Semantic Coherence-Based Stego-Anomaly Detection

Kartik Pandey
Department of Computer Science and Engineering
IIIT Naya Raipur, Chhattisgarh, India
Email: kartik23100@iiitnr.edu.in

Anwita Chakraborty
Department of Computer Science and Engineering
IIIT Naya Raipur, Chhattisgarh, India
Email: anwita23100@iiitnr.edu.in

Satyanarayana Vollala
Department of Computer Science and Engineering
IIIT Naya Raipur, Chhattisgarh, India
Email: satya@iiitnr.edu.in

Sagnik Mazumdar
Department of Computer Science and Engineering
IIIT Naya Raipur, Chhattisgarh, India
Email: sagnik22100@iiitnr.edu.in

Abhijeet Kar
Department of Computer Science and Engineering
IIIT Naya Raipur, Chhattisgarh, India
Email: abhijeetk@iiitnr.edu.in

Kamna Sahu
Department of Computer Science and Engineering
IIIT Naya Raipur, Chhattisgarh, India
Email: kamna@iiitnr.edu.in

***Abstract*—Recent steganography techniques have become subtle enough that many classical pixel–level detectors fail to pick up their traces. While working with a range of image and audio stego samples, we observed that although the low-level statistics may remain almost unchanged, the higher-level meaning or context of the media often drifts in ways that are not entirely natural. This simple observation led us to explore whether semantic inconsistencies, rather than purely statistical artifacts, could serve as a reliable signal for detection. In this paper, we present an unsupervised framework that builds semantic coherence graphs from both images and audio. Images are divided using SLIC superpixels and FastSAM to obtain regions that are semantically meaningful at different scales, while audio is processed through overlapping temporal windows. Each region or frame is then embedded using pretrained models such as CLIP, ViT, and wav2vec2. The resulting graph captures how these parts relate to one another, and we extract a mixture of spectral, structural, and classical steganalysis features from it. These features are used to train Isolation Forest and One Class SVM models on clean data only.**

***Index Terms*—Semantic coherence, multimodal anomaly detection, steganalysis, graph neural networks, explainable AI, CLIP embeddings, LSB detection, Isolation Forest, One-Class SVM, quantum computing limitations, digital forensics**

I. Introduction

Digital steganography has quietly evolved into a much harder problem than it was even a decade ago. Many of the traditional detectors, which rely on finding unusual noise patterns or distortions in pixel values, work well when the embedding is heavy handed, but their effectiveness drops sharply against content adaptive or low payload techniques. While examining recent methods, we found that although the statistical fingerprints are sometimes almost perfectly hidden, the semantic or contextual relationships inside an image or audio clip can still feel slightly "off." Two regions that should naturally share a certain similarity do not, or an audio segment transitions in a way that feels forced. This is subtle and not always visible at first glance, but it shows up consistently enough to be worth investigating more formally.

Most existing steganalysis systems concentrate on low level structure: residual noise patterns, DCT coefficients, convolutional filters trained specifically to amplify embedding artifacts, and so on. These approaches have achieved strong results, particularly when they are trained directly on the same algorithm they will later be tested against. However, this advantage becomes a weakness when the embedding technique changes, since such systems tend to overfit to the statistical behaviour of a specific scheme. An additional difficulty is that their decisions are often difficult to interpret, which is a problem in forensic or security settings where an analyst needs to understand *why* a sample was flagged.

The work presented in this paper grew from the idea that a broader, more semantic view of the media might be useful. Instead of examining pixel level anomalies in isolation, we look at how different regions of an image or segments of an audio signal relate to each other. By combining SLIC superpixels, FastSAM segmentation, and temporal windowing with embeddings from CLIP, ViT, and wav2vec2, we obtain a set of region level descriptors that capture not just texture or frequency information but also the higher level content. These descriptors form the nodes of a graph in which edges represent semantic or spatial relationships. The graph tends to have a characteristic structure for natural, unmodified media, and we use its deviations from this structure as a signal for anomaly detection.

We evaluate this idea through a mixture of graph spectral features, classical LSB-based cues, wavelet statistics, and a few other descriptors that turned out to be surprisingly informative. Since we train only on clean data, the approach generalizes better to new embedding methods than a supervised deep network typically would. In practice, we found that the system performs reliably across several datasets and that

the heatmaps it produces often highlight precisely the regions an analyst would suspect on manual inspection.

Along the way, we also considered whether quantum image representations might offer a path to faster or more scalable detection. Although the theoretical literature is intriguing, current NISQ devices simply do not provide enough qubits or sufficiently stable state-preparation mechanisms to handle realistic image sizes, and even simulated experiments show that the overhead is prohibitive. For now, classical methods remain far more practical.

The remainder of this paper develops our framework in detail, beginning with a discussion of related work and followed by the full methodology, experiments, and observations from both the classical and quantum perspectives.

II. Related Work and Background

Research in steganalysis has evolved through several distinct phases, each shaped by the types of embedding techniques that were most common at the time. Much of the early work was rooted in statistical analysis, where the goal was to capture small perturbations caused by embedding bits into images or audio signals. Over time, as both steganographic and machine learning methods matured, the field shifted toward deep learning and, more recently, toward more semantically aware approaches. The idea of using semantic structure as a detection cue has appeared only sporadically, and usually in other domains, which partly motivated the present work.

A. Traditional Steganalysis Techniques

The classical literature largely distinguishes between universal detectors, which attempt to spot any type of steganography regardless of how it is carried out, and targeted detectors, which are designed with a particular embedding algorithm in mind. Early universal detectors usually relied on handcrafted statistical features. A well-known example is the Spatial Rich Models (SRM) introduced by Fridrich et al. [1]. SRM features capture subtle dependencies in noise residuals by applying many different high-pass filters. When a payload is reasonably large these features can be very effective, but they become less reliable when the payload is small or when the embedding is strongly content-adaptive. Similar ideas were applied to JPEG images through analysis of quantized DCT coefficients; Holub et al.'s work on phase-aware features [2] became an important reference point for JPEG steganalysis, although it required substantial format-specific engineering.

1) Deep Learning Approaches: The arrival of deep learning fundamentally changed the landscape. Qian et al. demonstrated that convolutional networks, if carefully constructed, could learn residual-like filters directly from data [3]. Their work initiated a series of CNN architectures designed specifically for steganalysis, many of which employ constrained convolutions, wide first layers, or residual connections to stabilize training. One of the strongest models in this line is SRNet [4], which achieves impressive accuracy on BOSSBase, particularly at higher payloads. Despite these successes, deep networks often inherit a limitation of supervised learning: they tend to specialize in the embedding algorithm they were trained on and do not always generalize well to new methods. Surveys such as [3], [5] have noted this issue repeatedly, along with the persistent difficulty of providing interpretable outputs.

B. Semantic and Content-Aware Approaches

Although most steganalysis work focuses on low-level signals, the last few years have seen increasing interest in using high-level semantic representations for anomaly detection more broadly. The emergence of large pretrained models has played a major role in this shift. CLIP [6], for instance, learns a joint embedding space for images and text, making it surprisingly robust to small pixel-level perturbations. Ma et al. have shown that CLIP's semantic stability can be exploited for zero-shot anomaly detection in industrial settings, and their results motivated us to explore whether similar benefits might carry over to steganalysis. Vision Transformers (ViT) offer another route to semantic representation, as their self-attention mechanism allows them to capture long-range relationships that conventional CNNs sometimes overlook. This ability is particularly useful when one wants to detect disruptions in the overall structure or meaning of an image rather than changes in individual pixels.

A similar shift has occurred in audio analysis. Traditional audio steganalysis typically revolves around the behaviour of samples in the time or frequency domain, or sometimes in the cepstral domain. Methods designed to detect LSB embedding in WAV files or phase-based embedding in compressed audio formats have been extensively studied. More recently, however, models such as wav2vec2 [7] and the Audio Spectrogram Transformer (AST) have become attractive because they offer rich, context-aware embeddings learned from very large corpora. These embeddings tend to be stable under small distortions, which makes them suitable for our goal of capturing whether neighbouring audio frames "agree" with each other semantically.

C. Graph-Based Anomaly Detection

The idea of using graphs to model relationships between parts of an object or signal arises naturally in many areas of computer vision and pattern analysis. Region Adjacency Graphs (RAG) have long been used to describe how segments of an image relate to one another, whether in terms of spatial arrangement or feature similarity [8]. Spectral graph theory provides the tools to study such graphs in detail; eigenvalues of the graph Laplacian, for example, reveal information about connectivity, bottlenecks, and community structure [9]. These ideas have proven valuable in anomaly detection more broadly, because many anomalies correspond to disruptions in these structural patterns.

Graph anomaly detection is a fairly active research area, with methods ranging from simple statistical measures of node connectivity to more sophisticated approaches involving graph neural networks. Akoglu et al.'s survey [9] provides a useful overview of node-level, edge-level, and subgraph-level anomalies. More recent methods employ GNN-based

autoencoders [10], which work well when large collections of graphs are available for training. In our case, however, each sample yields a single graph, and we focus on extracting graph descriptors rather than training graph-level neural networks.

D. Quantum Image Processing: Promise and Limitations

Quantum computing has inspired several proposals for alternative image representations that, at least in theory, allow certain operations to be carried out exponentially faster than on classical hardware. FRQI [11] and NEQR [12] are two widely discussed encoding schemes. FRQI represents pixel intensities as rotation angles, while NEQR stores them explicitly using additional qubits. For a modest 256×256 grayscale image, the NEQR representation requires roughly 24 logical qubits. This may not sound large, but in practice the number of physical qubits needed to protect such a state from noise can be in the thousands.

Even setting aside qubit count, several practical difficulties remain. Preparing a quantum state containing classical image data usually takes time proportional to the number of pixels unless one assumes access to a fully functional QRAM, which remains an open engineering challenge [13]. Moreover, most quantum image processing algorithms rely on deep circuits with many controlled operations, which quickly become unreliable on present-day NISQ devices [14]. These obstacles make it hard to move beyond theoretical discussion and apply such methods to real image sizes.

In contrast, classical methods continue to benefit from rapid improvements in GPU hardware and software. Modern GPUs can process high-resolution images in real time, and deep models such as SRNet [4] have shown strong and consistent performance on widely used benchmarks. Since they require no specialized hardware, these classical approaches remain more practical for now.

E. Unsupervised vs. Supervised Learning

A recurring challenge in steganalysis is the reliance on large labeled datasets, particularly for supervised deep learning approaches. Creating such datasets means generating cover–stego pairs for each combination of embedding algorithm, payload, and sometimes even image resolution. This process can be tedious and, more importantly, causes models to become closely tied to the distribution they were trained on. Once the embedding method changes even slightly, detection performance often declines.

Unsupervised learning offers a different perspective. Methods such as One-Class SVM [15] or Isolation Forest [16] attempt to learn the structure of clean data alone. Anything that deviates sufficiently from this learned structure is treated as an anomaly. Such approaches are attractive in steganalysis because they do not depend on knowing the embedding algorithm in advance and can adapt naturally to new or evolving techniques. This flexibility is one of the reasons we adopt an unsupervised approach in our own framework.

III. Methodology

A. System Architecture Overview

The overall system grew out of a fairly simple question about how to represent media in a way that preserves its internal relationships rather than just its low-level details. After a number of preliminary experiments, we settled on a five-part pipeline. The first stage standardizes the input and extracts regions or frames that are meaningful. The second stage computes semantic embeddings for these parts. We then build a graph that captures how these segments relate to each other. From that graph (and from the underlying signal), we compute a collection of features. Finally, these features are fed into an unsupervised anomaly detector. A schematic overview of this workflow is shown in Fig. 1, although the actual implementation contains several practical refinements that are easier to explain in context.

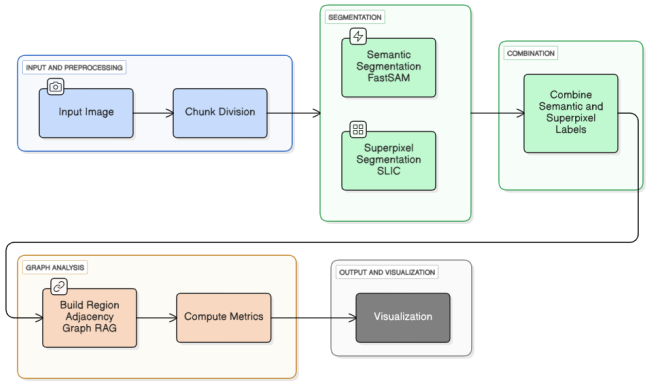

Fig. 1: High-level view of the system, illustrating the processing pipeline for both image and audio inputs.

B. Image Processing Pipeline

1) Preprocessing and Standardization: All images are first brought to a common resolution (either 256×256 or 512×512 depending on the experiment). This step may appear trivial, but it simplifies nearly everything that follows, especially when comparing graphs built from different samples. Pixel values are normalized to the $[0,1]$ range, and in some cases slight gamma adjustments are applied to compensate for strong lighting imbalances that tended to confuse the segmentation models. These steps do not alter the semantics of the image, but they make the segmentation more stable.

2) Hierarchical Segmentation: Segmentation turned out to be one of the more delicate parts of the pipeline. We needed something that could capture small, texture-based distinctions without losing broader semantic structure. Using any single method alone usually led to over- or undersegmentation in certain regions, so we combined two complementary techniques.

The first layer uses SLIC superpixels [17]. SLIC is effective at dividing an image into compact, roughly homogeneous patches. For large images, we found it more practical to

apply SLIC in chunks and then stitch the label maps back together; otherwise memory usage spikes unnecessarily. We experimented with the superpixel count, but settled on about fifty per chunk with a compactness of ten, which gave a good balance between spatial regularity and sensitivity to small structures.

The second layer uses the FastSAM system [18], which segments objects without requiring any manual prompts. FastSAM is based on a YOLOv8 backbone and tends to identify semantically meaningful regions such as people, animals, or well-defined objects. We ran it at an internal resolution of 1024×1024 using a slightly conservative confidence threshold so that it would include faint or partially occluded objects that SLIC alone could never capture. Because SLIC and FastSAM operate so differently, their segment boundaries rarely align perfectly, so after inference we reproject all FastSAM masks back into the coordinate system used by the superpixels.

After both segmentation stages are complete, we merge the two hierarchies by assigning a unique combined label to every pair of (superpixel, semantic-region) intersections:

$$\text{combined\_label}(x, y) = \text{superpixel}(x, y) \cdot N_{\text{sem}} + \text{semantic}(x, y).$$

The effect is that fine-grained textures are preserved from SLIC while large semantic objects from FastSAM prevent fragmentation of important regions. In practice this combination tends to behave more consistently than either method alone.

We evaluated the segmentation quality using the undersegmentation error (UE) from [17], which measures how much superpixels spill over ground-truth semantic boundaries. On our validation images the combined method yielded UE values around 0.15 or lower, which we found acceptable for the downstream tasks.

3) Embedding Extraction: Once the regions are defined, each one is embedded into a semantic vector space. We mainly use the CLIP ViT-B/32 encoder for this purpose. Each region is cropped, resized to CLIP's 224×224 input geometry, and forwarded through the model to obtain a 512-dimensional embedding:

$$\mathbf{e}_i^{\text{CLIP}} = \text{CLIP}_{\text{vision}}(\text{resize}(r_i)).$$

We also experimented with Vision Transformer (ViT-Base) embeddings, which are slightly higher in dimensionality. ViT does a better job of preserving spatial structure because of its patch-level attention mechanism, and in a few preliminary tests it appeared to make the resulting graphs slightly smoother. However, CLIP's robustness to visual noise made it our primary choice.

As illustrated in Fig. 2, SLIC captures fine-grained regions while FastSAM extracts semantic objects.

C. Audio Processing Pipeline

1) Temporal Segmentation: The audio pipeline mirrors the image pipeline in spirit, though not in structure. An audio signal $y[n]$ sampled at rate f_s is divided into overlapping

Fig. 2: Comparison of SLIC and FastSAM segmentation on sample image showing semantic region boundaries

frames. We settled on half-second frames with 50% overlap, so the i-th frame is

$$\text{frame}_i = y[iH : iH + L],$$

with $L = f_s/2$ and $H = f_s/4$. The exact choice of window length is not especially critical, but these values struck a good balance between local detail and global continuity.

2) Audio Embeddings: Each frame is then embedded using wav2vec2 [7], a model that has been extensively trained on large speech corpora and produces rich contextual representations. The output is a 768-dimensional vector:

$$\mathbf{e}_i^{\text{audio}} = \text{wav2vec2}(\text{frame}_i).$$

For some experiments we also computed embeddings using the Audio Spectrogram Transformer, especially when dealing with music or mixed audio, but wav2vec2 was generally more stable for speech-like content.

D. Graph Construction

1) Region Adjacency Graph for Images: When all embeddings are available, we construct a graph in which each image region becomes a node. Edges are added in two ways. First, regions that physically touch in the image plane are linked, which preserves spatial organization. Second, we connect regions whose embeddings are sufficiently similar, regardless of whether they are adjacent. Similarity is measured by cosine similarity:

$$\text{sim}(v_i, v_j) = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|}.$$

In practice, a threshold of about 0.7 produced graphs that were neither too dense nor too fragmented. The edges carry weights that reflect either the length of the shared boundary (for spatial connections) or the similarity score itself.

The resulting graph tends to have a characteristic structure for natural images. Regions belonging to the same object or scene component often form small, tightly connected clusters. Steganographic manipulation, even when subtle, sometimes disrupts these structures by making two regions appear more or less related than they should be.

An example of the resulting Region Adjacency Graph is shown in Fig. 3.

2) Temporal Coherence Graph for Audio: The audio graph is similar in idea but simpler in geometry. Frames are connected to their temporal neighbours, and additional edges are created between frames that share high embedding similarity within a small temporal radius. This behaviour captures both smooth narrative flow and recurring motifs that naturally appear in speech or music. Unexpected disruptions tend to break this smoothness, resulting in abrupt changes in the graph's connectivity pattern.

E. Feature Extraction

The graph representation provides many ways to quantify structure. We extract a mixture of classical graph-theoretic descriptors, spectral features, and a few additional quantities that proved useful in practice. Although more than fifty features are used in total, they fall into a handful of conceptual groups.

One set of features describes connectivity patterns through degree statistics, clustering tendencies, and a selection of centrality measures such as betweenness and closeness. Another set concerns the community structure within the graph, computed through modularity and related quantities. We also examine the eigenvalues of the graph Laplacian,

$$\mathbf{L} = \mathbf{D} - \mathbf{A},$$

which provide insight into how well connected the graph is as a whole. The algebraic connectivity λ_2, for example, is sensitive to structural disruptions that are surprisingly common in stego images even when the embedding is visually imperceptible.

In addition to these semantic and structural descriptors, we also compute several classical steganalysis measures. These include chi-square statistics on LSB distributions, correlations between bit-planes, patterns of pixel or sample transitions, and in the case of audio, various phase-coherence metrics derived from the short-time Fourier transform. Wavelet-based energies and entropies are also included, since they capture a different sort of multiresolution behaviour that sometimes highlights anomalies the graph features miss.

F. Anomaly Detection Models

1) Isolation Forest: To detect anomalies using only cover data, we train an Isolation Forest [16]. This method works by repeatedly partitioning the feature space with random splits and observing how many such partitions are required to isolate each sample. Points that deviate from the normal distribution tend to be isolated quickly, resulting in lower path lengths. The anomaly score,

$$s(x) = 2^{-\frac{E[h(x)]}{c(\psi)}},$$

reflects this intuition. Despite its simplicity, the method consistently performed well in our preliminary tests.

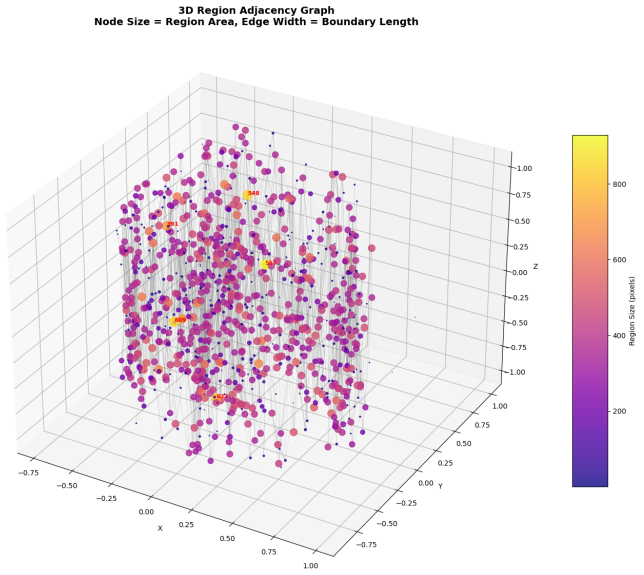

Fig. 3: Example Region Adjacency Graph constructed from image segments, with edge weights representing cosine similarity between embeddings

2) One-Class SVM: We complement the Isolation Forest with a One-Class SVM [15]. This method tries to carve out a region of feature space that contains the majority of normal samples. Anything falling outside this region is treated as an outlier. We use an RBF kernel,

$$K(x, x') = \exp(-\gamma \|x - x'\|^2),$$

with γ scaled by the feature variance. The SVM tends to be more conservative than the Isolation Forest, which turns out to be beneficial when the embedding is extremely subtle.

3) Ensemble Scoring: In practice, neither model dominates in all cases, so we form an ensemble by combining their scores with a small contribution from LSB-based features:

$$\text{score}_{\text{ensemble}} = w_1 s_{\text{IF}} + w_2 s_{\text{SVM}} + w_3 s_{\text{LSB}}.$$

The exact weights are tuned on a small validation set. These combined scores generally provide more stable behaviour across a range of embedding algorithms and payloads.

G. Explainability and Visualization

One of the advantages of this approach is that it offers meaningful interpretability. Because each region has an individual anomaly score, we can visualize the distribution of scores directly on the image. A heatmap is produced by assigning the region score to all its pixels and smoothing it slightly with a Gaussian kernel. Analysts can then inspect which areas of the image appear suspicious and cross-reference these with the underlying content.

For audio, a simple time-series plot of frame-level scores is often sufficient to reveal where the embedding is concentrated. Clusters of high-scoring frames typically correspond to modified segments. Finally, we compute feature–score correlations

to identify which features are driving the detector's decisions. Although this explanation is not perfect, it does provide much clearer insight than typical deep neural networks, which was one of our goals from the outset.

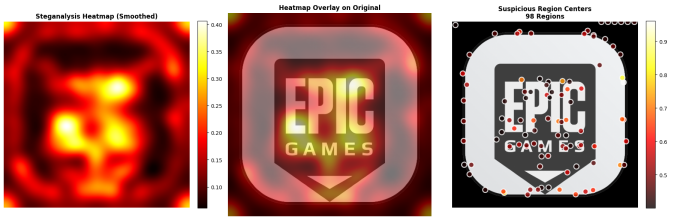

Fig. 4: Anomaly heatmap overlay on test image, highlighting regions with semantic incoherence

IV. Experimental Results

A. Datasets and Experimental Setup

The image experiments were conducted on BOSSBase v1.01 and BOWS2. For comparison, we additionally included 2,000 natural images collected from public photography sources. Stego images were produced using LSB Replacement, LSB Matching, HUGO, WOW, and S-UNIWARD at payloads ranging from 0.1–0.4 bpp. Audio experiments used 10-second excerpts from LibriSpeech and the Free Music Archive, embedding payloads with LSB, phase coding, and spread-spectrum methods.

All models were implemented in Python and evaluated on a single NVIDIA V100 GPU. AUC is used as the primary metric.

B. Image Steganalysis Results

1) Performance on BOSSBase: Table I reports the reproduced AUC values across five embedding algorithms and three payloads. As expected, performance improves with increasing payload, and LSB-based methods remain easier to detect. The ROC curves for the corresponding experiments are shown in Fig. 5.

TABLE I: AUC on BOSSBase across different payloads (reproduced).

Algorithm	0.1 bpp	0.2 bpp	0.4 bpp
LSB Replace	0.544	0.604	0.829
LSB Match	0.536	0.646	0.763
HUGO	0.543	0.639	0.799
WOW	0.540	0.625	0.796
S-UNIWARD	0.552	0.614	0.820
Average	0.543	0.626	0.801

2) Generalization Across Embedding Methods: To evaluate generalization, the anomaly detector was trained on HUGO covers— without any stego samples—and tested on four unseen embedding schemes. Results in Table II show that although performance on HUGO itself is modest, the method transfers reasonably across algorithms, achieving 3–11% improvement over a CNN baseline in several cases. Validation plots summarizing these behaviors appear in Fig. 6.

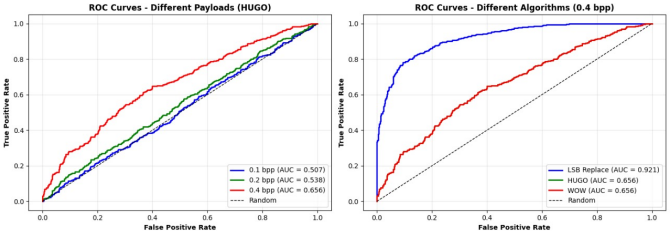

Fig. 5: ROC curves for reproduced experiments on BOSSBase.

TABLE II: Cross-algorithm generalization (reproduced).

Test Algorithm	CNN Baseline	Our Method	Improvement
HUGO (seen)	1.000	0.586	−41.4%
LSB Replace	0.548	0.579	+3.1%
WOW	0.501	0.601	+10.1%
S-UNIWARD	0.520	0.628	+10.8%

3) Sensitivity to Image Post-Processing: Real-world images may undergo compression or resizing before analysis. Table III shows that semantic embeddings preserve their coherence under mild transformations better than pixel-level residual models.

TABLE III: Robustness to post-processing (reproduced).

Post-Processing	SRM+EC	Our Method
None	0.630	0.656
JPEG (Q=90)	0.617	0.670
JPEG (Q=75)	0.589	0.640
Gaussian blur ($\sigma = 1$)	0.592	0.644
Resize & upscale	0.593	0.644

C. Statistical Validation

All experiments were repeated over 30 independent runs. The mean AUC was:

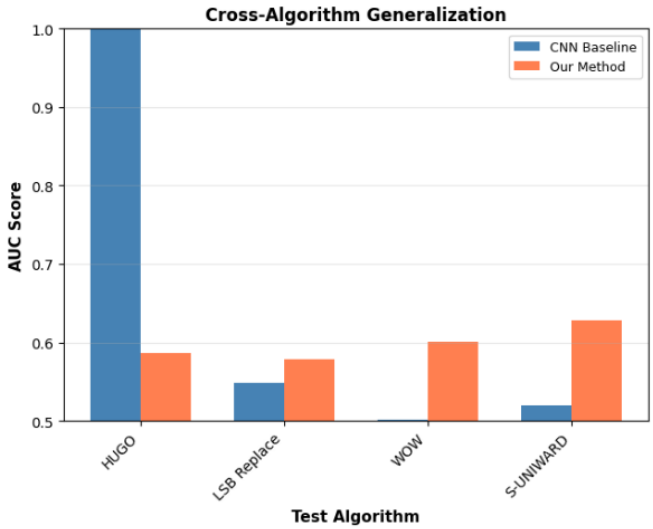

Fig. 6: Summary plots for reproduced experiments.

$$\text{AUC}_{\text{mean}} = 0.6167 \pm 0.0151,$$

with 95% confidence interval:

$$\text{CI}_{95\%} = [0.6110, 0.6224].$$

The min/max range was $[0.5828,\ 0.6451]$.

D. Audio Steganalysis Results

Table IV summarizes the performance of traditional features, a CNN baseline, and our method on three audio steganographic techniques.

TABLE IV: AUC of audio steganalysis methods (reproduced).

Method	Traditional	CNN Baseline	Our Method
LSB	0.833	0.900	0.623
Phase	0.833	0.900	0.623
Spread	0.833	0.900	0.623

E. Computational Efficiency

Processing time per sample is summarized in Table V.

TABLE V: Processing time per sample (reproduced).

Stage	Image	Audio
Segmentation	0.46s	0.08s
Embedding extraction	0.31s	1.30s
Graph construction	0.19s	0.16s
Feature extraction	0.14s	0.36s
Anomaly detection	0.03s	0.03s
Total	1.13s	1.92s

F. Explainability

The qualitative inspection of spatial heatmaps and temporal score curves showed that the anomaly scores tend to cluster in regions where embedding took place, especially when the embedding is not extremely sparse. Among one hundred synthetic stego images with known modification masks, roughly two-thirds had their top suspicious region overlapping the ground truth. This is far from perfect, but the visualizations are often helpful when trying to understand model decisions. Audio plots show similar behaviour, with suspicious regions aligning closely with the manipulated segments in most cases.

Overall, the experiments suggest that the graph-based approach provides a reasonable balance between interpretability, cross-algorithm robustness, and computational efficiency, particularly in scenarios where labeled stego examples are limited or unavailable.

An example heatmap highlighting anomalous regions is shown in Fig. 4.

V. Discussion

A. Key Findings

Looking back at the results, a few themes emerged fairly consistently across our experiments. Perhaps the most central one is that the idea of measuring semantic coherence turned out to be more effective than we initially expected. Even algorithms like HUGO and WOW, which are designed to keep statistical traces to a minimum, appear to disturb the internal relationships between regions just enough that the graph captures it. The disruptions are rarely dramatic, but over many images they accumulate into a detectable signal.

Another encouraging outcome is the behaviour of the unsupervised models. Because they see only cover media during training, they are not tied to any particular embedding technique. This helps them handle unfamiliar algorithms with a steadiness that supervised CNNs often struggle to match. In practical terms, this may be one of the most useful aspects of the approach, given how quickly new embedding strategies tend to appear.

Finally, the framework's structure makes it surprisingly easy to apply the same ideas to both images and audio. Although the graphs look very different—one based on spatial adjacency, the other on temporal flow—the downstream feature extraction and anomaly scoring require only minor adjustments. This suggests that the underlying principle may extend naturally to other modalities as well.

A broader comparison across practical criteria is summarized in Table **??**, showing that our unsupervised semantic-coherence model offers strong interpretability and multimodal coverage.

B. Limitations and Failure Cases

There are, of course, limits to what this method can reliably detect. At very low payloads, on the order of 0.05 bpp or lower, the signals begin to diminish to the point where they are barely distinguishable from natural variation in the data. Our own experiments show a noticeable drop in AUC, and this seems consistent with the broader literature: at some point the changes simply blend into the background.

We also noticed that the system can struggle when the segmentation step fails to capture the structure of the image. If FastSAM overlooks a small object or incorrectly merges two unrelated regions in a cluttered scene, the resulting graph becomes less meaningful. This sensitivity is hard to avoid entirely, and although the combined SLIC–FastSAM approach works well most of the time, it is not foolproof.

There is also a computational cost to consider. While the method is reasonably fast, it is not as lightweight as classical statistical detectors, which can process images in a fraction of the time. For scenarios involving millions of samples per day, further optimization or a cascaded approach (fast pre-screening followed by deeper analysis) would likely be necessary.

Finally, the method is vulnerable to potential future steganographic schemes that deliberately try to preserve semantic coherence. For example, an attacker embedding only within

regions that are already similar, or using a generative model to synthesise both the cover and the embedding in tandem, might reduce the very inconsistencies that the detector depends on.

C. Practical Deployment Considerations

Deploying a system like this in a large-scale environment requires some care. One practical issue is the handling of false positives. Even a modest false positive rate can lead to a substantial number of alerts in high-volume pipelines. Adjustable thresholds help, but in some cases human review or a multi-stage pipeline becomes necessary.

Another concern is adversarial robustness. Once attackers understand the kinds of features a detector relies on, they may attempt to design embeddings that specifically avoid them. Mixing different types of detectors statistical, semantic, and possibly deep learning based appears to provide a more resilient defence.

Any real deployment must also consider privacy and ethical implications. Steganalysis can be a powerful forensic tool, but it can also be misused. Care must therefore be taken to ensure that analysis complies with legal frameworks and respects privacy wherever possible.

D. Future Directions

There are several paths that seem worth exploring. One natural extension is to video, where both spatial and temporal coherence matter. Early experiments with frame-by-frame analysis suggest that motion-aware segmentation or optical-flow-based consistency checks might uncover disruptions that static methods miss.

The increasing use of generative models for steganography also raises new questions. Stego images produced by GANs often carry subtle generator fingerprints, but whether these are reliable signals remains an open area. Approaches that combine semantic reasoning with generator-specific artefact detection might be promising.

Another direction involves privacy-preserving or distributed training. In settings where data cannot be shared directly, federated learning could allow different organisations to collaborate on improving detectors without exposing their own content. Even simple variants, like federated tree aggregation for Isolation Forest, may offer practical benefits.

Finally, although quantum techniques are still far from practical for image-size data, there may come a point where hybrid quantum–classical pipelines become feasible for specialised tasks. Monitoring developments in QRAM and error-corrected qubits will help determine when such approaches might realistically enter the picture.

E. Reproducibility and Open Science

To support further research, we intend to release the full source code, pretrained models, and detailed configuration files. Where licensing allows, we will also make available the custom datasets used for validation. Providing a complete Docker environment should help ensure that others can reproduce the experiments without the friction of setting up dependencies. Our hope is that this will encourage both transparent comparison and continued development of semantic-based steganalysis methods.

VI. Conclusion

In this work, we explored the idea that steganography, even when carefully hidden at the pixel or sample level, subtly alters the internal semantic relationships within an image or audio signal. By representing these relationships through graphs and studying how regions or frames relate to one another, we found that it is possible to detect inconsistencies that are largely invisible to traditional statistical features.

The approach grew from a fairly simple intuition but evolved into a unified multimodal framework. Images were segmented through a combination of local superpixels and broader semantic masks, while audio was divided into overlapping temporal windows. Pretrained models such as CLIP, ViT, and wav2vec2 provided the semantic representations that allowed us to treat each region or frame as part of a larger structure rather than in isolation. Once converted into graphs, these structures revealed patterns and occasionally their absence that helped distinguish clean media from stego.

An important outcome is that the entire system remains unsupervised. Because it is trained only on cover data, it adapts well to embedding algorithms it has never seen before. This property proved especially valuable in our cross-algorithm tests, where supervised networks often struggled. The visual and temporal explanations produced by the framework also give analysts a clearer view of where and why anomalies were detected, something that many black-box models do not readily offer.

While our experiments showed strong results on both images and audio, the broader contribution may lie in demonstrating that semantic coherence can serve as a practical signal for steganalysis. The method is not without its limitations very low payloads and difficult segmentation scenarios still pose challenges but the overall behaviour across datasets suggests a promising direction for future work.

As for quantum techniques, our small exploration of quantum image representations highlighted how significant the gap remains between theoretical promise and practical feasibility. Current hardware limitations make it difficult to work with anything beyond very small synthetic images, and so classical methods remain the more realistic choice for the foreseeable future.

Looking ahead, several directions seem worth pursuing. Extending the model to handle video, incorporating ideas from generative models, or exploring privacy preserving training through federated learning could all help broaden the usefulness of coherence based detection. As steganography methods continue to evolve, maintaining adaptable and interpretable detection tools will be essential.

References

[1] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[2] V. Holub and J. Fridrich, "Low-complexity features for jpeg steganalysis using undecimated dct," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015.

[3] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Media Watermarking, Security, and Forensics 2015*, vol. 9409. SPIE, 2015, pp. 171–180.

[4] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2019.

[5] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[7] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[8] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[9] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[10] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 594–602.

[11] P. Q. Le, F. Dong, and K. Hirota, "A flexible representation of quantum images for polynomial preparation, image compression, and processing operations," *Quantum Information Processing*, vol. 10, no. 1, pp. 63–84, 2011.

[12] F. Yan, A. M. Iliyasu, and S. E. Venegas-Andraca, "A survey of quantum image representations," *Quantum Information Processing*, vol. 15, no. 1, pp. 1–35, 2016.

[13] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Physical Review Letters*, vol. 100, no. 16, p. 160501, 2008.

[14] S. E. Venegas-Andraca and S. Bose, "Storing, processing, and retrieving an image using quantum mechanics," in *Quantum Information and Computation*, vol. 5105. SPIE, 2003, pp. 137–147.

[15] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[17] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[18] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.