteractive Field

In Section 3, we employ a Chebyshev polynomial approximation for $\Phi_{\phi,r}(\Delta)$, since eigendecomposition of the graph Laplacian Δ is computationally prohibitive. In this section, we provide the theoretical justification for the approximation bound in Theorem 2. Since the counteractive field is instantiated as a spectral operator and does not introduce additional non-analytic transformations beyond the parameterized spectral response, its approximation error can be analyzed within the same Chebyshev framework as prior Matérn-based graph filters Borovitskiy et al. (2021); Defferrard et al. (2016); Xu & Markovich (2025).

Following standard Chebyshev approximation analysis, we assume that $m_{\phi,r}(\lambda)$ is a bounded, analytic spectral filter with no closer singularity than that of the Matérn base term.

Theorem 2: *Given undirected graph $G = \{V, E\}$, the Chebyshev approximation in Eq. (14) admits the following error bound:*

$$\mathcal{O}\left(\left(\frac{\kappa^2 d_{\max}}{16\nu + \kappa^2 d_{\max}}\right)^K\right), \tag{31}$$

where K is the degree of the Chebyshev polynomial and $d_{\max}$ is the maximum degree of the graph.

Proof. In our formulation, the function to approximate on each channel r is:

$$h_{\phi,r}(\lambda) = m_{\phi,r}(\lambda)\left(\frac{2\nu}{\kappa^2} + \lambda\right)^{-\nu}. \tag{32}$$

Compared with the standard Matérn kernel, the additional factor $m_{\phi,r}(\lambda)$ shapes the spectral distribution of the counter signal. Since the nearest singularity of $h_{\phi,r}$ is at $\lambda_s = -\frac{2\nu}{\kappa^2}$, which lies outside the non-negative spectrum of the graph Laplacian, $h_{\phi,r}(\lambda)$ is analytic over the spectral domain of interest.

Next, we rescale the spectrum of Δ to the standard Chebyshev approximation domain:

$$\tilde{\Delta} = \frac{2\Delta - (\lambda_{\max} + \lambda_{\min})I}{\lambda_{\max} - \lambda_{\min}}, \quad \tilde{\lambda} = \frac{2\lambda - (\lambda_{\max} + \lambda_{\min})}{\lambda_{\max} - \lambda_{\min}}. \tag{33}$$

To bound the approximation error, we consider the Bernstein ellipse E_ρ in the complex plane. In Chebyshev approximation, the convergence rate is governed by the nearest singularity, equivalently by the ellipse parameter ρ. That is, a larger ρ leads to a faster exponential decay of the Chebyshev approximation error Trefethen (2019).

Since $h_{\phi,r}$ is analytic on E_ρ, the classical Chebyshev approximation bound Demanet & Ying (2010); Tadmor (1986) implies that there exists a degree-K polynomial

$$p_{K,\phi,r}(\tilde{\lambda}) = \sum_{k=0}^{K} c_{k,r}(\phi)\, T_k(\tilde{\lambda}), \tag{34}$$

that has the following approximation error on the rescaled interval $[-1, 1]$:

$$|h_{\phi,r}(x) - p_{K,\phi,r}(x)| \leq C\frac{M_{\phi,r}(\rho)}{\rho - 1}\rho^{-K}. \tag{35}$$

Here, $M_{\phi,r}(\rho) = \max_{z\in E_\rho} |h_{\phi,r}(z)|$ and can be integrated into a constant $C_{\phi,r} > 0$. Therefore, the approximation error can be simplified as:

$$\|h_{\phi,r} - p_{K,\phi,r}\|_\infty \leq C_{\phi,r}\rho^{-K}. \tag{36}$$

To make the convergence factor explicit, we rescale the singularity $\lambda_s = -\frac{2\nu}{\kappa^2}$ using Eq. (33) as:

$$\tilde{\lambda}_s = \frac{2\lambda_s - (\lambda_{\max} + \lambda_{\min})}{\lambda_{\max} - \lambda_{\min}} = -\frac{4\nu}{\kappa^2 \lambda_{\max}} - 1. \tag{37}$$

Using the standard bound in Zhang (2007), we further have:

$$\tilde{\lambda}_s \leq -\frac{8\nu}{\kappa^2 d_{\max}} - 1, \tag{38}$$

where d_{max} denotes the maximum degree of the graph. On this basis, the distance from the spectrum of Δ to the nearest singularity can be given as:

$$\rho = |\tilde{\lambda}_s| + \sqrt{\tilde{\lambda}_s^2 - 1} \;\geq\; \frac{8\nu}{\kappa^2 d_{\max}} + 1 + \sqrt{\left(\frac{8\nu}{\kappa^2 d_{\max}} + 1\right)^2 - 1} \;\geq\; \frac{16\nu + \kappa^2 d_{\max}}{\kappa^2 d_{\max}}. \tag{39}$$

Applying the approximation bound from Trefethen (2019) to Eq. (39) & (35), we have:

$$\|h_{\phi,r} - p_{K,\phi,r}\|_\infty = \mathcal{O}(\rho^{-K}) = \mathcal{O}\left(\left(\frac{\kappa^2 d_{\max}}{16\nu + \kappa^2 d_{\max}}\right)^K\right) \tag{40}$$

Compared with Trefethen (2019); Lin et al. (2024); Xu & Markovich (2025), the noise counter inherits the same exponential Chebyshev convergence behavior as the base Matérn kernel, while allowing potentially tighter bounds through the control of $M_{\phi,r}(\rho)$, which naturally aligns with the cancellation of task-irrelevant information.

8 Preliminary

8.1 Compact Graph Representation Learning

Compact graph representation learning seeks representations that preserve task-relevant information and structures while discarding task-irrelevant content. Beyond performance gains, this objective further promises improved generalization, adversarial robustness, and causal interpretability, making it a central concern across diverse domains.

To this end, traditional methods rely on hand-crafted criteria such as highest density Fang et al. (2019b) or common substructures across graphs Ketkar et al. (2005) to identify components for retention or removal. With the rapid progress in graph neural networks, more recent works learn to compress graphs in a data-driven manner, including graph sparsification Zhang et al. (2025), pooling Wang et al. (2024b), and condensation Luo et al. (2025). Despite their empirical effectiveness, existing analyses suggest that these methods often optimize for the task at the expense of discarding fundamental graph structures Limbeck et al. (2025), primarily due to the lack of a unified characterization of task-irrelevant propagation.

To complement this, another line of research Yu et al. (2021); Hoang & Lee (2025); Wang & Fan (2024); Ding et al. (2024a) follows the information bottleneck (IB) principle Tishby et al. (2000), which seeks a representation Z that preserves information relevant to the target Y while discarding unnecessary details from the input X:

$$\mathcal{L}_{IB} := I(X; Z) - \beta \cdot I(Y; Z). \tag{41}$$

By leveraging or reformulating Eq. (41), these approaches attempt to extract critical subgraphs or compressed representations incorporating sufficient information about labels as the results. Despite their brilliant achievements, faithfully instantiating IB on graphs remains challenging, since existing estimation paradigms are mainly developed under Euclidean *i.i.d.* settings and are difficult to align with non-Euclidean topology and inter-node dependencies Wu et al. (2020). More importantly, even with a faithful adaptation, they still struggle at separating the broadcast detrimental signals from task-relevant cues, which inevitably intensifies the risk to cause collateral damage to task-relevant structures.

8.2 Graph Diffusion and Stochastic Propagation

Graph diffusion provides a fundamental continuous-time view for graph representation learning Eliasof et al. (2021). By interpreting graph topology and layer-wise propagation as spatial and temporal discretizations, prior works reformulate message passing with differential operators for various tasks. Given a graph G and its node embeddings $\mathbf{H}$, a typical diffusion equation describes message passing as Chamberlain et al. (2021):

$$\frac{\partial \mathbf{H}(t)}{\partial t} = \mathrm{div}[\mathcal{G}_\theta(\mathbf{H}(t), t)\nabla \mathbf{H}(t)], \tag{42}$$

where div and ∇ are the divergence and gradient operators, respectively. $\mathcal{G}_\theta$ denotes a GNN parameterized by θ, with t indexing the propagation depth. Eq. (42) interprets message passing as a graph diffusion process, analogous to particle diffusion $\frac{\partial \mathbf{X}}{\partial t} = \Delta(G(\mathbf{X}, t)\mathbf{X})$ Crank & Nicolson (1947), thus rewrites feature propagation as a partial differential equation. On this basis, subsequent works Lin et al. (2024); Xu & Markovich (2025) further extend the deterministic graph diffusion to stochastic propagation for uncertainty estimation:

$$d\mathbf{H}(t) = \underbrace{f(\mathbf{H}(t))dt}_{\textit{Drift term}} + \underbrace{g(\mathbf{H}(t))d\mathbf{W}(t)}_{\textit{Stochastic term}}. \tag{43}$$

Here, the drift term models the deterministic graph diffusion process, while the stochastic term serves as a structured carrier of perturbation, quantifying epistemic uncertainty on graphs. In particular, the key is not to restrict the stochastic term to a specific form, but to employ a sound prior to capture the targeted propagation, which can then be approximated in a computationally tractable manner. This suggests that the propagation of task-irrelevant signals can be characterized as a structured diffusion process on graphs. Motivated by this, we build on the stochastic diffusion perspective of GNSD Lin et al. (2024), but reinterpret $g(\mathbf{H}(t))\, d\mathbf{W}(t)$ as a structured carrier of task-irrelevant propagation.