Enhancing Image Steganography via Stego Generation and Selection

IEEE ICASSP 2021

Tingting Song, Minglin Liu, Weiqi Luo*, Peijia Zheng

Sun Yat-sen University

Outline

1/ Traditional & Adversarial Embedding

2/ Proposed Method

3/ Experiments

4/ Conclusion

Outline

1/ Traditional & Adversarial Embedding

2/ Proposed Method

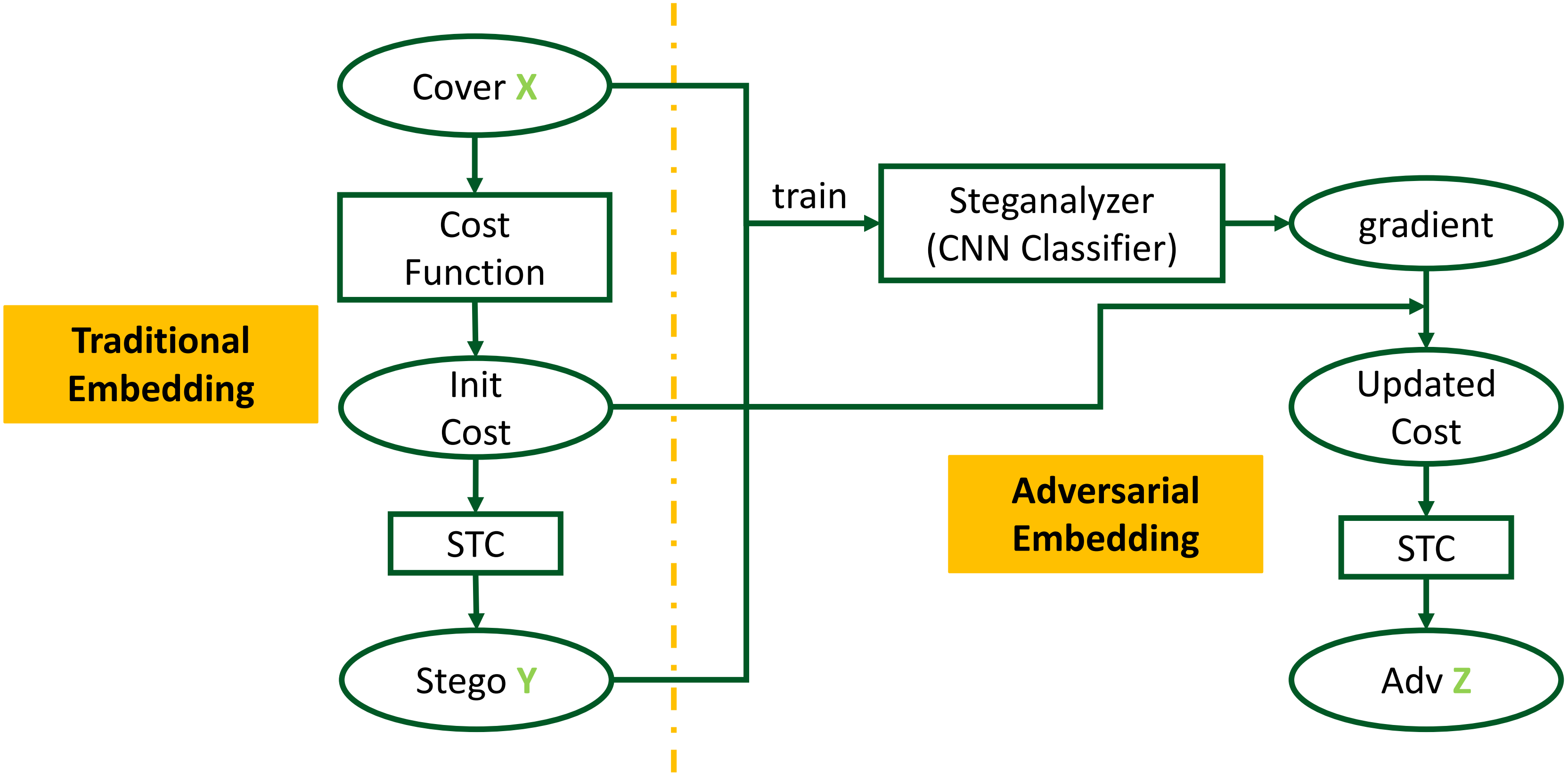
3/ Experiments

4/ Conclusion

Traditional & Adversarial Embedding

- Traditional Embedding :
  - Embedding messages according to a handcrafted cost function

- Adversarial Embedding:
  - Automatic learning by adversarial attack
  - Modifying costs according to the gradient of networks
  - Aim to deceive CNN-based steganalyzers


Cover X
Cost Function
Traditional Embedding
Init Cost
STC
Stego Y
train
Steganalyzer (CNN Classifier)
gradient
Updated Cost
Adversarial Embedding
STC
Adv Z

Outline

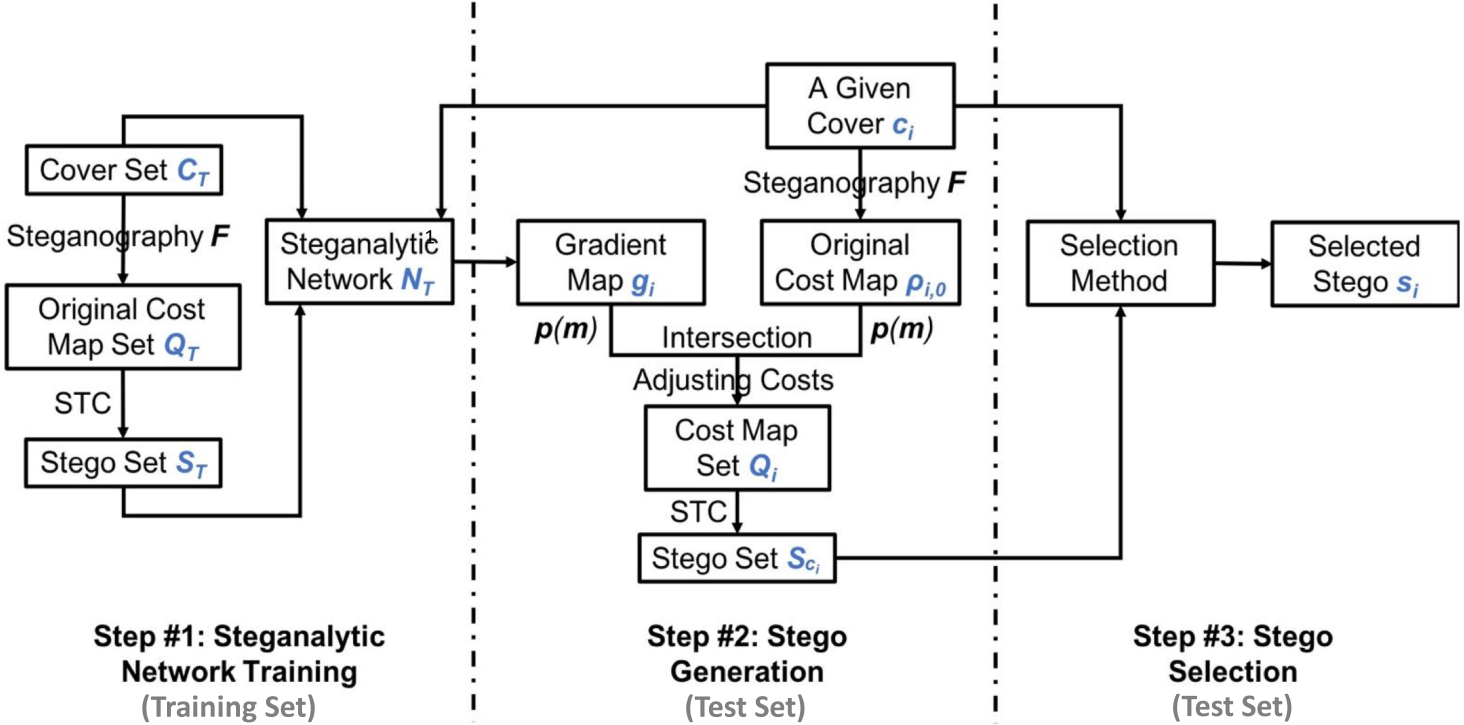
1/ Traditional & Adversarial Embedding

2/ Proposed Method

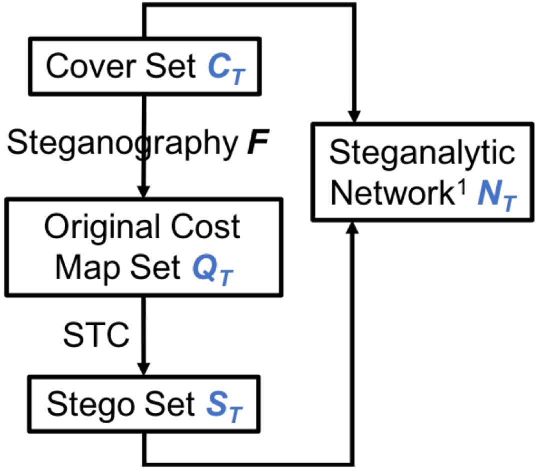
3/ Experiments

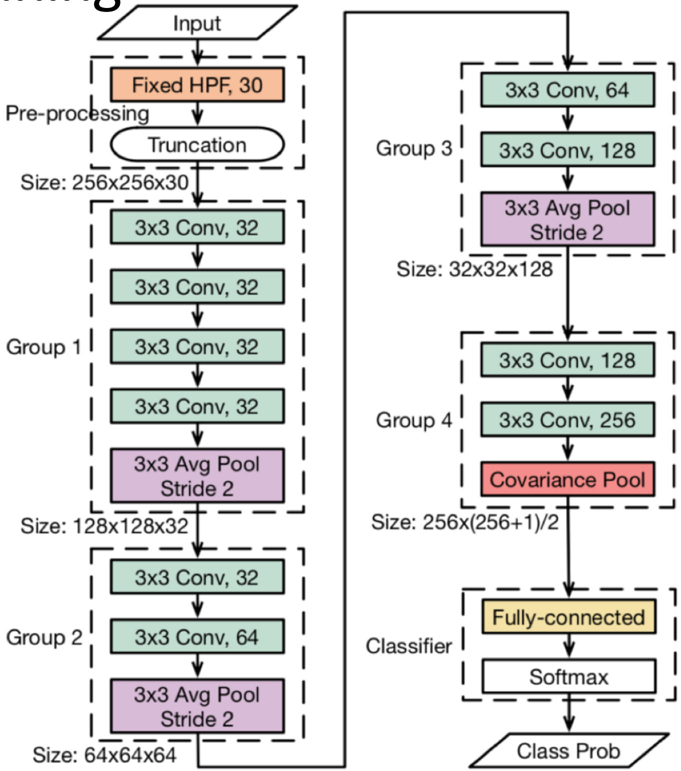
4/ Conclusion

Proposed Framework



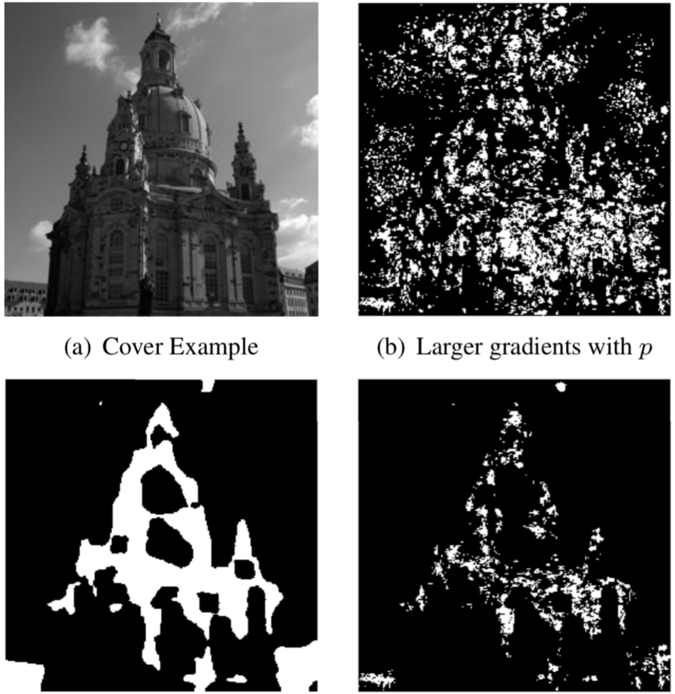
- Step #1: Steganalytic Network Training





[1] X. Deng, B. Chen, W. Luo, and D. Luo, “Fast and effective global covariance pooling network for image steganalysis,” in ACM Workshop on Information Hiding and Multimedia Security, 2019, pp. 230–234.

- Step #2: Stego Generation

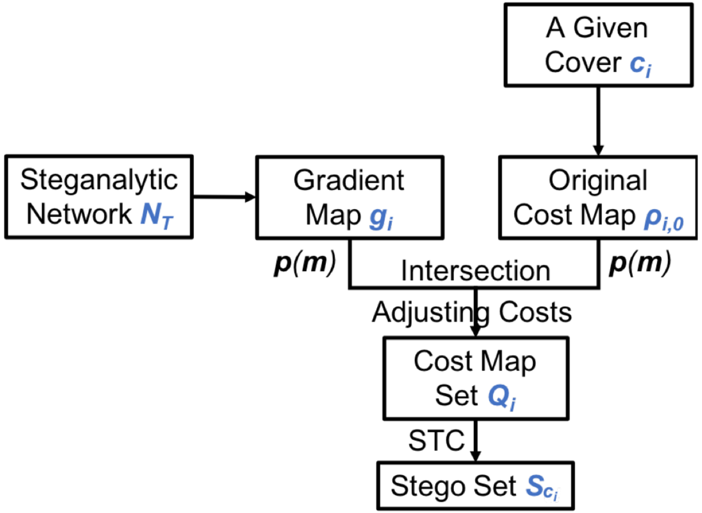


(a) Cover Example

(b) Larger gradients with p

(c) Smaller costs with p

(d) Selected Units



A Given Cover c_i

Steganalytic Network N_T

Gradient Map g_i

Original Cost Map $\rho_{i,0}$

$p(m)$

Intersection

$p(m)$

Adjusting Costs

Cost Map Set Q_i

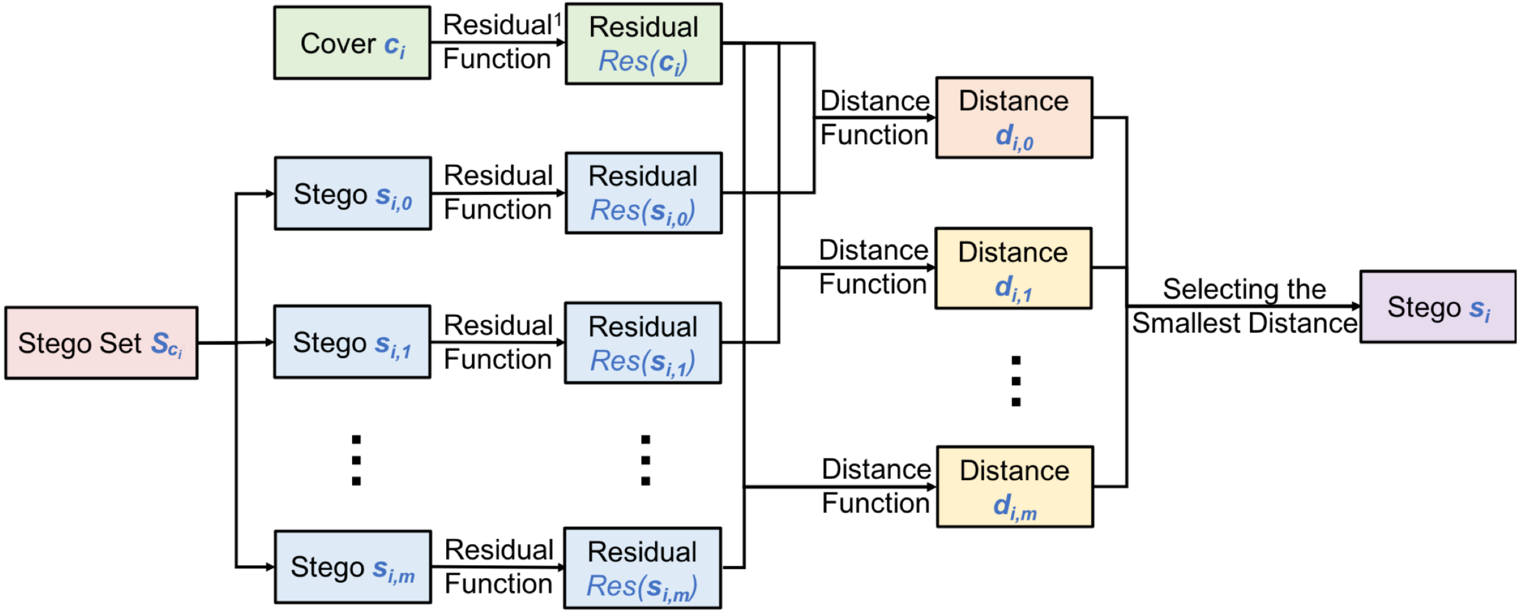
STC

Stego Set S_{c_i}

$$\rho^{+}_{i,j}(x,y) = \begin{cases} \rho^{+}_{i,0}(x,y) & g_i(x,y) < 0 \\ \rho^{+}_{i,0}(x,y) + \alpha & g_i(x,y) > 0 \end{cases}$$

$$\rho^{-}_{i,j}(x,y) = \begin{cases} \rho^{-}_{i,0}(x,y) + \alpha & g_i(x,y) < 0 \\ \rho^{-}_{i,0}(x,y) & g_i(x,y) > 0 \end{cases}$$

- Step #3: Stego Selection



Selection Method

[1] Bolin Chen, Weiqi Luo, and Jiwu Huang, “Universal stego post-processing for enhancing image steganography,” Journal of Information Security and Applications, vol. 55, pp.102664, 2020.

Outline

1/ Traditional & Adversarial Embedding

2/ Proposed Method

3/ Experiments

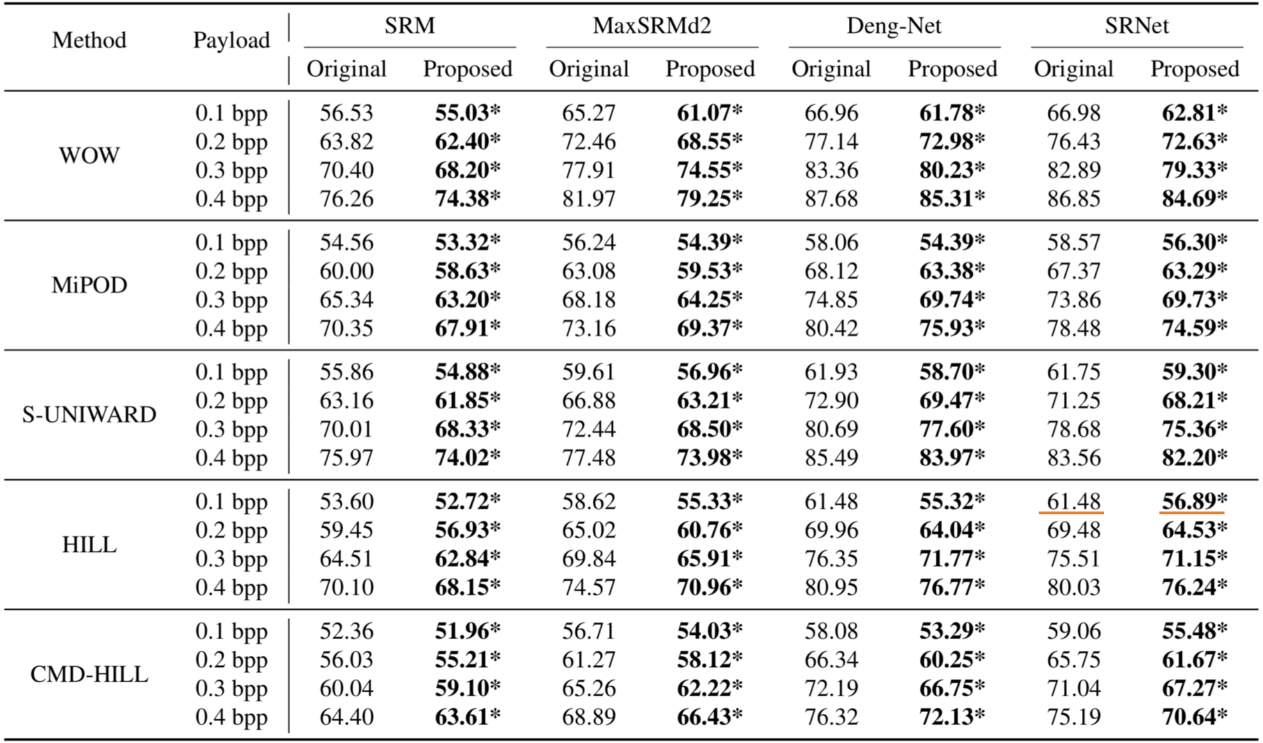
4/ Conclusion

Settings

- Database:
  - 10,000 images in BOSSBase-v1.01 & 10,000 image in BOWS2 are resized to 256x256, random shuffle
  - 8000 for training, 2000 for evaluation --- step #1
  - 10,000 for testing --- step #2 & #3

- Key Parameters
  - Number of new generated stego: m = 100
  - Cost enhance parameter: α = 2
  - Top gradient & low cost: p ∈ [0.25, 1.25] $\times payload$

Security Performances

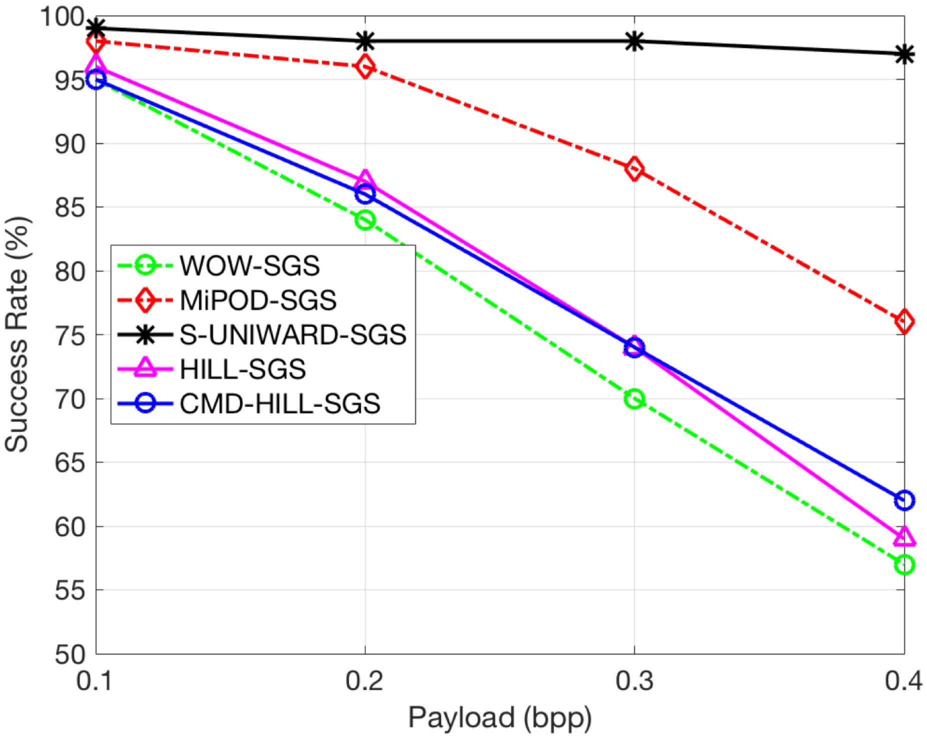


Method	Payload	SRM		MaxSRMd2		Deng-Net		SRNet	
		Original	Proposed	Original	Proposed	Original	Proposed	Original	Proposed
WOW	0.1 bpp	56.53	**55.03***	65.27	**61.07***	66.96	**61.78***	66.98	**62.81***
	0.2 bpp	63.82	**62.40***	72.46	**68.55***	77.14	**72.98***	76.43	**72.63***
	0.3 bpp	70.40	**68.20***	77.91	**74.55***	83.36	**80.23***	82.89	**79.33***
	0.4 bpp	76.26	**74.38***	81.97	**79.25***	87.68	**85.31***	86.85	**84.69***
MiPOD	0.1 bpp	54.56	**53.32***	56.24	**54.39***	58.06	**54.39***	58.57	**56.30***
	0.2 bpp	60.00	**58.63***	63.08	**59.53***	68.12	**63.38***	67.37	**63.29***
	0.3 bpp	65.34	**63.20***	68.18	**64.25***	74.85	**69.74***	73.86	**69.73***
	0.4 bpp	70.35	**67.91***	73.16	**69.37***	80.42	**75.93***	78.48	**74.59***
S-UNIWARD	0.1 bpp	55.86	**54.88***	59.61	**56.96***	61.93	**58.70***	61.75	**59.30***
	0.2 bpp	63.16	**61.85***	66.88	**63.21***	72.90	**69.47***	71.25	**68.21***
	0.3 bpp	70.01	**68.33***	72.44	**68.50***	80.69	**77.60***	78.68	**75.36***
	0.4 bpp	75.97	**74.02***	77.48	**73.98***	85.49	**83.97***	83.56	**82.20***
HILL	0.1 bpp	53.60	**52.72***	58.62	**55.33***	61.48	**55.32***	61.48	**56.89***
	0.2 bpp	59.45	**56.93***	65.02	**60.76***	69.96	**64.04***	69.48	**64.53***
	0.3 bpp	64.51	**62.84***	69.84	**65.91***	76.35	**71.77***	75.51	**71.15***
	0.4 bpp	70.10	**68.15***	74.57	**70.96***	80.95	**76.77***	80.03	**76.24***
CMD-HILL	0.1 bpp	52.36	**51.96***	56.71	**54.03***	58.08	**53.29***	59.06	**55.48***
	0.2 bpp	56.03	**55.21***	61.27	**58.12***	66.34	**60.25***	65.75	**61.67***
	0.3 bpp	60.04	**59.10***	65.26	**62.22***	72.19	**66.75***	71.04	**67.27***
	0.4 bpp	64.40	**63.61***	68.89	**66.43***	76.32	**72.13***	75.19	**70.64***

Success Rate

$$R_S = \frac{\sum_{\forall c_i \in C} I(F - SGS(c_i) \neq s_{i,0})}{|C|}$$

- $F - SGS$ ：enhancing steganography F using the proposed method
- $F - SGS(c_i)$ ：final selected stego for an input cover c_i
- $I(*)$ ：indicator function
- $|C|$ ：number of elements in test set C



Cost Modification Rate

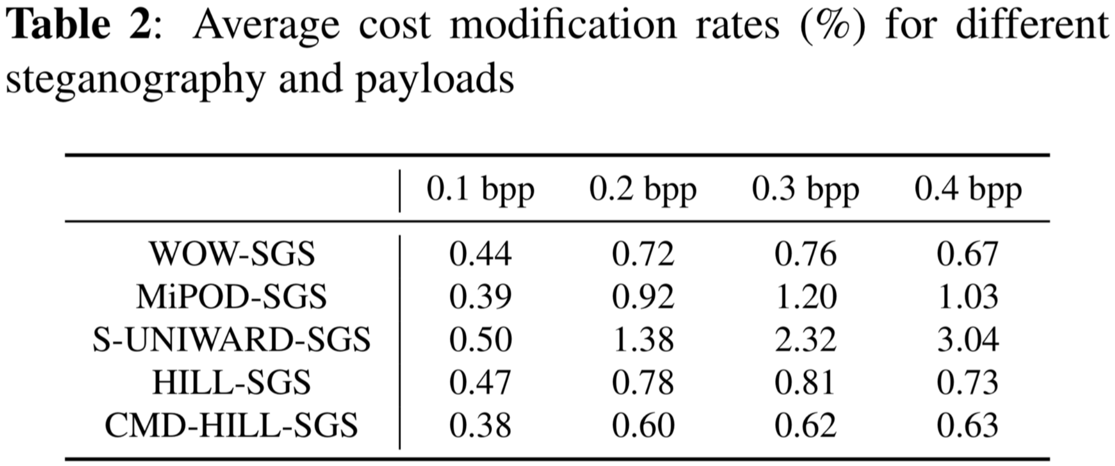


Table 2: Average cost modification rates (%) for different steganography and payloads

	0.1 bpp	0.2 bpp	0.3 bpp	0.4 bpp
WOW-SGS	0.44	0.72	0.76	0.67
MiPOD-SGS	0.39	0.92	1.20	1.03
S-UNIWARD-SGS	0.50	1.38	2.32	3.04
HILL-SGS	0.47	0.78	0.81	0.73
CMD-HILL-SGS	0.38	0.60	0.62	0.63

Over 99% of original costs would not changed !

Outline

1/ Traditional & Adversarial Embedding

2/ Proposed Method

3/ Experiments

4/ Conclusion

Contributions

- New framework for enhancing existing steganography
- Great security improvement

Future Works

- Apply in JPEG steganography
- More pre-trained classifiers
- Other generate methods and select methods

THANKS

Enhancing Image Steganography via Stego Generation and Selection

Email: songtt3@mail2.sysu.edu.cn