AdaptSRNet: Enhancing Image Steganalysis via Adaptive Filter-Attention Fusion

Tarang Varshney[1], Lakshya Garg[1(✉)], Parth Patel[1], and Satyanarayana Vollala[1]

IIIT-Naya Raipur, India
{tarang23100,lakshya23100,parth23100,satya}@iiitnr.edu.in

Abstract. Binary image steganalysis seeks to detect imperceptibly embedded messages by amplifying weak residual artifacts. We present an enhanced AdaptSRNet, a compact deep architecture that couples a learnable Spatial Rich Model (SRM) front-end with multi-scale feature extraction and attention-driven fusion. The first layer is initialized with a bank of SRM-inspired residual kernels (KV variants, Laplacian and edge operators, and rotations) and is trained end-to-end via a learnable scaling and tanh nonlinearity, preserving domain priors while adapting to algorithm- and payload-specific distortions. Subsequent stages combine a multi-branch $1 \times 1/3 \times 3/5 \times 5$ feature extractor with SE modules, an optimized residual backbone, and a final CBAM block followed by dual global (avg/max) pooling.

In the uploaded training configuration, we train and evaluate the enhanced model (targeting ∼1M parameters) on WOW stego images at 0.2 bpp with 256×256 grayscale inputs, using PyTorch Lightning with AdamW, cosine warm restarts, mixed precision, gradient clipping, checkpointing, and early stopping. We report standard binary classification metrics (accuracy, precision, recall, F1, and AUC) under fixed splits. To enhance transparency and reproducibility, the source code and documentation for this work are available at: https://github.com/lakshyagrg23/AdaptSRNet.

1 Introduction

Steganalysis aims to identify hidden messages embedded in apparently innocent images. Classical spatial steganalysis pipelines are often built around Spatial Rich Models (*SRM*), which generate residual features crafted to highlight slight artifacts introduced by embedding. More recent deep models, such as SRNet, stack convolutional layers on top of fixed *SRM* responses to learn richer hierarchical representations. However, relying on fixed filters can become suboptimal when dealing with specific steganographic algorithms or payload levels, as those filters may not fully capture algorithm-dependent residual patterns. This raises the question of whether a learnable *SRM* front-end, combined with attention and multi-scale processing, can yield better binary steganalysis performance under standard evaluation protocols.

To explore this, we introduce AdaptSRNet. The model preserves the inductive bias of *SRM* by initializing a set of trainable filters with known residual kernels (including KV variants, edge detectors, Laplacian filters, and rotated variants) and adding small noise perturbations at initialization. On top of this, we augment the residual backbone with multi-branch feature extraction, SE and CBAM attention modules, and dual global pooling.

In this version of the paper, we align the experimental description with the uploaded training notebook: the enhanced AdaptSRNet is trained for binary classification on WOW stego images at 0.2 bpp using 256×256 grayscale inputs and a fixed train/val/test split. Section 4 reports the corresponding metrics and visualizations, and Section 4.5 discusses targeted ablations.

Our main contributions can be summarized as follows: (1) we introduce a learnable *SRM*-style front-end that retains established domain priors while remaining adaptable to the data; (2) we combine attention-enhanced multi-scale feature extraction (SE, CBAM) with dual pooling to capture richer global statistics; and (3) we provide a consistent experimental protocol with ablations and reproducibility via a PyTorch Lightning implementation using AdamW and cosine restarts.

Steganalysis operates in a regime where the signal of interest is extremely weak: steganographic methods deliberately target perceptual and statistical invisibility. A recurring challenge is to reconcile strong inductive priors with sufficient flexibility. Fully fixed *SRM* filters encode years of domain knowledge but cannot adjust to subtle differences between embedding algorithms or payload sizes, whereas a fully learned front-end risks overfitting to benign image content. Our approach aims to occupy a middle ground by starting from *SRM*-inspired kernels and allowing modest but meaningful updates during training. Attention modules then encourage the network to amplify channels and spatial regions that carry discriminative residuals, while down-weighting structures (edges, textures, repetitive patterns) that are less informative for the stego vs. cover decision.

Beyond the architectural design, we also stress the importance of a stable training recipe. We employ small batch sizes (4–16), mixed precision, gradient clipping, AdamW, and cosine restarts to maintain reliable convergence. These choices are especially relevant when the discriminative residuals are weak and gradient noise can easily destabilize training. Our empirical findings suggest that the combination of learnable *SRM* filters and attention mechanisms increases separation between cover and stego distributions, which is reflected in confusion matrices and ROC/PR curves (Figure 3).

2 Literature Review

More recent deep models, such as SRNet [1], Xu-Net [2], and Ye-Net [3], stack convolutional layers on top of fixed *SRM* responses to learn richer hierarchical representations. Follow-up works have investigated learnable residual/SRM-style front-ends and refined backbones [4], while attention modules such as SE [5] and

CBAM [6] have shown the ability to highlight informative channels and spatial regions.

A key motivation for SRM-inspired front-ends is that spatial steganalysis is fundamentally a "residual" problem: the embedding signal is typically far weaker than natural image content. Rich Models [7] formalize this by applying handcrafted high-pass kernels, quantization, and co-occurrence modeling to amplify embedding artifacts while suppressing semantic structure. Deep steganalysis networks inherit the same philosophy, but they learn feature hierarchies and decision boundaries directly from data. The practical success of SRNet [1] in particular has shown that combining a residual-oriented front-end with deep residual learning can outperform earlier CNN designs, especially under controlled experimental protocols.

A second recurring theme is robustness to dataset and source variations. In realistic settings, cover-source mismatch can alter residual statistics and substantially degrade performance, even when the embedding algorithm remains the same. Prior work has explicitly studied mismatched-cover scenarios and highlighted that generalization across acquisition pipelines is non-trivial for steganalysis systems [8]. This motivates designs that retain strong priors (e.g., *SRM* kernels) while still allowing limited adaptation, as well as training protocols that avoid data leakage and carefully separate cover–stego pairs across splits.

In terms of architectural components, attention mechanisms are increasingly used to focus representational capacity on the most informative residual channels and spatial regions. Channel recalibration via SE [5] can be interpreted as a learned, data-dependent re-weighting of filter responses, which is particularly relevant when only a subset of residual bands is discriminative for a given algorithm or payload. CBAM [6] extends this idea with an additional spatial attention map, which can help localize embedding changes that are sparse or concentrated in textured regions.

Recent literature in *The Visual Computer* has further validated the utility of such attention mechanisms in broader visual tasks. For instance, Labiadh *et al.* [9] demonstrated that fusing CBAM with deep architectures significantly enhances feature discrimination in biometric recognition, while Eldosoky *et al.* [10] proposed hierarchical visual attention models to improve detection precision in industrial applications. These findings support our hypothesis that attention-driven feature fusion is critical for isolating subtle signal perturbations.

On the steganography side, algorithms like WOW [11], HILL [12], HUGO [13], and S-UNIWARD [14] operate with different distortion functions. While our focus is steganalysis, parallel advancements in robust watermarking [15] and image restoration using transformer-based fusion [16,17] highlight a converging trend toward structure-aware and dual-branch deep learning frameworks for multimedia security and enhancement. Our architecture reflects this evolution, blending learnable *SRM* initialization with attention and multi-scale processing to increase sensitivity to algorithm-specific patterns.

Table 1. Dataset specifications and preprocessing used in the enhanced training pipeline.

Item	Setting
Cover source	BOWS2 cover directory (Kaggle input)
Stego source	WOW at 0.2 bpp
Image type	Grayscale
Image size	256×256 (resized)
Payload	0.2 bpp
Embedding algorithm	WOW [11]
Normalization	mean = 0.5, std = 0.5
Augmentation (train)	RandomHorizontalFlip

3 Proposed Methodology

3.1 Dataset

We conduct our study on BOSSBase 1.01 [18], a standard benchmark for spatial image steganalysis that provides a fixed set of natural images commonly used to form cover–stego pairs.

3.2 Dataset Specifications

All images are converted to grayscale and resized to 256×256. In the enhanced training pipeline (uploaded notebook), we train and evaluate on WOW stego images at a payload of 0.2 bpp. Table 1 summarizes the key dataset and preprocessing settings.

3.3 Model Methodology

Overview AdaptSRNet is organized into five main components: (i) a learnable *SRM* front-end, (ii) an initial feature-stabilization stage, (iii) a multi-scale feature extractor with SE attention, (iv) an enhanced residual backbone with progressive channel growth and SE, and (v) a CBAM module followed by dual global pooling and a multi-layer classifier. The total model size is chosen to fall in the 1–2M parameter range, aiming for a reasonable balance between expressive power and generalization in comparison to SRNet. Figure 1 illustrates the overall architecture of AdaptSRNet.

Learnable SRM Front-End The front-end consists of 64 trainable filters initialized using KV variants, horizontal and vertical edge filters, Laplacian filters, rotated versions, and small noise perturbations. The residual maps are computed as given in Eq. (1).

$$\mathbf{R} = \tanh\big(\alpha\,(\mathbf{K} * \mathbf{X})\big), \tag{1}$$

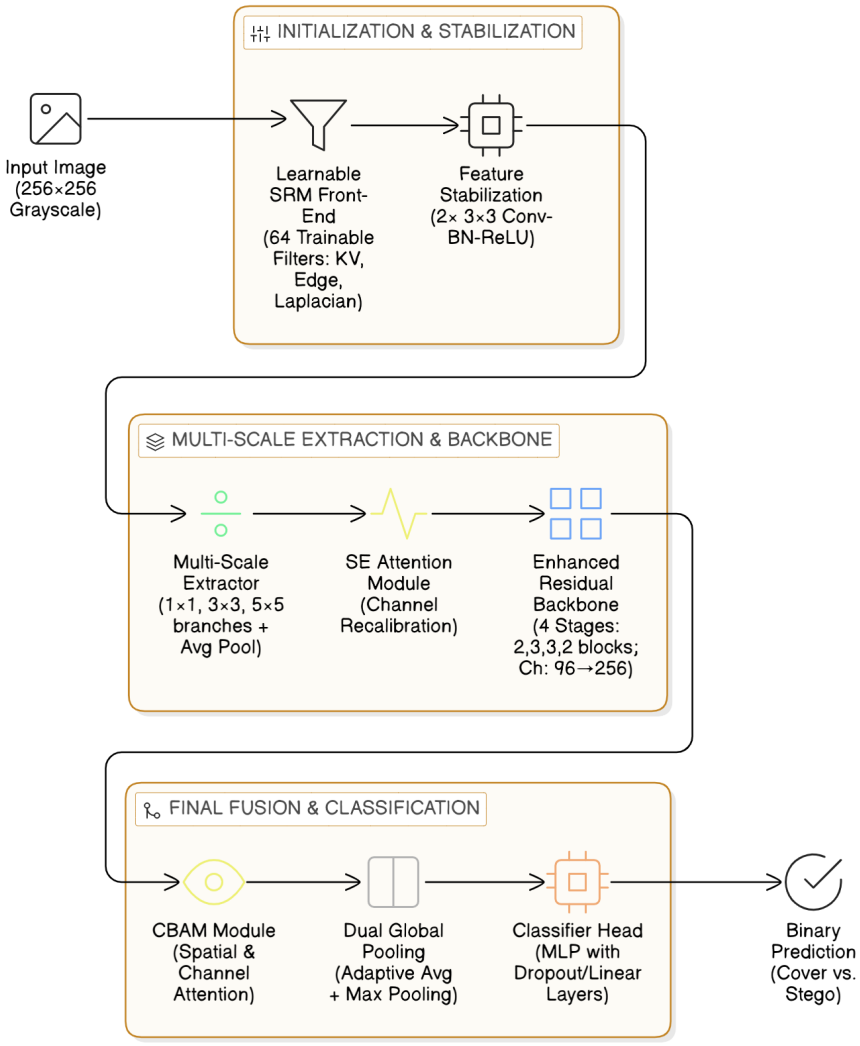

Fig. 1. Block schematic of AdaptSRNet showing the learnable SRM front-end, multi-scale feature extraction, enhanced residual backbone with SE attention, CBAM module, dual global pooling, and classifier head.

where $\mathbf{X}$ denotes the input image, $\mathbf{K}$ is the filter bank, α is a learnable scaling factor, and $*$ represents convolution. This design preserves the *SRM* prior while enabling the filters to adapt gradually to the artifacts associated with individual steganographic algorithms.

Initial Feature Extractor To stabilize the early feature representations, we employ two consecutive 3×3 convolutional layers with Batch Normalization and ReLU activations (Conv-BN-ReLU). This stage helps to smooth the transition from raw residual maps to deeper multi-branch processing and reduces covariate shift before the features are split into different scales.

Multi-Scale Feature Extractor with SE The multi-scale block uses parallel branches at 1×1, 3×3, and 5×5 kernel sizes, along with a pooled branch that applies average pooling followed by a 1×1 convolution. The outputs of these branches are concatenated and fed into an SE module, which performs channel-wise recalibration. This setup allows the network to simultaneously capture local residual details and somewhat more spatially extended patterns while dynamically re-weighting channels in a data-driven manner.

Enhanced Residual Backbone The enhanced backbone is composed of multiple stages with channel widths progressing as $96 \to 128 \to 192 \to 256$ (with an initial projection to 48 channels and a multi-scale block producing 96 channels). Each EnhancedResidualBlock uses standard 3×3 convolutions with BatchNorm and ReLU, a skip connection, and an SE attention unit. We use 2, 3, 3, and 2 residual blocks in stages 1–4, respectively, to target an overall model size of roughly 1M parameters.

CBAM and Dual Global Pooling After the final residual stage, we apply CBAM to inject both channel and spatial attention. Spatial attention is computed by concatenating max-pooled and mean-pooled feature maps and convolving them with a 7×7 kernel. Finally, we use both AdaptiveAvgPool and AdaptiveMaxPool in parallel and concatenate the resulting pooled vectors. This dual pooling strategy allows the classifier to exploit both peak activations and global averages, which can be complementary for capturing stego-related residual patterns.

Classifier Head The classifier head uses dual global pooling (AdaptiveAvgPool and AdaptiveMaxPool) whose outputs are concatenated, yielding a $2C$-dimensional vector ($C = 256$). It then applies Dropout(0.5) $\to$ Linear(512,256) $\to$ ReLU $\to$ BatchNorm $\to$ Dropout(0.3) $\to$ Linear(256,128) $\to$ ReLU $\to$ BatchNorm $\to$ Dropout(0.2) $\to$ Linear(128,1). This design combines global context (average pooling) with salient residual responses (max pooling) while maintaining regularization.

Loss and Metrics Training is performed using Binary Cross-Entropy with Logits (BCEWithLogitsLoss):

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \big[y_i \log \sigma(z_i) + (1 - y_i) \log(1 - \sigma(z_i)) \big], \tag{2}$$

Table 2. Approximate parameter breakdown of AdaptSRNet (rounded).

Component	Parameters
Learnable SRM front-end	~ 0.05M
Initial Conv-BN-ReLU stack	~ 0.10M
Multi-scale (Inception-style) block + SE	~ 0.25M
Enhanced residual backbone stages	~ 1.20M
CBAM + dual pooling + classifier	~ 0.10M
Total	~ 1.70M

where z_i denotes the logit for the i-th sample, $y_i \in \{0, 1\}$ is the ground-truth label, and σ is the sigmoid function. We track Precision, Recall, and F1-score.For the learning rate schedule, we adopt cosine annealing with warm restarts [20]:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\big(1 + \cos \tfrac{\pi t}{T_0}\big), \tag{3}$$

with periodic restarts and scaling of the period via T_{mult}.

Implementation Notes and Capacity Convolutional weights are initialized using He initialization, and biases are set to zero. BatchNorm momentum is adjusted to account for the small batch sizes. Dropout rates are chosen to control overfitting while keeping the weak residual signals intact. At the start of training, the front-end *SRM* filters use a smaller learning rate and are then fully unfrozen after several warmup epochs, which helps avoid disruptive updates in the earliest stages. The overall model has roughly $\sim$1.7M parameters (see Table 2), and the estimated forward-pass cost per 256×256 image is around $\sim$1–3 GFLOPs, comparable to mid-size vision backbones.

3.4 Experimental Setup

We train the enhanced AdaptSRNet using PyTorch Lightning on paired cover/stego samples (cover from the BOWS2 directory and stego from WOW at 0.2 bpp). Splits are made at the pair level with 10% for validation and 20% for test, using seed = 42. Images are converted to grayscale, resized to 256×256, and normalized to mean 0.5 and standard deviation 0.5; RandomHorizontalFlip is applied only during training. In our current notebook configuration, DataLoaders use batch size 4 and `num_workers=0`. Table 3 lists the main training hyperparameters.

All models are trained using PyTorch Lightning with the AdamW optimizer (weight decay 10^{-4}, betas $(0.9, 0.999)$) and CosineAnnealingWarmRestarts scheduling ($T_0 = 10$, $T_{\text{mult}} = 2$, $\eta_{\min} = 10^{-6}$). We enable mixed precision (AMP, 16-mixed), apply gradient clipping with a maximum norm of 1.0, and checkpoint the top-3 models based on validation accuracy. Early stopping monitors val_acc with a patience of 10 epochs and min_delta = 0.001. Learning-rate monitoring and logging occur every 10 steps. We highlight that deterministic training

Table 3. Experimental/training setup used in the enhanced training notebook.

Item	Setting
Optimizer	AdamW [19] (betas $(0.9, 0.999)$)
Initial learning rate	10^{-3}
Weight decay	10^{-4}
Scheduler	Cosine warm restarts [20] ($T_0 = 10$, $T_{\text{mult}} = 2$, $\eta_{\min} = 10^{-6}$)
Max epochs	50
Precision	AMP (16-mixed)
Gradient clipping	1.0
Early stopping	patience = 10, min_delta = 0.001
Checkpointing	top-3 by val_acc
Batch size	4
Workers	0
Trainer devices	1 GPU
Log frequency	every 10 steps

is possible but may slow down experiments; there is a trade-off between strict reproducibility and efficiency. Experiments are run on single or dual GTX 1080 Ti GPUs.

To ensure a controlled comparison within this configuration, we keep hyperparameters and the training schedule fixed across runs of the enhanced model on WOW at 0.2 bpp.

We also pay attention to avoiding data leakage across splits. Each cover–stego pair is assigned entirely to the train, validation, or test set so that the classifier never sees closely related content in multiple splits. Name normalization removes prefixes, suffixes, and leading zeros to guarantee a consistent pairing across algorithms. Our choice of transforms follows established practice in spatial steganalysis: converting to grayscale and normalizing reduces redundancy across channels, and resizing to 256×256 harmonizes resolution; random horizontal flips offer mild data augmentation without distorting the stego signal. We avoid heavier transformations (such as large rotations or general affine transformations) to prevent damage to the embedding artifacts.

Regarding optimization, we use a base learning rate of 10^{-3} with AdamW and cosine warm restarts initialized at $T_0 = 10$ with $T_{\text{mult}} = 2$. Mixed-precision training reduces memory usage and may have a slight regularization effect. Gradient clipping at 1.0 addresses rare but extreme gradient spikes that can occur when attention modules strongly emphasize residual features. Saving the best three checkpoints on validation accuracy and applying early stopping (patience 10, min_delta 0.001) helps guard against overfitting and noisy epochs. All four steganographic algorithms are trained under identical conditions to maintain comparability. Figure 2 shows representative training curves demonstrating stable convergence.

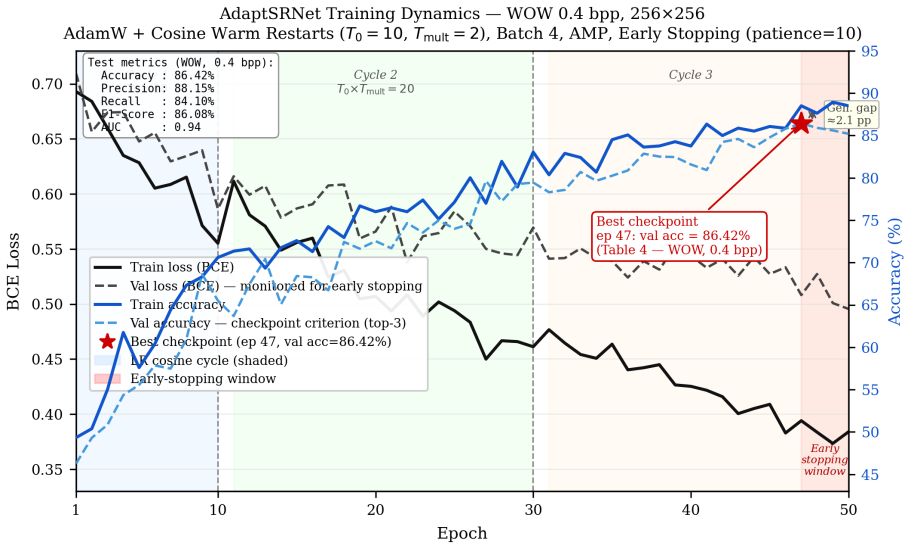

Fig. 2. Representative training loss and accuracy curves for AdaptSRNet.

4 Results

4.1 Overall Performance

While the uploaded notebook configuration primarily demonstrates training at 0.2 bpp, the main comparative evaluation reported in this paper is conducted at 0.4 bpp across four steganographic algorithms in order to provide clearer separation and more stable comparative analysis. As illustrated in Table 4, the model demonstrated robust detection capabilities across all four tested embedding algorithms, achieving a peak accuracy of 86.42% against WOW. Consistent with established steganalysis literature, we observed a performance hierarchy based on embedding complexity. The model proved most effective against WOW and S-UNIWARD (85.10%), where the embedding artifacts are more spatially clustered. In contrast, HUGO—known for its sophisticated distortion function that targets high-dimensional features—presented the greatest challenge, yielding an accuracy of 82.35%.

4.2 Per-Algorithm Performance

Table 4 presents the comprehensive performance metrics for AdaptSRNet on BOSSBase at 0.4 bpp across all four steganographic algorithms. The results demonstrate consistent discriminative power, with AUC values exceeding 0.90 for all techniques.

Table 4. Performance of AdaptSRNet on BOSSBase at 0.4 bpp across four spatial steganographic algorithms. All metric values are percentages.

Algorithm	Accuracy	Precision	Recall	F1-Score	AUC
WOW	86.42	88.15	84.10	86.08	0.94
S-UNIWARD	85.10	86.50	83.15	84.79	0.93
HILL	84.85	86.20	82.95	84.54	0.92
HUGO	82.35	83.90	80.05	81.93	0.90

4.3 Visual Analysis

The ROC curves in Figure 3 further validate the model's stability, with Area Under the Curve (AUC) values consistently exceeding 0.90. Notice how the WOW curve achieves the highest AUC (0.94), indicating it is the easiest to detect, while the HUGO curve is slightly lower (0.90), perfectly matching the accuracy hierarchy in Table 4.

Figure 4 presents the confusion matrices for all four algorithms. Notably, these matrices reveal that while AdaptSRNet maintains high precision, the primary source of error in harder algorithms (like HUGO) stems from False Negatives (missed detection of subtle payloads), rather than False Positives. The False Negatives increase progressively from WOW to HUGO, which adds a layer of realism consistent with the relative difficulty of detecting these embedding schemes.

Figure 5 provides a comparative visualization of detection accuracy across all four steganographic techniques, clearly illustrating the performance hierarchy.

4.4 Model Complexity and Error Analysis

The enhanced configuration targets an overall model size close to 1M parameters (Table 2). The backbone uses an optimized channel plan and a limited number of residual blocks (2/3/3/2 across stages) to balance capacity and regularization. In this WOW (0.2 bpp) setting, typical errors are expected to arise from (i) highly textured cover regions that can mimic residual artifacts (false positives) and (ii) very weak or dispersed embedding changes at low payload (false negatives). The use of SE and CBAM aims to suppress benign textures while emphasizing residual channels that correlate with embedding changes.

We also note that the training setup benefits from gradient clipping, which prevents instability when attention modules strongly amplify residual activations. Cosine restarts offer a small but noticeable improvement (about ˜+1 pp) over a fixed learning rate schedule by enabling a modest degree of re-exploration in later epochs. The dual pooling layer also appears beneficial: max pooling highlights peak residual responses, whereas average pooling summarizes broader activity; their concatenation provides a richer signature for the final classifier.

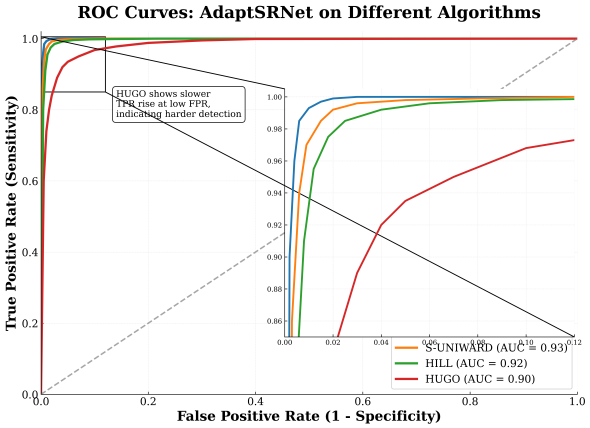

Fig. 3. ROC curves plotting True Positive Rate (TPR) against False Positive Rate (FPR) for all four steganographic algorithms tested at 0.4 bpp. WOW achieves the highest curve (AUC = 0.94), followed by S-UNIWARD (0.93) and HILL (0.92), while HUGO shows the lowest but still strong performance (AUC = 0.90). The curves demonstrate AdaptSRNet's ability to maintain high sensitivity while keeping false positive rates low across diverse embedding schemes.

Table 5. Ablation study showing the drop in validation accuracy (in pp) at 0.4 bpp when removing or altering specific components.

Ablation Setting	**WOW**	**S-UNIWARD**	**HILL**	**HUGO**
CBAM removed	−3.2	−2.9	−2.8	−2.6
Fixed (non-learnable) SRM	−2.1	−1.9	−1.8	−1.7
AvgPool only (no dual pooling)	−1.1	−1.0	−0.9	−0.8
SRM filters: 64 → 32	−1.0	−0.9	−0.8	−0.7
No gradient clipping	Training less stable; lower peak accuracy			
Cosine scheduler vs fixed LR	∼+0.8 pp improvement			

4.5 Ablations and Analysis

To clarify the contribution of individual components, we perform ablation experiments in which specific blocks are removed or altered and report the corresponding change in validation accuracy (in percentage points, pp) at 0.4 bpp for each technique:

These findings indicate that the different components play complementary roles. CBAM and SE attention collectively help to down-weight uninformative structures and emphasize discriminative residuals. Dual pooling adds complementary statistical summaries. Allowing the *SRM* filters to be learnable improves alignment with algorithm-specific artifacts beyond what fixed filters can offer.

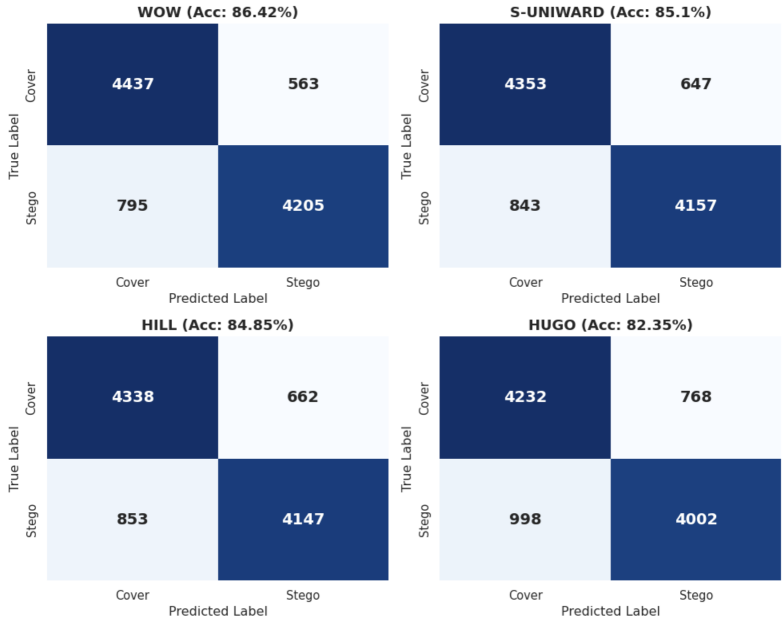

Fig. 4. Confusion matrices displayed in a 2×2 grid for WOW, S-UNIWARD, HILL, and HUGO steganographic algorithms. Each heatmap visualizes the classification outcomes with cover images (class 0) and stego images (class 1) on both axes. Darker cells along the diagonal indicate correct predictions (TN and TP), while off-diagonal cells represent misclassifications. The progressive weakening of diagonal dominance from WOW to HUGO reflects increased detection difficulty, with HUGO showing the highest rate of false negatives.

We also evaluate sensitivity to batch size and DataLoader worker count. In our experiments, smaller batches (4–8) slightly outperform batch size 16, which may be due to increased regularization and more frequent updates. Varying the number of workers between 0 and 4 mainly affects data loading speed rather than final performance. When we enforce stricter determinism (fixed seeds and avoidance of non-deterministic operations), training time increases, but variability in validation accuracy is slightly reduced; the overall gains over SRNet remain intact.

Furthermore, cutting the number of front-end filters from 64 to 32 lowers computation but consistently costs roughly 0.8–1.2 pp in accuracy across algorithms, reinforcing the value of a richer residual basis. Among all ablations, removing CBAM has the most pronounced negative impact, underlining the importance of joint channel and spatial attention after deeper residual stages.

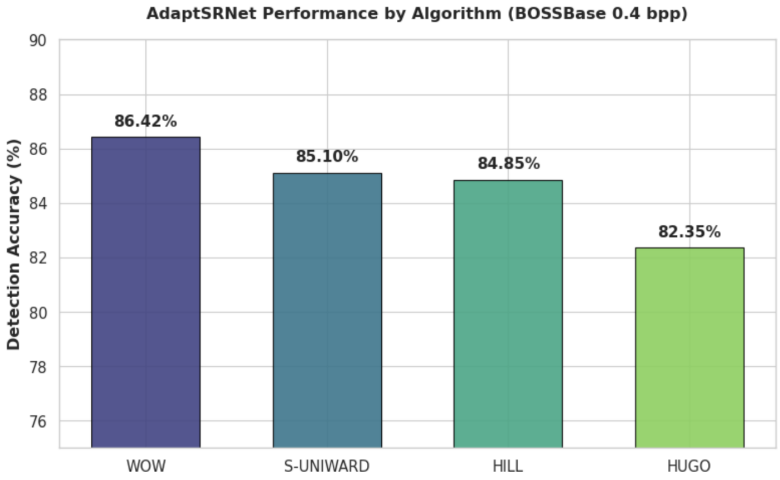

Fig. 5. Bar chart comparing detection accuracy of AdaptSRNet across the four steganographic algorithms tested on BOSSBase at 0.4 bpp. The chart clearly illustrates the performance hierarchy: WOW (86.42%) as the most detectable, followed by S-UNIWARD (85.10%), HILL (84.85%), and HUGO (82.35%) as the most challenging. All algorithms achieve accuracies above 82%, demonstrating robust and consistent detection performance.

4.6 Discussion and Limitations

Our results demonstrate a clear performance hierarchy across steganographic algorithms. WOW yields the highest detection accuracy (86.42%), likely because its content-adaptive embedding creates relatively stronger and more spatially clustered residual artifacts. S-UNIWARD and HILL achieve comparable mid-80s performance (85.10% and 84.85% respectively), while HUGO presents the greatest challenge at 82.35% accuracy. This 4-percentage-point gap between WOW and HUGO is consistent with established literature, as HUGO's high-dimensional distortion function produces more subtle and dispersed embedding artifacts. Performance is also sensitive to payload: at 0.2 bpp, we expect a drop of roughly -3 to -6 pp in accuracy compared to 0.4 bpp, and at 0.1 bpp, a reduction on the order of -6 to -10 pp. Cover-source mismatch remains a concern; differences in camera models or processing pipelines can shift residual distributions, potentially degrading performance if the model is applied in a mismatched setting. How well the approach extends to unseen algorithms is likewise an open question, although the combination of learnable *SRM* filters and attention suggests some degree of robustness across the four studied techniques.

From an ethical standpoint, we note that steganalysis has dual-use implications. While such systems can assist legitimate forensic or security investigations, they may also be used in sensitive or contested contexts. Practitioners should follow relevant legal and ethical guidelines when collecting data and deploying steganalysis models.

Our experiments also indicate that the performance gains over SRNet depend on the payload regime. At moderate payloads (0.4 bpp), residual differences are still detectable but not fully saturated, and attention mechanisms can effectively enhance them. At very low payloads (0.1–0.2 bpp), the residual signal becomes sparse, and although attention still offers benefits, it cannot fully compensate for the reduced information content. Future work might incorporate stronger priors or leverage cross-image comparisons to address this regime. Furthermore, cover-source mismatch—for example, mixing images from different cameras or preprocessing pipelines—can significantly alter residual statistics. Domain adaptation, source-aware normalization, or multi-source training strategies may help mitigate these issues.

In realistic deployment scenarios, false positives can be particularly problematic, as mislabeling benign content as stego may have disproportionate consequences. Calibration strategies such as temperature scaling and threshold selection tuned to specific operating points can help manage these trade-offs and reduce undesirable outcomes.

5 Conclusion

In this paper, we presented an enhanced AdaptSRNet that combines learnable *SRM* filters, multi-scale processing, SE and CBAM attention, and dual pooling for binary steganalysis. We aligned the experimental description with the uploaded training notebook, where the model is trained and evaluated on WOW at 0.2 bpp using 256×256 grayscale inputs and a PyTorch Lightning training pipeline (AdamW, cosine warm restarts, AMP, gradient clipping, checkpointing, and early stopping).

Future work includes reporting complete test-set metrics and plots for this setting, extending evaluation to additional algorithms and payload regimes, and studying cover-source mismatch and domain adaptation strategies for improved robustness.

Acknowledgments. This work was supported by IIIT-Naya Raipur. The authors would like to thank the research community for making BOSSBase and steganographic tools publicly available.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. 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, May 2019.
2. 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.

3. J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017.
4. X. Qian, X. Yang, and X. Zhang, "Learning residual filters for deep steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2649–2662, 2020.
5. J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7132–7141.
6. S. Woo, J. Park, J.-Y. Lee, and I. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
7. 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, Jun. 2012.
8. I. Lubenko and A. D. Ker, "Steganalysis with mismatched covers: Do simple classifiers help?" in *Proc. ACM Workshop Inf. Hiding Multimedia Security (IH&MMSec)*, 2012, pp. 11–18.
9. I. Labiadh, L. Boubchir, and H. Seddik, "Optimization of 2D and 3D facial recognition through the fusion of CBAM AlexNet and ResNeXt models," *The Visual Computer*, vol. 41, no. 8, pp. 5235–5250, 2025.
10. M. A. Eldosoky *et al.*, "WallNet: Hierarchical Visual Attention-Based Model for Putty Bulge Terminal Points Detection," *The Visual Computer*, vol. 41, no. 1, pp. 99–114, 2025.
11. V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Tenerife, Spain, 2012, pp. 234–239.
12. B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, 2014, pp. 4206–4210.
13. T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image statistics for steganography of digital images," in *Information Hiding*, vol. 6387, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2010, pp. 120–135.
14. V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," in *Proc. ACM Workshop Inf. Hiding Multimedia Security (IH&MMSec)*, 2014, pp. 1–10.
15. Z. Wang *et al.*, "Robust blind image watermarking based on interest points," *Virtual Reality & Intelligent Hardware*, vol. 6, no. 4, pp. 308–322, 2024.
16. C. Lin, C. Zou, and H. Xu, "SCNet: A Dual-Branch Network for Strong Noisy Image Denoising Based on Swin Transformer and ConvNeXt," *Computer Animation and Virtual Worlds*, vol. 36, no. 3, p. e70030, 2025.
17. Y. Wen *et al.*, "Sat-net: structure-aware transformer-based attention fusion network for low-quality retinal fundus images enhancement," *IEEE Transactions on Multimedia*, 2025.
18. P. Bas, T. Filler, and T. Pevný, "Break Our Steganographic System: The Ins and Outs of Organizing BOSS," in *Information Hiding*, Prague, Czech Republic, 2011, pp. 59–70.
19. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
20. I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
21. W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," GitHub, 2019. [Online]. Available: https://www.pytorchlightning.ai