# KGMark: A Diffusion Watermark for Knowledge Graphs

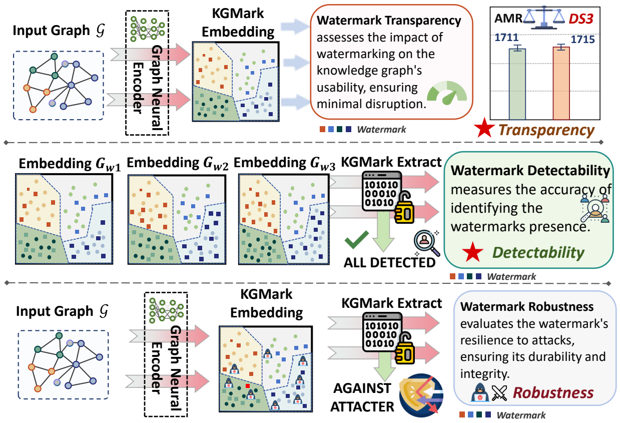
Hongrui Peng [*] [1] Haolang Lu [*] [1] Yuanlong Yu [1] Weiye Fu [1] Kun Wang [2] Guoshun Nan [1]

## Abstract

Knowledge graphs (KGs) are ubiquitous in numerous real-world applications, and watermarking facilitates protecting intellectual property and preventing potential harm from AI-generated content. Existing watermarking methods mainly focus on static plain text or image data, while they can hardly be applied to dynamic graphs due to spatial and temporal variations of structured data. This motivates us to propose KGMark, the first graph watermarking framework that aims to generate robust, detectable, and transparent diffusion fingerprints for dynamic KG data. Specifically, we propose a novel clustering-based alignment method to adapt the watermark to spatial variations. Meanwhile, we present a redundant embedding strategy to harden the diffusion watermark against various attacks, facilitating the robustness of the watermark to the temporal variations. Additionally, we introduce a novel learnable mask matrix to improve the transparency of diffusion fingerprints. By doing so, our KGMark properly tackles the variation challenges of structured data. Experiments on various public benchmarks show the effectiveness of our proposed KGMark.

## 1. Introduction

The growing adoption of generative models has significantly expanded the creation and utilization of synthetic data (Bauer et al., 2024), including structured formats such as time-series data (Das et al., 2024), tabular data (Vero et al., 2024), and graphs (Han et al., 2025; Wang et al., 2023). Among these, Knowledge graphs (KGs) (Ji et al., 2022) are especially crucial due to their ability to represent complex relationships and semantic hierarchies (An et al., 2025), making them indispensable for applications such as semantic search (Chen et al., 2024), question answering (Yin et al., 2024; Zhou et al., 2024), and recommendation systems (Fan et al., 2019; Wang et al., 2019). Deep learning-based models such as GraphRNN (You et al., 2018), GraphVAE (Simonovsky & Komodakis, 2018), MolGAN (Cao & Kipf, 2022), and DiGress (Vignac et al., 2023) have emerged as powerful tools for generating high-quality synthetic graphs, supporting applications such as graph classification, molecular design, and structure-preserving data augmentation in machine learning pipelines.



Watermark Transparency assesses the impact of watermarking on the knowledge graph's usability, ensuring minimal disruption.

Watermark Detectability measures the accuracy of identifying the watermarks presence.

Watermark Robustness evaluates the watermark's resilience to attacks, ensuring its durability and integrity.

*Figure 1.* Overview of our KGMark. KGMark implements a KGE watermarking scheme that preserves transparency, enables reliable detection, and remains robust against various post-editing attacks.

However, existing synthetic graph generation methods may inadvertently embed biases (Shomer et al., 2023; Dong et al., 2022) or misleading (Yang et al., 2024a) information and are vulnerable to malicious alterations by attackers (Zhang et al., 2019), potentially introducing harmful content that compromises analyses or even facilitates the exploitation of real-world systems (Jiang et al., 2024; Wang et al., 2025). Furthermore, presenting synthetic graphs as original research can violate intellectual property rights, undermining trust in academic and commercial environments (Smits & Borghuis, 2022; Zhu et al., 2024b). Ensuring *traceability*, *integrity*, and *copyright* protection of synthetic counterparts causes a significant amount of compute and memory footprints due to the notorious designs (LIANG et al., 2024; Charpenay & Schockaert, 2025), which even unnerves large corporations, let alone individual researchers.

Watermarking techniques (Radford et al., 2021; Podell et al., 2023) have proven to be effective in ensuring the authentic-

[*]Equal contribution [1]Beijing University Of Posts and Telecommunications, Beijing, China [2]Nanyang Technological University, Singapore. Correspondence to: Guoshun Nan <nanguo2021@bupt.edu.cn>, Kun Wang <wang.kun@ntu.edu.sg>.

*Proceedings of the $42^{nd}$ International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

ity of synthetic data in image and text generation (Wen et al., 2023; Zhao et al., 2023a). However, off-the-shelf watermarking algorithms (Liu et al., 2024) appear to lag behind in the era of graph structures, despite their increasing importance in various domains. This gap makes watermarking for KGs an unsolved and looming challenge.

The core challenges of graph watermarking lie in achieving robustness against spatial-temporal variations and maintaining transparency. Embedded watermarks must preserve KG semantics while resisting structural perturbations caused by dynamic node/edge updates (e.g., temporal evolution in recommendation systems (Wang et al., 2019)) and graph isomorphism (e.g., node relabeling or spatial rearrangements (Bouritsas et al., 2023)). Conventional watermarking methods (Bouritsas et al., 2023), designed for static and unstructured data, often exhibit limited robustness under the spatial-temporal variations of dynamic KGs, posing challenges in mitigating risks of AI-generated KG misuse (Liu et al., 2025). The inherent heterogeneity of knowledge graphs further necessitates embedding watermarks at the knowledge graph embedding (KGE) level to reconcile semantic fidelity with detection resilience (Le et al., 2024).

To address these challenges, we propose KGMark, the first watermarking scheme specifically designed for knowledge graphs, based on diffusion models. KGMark employs graph alignment with a learnable mask matrix to spatially adapt watermarks to structural variations, ensuring seamless integration between topology and embedded signals. For temporal robustness, we design redundant embedding strategies coupled with likelihood estimation, enabling resilient extraction despite incremental graph updates. KGMark requires no prior assumptions about graph stability or isomorphism, making it agnostic to node ordering and scalable to evolving KGs. The watermark is detectable exclusively through secure keys, ensuring resistance to adversarial attacks while preserving KG utility (Nandi et al., 2024).

Through rigorous testing, we demonstrate that KGMark achieves: ❶ high detectability, with a watermark detection AUC up to 0.99; ❷ maintaining KG quality and limiting downstream task performance loss to within the range of $0.02\% \sim 9.7\%$; and ❸ high robustness, retaining an AUC of around $0.95$ against various post-editing attacks.

Briefly, our key contributions are summarized as follows:

➠ We introduce KGMark, *the first watermarking scheme designed for KGE,* embedding robust and transparent watermarks into graph structures to protect intellectual property and ensure data integrity.

➠ We propose effective solutions to key challenges, including redundancy-based embedding and the Learnable Adaptive Watermark Matrix, which significantly enhance the performance of KGMark.

➠ We demonstrate through rigorous experiments the superior performance of KGMark, showing its robustness, transparency, and high detectability across various post-editing attack scenarios.

## 2. Related Work

### 2.1. Watermarking AI-generated Content

As AI technologies continue to evolve and generate increasingly sophisticated content, the necessity for robust watermarking solutions for AI-generated content has gained prominence (Zhao et al., 2023b; Zhu et al., 2024a). This technique serves a critical role in addressing issues such as copyright infringement (Zhang et al., 2024b), misinformation, and the ethical implications surrounding the use of AI in creative processes. In particular, watermarking helps to distinguish AI-generated work from human-created content (Asnani et al., 2024), thereby enhancing credibility in the digital landscape (Barman et al., 2024).

### 2.2. Watermarking Synthetic Unstructured Data

Watermarking synthetic unstructured data, such as plain text (Dathathri et al., 2024) and images (Zhang et al., 2024a; Wen et al., 2024), is crucial for ensuring data integrity, protecting intellectual property, and preventing misuse. Common methods include pre-applying watermarks to training datasets, enabling models to generate inherently watermarked data (Yu et al., 2021; Zhao et al., 2023c), or modifying the sampling process in large language models to subtly alter output word distributions and encode watermarks (Kirchenbauer et al., 2023). A widely used approach in diffusion models involves embedding watermarks into the initial noise vector or latent space without retraining (Wen et al., 2024; Yang et al., 2024b; Yang, 2024), ensuring transparency and robustness through reversibility. In this work, we conduct the first systematic exploration of adaptively incorporating this method into knowledge graphs.

## 3. Proposed Method: KGMark

### 3.1. Preliminary of Latent Diffusion Model

The core idea of the Diffusion Model can be summarized in two phases: the forward diffusion process and the reverse diffusion process (Chang et al., 2023; Rombach et al., 2021). In the forward diffusion process, the latent representation $\mathcal{Z}_0$ is progressively corrupted by Gaussian noise over $\mathcal{T}$ steps, resulting in:

$$q(\mathcal{Z}_t|\mathcal{Z}_{t-1}) = \mathcal{N}(\mathcal{Z}_t; \sqrt{1-\beta_t}\mathcal{Z}_{t-1}, \beta_t I) \tag{1}$$

As $t \to \mathcal{T}$, $\mathcal{Z}_\mathcal{T}$ converges to an approximate *standard Gaussian distribution*. The reverse diffusion process starting from noise $\mathcal{Z}_\mathcal{T}$, the model denoises it step-by-step to recover $\mathcal{Z}_0$:

$$p_\theta(\mathcal{Z}_{t-1}|\mathcal{Z}_t) = \mathcal{N}(\mathcal{Z}_{t-1}; \mu_\theta(\mathcal{Z}_t, t), \Sigma_\theta(\mathcal{Z}_t, t)) \tag{2}$$

Our method employs a latent diffusion model (LDM) with

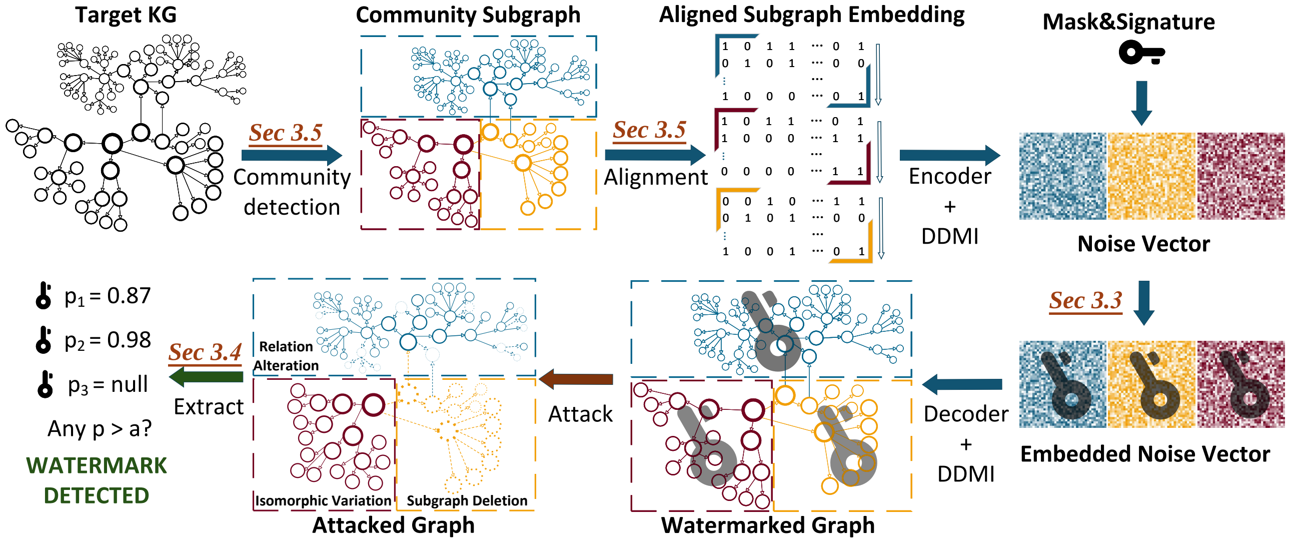


*Figure 2.* Pipeline of the proposed KGMark. The target KGE undergoes community detection and alignment before watermark embedding, enabling robust watermark extraction under attacks (Relation Alteration, Subgraph Deletion, and Isomorphic Variation).

DDIM (Song et al., 2022) for sampling. DDIM's deterministic inversion process allows efficient recovery of the initial noise vector $\mathcal{Z}_T^{\text{INV}}$, which is then used for watermark embedding, ensuring imperceptibility and integrity preservation in the latent space. The precise control offered by DDIM's reverse diffusion process is key to balancing watermark robustness and reconstruction fidelity.

### 3.2. Overview of KGMark

The proposed watermarking framework embeds a watermark into a KGE $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The process begins by encoding $\mathcal{G}$ into a latent representation $\mathcal{Z}_0$ using a graph encoder, trained under a VAE framework.

**Watermark Embedding:** A signature $\mathcal{S}$ is embedded into designated latent subspaces:

$$\Delta = \mathbf{F}(\mathcal{Z}_T^{\text{INV}}) \cdot (1 - \mathbb{M}) + \mathbf{F}(\mathcal{S}) \cdot \mathbb{M}, \mathbb{M} \in \{0,1\}^{m \times n}, \quad (3)$$

where $\mathbf{F}(\cdot)$ denotes the Fourier transform, and $\mathbb{M}$ is a mask matrix defining embedding regions for $\mathcal{S}$. The watermarked latent vector is then derived via inverse Fourier transform:

$$\mathcal{Z}_T^w = \mathbf{F}^{-1}(\Delta), \quad \mathcal{S} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}), \quad (4)$$

Where $\sigma^2$ determines the variance of the *normal distribution*. $\mathcal{Z}_T^w$ undergoes reverse diffusion, yielding $\mathcal{Z}_0^w$, which is decoded to reconstruct the watermarked graph $\mathcal{G}^w$.

**Attack Modeling:** We define the attack intensity by the total perturbation $\delta$ applied to the graph $\mathcal{G}$, where $\delta = \sum_k \delta_k$, and each $\delta_k$ denotes the perturbation (e.g., post-editing operations) applied to a subgraph $\mathcal{G}_{sub}$.

The adversary aims to invalidate the watermark by reducing the similarity between the extracted watermark and the original signature $\mathcal{S}$. Let $T(\cdot)$ denote the watermark extraction function. The attack objective is then formulated as:

$$\min_{\delta} \text{sim}(T(\mathcal{G}^w + \delta), \mathcal{S}) + \gamma \cdot \sum_k \|\delta_k\|_q, \quad (5)$$

where $\text{sim}(\cdot, \cdot)$ denotes a similarity measure, and $\gamma$ is a trade-off parameter that controls the allowed perturbation budget. A successful attack reduces the similarity score while maintaining the semantic validity of subgraphs.

As illustrated in Figure 2, isomorphic variations preserve local subgraph structure, while stronger perturbations compromise the global topology. This highlights another unique characteristic of graph-based watermarking: *the adversary can strategically degrade global structural fidelity while preserving local usability*.

**Watermark Extraction:** For watermark extraction, $\mathcal{G}^w$ is encoded and inverted to approximate $\mathcal{Z}_T^{\text{INV}}$. The Fourier transform of $\mathcal{Z}_T^{\text{INV}}$ is computed as:

$$\mathcal{Y} = \mathbf{F}(\mathbf{Z}_T^{\text{INV}}), \quad \text{s.t.} \quad \mathbb{E}[\mathcal{Y}_i] = \mu_i, \ \text{Var}[\mathcal{Y}_i] = \sigma_i^2, \quad (6)$$

A test statistic $\mathcal{T}$ is computed by comparing $\mathcal{Y}$ with $(\mathcal{S}, \mathbb{M})$. The P-value is evaluated using a noncentral $\chi^2$ distribution, and the watermark is detected if the P-value is below a predefined significance level.

To fulfill the three essential properties of a watermarking **transparency** during embedding, **robustness** against adversarial perturbations, and **detectability** upon extraction, KGMark introduces three dedicated designs across the embedding, extraction, and attack resilience stages.

Specifically, we present: (1) a *Learnable Adaptive Watermark Mask Matrix* (Sec. 3.3) to ensure transparent watermark embedding, (2) a *Defending Against Isomorphism and*

*Structural Variations* module (Sec. 3.5) to enhance watermark robustness under perturbations, and (3) a *Likelihood-Based Watermark Verification* mechanism (Sec. 3.4) validate the embedded watermark.

### 3.3. Learnable Adaptive Watermark Mask Matrix

*Principle 3.1 (**Latent Space Equilibrium**). To preserve the latent space equilibrium during watermark embedding, we decompose the total perturbation into two parts: (1) the inherent noise from DDIM inversion, and (2) the additional distortion from watermark embedding. Let $\mathcal{Z}^{INV}_{\mathcal{T}-k_j}$ be the $k_j$ step latent from DDIM inversion, $\widehat{\mathcal{Z}}_{\mathcal{T}-k_j}$ the latent from DDIM without watermark, and $\mathcal{Z}^{w}_{\mathcal{T}-k_j}$ the latent with watermark. The total deviation satisfies:*

$$\sum_{j=1}^{\mathcal{T}} \underbrace{\left\| \widehat{\mathcal{Z}}_{\mathcal{T}-k_j} - \mathcal{Z}^{\text{INV}}_{\mathcal{T}-k_j} \right\|^2}_{\text{DDIM sampling loss}} + \underbrace{\left\| \mathcal{Z}^{w}_{\mathcal{T}-k_j} - \widehat{\mathcal{Z}}_{\mathcal{T}-k_j} \right\|^2}_{\text{Watermark embedding loss}} \leq \epsilon. \tag{7}$$

*This formulation isolates the unavoidable sampling error from the controllable embedding distortion, guiding watermark design to minimize overall latent drift.*

To operationalize ***Latent Space Equilibrium***, we design a learnable masking mechanism that controls how watermark signals are injected into the latent space. Watermark embedding inevitably introduces reconstruction loss $\mathcal{L}$ for $\mathcal{G}$, quantified by the discrepancy between the $\mathcal{Z}^{\text{INV}}_{\mathcal{T}}$ and $\mathcal{Z}^{w}_{\mathcal{T}}$ (Fridrich et al., 2002). Specifically, we introduce a learnable, adaptive mask matrix $\mathbb{M}$, optimized for the structure of $\mathcal{G}$. $\mathbb{M}$ is trained to minimize $\mathcal{L}$ in Equation (8), which quantifies the discrepancy between the $\mathcal{Z}^{\text{INV}}_{\mathcal{T}}$ and $\mathcal{Z}^{w}_{\mathcal{T}}$.

$$\mathcal{L} = \sum_{j\in[1,\mathcal{T}]} \left\| \mathcal{Z}^{\text{INV}}_{\mathcal{T}-k_j} - f^{k_j}_{\text{DDIM}}\left( f_w(\mathcal{Z}^{\text{INV}}_{\mathcal{T}}, \mathcal{S}, \mathbb{M}), \mathcal{T}\right) \right\|^2, \tag{8}$$

where $\mathcal{T}$ denotes the total diffusion steps, and $k_j$ satisfies $0 < k_j < \mathcal{T}$. The function $f_w$ represents the watermark embedding function, while $f^{k_j}_{\text{DDIM}}$ denotes the DDIM sampling process at time step $k_j$.

To further refining, we adopt a ”sample-then-embed” strategy with a correction term $\alpha\mathcal{S}\cdot\mathbb{M}$ ($\alpha$ is a tunable coefficient), ensuring better alignment in the latent space.

$$\begin{aligned} \mathcal{L} = \sum_{j\in[1,\mathcal{T}]} \Big\| \mathcal{Z}^{\text{INV}}_{\mathcal{T}-k_j} - \\ \left[ f_w\left( f^{k_j}_{\text{DDIM}}(\mathcal{Z}^{\text{INV}}_{\mathcal{T}}, \mathcal{T}), \mathcal{S}, \mathbb{M}\right) + \alpha\mathcal{S}\cdot\mathbb{M}\right] \Big\|^2. \end{aligned} \tag{9}$$

As $\mathcal{L}$ shown in Equation (9), $\mathcal{S} \sim \mathcal{N}(0, \mathbf{I})$ represents a noise vector sampled from a standard normal distribution, where $\mathbf{I}$ is the identity matrix. The adaptive nature of $\mathbb{M}$ ensures that it is specific to each graph $\mathcal{G}$, making the watermark embedding robust to structural variations.

To characterize the sparsity of the learnable mask matrix $\mathbb{M}$, we define its density as:

$$Density(\mathbb{M}) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{M}_{ij}}{m \times n}, \tag{10}$$

where $m \times n$ is the total number of elements in $\mathbb{M}$. This metric quantifies the proportion of nonzero entries, balancing watermark imperceptibility and robustness against attacks.

During watermark extraction, we evaluate all candidate $\mathbb{M}$. The decision rule is formulated as Equation (11):

$$\mathcal{G}_{\text{classified}} = \begin{cases} \top, & \exists \mathbb{M} \text{ s.t. } d\big(f_{\text{ex}}(\mathbb{M}, \mathcal{G}), \mathcal{S}\big) \leq \delta, \\ \bot, & \forall \mathbb{M},\ d\big(f_{\text{ex}}(\mathbb{M}, \mathcal{G}), \mathcal{S}\big) > \delta, \end{cases} \tag{11}$$

where $f_{\text{ex}}(\mathbb{M}, \mathcal{G})$ denotes the extracted signature from $\mathcal{G}$ using matrix $\mathbb{M}$, $d(\cdot, \cdot)$ is the distance metric quantifying the difference between signatures, and $\delta$ is the threshold determining signature validity.

### 3.4. Likelihood-Based Watermark Verification

In KGMark, watermark detection utilizes a likelihood ratio test, where the null hypothesis $\mathcal{H}_0$ assumes the noise vector $\mathcal{Y}$ follows a Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2\mathcal{I}_{\mathbb{C}})$. Under the alternative hypothesis $\mathcal{H}_1$, the graph contains a watermark, introducing a detectable deviation from this distribution.

To integrate the distance metric $d(\cdot, \cdot)$ and the signature extraction function $f_{\text{ex}}$, we define the residual vector $\mathcal{R}$ as the difference between the extracted signature and the optimal reference signature $\mathcal{K}^* \in \mathcal{S}$:

$$\mathcal{R} = f_{\text{ex}}(\mathbb{M}, \mathcal{G}) - \mathcal{K}^*. \tag{12}$$

The distance metric $d(\cdot, \cdot)$ thus measures the magnitude of $\mathcal{R}$, which quantifies the deviation between the extracted signature and the expected reference signature. The likelihood of observing $\mathcal{R}$ under $\mathcal{H}_0$ is given by Equation (13):

$$\mathcal{L}(\mathcal{R}|\mathcal{H}_0) = \prod_{i\in\mathbb{M}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{|\mathcal{R}_i|^2}{2\sigma^2}\right), \tag{13}$$

where $\sigma^2 = \frac{1}{|\mathbb{M}|}\sum_{i\in\mathbb{M}} |\mathcal{R}_i|^2$ is the estimated variance of the residual vector, computed as the mean squared deviation of the residuals within the mask region $\mathbb{M}$.

The likelihood ratio test statistic $\lambda$ is defined as the logarithmic ratio of the likelihoods under $\mathcal{H}_0$ and $\mathcal{H}_1$:

$$\lambda = -2\log\frac{\sup_{\mathcal{H}_0} \mathcal{L}(\mathcal{R}|\mathcal{H}_0)}{\sup_{\mathcal{H}_1} \mathcal{L}(\mathcal{R}|\mathcal{H}_1)}. \tag{14}$$

For watermark detection, we employ a simplified test statistic $\hat{\mathcal{T}}$, defined in Equation (15), which directly quantifies the deviation of $\mathcal{R}$ from the expected values under $\mathcal{H}_0$:

$$\hat{\mathcal{T}} = \frac{1}{\sigma^2}\sum_{i\in\mathbb{M}} |\mathcal{R}_i|^2. \tag{15}$$

Under $\mathcal{H}_0$, $\hat{\mathcal{T}}$ follows a noncentral chi-squared distribution with degrees of freedom $|\mathbb{M}|$ and noncentrality parameter $\lambda = \frac{1}{\sigma^2}\sum_i |\mathcal{K}_i^*|^2$ (Patnaik, 1949). The p-value for watermark detection is then calculated using the cumulative distribution function (CDF) of this distribution (Glasserman, 2004), as shown in Equation (16):

$$\begin{aligned} p &= \Pr\left(\chi^2_{|\mathbb{M}|,\lambda} \leq \hat{\mathcal{T}} \mid \mathcal{H}_0\right) \\ &= \int_0^{\hat{\mathcal{T}}} \frac{1}{\Gamma\left(\frac{|\mathbb{M}|}{2}\right)} \left(\frac{x}{2}\right)^{\frac{|\mathbb{M}|}{2}-1} \exp\left(-\frac{x+\lambda}{2}\right) dx. \end{aligned} \tag{16}$$

If the $p$ is below a significance level $\alpha$, $\mathcal{G}$ is considered watermarked; otherwise, the null hypothesis $\mathcal{H}_0$ is not rejected.

### 3.5. Defending Isomorphism and Structural Variations

Real-world graph applications encounter isomorphism and structural variations, which are unique to graph-structured data and can be exploited for attacks (Yu et al., 2023). Isomorphism refers to graphs that preserve their overall structural connectivity but differ in node ordering or node-specific properties. Structural variations involve more substantial modifications that can alter the graph's topology or the relationship between nodes. See Appendix A for details.

**Isomorphism Variations:** Graph isomorphism (Yan & Han, 2002) transformations alter the latent space representation $z_0$ of a graph, despite preserving its inherent properties (Bouritsas et al., 2023). To address this, we ensure $z_0$ remains invariant under adjacency matrix reorderings via a *graph alignment* procedure. Let $V(\mathcal{G})$ denote the set of vertices, and $A$ the adjacency matrix. Vertices are reordered by degree $\deg(v_i)$ and clustering coefficient $C(v_i)$:

$$\begin{aligned} \deg(v_i) > \deg(v_j) &\implies i < j, \\ \deg(v_i) = \deg(v_j) &\implies C(v_i) \geq C(v_j). \end{aligned} \tag{17}$$

Graph attributes (vertex features, edge weights, adjacency matrix) are adjusted to the new vertex order, preserving structural consistency.

**Structural Variations:** Let $\Delta A \in \{0,1\}^{n\times n}$ denote the perturbation matrix, where $\|\Delta A\|_0 = \delta$ quantifies the attack intensity as the number of edge modifications. Structural variations introduce perturbations at multiple scales of graph topology, formalized by the divergence metric:

$$\mathcal{D}(\mathcal{G},\tilde{\mathcal{G}}) = \underbrace{\|A-\tilde{A}\|_F}_{\text{local perturbations}} + \alpha \underbrace{\|L^\dagger - \tilde{L}^\dagger\|_2}_{\text{global spectral shifts}}, \tag{18}$$

where $L^\dagger$ is the pseudoinverse of the graph Laplacian $L$, and $\alpha$ balances local and global contributions. This divergence aligns with the attacker's objective in Equation (5), where the constraint $\|\Delta A\|_0 \leq \delta$ limits the attack intensity, and $sim(T(\mathcal{G}^w + \Delta A), \mathcal{S})$ measures degradation of the extracted watermark.

The local term $\|A - \tilde{A}\|_F$ captures edge-level modifications, including Gaussian noise, Relation Alteration, and Triple Deletion. The global term $\|L^\dagger - \tilde{L}^\dagger\|_2$ reflects low-frequency structural changes often introduced by smoothing attacks that diffuse information across the graph (Table 1).

*Principle 3.2 (**Information-Theoretic Robustness**). The watermark embedding $\mathcal{W}$ must ensure that the mutual information $I(\mathcal{S}; T(\mathcal{G}^w + \Delta A))$ between the original watermark $\mathcal{S}$ and the extracted signature under attack satisfies:*

$$\inf_{\|\Delta A\|_0 \leq \delta} I(\mathcal{S}; T(\mathcal{G}^w + \Delta A)) \geq \beta, \tag{19}$$

*where $\beta > 0$ is a lower bound guaranteeing detectable information retention. This is achieved by enforcing that the watermark encoding $\mathcal{W}(\mathcal{G})$ maximizes the effective information capacity $C(\mathcal{W})$ under adversarial constraints:*

$$C(\mathcal{W}) = \min_{\Delta A} \left[H(\mathcal{S}) - H(\mathcal{S} \mid T(\mathcal{G}^w + \Delta A))\right], \tag{20}$$

*where $H(\cdot)$ denotes entropy. The principle mandates that critical graph substructures (e.g., high-centrality communities) encode $\mathcal{S}$ with minimal entropy loss $H(\mathcal{S} \mid T(\cdot))$, ensuring robustness against $\mathcal{D}(\mathcal{G}, \tilde{\mathcal{G}})$.*

To satisfy *Principle 3.2*, we partition $\mathcal{G}$ into $l$ communities $\{\mathcal{C}_i\}_{i=1}^l$with vertices ranked by centrality $\eta(v)$. The watermark is embedded as:

$$\mathcal{W}(\mathcal{G}) = \bigcup_{i=1}^{l} \Phi(\mathcal{C}_i) \cup \bigcup_{v\in\mathcal{C}_i} \Psi(v), \tag{21}$$

where $\Phi(\mathcal{C}_i)$ injects $\mathcal{S}$ into the community's spectral profile (resilient to $\|L^\dagger - \tilde{L}^\dagger\|_2$), and $\Psi(v)$ encodes $\mathcal{S}$ via edge-weight distributions around high-centrality vertices (resistant to $\|A - \tilde{A}\|_F$). This dual encoding maximizes $C(\mathcal{W})$ by distributing $\mathcal{S}$ across both low-frequency (global) and high-frequency (local) structural invariants.

## 4. Experiments

The experimental evaluation of KGMark assesses its effectiveness in terms of watermark **transparency**, **detectability**, and **robustness** across various attack scenarios and datasets.

### 4.1. Experiment setup

**Datasets.** We evaluate our approach using three public datasets representing diverse real-world scenarios: Last-FM (music) (Çano et al., 2017), MIND (news) (Wu et al., 2020), and Alibaba-iFashion (e-commerce) (Chen et al., 2019). Table 2 provides a summary of these datasets.

**Variants.** Since KGMark is the first watermarking scheme for Knowledge Graph Embedding (KGE), we explored multiple variants in the experiments. ❶ **W/O LAWMM**: Using a fixed watermark mask matrix with the same density, instead of applying the Learnable Adaptive Watermark Mask

*Table 1.* Watermark Detectability & Robustness (Relation Alteration, Triple Deletion, two adversarial attacks (L2 Metric, NEA), Isomorphism Variation (IsoVar)). We evaluate the detectability of the watermark under clean samples and its robustness under editing attacks with structural and isomorphism variations. Includes four baselines and three variants compared with KGMark.

| Datasets | Method | Clean | Relation Alteration | | | Triple Deletion | | | Adversarial | | IsoVar | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 10% | 30% | 50% | 10% | 30% | 50% | L2 Metric | NEA | | |
| AliF | DwtDct | 0.9837 | 0.9730 | 0.8813 | 0.8371 | 0.9457 | 0.8577 | 0.7724 | 0.9577 | 0.9638 | 0.6039 | 0.8776 |
| | DctQim | 0.9749 | 0.9548 | 0.8623 | 0.8139 | 0.9121 | 0.8049 | 0.7073 | 0.9203 | 0.9278 | 0.5867 | 0.8465 |
| | TreeRing | 0.9814 | 0.9207 | 0.8321 | 0.7392 | 0.9442 | 0.8599 | 0.8091 | 0.9621 | 0.9584 | 0.6257 | 0.8632 |
| | GaussianShading | 0.9882 | 0.9481 | 0.8745 | 0.7998 | 0.9217 | 0.8846 | 0.7850 | 0.9364 | 0.9512 | 0.6094 | 0.8699 |
| | W/O LAWMM | 0.9980 | 0.9623 | 0.9352 | 0.9147 | 0.9795 | 0.9592 | **0.9341** | 0.9817 | 0.9706 | 0.9895 | 0.9625 |
| | Only CL | 0.9942 | 0.9537 | 0.9216 | 0.8864 | 0.9214 | 0.8745 | 0.8063 | 0.9426 | 0.9535 | 0.9635 | 0.9218 |
| | Only VL | 0.9828 | 0.9765 | 0.9433 | 0.9172 | 0.9670 | 0.9148 | 0.8592 | 0.9521 | 0.9676 | 0.9787 | 0.9459 |
| | KGMark | **0.9991** | **0.9810** | **0.9564** | **0.9207** | **0.9829** | **0.9669** | 0.9320 | **0.9841** | **0.9809** | **0.9933** | **0.9697** |
| MIND | DwtDct | 0.9793 | 0.9563 | 0.8827 | 0.8161 | 0.9334 | 0.8413 | 0.7610 | 0.9358 | 0.9291 | 0.6348 | 0.8669 |
| | DctQim | 0.9785 | 0.9542 | 0.8703 | 0.8269 | 0.9046 | 0.8071 | 0.6993 | 0.9209 | 0.9198 | 0.5708 | 0.8452 |
| | TreeRing | 0.9862 | 0.9128 | 0.8554 | 0.8171 | 0.9704 | 0.8617 | 0.7831 | 0.9682 | 0.9543 | 0.5763 | 0.8685 |
| | GaussianShading | 0.9903 | 0.9076 | 0.8455 | 0.7930 | 0.9334 | 0.8746 | 0.8284 | 0.9767 | 0.9681 | 0.5845 | 0.8702 |
| | W/O LAWMM | 0.9984 | 0.9896 | 0.9743 | 0.9335 | **0.9895** | 0.9609 | 0.9328 | **0.9915** | 0.9864 | 0.9705 | 0.9727 |
| | Only CL | 0.9973 | 0.9792 | 0.9515 | 0.9287 | 0.9624 | 0.9074 | 0.8449 | 0.9697 | 0.9713 | 0.9846 | 0.9497 |
| | Only VL | 0.9956 | 0.9814 | 0.9658 | **0.9346** | 0.9647 | 0.9154 | 0.8804 | 0.9683 | 0.9618 | **0.9861** | 0.9554 |
| | KGMark | **0.9987** | **0.9907** | **0.9751** | 0.9314 | 0.9781 | **0.9726** | **0.9576** | 0.9849 | **0.9883** | 0.9842 | **0.9762** |
| Last-FM | DwtDct | 0.9801 | 0.9636 | 0.8834 | 0.8229 | 0.9570 | 0.8353 | 0.7415 | 0.9596 | 0.9678 | 0.6407 | 0.8752 |
| | DctQim | 0.9842 | 0.9527 | 0.8773 | 0.8062 | 0.9083 | 0.7972 | 0.7125 | 0.9144 | 0.9161 | 0.5938 | 0.8463 |
| | TreeRing | 0.9879 | 0.9167 | 0.8353 | 0.7982 | 0.9316 | 0.9024 | 0.8519 | 0.9553 | 0.9487 | 0.6109 | 0.8738 |
| | GaussianShading | 0.9795 | 0.9356 | 0.8658 | 0.8303 | 0.9519 | 0.9145 | 0.8667 | 0.9638 | 0.9594 | 0.6551 | 0.8922 |
| | W/O LAWMM | **0.9982** | 0.9929 | **0.9857** | **0.9468** | 0.9785 | **0.9420** | **0.9057** | 0.9794 | **0.9852** | 0.9970 | **0.9711** |
| | Only CL | 0.9962 | 0.9893 | 0.9367 | 0.9073 | 0.9773 | 0.9206 | 0.8737 | 0.9156 | 0.9205 | 0.9901 | 0.9427 |
| | Only VL | 0.9929 | 0.9726 | 0.9308 | 0.9034 | **0.9824** | 0.9275 | 0.8661 | 0.9053 | 0.9112 | 0.9912 | 0.9383 |
| | KGMark | 0.9976 | **0.9973** | 0.9844 | 0.9421 | 0.9796 | 0.9350 | 0.9031 | **0.9886** | 0.9814 | **0.9977** | 0.9707 |

*Table 2.* We use three KG datasets from different domains.

| Name | Entities | Relations | Triples |
|---|---|---|---|
| Alibaba-iFashion (AliF) | 59,156 | 51 | 279,155 |
| MIND | 24,733 | 512 | 148,568 |
| Last-FM | 58,266 | 9 | 464,567 |

Matrix (LAWMM). ❷ **Only CL**: Applying the watermark exclusively in the Community Layer, selecting specific vertices or vertex groups within each community. ❸ **Only VL**: Applying the watermark exclusively in the Vertex Layer, selecting multiple vertices or vertex groups within a specific community. ❹ **W/O Watermark (Control)**: Reconstructing the knowledge graph without any watermark.

**Baselines.** As the first watermarking framework designed for diffusion KGE, we introduce four additional baselines to validate KGMark's effectiveness. **TreeRing** (Wen et al., 2023) and **GaussianShading** (Yang et al., 2024b) are node-level watermarking methods designed for diffusion models (Images), embedding watermarks by replacing 5% of nodes in the graph. **DwtDct** (Cox et al., 2007) and **DctQim** (Chen & Wornell, 2001) are classical watermarking techniques that modify transformed coefficients to balance imperceptibility and robustness.

**Implementation details.** We first employ the RotatE (Sun et al., 2019) model to embed the knowledge graph, with an embedding dimension of 4096. Our watermarking method is applied to the above-processed datasets, and a subsequent series of related experiments is carried out. All experiments are conducted on a single NVIDIA A800.

### 4.2. Detectability

To evaluate the detectability of KGMark, we calculate the false positive rate (FPR), true positive rate (TPR), and their corresponding average AUC values, as summarized in Table 1. The evaluations are conducted under a configuration where the DDIM inference steps are set to 75, and the predefined significance level is fixed at $5 \times 10^{-5}$. To further examine the impact of significance levels and DDIM inference steps on watermark detectability, we perform controlled experiments by varying these parameters.

Table 1 summarizes the detection performance of KGMark under various attack scenarios. Across all settings, KGMark consistently maintains high detectability. We also evaluate an ablated variant (W/O LAWMM) that removes the Learnable Adaptive Watermark Mask, which is designed to preserve transparency while balancing embedding detectability. Interestingly, this variant exhibits slightly higher detectability in some cases, likely due to stronger watermark signal strength without the masking constraint. However, the full KGMark still achieves comparable or superior performance overall. This result also highlights the effectiveness of LAWMM in achieving a favorable trade-off between watermark detectability and semantic transparency.

*Table 3.* Watermark Transparency. We evaluate the transparency of watermarking from two dimensions: the **similarity** of knowledge graph embedding before and after watermarking and the **quality** of watermarked knowledge graph. We **mark** the best transparency score only for the four baselines and two KGMark variants, as the original KG and W/O Watermark serve as control groups.

| **Datasets** | **Method** | **Cosine Similarity ↑** | | | **KG Quality Metric @ 75 Steps** | | | |
|---|---|---|---|---|---|---|---|---|
| | | 50 Steps | 65 Steps | 75 Steps | GMR ↓ | HMR ↓ | AMR ↓ | Hits@10 ↑ |
| AliF | Original KG | - | - | - | 1.828 | 1.162 | 135.459 | 0.8980 |
| | W/O Watermark | 0.7971 | 0.8797 | 0.9674 | 3.026 | 1.579 | 141.412 | 0.8318 |
| | DwtDct | 0.7215 | 0.7928 | 0.8251 | 5.096 | 1.699 | 157.036 | 0.6933 |
| | DctQim | 0.7509 | 0.7633 | 0.7653 | 5.104 | 1.654 | 161.142 | 0.7385 |
| | TreeRing | 0.7761 | **0.8431** | 0.9071 | 3.928 | 1.618 | 152.634 | 0.8017 |
| | GaussianShading | 0.2879 | 0.3226 | 0.3538 | 6.641 | 1.798 | 172.813 | 0.5137 |
| | W/O LAWMM | 0.7662 | 0.7838 | 0.8643 | 3.457 | 1.624 | 147.305 | 0.7871 |
| | KGMark | **0.7839** | 0.8309 | **0.9482** | **3.046** | **1.580** | **141.904** | **0.8296** |
| MIND | Original KG | - | - | - | 7.197 | 1.975 | 155.656 | 0.6649 |
| | W/O Watermark | 0.7345 | 0.8290 | 0.9579 | 9.853 | 2.215 | 168.906 | 0.5734 |
| | DwtDct | 0.7831 | 0.8244 | 0.8312 | 11.328 | 2.328 | 188.992 | 0.5216 |
| | DctQim | 0.7549 | 0.7574 | 0.7703 | 12.037 | 2.753 | 205.483 | 0.4835 |
| | TreeRing | 0.7976 | 0.8108 | 0.8581 | 11.102 | 2.297 | 182.925 | 0.5108 |
| | GaussianShading | 0.2843 | 0.3196 | 0.3728 | 14.523 | 3.312 | 234.064 | 0.3940 |
| | W/O LAWMM | 0.7469 | 0.8235 | 0.8902 | 10.819 | 2.238 | 173.194 | 0.5332 |
| | KGMark | **0.8083** | **0.8533** | **0.9397** | **10.508** | **2.226** | **169.305** | **0.5683** |
| Last-FM | Original KG | - | - | - | 3.571 | 1.202 | 1711.695 | 0.8436 |
| | W/O Watermark | 0.8293 | 0.9157 | 0.9410 | 4.455 | 1.452 | 1715.907 | 0.8431 |
| | DwtDct | 0.7215 | 0.7928 | 0.8433 | 4.519 | 1.502 | 1734.823 | 0.8221 |
| | DctQim | 0.7509 | 0.7633 | 0.7679 | 5.139 | 1.704 | 2043.249 | 0.7264 |
| | TreeRing | 0.7262 | 0.7896 | 0.8364 | 4.733 | 1.652 | 1772.961 | 0.8149 |
| | GaussianShading | 0.3252 | 0.3649 | 0.4184 | 6.349 | 2.016 | 2192.492 | 0.6463 |
| | W/O LAWMM | 0.7628 | 0.8809 | 0.8948 | 4.456 | 1.452 | 1716.828 | 0.8430 |
| | KGMark | **0.8876** | **0.9051** | **0.9161** | **4.455** | **1.452** | **1716.365** | **0.8430** |

**Obs.❶ KGMark's detectability remains consistently high with optimal DDIM steps and significance levels.** Figure 3a illustrates the impact of significance levels and DDIM inference steps on the AUC values. Within the significance level range of $1e-5 \sim 1e-4$, increasing the DDIM steps significantly enhances detection performance by reducing watermark information loss. Meanwhile, the AUC values remain exceptionally stable, approaching 1 for configurations with higher steps, highlighting KGMark's robust and consistent performance under suitable conditions.

However, for higher significance levels $1e-3 \sim 1e-2$, the AUC values drop more sharply. This decline is attributed to higher inference steps reducing p-values for negative samples, thereby increasing the false positive rate (FPR) and accelerating the drop in detection accuracy.

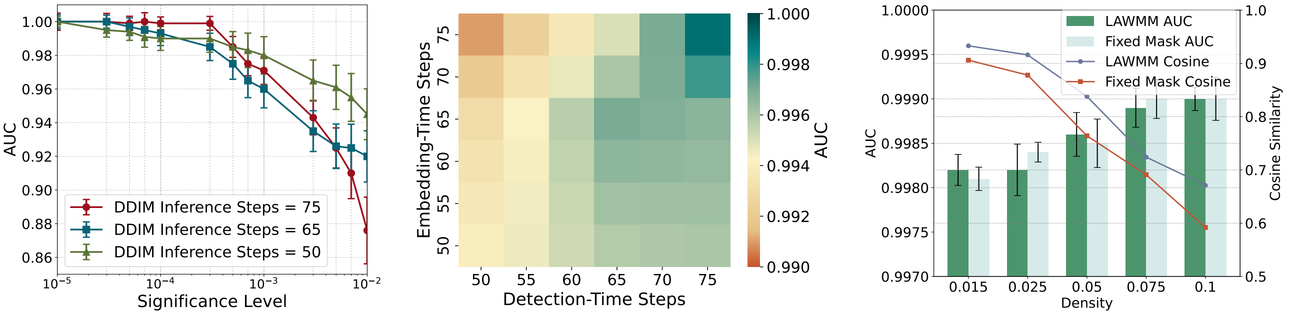
**Obs.❷ Increasing detection-stage DDIM steps boosts detectability, while stage alignment further enhances performance.** Results in Figure 3b demonstrate that higher DDIM steps in the detection stage yield marginal improvements in AUC, with values consistently exceeding 0.996 across most configurations. For example, as detection steps increase from lower to higher ranges, the AUC improves slightly, showcasing the positive impact of precise parameter tuning during detection. However, inconsistencies between embedding and detection steps introduce minor declines in performance, as seen in higher-step configurations where AUC decreases from $0.99785 \sim 0.9967$. This highlights the critical importance of alignment between the two stages. Notably, the detection stage exerts a more significant influence on watermark retrieval accuracy, reinforcing the need for fine-grained control at this stage to achieve optimal detectability.

### 4.3. Transparency

We evaluate the transparency of KGMark, which measures the extent to which the watermark preserves the structural and functional integrity of the original KGE.

**Obs.❸ KGMark achieves strong transparency by minimally impacting KG structure and usability.** As shown in Table 3, KGMark achieves high cosine similarity scores (e.g., 0.9482 for AliF at 75 steps) compared to the non-watermarked version, indicating that the watermark does not significantly distort the KG's inherent relationships. This demonstrates that the watermark is seamlessly integrated without disrupting the KG's core structure.

To further evaluate the KG's quality post-watermarking, we measure forecast task performance using GMR, HMR, AMR, and Hits@10 metrics. KGMark's results are nearly



(a) Watermark Detectability

(b) Ablation on DDIM Steps

(c) Effect of Mask Density

*Figure 3.* Ablation. (a) AUC under varying significance levels and DDIM inference steps. (b) Different DDIM inference steps during embedding and detection. (c) Effect of watermark mask density on all datasets, with DDIM steps set to 75 and significance level $5e{-}5$.

on par with the original KG and the non-watermarked version. For example, in AliF, KGMark achieves a Hits@10 score of 0.8296, slightly lower than the non-watermarked score of 0.8318. Similar trends are observed across MIND and Last-FM, where KGMark consistently outperforms the version without LAWMM. These findings confirm that KGMark preserves the KG's utility for downstream tasks, ensuring the watermarked KG remains functional and reliable.

Experimental results in Figure 3c show that the density of the watermark mask matrix also influences watermark transparency. Higher density causes a slight increase in detectability. However, this comes at the cost of transparency, as cosine similarity (LAWMM_cosine) drops from 0.933 to 0.671, reflecting degradation in KG integrity. This trade-off underscores the need to balance density for optimal detectability and transparency.

## 4.4. Robustness

We next assess the robustness of KGMark, focusing on its ability to withstand a wide range of post-editing attacks without compromising watermark integrity. To evaluate the robustness of our watermarking scheme, we systematically analyze its performance under five post-editing attacks with varying intensities: Gaussian noise injection, Gaussian smoothing(Hu et al., 2024), relation alteration, triple deletion, and graph isomorphism variation. Attack intensity is quantified by the proportion of affected entities or triples, ranging from 10% to 50%, simulating real-world adversarial scenarios. We further incorporate two stronger adversarial attacks into our evaluation: NEA (Bojchevski & Günnemann, 2019), a graph poisoning attack targeting node embeddings, and the L2 Metric attack (Bhardwaj et al., 2021), which perturbs embeddings via instance attribution analysis. These attacks introduce fine-grained and targeted perturbations, enabling rigorous robustness evaluation under realistic and challenging adversarial settings.

*Table 4.* Watermark Detectability & Robustness (Gaussian Noise & Smoothing). We impose two common attacks against images on KGE and evaluate the robustness of these two attacks under different attack intensities.

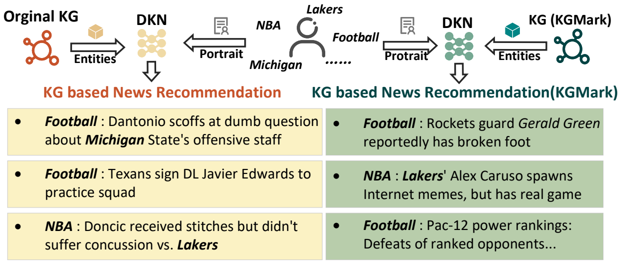
| Datasets | Method | Gaussian Noise | | | Smoothing | | |
|---|---|---|---|---|---|---|---|
| | | 10% | 30% | 50% | 10% | 30% | 50% |
| AliF | DwtDct | 0.98 | 0.89 | 0.86 | 0.94 | 0.90 | 0.81 |
| | DwtQim | 0.93 | 0.84 | 0.69 | 0.90 | 0.82 | 0.77 |
| | TR | 0.92 | 0.86 | 0.81 | 0.92 | 0.86 | 0.78 |
| | GaussianShading | 0.94 | 0.91 | 0.89 | 0.90 | 0.83 | 0.79 |
| | W/O LAWMM | 0.98 | 0.95 | **0.92** | **0.95** | 0.91 | 0.88 |
| | Only CL | 0.98 | 0.94 | 0.90 | 0.93 | 0.90 | 0.84 |
| | Only VL | 0.96 | 0.92 | 0.87 | 0.91 | 0.87 | 0.82 |
| | KGMark | **0.98** | **0.96** | 0.91 | 0.94 | **0.92** | **0.89** |
| MIND | DwtDct | 0.96 | 0.90 | 0.83 | 0.96 | 0.87 | 0.81 |
| | DwtQim | 0.93 | 0.81 | 0.72 | 0.94 | 0.85 | 0.79 |
| | TR | 0.91 | 0.86 | 0.77 | 0.94 | 0.91 | 0.83 |
| | GaussianShading | 0.94 | 0.89 | 0.85 | 0.90 | 0.86 | 0.83 |
| | W/O LAWMM | 0.98 | 0.95 | 0.91 | 0.95 | 0.92 | 0.89 |
| | Only CL | 0.98 | 0.93 | 0.90 | 0.96 | 0.92 | 0.87 |
| | Only VL | 0.94 | 0.93 | 0.91 | 0.95 | 0.91 | 0.89 |
| | KGMark | **0.99** | **0.96** | **0.92** | **0.96** | **0.93** | **0.90** |
| Last-FM | DwtDct | 0.96 | 0.90 | 0.85 | 0.95 | 0.85 | 0.77 |
| | DwtQim | 0.95 | 0.84 | 0.74 | 0.95 | 0.92 | 0.82 |
| | TR | 0.93 | 0.89 | 0.86 | 0.92 | 0.89 | 0.85 |
| | GaussianShading | 0.96 | 0.92 | 0.89 | 0.92 | 0.89 | 0.86 |
| | W/O LAWMM | 0.99 | 0.96 | 0.93 | 0.95 | 0.93 | 0.91 |
| | Only CL | 0.97 | 0.96 | 0.92 | 0.93 | 0.91 | 0.89 |
| | Only VL | 0.98 | 0.96 | 0.93 | 0.94 | 0.90 | 0.86 |
| | KGMark | **0.99** | **0.97** | **0.93** | **0.95** | **0.93** | **0.91** |

**Obs.❹ KGMark's hierarchical embedding ensures robustness by mitigating both local perturbations and structural disruptions.** As shown in Table 4 and Table 1, the robustness of the watermark varies significantly across method variants. The standalone Community Layer (Only CL) and Vertex Layer (Only VL) exhibit weaker resilience compared to the full KGMark. For instance, under triple deletion attacks on AliF at 50% intensity, Only CL and Only VL achieve AUC scores of 0.8063 and 0.8592, respectively, while KGMark maintains a robust score of 0.9320. This underscores the necessity of combining coarse-grained (Community Layer) and fine-grained (Vertex Layer) embedding strategies. Notably, the variant without LAWMM (W/O LAWMM) demonstrates comparable robustness to KGMark, as LAWMM is primarily designed for transparency and does not compromise robustness.

Our analysis reveals that KGMark is particularly sensitive to triple deletion and high-intensity smoothing attacks while maintaining strong robustness against other attack types. Under triple deletion attacks on AliF, the AUC drops from 0.9829 at 10% intensity to 0.9320 at 50% intensity, reflecting a significant but manageable decline. Similarly, under smoothing attacks, the AUC decreases from 0.94 at 10% intensity to 0.89 at 50% intensity on AliF. In contrast, KGMark demonstrates exceptional resilience to Gaussian noise and relation alteration, achieving an AUC of 0.93 at 50% intensity under Gaussian noise attacks on Last-FM. These results validate that KGMark effectively balances robustness and usability, even under aggressive post-editing attacks.

### 4.5. Case Study

We conduct a case study on the downstream task of news recommendation using the MIND dataset, comparing the performance of knowledge graph embeddings with and without the proposed watermark. The experiments are carried out using the Deep Knowledge-Aware Network (DKN) model (Wang et al., 2018), which has been trained for 10 epochs on the MIND dataset. To simulate a controlled user profile, we restrict the user's click history by focusing on a strong interest in sports news. Additionally, we randomly sample 5 news items from the model's output for evaluation.



| KG based News Recommendation | KG based News Recommendation(KGMark) |
|---|---|
| • ***Football*** : Dantonio scoffs at dumb question about ***Michigan*** State's offensive staff | • ***Football*** : Rockets guard *Gerald Green* reportedly has broken foot |
| • ***Football*** : Texans sign DL Javier Edwards to practice squad | • ***NBA*** : ***Lakers***' Alex Caruso spawns Internet memes, but has real game |
| • ***NBA*** : Doncic received stitches but didn't suffer concussion vs. ***Lakers*** | • ***Football*** : Pac-12 power rankings: Defeats of ranked opponents... |

*Figure 4.* Case Study. Workflow and representative results.

The results, shown in Figure 4, highlight that the content and subcategories of targeted recommendations remain consistent with and without watermarking. Despite the watermark, the recommendation system focuses on sports news and identifies trending topics, such as the "Pac-12 power rankings." Entity recognition remains stable, with consistent identification of sports figures, teams, and locations across both versions. Additionally, the recommendation logic and hot topic recommendations remain unchanged, indicating no impact from the watermarking process.

## 5. Discussion

### 5.1. Variational Autoencoder

KGMark's implementation employed a VAE based on the Relational Graph Attention Network (RGAT) (Busbridge et al., 2019), which extends GAT (Veličković et al., 2018) by incorporating relational dependencies. Given a KG $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, RGAT updates node representations as:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{N}r(v)} \alpha_{vu}^r \mathbf{W}^r \mathbf{h}_u^{(l)}\right), \qquad (22)$$

where $\mathcal{N}_r(v)$ denotes neighbors connected via relation $r$, $\mathbf{W}^r$ is a relation-specific transformation, and $\alpha_{vu}^r$ is the learned attention weight.

Relational graph neural networks (Schlichtkrull et al., 2017) (e.g., RGAT, RGCN) better enhance the expressiveness of the latent space $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathcal{G})$by capturing structural dependencies. However, our experiments show that relation-based models are vulnerable to adversarial attacks, which disrupt attention weights and cause representation shifts:

$$\|\tilde{\mathbf{h}}_v - \mathbf{h}_v\|_2^2 \approx \sum_{r\in\mathcal{R}}\left(\sum_{u\in\mathcal{N}_r(v)\setminus\tilde{\mathcal{N}_r}(v)} (\alpha_{vu}^r \mathbf{W}^r \mathbf{h}_u)^2\right) \qquad (23)$$

where $\tilde{\mathcal{N}_r}(v)$ is the modified neighborhood. This degradation also impacts VAE reconstruction, reducing watermark detectability. Using non-relational models such as GCN or GAT mitigates this vulnerability but lowers latent space expressiveness, leading to suboptimal reconstruction:

$$D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathcal{G})\|p(\mathbf{z}))\uparrow, \quad \mathbb{E}_{q_\phi(\mathbf{z}|\mathcal{G})}[\log p_\theta(\mathcal{G}|\mathbf{z})]\downarrow. \qquad (24)$$

Thus, relational modeling improves representation quality but increases attack susceptibility, requiring a balance between robustness and expressiveness.

### 5.2. Embedding Dimensions and DDIM Generality

The dimensionality of embeddings affects watermark detectability and robustness, requiring a balance with task performance. While DDIM inversion is central to our method, compatibility with advanced samplers remains an open problem. Future work will explore conditional sampling to improve reconstruction quality and watermark transparency.

## 6. Conclusion

In this paper, we introduce KGMark, a novel watermarking scheme for knowledge graphs, leveraging diffusion models to ensure robustness, transparency, and detectability. Our method addresses key challenges in graph integrity, offering a secure solution that maintains semantic fidelity even in dynamic environments. KGMark provides a foundation for securing the integrity and ownership of synthetic knowledge graphs, with potential applications spanning academic research to commercial deployments in fields like recommendation systems and GraphRAG.

## Acknowledgements

We would like to thank the anonymous reviewers for their valuable feedback. We are also grateful to the open-source community for providing benchmark datasets and foundational libraries that made this research possible. This work was supported in part by the National Natural Science Foundation of China (No.62471064), the National Natural Science Foundation of China (No.62301072) and the National Natural Science Foundation of China (No.62271066).

## Impact Statement

This work proposes KGMark, the first diffusion watermarking framework for knowledge graphs, aiming to enhance intellectual property protection and traceability in AI-generated KGE data. Given the increasing reliance on synthetic KGs in downstream applications such as recommendation systems, semantic search, and language model augmentation, our method enables robust, transparent, and detectable watermarking at the embedding level. We do not foresee immediate negative societal impact from this work. However, we acknowledge that any technology for ownership control may raise ethical concerns if misused, such as improper surveillance or censorship. To mitigate this, we focus strictly on benign applications such as integrity verification and content provenance.

## References

An, B., Ding, M., Rabbani, T., Agrawal, A., Xu, Y., Deng, C., Zhu, S., Mohamed, A., Wen, Y., Goldstein, T., and Huang, F. Waves: benchmarking the robustness of image watermarks. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2025.

Asnani, V., Collomosse, J., Bui, T., Liu, X., and Agarwal, S. Promark: Proactive diffusion watermarking for causal attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10802–10811, 2024.

Barman, N. R., Sharma, K., Aziz, A., Bajpai, S., Biswas, S., Sharma, V., Jain, V., Chadha, A., Sheth, A., and Das, A. The brittleness of ai-generated image watermarking techniques: Examining their robustness against visual paraphrasing attacks. *arXiv preprint arXiv:2408.10446*, 2024.

Bauer, A., Trapp, S., Stenger, M., Leppich, R., Kounev, S., Leznik, M., Chard, K., and Foster, I. Comprehensive exploration of synthetic data generation: A survey, 2024.

Bhardwaj, P., Kelleher, J., Costabello, L., and O'Sullivan, D. Adversarial attacks on knowledge graph embeddings via instance attribution methods. *arXiv preprint arXiv:2111.03120*, 2021.

Bojchevski, A. and Günnemann, S. Adversarial attacks on node embeddings via graph poisoning. In *International conference on machine learning*, pp. 695–704. PMLR, 2019.

Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2023.

Busbridge, D., Sherburn, D., Cavallo, P., and Hammerla, N. Y. Relational graph attention networks, 2019.

Çano, E., Morisio, M., et al. Music mood dataset creation based on last. fm tags. In *2017 International Conference on Artificial Intelligence and Applications, Vienna, Austria*, pp. 15–26, 2017.

Cao, N. D. and Kipf, T. Molgan: An implicit generative model for small molecular graphs, 2022.

Chang, Z., Koulieris, G. A., and Shum, H. P. H. On the design fundamentals of diffusion models: A survey, 2023.

Charpenay, V. and Schockaert, S. Capturing knowledge graphs and rules with octagon embeddings. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2025.

Chen, B. and Wornell, G. A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. on Information Theory*, 47(4):291–314, 2001.

Chen, S., Fang, H., Cai, Y., Huang, X., and Sun, M. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc.

Chen, W., Huang, P., Xu, J., Guo, X., Guo, C., Sun, F., Li, C., Pfadler, A., Zhao, H., and Zhao, B. Pog: Personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 2662–2670. Association for Computing Machinery, 2019. ISBN 9781450362016. doi: 10.1145/3292500.3330652.

Cox, I., Miller, M., Bloom, J., Fridrich, J., and Kalker, T. *Digital watermarking and steganography*. Morgan kaufmann, 2007.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024.

Dathathri, S., See, A., Ghaisas, S., Huang, P.-S., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., Stanforth, R., Matejovicova, T., Hayes, J., Vyas, N., Merey, M. A., Brown-Cohen, J., Bunel, R., Balle, B., Cemgil, T., Ahmed, Z., Stacpoole, K., Shumailov, I., Baetu, C., Gowal, S., Hassabis, D., and Kohli, P. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Dong, Y., Wang, S., Wang, Y., Derr, T., and Li, J. On structural explanation of bias in graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 316–326, New York, NY, USA, 2022. Association for Computing Machinery.

Fan, W., Ma, Y., Yin, D., Wang, J., Tang, J., and Li, Q. Deep social collaborative filtering. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 305–313, New York, NY, USA, 2019. Association for Computing Machinery.

Fridrich, J., Goljan, M., and Du, R. Lossless data embedding—new paradigm in digital watermarking. *EURASIP Journal on Advances in Signal Processing*, 2002:1–12, 2002.

Glasserman, P. Monte carlo methods in financial engineering, 2004.

Han, H., Wang, Y., Shomer, H., Guo, K., Ding, J., Lei, Y., Halappanavar, M., Rossi, R. A., Mukherjee, S., Tang, X., He, Q., Hua, Z., Long, B., Zhao, T., Shah, N., Javari, A., Xia, Y., and Tang, J. Retrieval-augmented generation with graphs (graphrag), 2025.

Hu, Y., Jiang, Z., Guo, M., and Gong, N. A transfer attack to image watermarks. *arXiv preprint arXiv:2403.15365*, 2024.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

Jiang, P., Cao, L., Xiao, C., Bhatia, P., Sun, J., and Han, J. KG-FIT: Knowledge graph fine-tuning upon open-world knowledge. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. In *International Conference on Machine Learning*, pp. 17061–17084. PMLR, 2023.

Le, D., Zhong, S., Liu, Z., Xu, S., Chaudhary, V., Zhou, K., and Xu, Z. Knowledge graphs can be learned with just intersection features. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, pp. 26199–26214. PMLR, 2024.

LIANG, K., Liu, Y., Li, H., Meng, L., Liu, S., Wang, S., sihang zhou, and Liu, X. Clustering then propagation: Select better anchors for knowledge graph embedding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Liu, A., Pan, L., Lu, Y., Li, J., Hu, X., Zhang, X., Wen, L., King, I., Xiong, H., and Yu, P. S. A survey of text watermarking in the era of large language models, 2024.

Liu, J., Ke, W., Wang, P., Wang, J., Gao, J., Shang, Z., Li, G., Xu, Z., Ji, K., and Li, Y. Fast and continual knowledge graph embedding via incremental lora. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2025.

Nandi, A., Kaur, N., Singla, P., and Mausam. Dynasemble: Dynamic ensembling of textual and structure-based models for knowledge graph completion. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Patnaik, P. The non-central $\chi$ 2-and f-distribution and their applications. *Biometrika*, 36(1/2):202–232, 1949.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pp. 8748–8763. PMLR, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2021.

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks, 2017.

Shomer, H., Jin, W., Wang, W., and Tang, J. Toward degree bias in embedding-based knowledge graph completion. In *Proceedings of the ACM Web Conference 2023*, pp. 705–715, New York, NY, USA, 2023. Association for Computing Machinery.

Simonovsky, M. and Komodakis, N. Graphvae: Towards generation of small graphs using variational autoencoders. In Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., and Maglogiannis, I. (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2018*, pp. 412–422, Cham, 2018. Springer International Publishing.

Smits, J. and Borghuis, T. *Generative AI and Intellectual Property Rights*, pp. 323–344. T.M.C. Asser Press, The Hague, 2022.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2022.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks, 2018.

Vero, M., Balunović, M., and Vechev, M. Cuts: customizable tabular synthetic data generation. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024.

Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., and Frossard, P. Digress: Discrete denoising diffusion for graph generation, 2023.

Wang, H., Zhang, F., Xie, X., and Guo, M. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, pp. 1835–1844, 2018.

Wang, K., Liang, Y., Li, X., Li, G., Ghanem, B., Zimmermann, R., Zhou, Z., Yi, H., Zhang, Y., and Wang, Y. Brave the wind and the waves: Discovering robust and generalizable graph lottery tickets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3388–3405, 2023.

Wang, K., Zhang, G., Zhou, Z., Wu, J., Yu, M., Zhao, S., Yin, C., Fu, J., Yan, Y., Luo, H., et al. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585*, 2025.

Wang, X., He, X., Wang, M., Feng, F., and Chua, T.-S. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 165–174, New York, NY, USA, 2019. Association for Computing Machinery.

Wen, Y., Kirchenbauer, J., Geiping, J., and Goldstein, T. Tree-rings watermarks: Invisible fingerprints for diffusion images. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, pp. 58047–58063. Curran Associates, Inc., 2023.

Wen, Y., Kirchenbauer, J., Geiping, J., and Goldstein, T. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36, 2024.

Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., and Zhou, M. MIND: A large-scale dataset for news recommendation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3597–3606, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.331.

Yan, X. and Han, J. gspan: Graph-based substructure pattern mining. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pp. 721–724. IEEE, 2002.

Yang, R. Guise: Graph gaussian shading watermark, 2024.

Yang, Y., Chen, J., and Xiang, Y. A review on the reliability of knowledge graph: from a knowledge representation learning perspective. *World Wide Web*, 28(1):4, 2024a.

Yang, Z., Zeng, K., Chen, K., Fang, H., Zhang, W., and Yu, N. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12162–12171, 2024b.

Yin, H., Wang, Z., and Song, Y. Rethinking complex queries on knowledge graphs with neural link predictors. In *The Twelfth International Conference on Learning Representations*, 2024.

You, J., Ying, R., Ren, X., Hamilton, W., and Leskovec, J. GraphRNN: Generating realistic graphs with deep auto-regressive models. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, pp. 5708–5717. PMLR, 2018.

Yu, N., Skripniuk, V., Abdelnabi, S., and Fritz, M. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 14448–14457, 2021.

Yu, Z., Zhang, C., and Deng, C. An improved gnn using dynamic graph embedding mechanism: A novel end-to-end framework for rolling bearing fault diagnosis under variable working conditions. *Mechanical Systems and Signal Processing*, 200:110534, 2023.

Zhang, H., Zheng, T., Gao, J., Miao, C., Su, L., Li, Y., and Ren, K. Data poisoning attack against knowledge graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4853–4859. International Joint Conferences on Artificial Intelligence Organization, 2019.

Zhang, L., Liu, X., i Martin, A. V., Bearfield, C. X., Brun, Y., and Guan, H. Attack-resilient image watermarking using stable diffusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Zhang, X., Li, R., Yu, J., Xu, Y., Li, W., and Zhang, J. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11964–11974, 2024b.

Zhao, X., Ananth, P., Li, L., and Wang, Y.-X. Provable robust watermarking for ai-generated text, 2023a.

Zhao, X., Zhang, K., Su, Z., Vasan, S., Grishchenko, I., Kruegel, C., Vigna, G., Wang, Y.-X., and Li, L. Invisible image watermarks are provably removable using generative ai. *arXiv preprint arXiv:2306.01953*, 2023b.

Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N.-M., and Lin, M. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023c.

Zhou, Z., Zhang, Y., Yao, J., quanming yao, and Han, B. Less is more: One-shot subgraph reasoning on large-scale knowledge graphs. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhu, M., Chen, H., Yan, Q., Huang, X., Lin, G., Li, W., Tu, Z., Hu, H., Hu, J., and Wang, Y. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in Neural Information Processing Systems*, 36, 2024a.

Zhu, P., Takahashi, T., and Kataoka, H. Watermark-embedded adversarial examples for copyright protection against diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24420–24430, 2024b.

# A. Defending Against Isomorphism and Structural Variations

**Algorithm 1** Graph Alignment

```
Require: Input graph $G = (V, E)$ with vertex attributes and adjacency matrix $A$
Ensure: Normalized graph $\hat{G}$
For each $v \in V$:
    $d[v] \leftarrow \deg(v)$, $c[v] \leftarrow c(v)$    Compute the degree and clustering coefficient for each vertex
$V_o \leftarrow \text{Sort}(V, d, c)$    Sort $V$ first by degree $d[v]$ and then by clustering coefficient $c[v]$ in ascending order
$A' \leftarrow \text{Reorder}(A, V_o)$    Reorder rows and columns of $A$ according to the order in $V_o$
$\hat{G} \leftarrow \emptyset$
for $v_i \in V_o$ do
    $\hat{G} \leftarrow \hat{G} \cup \{v_i\}$    Add vertex $v_i$ to $\hat{G}$
end for
for $(v_i, v_j) \in E$ do
    $\hat{G} \leftarrow \hat{G} \cup \{(v_i, v_j)\}$    Add edge $(v_i, v_j)$ to $\hat{G}$
end for
for $v_i \in V_o$ do
    RearrangeAttributes$(v_i, \hat{G})$    Rearrange or update attributes of $v_i$ based on the new order in $\hat{G}$
end for
return $\hat{G}$
```

**Symbol Definitions:**

- $d[v]$: The degree of vertex $v$, i.e., the number of edges incident to $v$.
- $c[v]$: The clustering coefficient of vertex $v$ measures the degree to which vertices in a graph tend to cluster together.
- $\hat{G}$: The normalized graph after vertex and edge processing.
- RearrangeAttributes$(v_i, \hat{G})$: A function to reorder or update the attributes of vertex $v_i$ in the normalized graph $\hat{G}$ based on the sorted vertex list $V_o$.

Algorithm 1 outlines a graph alignment process designed to enhance the robustness of a graph against isomorphism and structural variations, which are common forms of attack. The process begins by calculating the degree and clustering coefficient for each node. These attributes reflect the node's connectivity and the density of its neighbors, which are crucial for determining the node's importance within the graph.

Next, the nodes are sorted based on these attributes, first by degree and then by clustering coefficient. This sorting ensures that nodes with lower connectivity are processed first, while more central nodes appear later. Following the sorting, the graph's adjacency matrix is reordered to match the new node order, preserving the graph's original relationships.

A new graph, $\hat{G}$, is created by adding nodes in the sorted order. As each node is added, its attributes are updated according to the new structure, ensuring consistency. Finally, the edges are added to the new graph based on the original connections, but aligned with the new node ordering.

The output is a normalized graph that is less susceptible to structural perturbations. This process helps improve the graph's resilience to attacks that alter its structure, making it more stable for applications like watermarking, where the goal is to embed and extract watermarks while maintaining robustness against manipulation.

Algorithm 2 describes a process for redundantly embedding a watermark into a graph based on its subgraphs. The aim is to enhance the robustness of the watermark against structural modifications and attacks by embedding it in multiple subgraph communities.

The process starts by computing the number of communities $l$ based on the graph's total number of vertices $|V|$ and a predefined smallest embedded size $s$. The graph $G$ is then partitioned into $l$ non-overlapping communities $C_1, C_2, \ldots, C_l$, ensuring that every vertex in the graph is assigned to exactly one community, and the communities do not overlap.

**Algorithm 2** Redundant Embedding Based on Subgraphs

```
**Input:** Graph $G = (V, E)$, Smallest embedded size $s$, Watermark $W$
**Output:** Watermarked graph $G'$
$l \leftarrow \lfloor |V|/s \rfloor$    Compute the number of communities
$C \leftarrow \{C_1, C_2, \ldots, C_l\}$
when $\bigcup_{i=1}^{l} C_i = V$ and $C_i \cap C_j = \emptyset, \forall i \neq j$    Partition $G$ into $l$ non-overlapping communities
**for** each community $c \in C$ **do**
    $V_{C_i} \leftarrow \{v \mid v \in C_i,$    selected based on a predefined strategy$\}$
    $c_{norm} \leftarrow$ GraphNormalization($c$, $A_c$)    Normalize the graph structure
    **for** each vertex $v \in V_c$ **do**
        EmbedWatermark($v$, $W$)    Embed watermark into selected vertices
    **end for**
**end for**

**return** $G'$
```

Next, for each community $c \in C$, the algorithm selects a subset of vertices, $V_{C_i}$, based on a predefined strategy, which could be influenced by factors such as node importance or connectivity. The subgraph corresponding to each community, $c$, is then normalized using the ‘GraphNormalization‘ function. This step ensures that the internal structure of each community is standardized, making the watermark embedding more robust to structural variations.

After normalizing the graph structure, the watermark is embedded into selected vertices of each community using the ‘EmbedWatermark‘ function. This embedding process is repeated for each vertex in every community. By embedding the watermark redundantly across different communities of the graph, the method increases the likelihood of watermark retention even if parts of the graph undergo modification or attack.

Finally, the watermarked graph $G'$ is returned, which contains the embedded watermark in a redundant and robust manner, ensuring better resilience to structural changes or adversarial perturbations. This approach is particularly useful for applications that require high security, such as digital watermarking in graphs, where the goal is to protect the integrity of the watermark despite potential attacks.

## B. Case Study in Embedding Space

To further demonstrate the effectiveness of our proposed watermarking method, a case study is conducted on the dataset AliF related to customer behavior analysis. We select 2400 entity embeddings from five communities (A-E) from AliF for watermark embedding and visualize the embedding vectors before and after processing by dimensionality reduction. The case background is presented in Table 5, the result of the visualization is shown in Figure 5

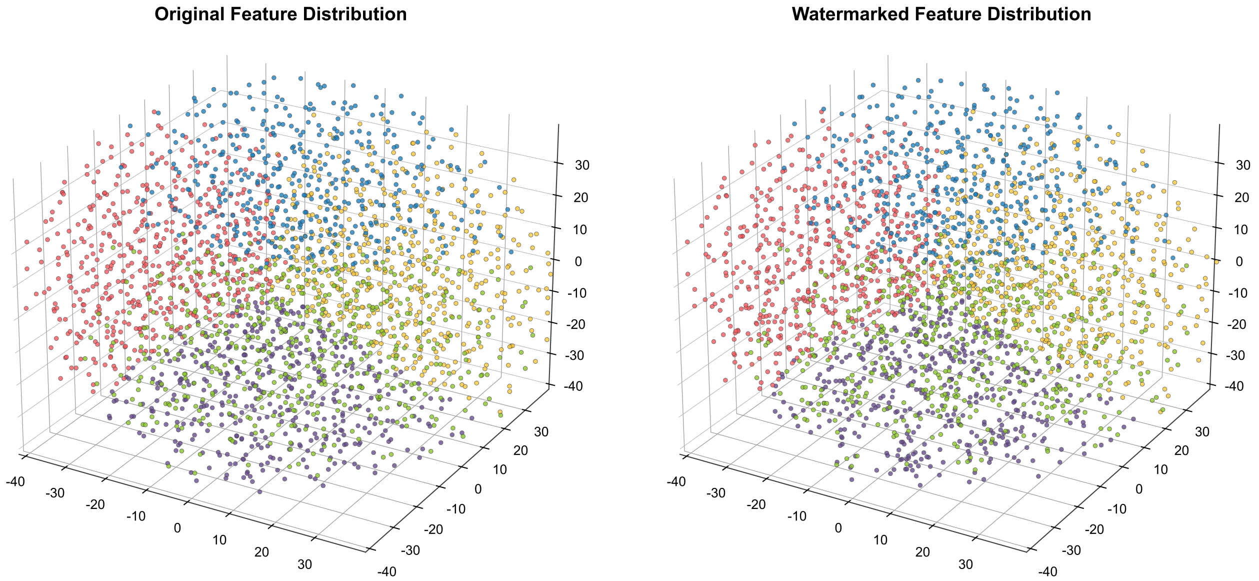
*Table 5.* Case Background.

| **Number of Entities** | | | | | **total** | **Dimensions of Embeddings** | **DDIM Inference Steps** | **Density of Mask** | **Cosine Similarity** |
|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | | | | | |
| 524 | 507 | 524 | 420 | 425 | 2400 | 4096 | 75 | 0.015 | 0.9535 |

For visualization purposes, we used the t-SNE technique to reduce the dimensionality of the embedding vectors from 4096 to 3. This technique is particularly suitable for mapping data from high-dimensional to two-dimensional or three-dimensional space while preserving local structural relationships between data points.

The similarity between points $x_i$ and $x_j$ in a high-dimensional space is defined by the conditional probability $p_{j|i}$ as follows, where $\|x_i - x_j\|^2$ represents the squared Euclidean distance. $\sigma_i$ is a parameter that controls the width of the neighborhood of the point $x_i$.

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)} \tag{25}$$


Original Feature Distribution

Watermarked Feature Distribution

*Figure 5.* Ablation study of the density of the watermark mask matrix on all datasets. Each subfigure corresponds to a different method.

To symmetrize the similarity, the joint probability is defined as follows, where $n$ is the total number of data points.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{26}$$

In a low-dimensional space, the similarity between the points $y_i$ and $y_j$ is defined by the t-distribution as:

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}} \tag{27}$$

t-SNE uses Kullback-Leibler (KL) divergence as a dissimilarity measure between a high-dimensional distribution $p_{ij}$and a low-dimensional distribution $q_{ij}$. The goal is to minimize the following loss function:

$$\mathcal{L} = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{28}$$

t-SNE optimizes the loss function $\mathcal{L}$ using Stochastic Gradient Descent (SGD). During the optimization, the position of the low-dimensional embedding $y_i$ is gradually adjusted, so that the low-dimensional similarity $q_{ij}$ is closer to the high-dimensional similarity $p_{ij}$.

In Figure 5, it can be found that the color gradient and position distribution of the points in the figure almost remain unchanged before and after the embedding of the watermark, indicating that the embedding of the watermark does not significantly change the geometric structure of the original embedding. In addition, both images show a spherical distribution that gradually diffuses from the center to the outside, and the density of the points remains the same, indicating that the characteristics of the original vector embedding are well preserved after embedding the watermark. The watermark embedding may introduce only slight perturbations in some local details or some dimensions, but these perturbations are not sufficient to cause significant effects on the overall distribution.

Then, we carry out watermark detection on the watermarked knowledge graph, and we select the significance level as 5e-5. We report their P-values to check whether the watermark is detected. In summary, our watermarking method performs well in terms of transparency. It neither destroys nor enhances the distribution characteristics of the original embeddings, but also achieves certain robustness and detectability.

*Table 6.* Case Result.

| Community | P-value | Is Detected |
|---|---|---|
| A | $8.29e-27$ | ✓ |
| B | $1.52e-26$ | ✓ |
| C | $4.20e-19$ | ✓ |
| D | $1.88e-15$ | ✓ |
| E | $3.27e-34$ | ✓ |

## C. Latent Diffusion Model

### C.1. Variational Autoencoder

To effectively capture the intricate structure of graph data, we employ a Variational Autoencoder (VAE). This advanced model maps the entity embeddings of the knowledge graph into a latent space, where the underlying patterns and relationships can be more efficiently represented and analyzed. To ensure that we can adequately and accurately capture the complex patterns inherent in the graph structure, we utilize multiple Relational Graph Attention (RGAT) modules. These modules are specifically designed to handle the relational nature of the graph data, allowing the model to focus on different types of relationships and entities within the graph. By leveraging the power of multiple RGAT modules, we can produce higher-quality latent representations that better encapsulate the essential characteristics of the graph structure. These embeddings are then sampled using the reparameterization trick. This technique enables us to introduce stochasticity in a differentiable manner, which is crucial for the training process of the VAE. Finally, the sampled embeddings are decoded back into the reconstructed entity embeddings, allowing us to evaluate the quality of the learned representations and the model's overall performance in capturing and reconstructing the graph data.

### C.2. Diffusion, DDIM and Inversion

Denoising Diffusion Implicit Models (DDIM) provide an efficient alternative to standard diffusion-based generative models by leveraging a non-Markovian forward and reverse process. This section introduces the DDIM sampling process, its inversion, and the necessary mathematical formulation and derivation.

**Forward Diffusion Process:**

The forward process in DDIM maps a data point $\mathbf{Z}_0 \in \mathbb{R}^d$ to a noisy latent variable $\mathbf{Z}_T$ over $T$ timesteps. It is defined as:

$$\mathbf{Z}_t = \sqrt{\alpha_t}\mathbf{Z}_0 + \sqrt{1-\alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{29}$$

where $\alpha_t$ is a scheduling parameter that controls the noise variance at timestep $t$.

**Reverse Diffusion Process:**

The reverse process reconstructs $\mathbf{Z}_0$ from $\mathbf{Z}_T$ by iteratively denoising the latent variable. DDIM modifies the reverse process to make it deterministic, defined as:

$$\mathbf{Z}_{t-1} = \sqrt{\alpha_{t-1}}\hat{\mathbf{Z}}_0 + \sqrt{1-\alpha_{t-1}}\epsilon_t, \tag{30}$$

where $\hat{\mathbf{Z}}_0$ is the predicted clean data at timestep $t$ and is obtained via:

$$\hat{\mathbf{Z}}_0 = \frac{\mathbf{Z}_t - \sqrt{1-\alpha_t}\epsilon_t}{\sqrt{\alpha_t}}. \tag{31}$$

The noise $\epsilon_t$ is predicted using a pre-trained noise estimation model, typically parameterized as $\epsilon_\theta(\mathbf{Z}_t, t)$.

**DDIM Sampling:**

DDIM employs a deterministic sampling strategy by reparameterizing the reverse process. Specifically, the update step becomes:

$$\mathbf{Z}_{t-1} = \sqrt{\alpha_{t-1}}\hat{\mathbf{Z}}_0 + \sqrt{1-\alpha_{t-1}} \cdot \eta \cdot \epsilon_t, \tag{32}$$

where $\eta \in [0, 1]$ controls the stochasticity of the process. Setting $\eta = 0$ results in a fully deterministic process, while $\eta > 0$ introduces controlled randomness.

**Inversion Process:**

DDIM supports exact inversion, enabling the recovery of $\mathbf{Z}_t$ from $\mathbf{Z}_{t-1}$. By rearranging Eq. (32), the inversion is expressed as:

$$\mathbf{Z}_t = \sqrt{\alpha_t}\hat{\mathbf{Z}}_0 + \sqrt{1 - \alpha_t} \cdot \eta \cdot \epsilon_t. \tag{33}$$

This exact inversion capability is crucial for tasks such as embedding watermarks or debugging generative processes.

## D. Watermark Embedding in Frequency Domain

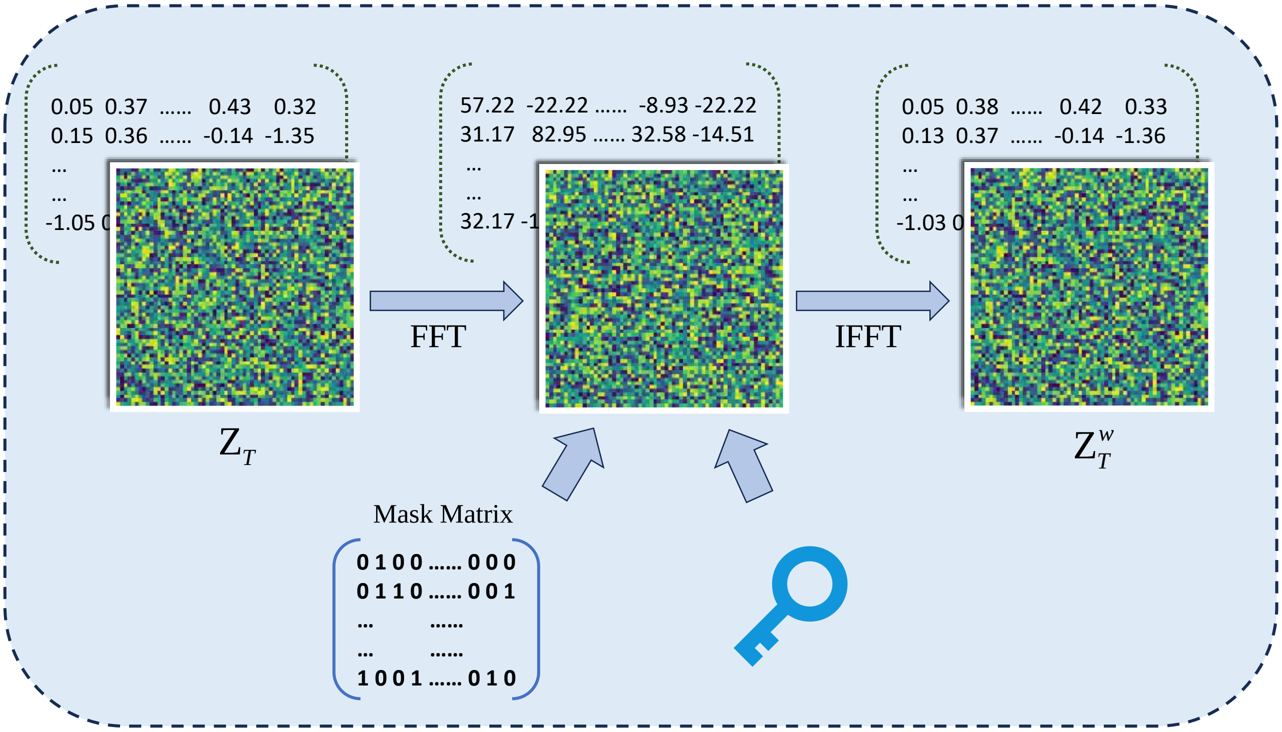


*Figure 6.* Watermark Embedding in Frequency Domain.

The theoretical basis of watermark embedding in the frequency domain is mainly based on the frequency domain transformation technology in signal processing. The essence of the Fourier transform we employ is to separate the time domain and frequency domain characteristics, and the Fourier transform decomposes the time domain signal into frequency domain components; that is, the signal is represented as a superposition of sine waves with different frequencies. In the frequency domain, Low-frequency components usually contain the main energy of the signal (such as the overall structure or low-resolution features). High-frequency components: Describe rapid changes in the signal (such as detail or texture information). The watermark is embedded in some specific frequency components of the frequency domain (such as middle frequency or high frequency), which can not only hide the watermark but also reduce the impact on the data's original structure. The frequency domain properties after the Fourier transform are robust to certain linear operations such as compression, smoothing, rotation, and cropping: for example, the rotation operation only causes a phase change in the frequency domain, while the amplitude spectrum remains stable. This makes the method of embedding the watermark in the frequency domain highly robust to these transformations.

Our method uses the energy distribution characteristics, perceptual sensitivity, and frequency domain stability of the Fourier transform frequency domain to embed the watermark into the frequency domain, and then realizes the flexible and adaptive selection of the frequency region through our LAWMM to achieve the goal of invisibility, robustness, and detectability of the embedded watermark. These theoretical foundations provide a solid foundation for frequency domain watermarking technology and also explain its advantages in the face of common attacks such as compression, geometric transformation, and smoothing.

## E. Metrics

### E.1. Detectability and Robustness

We use the AUC as an authoritative indicator for evaluating the detectability and robustness of our watermarking methods.

- **Area Under Curve (AUC)**: AUC refers to the area under the Receiver Operating Characteristic (ROC) curve, a widely used metric for evaluating classification models. The AUC score ranges from 0 to 1, with a value closer to 1 indicating better model performance. It represents the probability that a randomly chosen positive sample will be ranked higher than a randomly chosen negative sample. Mathematically, AUC can be expressed as:

$$\text{AUC} = \frac{1}{N_p N_n} \sum_{i=1}^{N_p} \sum_{j=1}^{N_n} \mathbb{I}(y_i > y_j) \tag{34}$$

  where $N_p$ and $N_n$ are the numbers of positive and negative samples, respectively, $y_i$ and $y_j$ are the predicted scores of the positive and negative samples, and $\mathbb{I}(y_i > y_j)$ is the indicator function that is 1 if $y_i > y_j$, and 0 otherwise.

### E.2. Transparency

We evaluate the transparency of watermarking from two dimensions: the **similarity** of knowledge graph embedding before and after watermarking and the **quality** of watermarked knowledge graph. Therefore, we used the following two categories of evaluation metrics.

We use cosine similarity to evaluate the **similarity** of knowledge graphs.

- **Cosine Similarity**: Cosine similarity is a similarity measure between two non-zero vectors in an inner product space. It is widely used in many fields, including information retrieval, natural language processing, and machine learning. The cosine similarity between two vectors $\mathbf{A}$ and $\mathbf{B}$ is defined as:

$$\text{cosine similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \tag{35}$$

For quality, we use the following four metrics. Given a set of test triples $\mathcal{T} = \{(h, r, t)\}$ in a knowledge graph $\mathcal{G}$, let $r(h, t)$ denote the rank of the correct entity $t$ (or $h$) among all candidates in a ranking task.

- **Geometric Mean Rank (GMR)**: GMR captures the overall ranking tendency in a multiplicative rather than additive manner, making it particularly suitable for datasets where ranking distributions exhibit heavy tails. It provides insights into the overall ranking distribution, making it useful for comprehensive model evaluation. The GMR is defined as follows:

$$\text{GMR} = \left( \prod_{(h,r,t) \in \mathcal{T}} r(h, t) \right)^{\frac{1}{|\mathcal{T}|}}. \tag{36}$$

- **Harmonic Mean Rank (HMR)**: HMR mitigates the influence of extremely high ranks, ensuring that poor predictions do not disproportionately impact the evaluation. Moreover, HMR is expressed directly in rank space, making it more interpretable while still benefiting from reciprocal weighting.

$$\text{HMR} = \frac{|\mathcal{T}|}{\sum_{(h,r,t) \in \mathcal{T}} \frac{1}{r(h,t)}}. \tag{37}$$

- **Arithmetic Mean Rank (AMR)**: AMR directly reflects the average rank of correct entities, making it easy to understand and compare across models. A major limitation of AMR is that it is heavily influenced by extremely high-rank values, which can distort performance evaluation.

$$\text{AMR} = \frac{1}{|\mathcal{T}|} \sum_{(h,r,t) \in \mathcal{T}} r(h, t). \tag{38}$$

- **Hits@k**: Hits@$k$ provides a direct measure of retrieval success within a fixed rank cutoff, making it useful for real-world applications requiring top-$k$ recommendations.

$$\text{Hits@}k = \frac{1}{|\mathcal{T}|} \sum_{(h,r,t)\in\mathcal{T}} \mathbb{I}[r(h,t) \leq k], \tag{39}$$

where $\mathbb{I}[\cdot]$ is the indicator function that returns 1 if the condition holds and 0 otherwise. This metric evaluates whether the correct entity appears within the top-$k$ ranks.

## F. Notations and Definitions

*Table 7.* Notations and Definitions

| Notation | Definition |
|---|---|
| $F(\cdot)$ | Fourier Transform |
| $M$ | Watermark mask matrix |
| $S$ | Watermark signature |
| $\sigma^2$ | Variance of the watermark signature |
| $Z_T^w$ | Watermarked latent representation |
| $Z_T^{INV}$ | Initial noise vector recovered via DDIM inversion |
| $L$ | Loss function in watermark embedding |
| $\epsilon$ | Threshold for latent space equilibrium |
| $\delta$ | Total perturbation in an attack |
| $\delta_k$ | Perturbation applied to each subgraph $G_k$ |
| $q$ | Norm type in the attack objective |
| $Y$ | Fourier transform result during watermark extraction |
| $\mu_i$ | Mean of $Y$ |
| $\sigma_i^2$ | Variance of $Y$ |
| $T$ | Test statistic for watermark detection |
| $\lambda$ | Likelihood ratio test statistic |
| $\hat{T}$ | Simplified test statistic |
| $p$ | P-value for watermark detection |
| $\alpha$ | Significance level for watermark detection |
| $\rho$ | Density of the watermark mask matrix |
| $\Phi(C_i)$ | Watermark embedding in the community layer |
| $\Psi(v)$ | Watermark embedding in the vertex layer |
| $C(W)$ | Effective information capacity of watermark encoding |
| $H(\cdot)$ | Entropy |
| $D(G, \tilde{G})$ | Divergence metric between graphs $G$ and $\tilde{G}$ |
| $L^\dagger$ | Pseudoinverse of the graph Laplacian |
| $\Delta A$ | Perturbation matrix for the graph |
| $\eta(v)$ | Centrality of vertex $v$ |
| $\hat{Z}_0$ | Predicted clean data in DDIM inversion |
| $\eta$ | Stochasticity control parameter in DDIM sampling |
| $\epsilon_t$ | Predicted noise in DDIM reverse process |
| $\alpha_t$ | Scheduling parameter in DDIM forward diffusion process |
| $\sigma_i$ | Parameter controlling neighborhood width in t-SNE |
| $p_{ij}$ | Joint probability in high-dimensional space |
| $q_{ij}$ | t-distribution similarity in low-dimensional space |
| $L$ | Loss function in t-SNE optimization |