Payload-Independent Direct Cost Learning for Image Steganography

Weixiang Li, Shihang Wu, Bin Li, Weixuan Tang, and Xinpeng Zhang

***Abstract*—Recent research has shown that architectures utilizing reinforcement learning (RL) are effective in cost-based image steganography. However, these architectures only learn embedding probabilities rather than costs, and are trained for a specific embedding payload, making it difficult to extend the trained model to serve other payloads. In this paper, we propose a payload-independent cost learning framework using RL called PICO-RL. This framework directly learns universal costs that can be applied to any payload. PICO-RL incorporates an optimal probability approximation (OPA) module that can calculate the required probability map for embedding simulation directly from a learned cost map for any payload, eliminating the need for time-consuming searches for a valid probability scaling parameter. Additionally, PICO-RL uses an advanced steganalysis environment network to provide more effective reward feedback for learning. During RL training, the learned cost maps of different payloads converge and eventually become similar under the OPA constraint, resulting in payload independence. Experimental results demonstrate that a well-trained PICO-RL model, which acts as a universal cost function, defines costs with superior security performance against steganalysis and has better coding compatibility when encoding with practical steganographic codes.**

***Index Terms*—Image steganography, steganalysis, automatic cost learning, reinforcement learning.**

I. Introduction

Steganography is a technique used for covert communication to hide messages without drawing suspicion [1], [2]. Mainstream image steganographic methods typically use a minimal-distortion paradigm [3]. With near-optimal steganographic codes [4], [5], there are three ways of defining embedding costs, i.e., hand-crafted [6]–[8], statistical model-based [9], [10] and deep learning (DL)-based [11]–[15] approaches.

As DL-based steganalysis models [16]–[18] surpass hand-crafted steganalytic methods [19], [20], the DL-based cost functions also achieve better steganographic security than the hand-crafted and statistical model-based ones. These DL-based cost functions are realized through generative adversarial network (GAN) [11], [12], reinforcement learning (RL) [13], [14] or adversarial perturbation (AP) [15]. While AP is used when baseline costs are initialized, GAN and RL learn baseline costs from scratch [11]–[14]. In GAN-based methods, two networks, namely a steganographic generative network and a steganalysis discriminative network, play a confrontation game with a trainable embedding simulator to learn embedding probabilities. However, these methods have limitations in that the continuous outputs of the simulator cannot well match discrete modifications, and the image-level loss function is too coarse to guide effective learning. Alternatively, SPAR-RL [13] based on RL uses a policy network (agent) to learn embedding policies (probabilities) by maximizing fine-grained pixel-level rewards from a steganalysis environment.

Despite achieving state-of-the-art security performance, SPAR-RL has some potential drawbacks. Firstly, it performs probability learning instead of direct cost learning. Converting the learned probabilities to costs for adaptation to steganographic codes can sometimes result in coding failures. One potential solution is to train the policy network to learn costs directly, but the conversion from costs to probabilities [3] required for embedding simulation can be time-consuming during training. Secondly, SPAR-RL is trained using a specific embedding payload, meaning that a payload-dependent model cannot yield costs that are universal for arbitrary payloads. Moreover, the quality of the learned costs is largely determined by the performance of the environment network. Therefore, adopting a state-of-the-art steganalysis network as the environment can further boost the security performance of SPAR-RL.

From the above analysis, it is obvious that there is still room for improvement in designing a DL-based cost function that is more compatible with coding, independent of payload and more secure. In this paper, we propose a Payload-Independent COst learning framework using RL (PICO-RL) with optimal probability approximation (OPA). To approximate the implicit relationship between the scaling parameter and the payload in Gibbs distribution, a power function is exploited and fitted with only a few image samples. This function efficiently facilitates OPA, i.e., probabilities for arbitrary payloads can be obtained directly from costs without time-consuming calculations. By utilizing OPA that corresponds to a type of universal cost map and minimizing a payload loss function that regulates the payload constraint during RL training, the learned costs with respect to arbitrary input payloads gradually converge and eventually become similar, thus ensuring payload

This work was supported in part by NSFC (Grant Nos. 62202310, U22B2047, 62002075), Guangdong Basic and Applied Basic Research Fundation (Grant Nos. 2019B151502001, 2023A1515011428), Shenzhen R&D Program (Grant No. JCYJ20200109105008228), and China Postdoctoral Science Foundation (Grant No. 2022M722192).

Weixiang Li, Shihang Wu and Bin Li are with the Guangdong Key Laboratory of Intelligent Information Processing and the Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China. (e-mail: liweixiang@szu.edu.cn; 2020285059@email.szu.edu.cn; libin@szu.edu.cn).

Weixuan Tang is with the Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China (e-mail: tweix@gzhu.edu.cn).

Xinpeng Zhang is with the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China (e-mail: xzhang@shu.edu.cn).

Corresponding author: Bin Li (e-mail: libin@szu.edu.cn).

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3294291

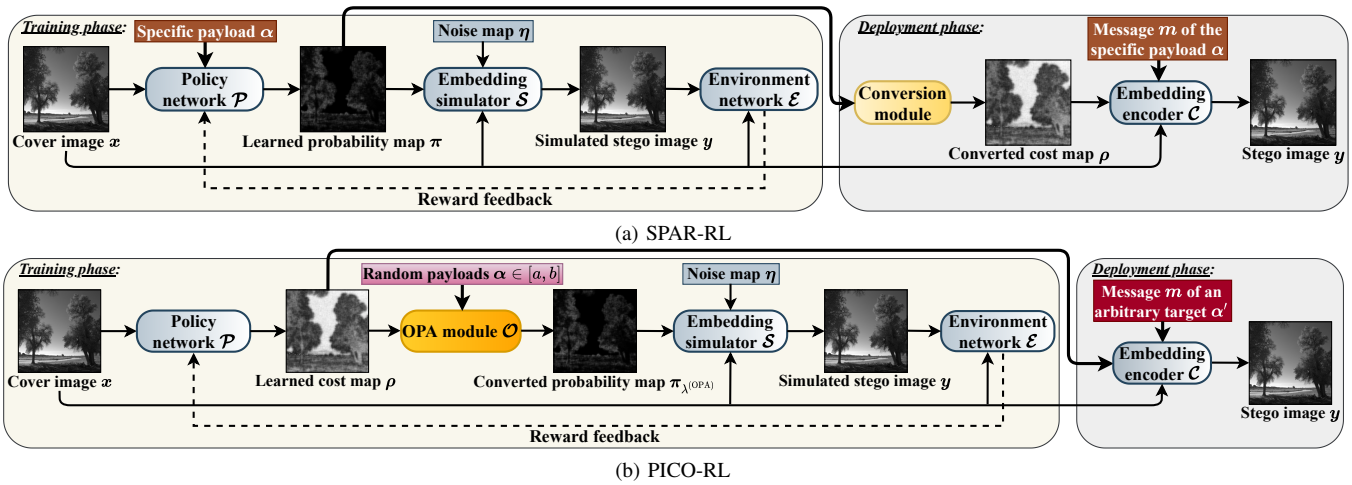


Fig. 1: Frameworks of SPAR-RL and the proposed PICO-RL. Compared with SPAR-RL, PICO-RL adds an OPA module $\mathcal{O}$ whose input payloads are randomly sampled from interval $[a, b]$ in the training phase, and removes the conversion module and works with arbitrary target payloads in the deployment phase thanks to the payload-independent direct cost learning.

independence. Besides, PICO-RL employs SRNet [16] as the environment to gain better reward feedback for learning. The main contributions of this paper are summarized as follows.

- We propose a novel framework, PICO-RL, for learning payload-independent embedding costs in image steganography. The framework includes a novel OPA module designed to directly obtain the probabilities required for embedding simulation and effectively constraint the learned costs for arbitrary payloads to be similar. By training with RL, a universal PICO-RL model can act as a practical cost function that yield costs applicable to all payloads.
- PICO-RL is the first framework for direct cost learning from scratch, making it more compatible with steganographic codes that require costs as input and reducing coding failures. The directly-learned costs also increase the maximum hiding capacity of pixels compared to SPAR-RL, allowing for more texture-adaptive embedding.
- Experimental results demonstrate that a well-trained PICO-RL model outperforms the hand-crafted cost functions and SPAR-RL by a large margin, especially in resisting DL-based steganalysis.

II. Related Works

A. Minimal-distortion and Optimal Probability

Under the minimal-distortion paradigm [3], the payload-limited sender problem is formulated by the optimization:

$$\min_{\boldsymbol{\pi}} E_{\boldsymbol{\pi}}(D) = \sum_{i=1}^{n} \sum_{t_i \in \mathcal{I}_i} \pi(t_i)\rho(t_i) \tag{1}$$

$$\text{s.t. } H(\boldsymbol{\pi}) = -\sum_{i=1}^{n} \sum_{t_i \in \mathcal{I}_i} \pi(t_i) \log_2 \pi(t_i) = m = \alpha \cdot n, \tag{2}$$

where $\rho(t_i)$ and $\pi(t_i)$ are the cost and probability of modifying a cover element x_i to a stego element $y_i = x_i + t_i$ by t_i, respectively. m, α and n are the bits of message, the payload in bit per pixel (bpp) and the number of cover elements, respectively. Assuming the cost of no modification be 0, the optimal probabilities $\boldsymbol{\pi}_\lambda$ derived by the costs $\boldsymbol{\rho}$ in a ternary form, i.e., $t_i \in \mathcal{I}_i = \{-1, 0, +1\}$, follow a Gibbs distribution:

$$\pi_\lambda(t_i) = \begin{cases} \frac{e^{-\lambda\rho_i}}{1+2\cdot e^{-\lambda\rho_i}}, & t_i = +1 \text{ or } -1 \\ \frac{1}{1+2\cdot e^{-\lambda\rho_i}}, & t_i = 0 \end{cases}, 1 \le i \le n, \tag{3}$$

where the scaling parameter λ ($\lambda > 0$) for obtaining $\boldsymbol{\pi}_\lambda$ is determined by (2) via a binary search [3].

B. SPAR-RL

Steganographic pixel-wise actions and rewards with RL (SPAR-RL) [13] contains a policy network $\mathcal{P}$, an embedding simulator $\mathcal{S}$ and an environment network $\mathcal{E}$ in the training phase, as shown in Fig. 1(a). $\mathcal{P}$ aims to learn an embedding probability map $\boldsymbol{\pi}$ of a cover image $\boldsymbol{x}$ under a specific payload α. $\mathcal{S}$ is used to generate the simulated stego $\boldsymbol{y}$ for feeding $\mathcal{E}$. $\mathcal{E}$ attempts to distinguish whether the input image is $\boldsymbol{x}$ or $\boldsymbol{y}$ and provides the feedback of pixel-wise rewards of modifications to $\mathcal{P}$. In the deployment phase, a well-trained policy network $\mathcal{P}$ is used to generate a probability map for a cover image. To adapt to practical codes $\mathcal{C}$, the probability map must be converted to a cost map by the Flipping Lemma [3]–[5], i.e., $\rho(t_i) = \ln(\pi(t_i)/\pi(0))$. Since SPAR-RL is trained with a specific α, the learned probability map is payload-dependent and its converted cost map cannot work well for other payloads that do not match the training payload.

III. PICO-RL Framework

In this section, we propose a Payload-Independent COst learning framework with RL (PICO-RL), which aims to yield direct and universal embedding costs. The overall framework is first outlined, and then a key module, i.e., optimal probability approximation (OPA), is introduced in detail.

A. Overall Framework

The overall framework of PICO-RL is illustrated in Fig. 1(b). Different from SPAR-RL, PICO-RL attempts to learn payload-independent direct costs by adding an OPA module $\mathcal{O}$

in the training phase to ensure training efficiency and payload independence. Without conversion, the directly-learned costs can work well with the encoder $\mathcal{C}$ for arbitrary target payloads in the deployment phase. Through direct cost learning, PICO-RL has a larger maximum hiding capacity for a textured pixel and makes the embedding more texture-adaptive. The training and deployment phases are provided in **Algorithm 1**, and the implementation details are as follows.

1) Learning Cost Map with Policy Network $\mathcal{P}$: Given a cover image $\boldsymbol{x}$, $\mathcal{P}$ outputs a cost map $\boldsymbol{\rho}$, where ρ_i is the cost of modifying x_i by $+1$ or -1. We use the same U-Net based policy network as that in SPAR-RL [13], except that the last group is replaced by the Sigmoid function to limit the value range of the output ρ_i to $[0, 1]$.

2) Obtaining Probability Map with OPA Module $\mathcal{O}$: Given a learned cost map $\boldsymbol{\rho}$ and a randomly selected payload α ranged in $[a, b]$ bpp, $\mathcal{O}$ outputs the converted probability map $\boldsymbol{\pi}_{\lambda^{(\text{OPA})}}$ directly. Under the constraint of OPA, a learned $\boldsymbol{\rho}$ has the potential to simultaneously meet any α. Using different α during training, the module can thus guide $\mathcal{P}$ to learn a payload-independent $\boldsymbol{\rho}$. Note that the value range of the ± 1 total changing probability derived by OPA is $[0, 2/3]$, which is different from $[0, 0.5]$ in SPAR-RL. This means that textured pixels can carry more information bits, allowing for more texture-adaptive embedding. The detailed design of OPA is given in Section III-B.

3) Generating Simulated Stego Image with Embedding Simulator $\mathcal{S}$: Given $\boldsymbol{x}$, $\boldsymbol{\pi}_{\lambda^{(\text{OPA})}}$, and a random noise map $\boldsymbol{\eta}$ whose elements follow an uniform distribution $\mathrm{U}(0, 1)$, $\mathcal{S}$ generates a simulated stego $\boldsymbol{y}$ by

$$y_i = \begin{cases} x_i + 1, & \text{if } \eta_i < \pi_{\lambda^{(\text{OPA})}}(+1) \\ x_i - 1, & \text{if } \eta_i > 1 - \pi_{\lambda^{(\text{OPA})}}(-1) \\ x_i, & \text{otherwise} \end{cases}, 1 \le i \le n. \quad (4)$$

4) Obtaining Rewards with Environment Network: The environment network $\mathcal{E}$ is used to distinguish between the input $\boldsymbol{x}$ and $\boldsymbol{y}$, and yield the pixel-wise rewards $\boldsymbol{r}$. Since the discrimination ability of $\mathcal{E}$ largely determines the merit of $\boldsymbol{r}$, we select the state-of-the-art steganalysis network SRNet [16] as $\mathcal{E}$. Following the reward feedback mechanism in SPAR-RL [13], the pixel-wise $\boldsymbol{r}$ w.r.t. the modifications $\boldsymbol{t}$ is given by

$$r_i = \beta \cdot \mathrm{sign}(t_i) \cdot g_i, \quad 1 \le i \le n, \quad (5)$$

where β is the reward magnitude, $t_i = y_i - x_i$ is the modification, and $g_i = \partial L_{\mathcal{E}} / \partial t_i$ is the modification gradient from the cross-entropy loss function of $\mathcal{E}$, i.e., $L_{\mathcal{E}} = -z'_0 \log_2 z_0 - z'_1 \log_2 z_1$. z_0 and z_1 are the Softmax outputs for denoting $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, while z'_0 and z'_1 are their ground-truth labels. A positive r_i is assigned when the sign of t_i has the same sign that increases $L_{\mathcal{E}}$, and $|r_i|$ is large with a large $|g_i|$. In this way, the reward feedback from $\mathcal{E}$ can guide $\mathcal{P}$ to learn a secure $\boldsymbol{\rho}$. Similar to [13], the total loss function $L_{\mathcal{P}}$ of $\mathcal{P}$ is composed by the weighted sum of the reward loss $L_{\boldsymbol{r}}$ and the payload loss L_α:

$$L_{\mathcal{P}} = L_{\boldsymbol{r}} + \gamma \cdot L_\alpha, \quad (6)$$

$$L_{\boldsymbol{r}} = -\frac{1}{n} \sum_{1}^{n} r_i \cdot \log_2 \pi_{\lambda^{(\text{OPA})}}(t_i), \quad (7)$$

$$L_\alpha = \left(-\frac{1}{n} \sum_{i=1}^{n} \sum_{t_i \in \mathcal{I}_i} \pi_{\lambda^{(\text{OPA})}}(t_i) \log_2 \pi_{\lambda^{(\text{OPA})}}(t_i) - \alpha \right)^2, \quad (8)$$

where the weighting parameter γ is to balance $L_{\boldsymbol{r}}$ and L_α.

Algorithm 1 Pseudo-code of the training and deployment.

Iterative Training Phase:

1: Generate $\boldsymbol{\rho}$ from a cover image $\boldsymbol{x}$ by $\mathcal{P}$;
2: Calculate $\boldsymbol{\pi}_{\lambda^{(\text{OPA})}}$ from $\boldsymbol{\rho}$ with a randomly selected α by $\mathcal{O}$ using (9);
3: Generate a simulated stego $\boldsymbol{y}$ from $\boldsymbol{x}$, $\boldsymbol{\pi}_{\lambda^{(\text{OPA})}}$ and $\boldsymbol{\eta}$ by $\mathcal{S}$ using (4);
4: Assign $\boldsymbol{r}$ from $\boldsymbol{x}$ and $\boldsymbol{y}$ by $\mathcal{E}$ using (5);
5: Update the parameters of $\mathcal{E}$ with the loss function $L_{\mathcal{E}}$;
6: Update the parameters of $\mathcal{P}$ with the loss function $L_{\mathcal{P}}$.

Deployment Phase:

1: Generate $\boldsymbol{\rho}$ from a cover image $\boldsymbol{x}$ by a well-trained $\mathcal{P}$;
2: Generate a stego image $\boldsymbol{y}$ from $\boldsymbol{x}$, $\boldsymbol{\rho}$ and a message $\boldsymbol{m}$ of any α' by the embedding encoder $\mathcal{C}$, i.e., $\boldsymbol{y} = \mathcal{C}(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{m})$.

The PICO-RL framework equipped with the OPA module has good scalability since other enhanced $\mathcal{P}$, $\mathcal{E}$ and reward feedback mechanism can be easily incorporated.

B. The Design of OPA Module

1) Motivation: As aforementioned, the OPA module $\mathcal{O}$ has two functions: obtaining probability maps directly from cost maps for arbitrary payloads and converging the learned cost maps of different payloads during training. Firstly, once the scaling parameter λ in (3) for the payload α in (2) is determined, the optimal probability map $\boldsymbol{\pi}_\lambda$ can be obtained directly from the cost map $\boldsymbol{\rho}$. However, there is no explicit function but a monotonic decreasing relationship to characterize $\lambda = f(\alpha)$, and current research applies a binary search to determine λ for α due to the monotonicity [3]. Secondly, each $\boldsymbol{\rho}$ leads to a specific type of $\lambda = f(\alpha)$, and conversely, each $\lambda = f(\alpha)$ corresponds to a specific type of $\boldsymbol{\rho}$. Based on the above analysis, we propose to approximate $\lambda = f(\alpha)$ with an explicit expression $\lambda^{(\text{OPA})} = \tilde{f}(\alpha)$, where $\tilde{f}$ is a function that approximates f. With this expression, different α can be obtained directly from α without search, and the approximate optimal probability map $\boldsymbol{\pi}_{\lambda^{(\text{OPA})}} = \varphi(\boldsymbol{\rho}, \tilde{f}(\alpha))$ needed for embedding simulation can be converted given a specific $\boldsymbol{\rho}$. Moreover, since $\lambda^{(\text{OPA})} = \tilde{f}(\alpha)$ actually corresponds to a certain type of $\boldsymbol{\rho}$, learning this $\boldsymbol{\rho}$ to simultaneously meet any α is feasible. This can be achieved by minimizing the payload loss L_α and the reward loss $L_{\boldsymbol{r}}$ that regulate the payload constraint and the security gain for different random α during RL training. The learned $\boldsymbol{\rho}$ of different α gradually converge and eventually become similar with satisfactory security, thus achieving payload independence.

2) Solution: We obtain $\lambda^{(\text{OPA})} = \tilde{f}(\alpha)$ via curve fitting with a few training samples. The curve fitting steps are as follows.

S1. ***Calculation and Normalization of λ.*** 100 images, accounting for 2.4‰ of the dataset, are randomly selected from SZUBase [13]. Their cost maps are defined by

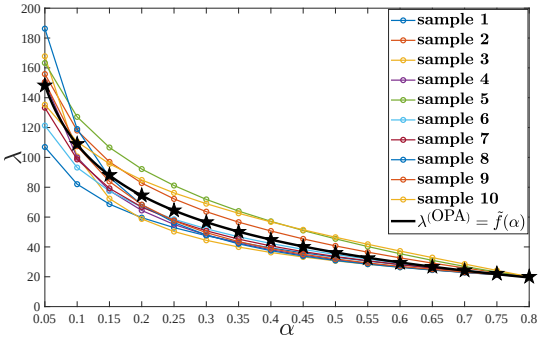


Fig. 2: Curves of normalized λ changed with α.

TABLE I: Fitting results of $\lambda^{(\mathrm{OPA})} = \tilde{f}(\alpha)$.

Type	Expression	R-square
Power	$\lambda^{(\mathrm{OPA})} = c_1 \cdot \alpha^{c_2} + c_3$	0.9999
Exponential	$\lambda^{(\mathrm{OPA})} = c_1 \cdot e^{c_2 \cdot \alpha} + c_3 \cdot e^{c_4 \cdot \alpha}$	0.9998
Rational	$\lambda^{(\mathrm{OPA})} = (c_1 \cdot \alpha^2 + c_2 \cdot \alpha + c_3)/(\alpha + c_4)$	0.9998

HILL[1] [6], and λ for each $\alpha \in \{0.05{\cdot}k \mid k = 1, 2, \cdots, 16\}$ bpp is searched. In order to make all curves of λ changed with α converge at $\alpha = 0.8$ via normalizing λ, we scale $\lambda_{\alpha=0.8}$ to 20 and then scale λ of other α by $20/\lambda_{\alpha=0.8}$, so that $\boldsymbol{\rho}$ scaled by $\lambda_{\alpha=0.8}/20$ can still educe the same $\boldsymbol{\pi}_\lambda$ in (3). Choosing 20 as the minimum λ is to ensure $e^{-\lambda\rho_i} \approx 0$ when $\rho_i \in [0, 1]$. In Fig. 2 we plot the curves of the normalized λ changed with α for 10 image samples, where the decreasing trends of all curves are roughly similar.

S2. ***Curve Fitting of*** λ. We average the normalized λ of the 100 images on each α, as marked by "★" in Fig. 2. Three types of $\lambda^{(\mathrm{OPA})} = \tilde{f}(\alpha)$ are designed for curve fitting in Table I, and they all fit well with large R-square values. We select the power function with the largest R-square value, but actually these functions have almost the same effect on the learning performance. One can observe that the fitted curve has the similar decreasing trend with other curves in Fig. 2.

With $\lambda^{(\mathrm{OPA})} = \tilde{f}(\alpha) = c_1 \cdot \alpha^{c_2} + c_3$, the converted probabilities $\boldsymbol{\pi}_{\lambda^{(\mathrm{OPA})}} = \varphi(\boldsymbol{\rho}, \tilde{f}(\alpha))$ can be obtained directly by

$$\pi_{\lambda^{(\mathrm{OPA})}}(t_i) = \begin{cases} \frac{e^{-(c_1 \cdot \alpha^{c_2} + c_3)\rho_i}}{1 + 2 \cdot e^{-(c_1 \cdot \alpha^{c_2} + c_3)\rho_i}}, & t_i = +1 \text{ or } -1 \\ \frac{1}{1 + 2 \cdot e^{-(c_1 \cdot \alpha^{c_2} + c_3)\rho_i}}, & t_i = 0 \end{cases}, \quad (9)$$

where $c_1 = 149.6$, $c_2 = -0.2163$, $c_3 = -137.4$ are the fitted parameters. Obviously, $\pi_{\lambda^{(\mathrm{OPA})}}(+1) + \pi_{\lambda^{(\mathrm{OPA})}}(-1) = 2/3$ if $\rho_i = 0$, resulting in the maximum hiding capacity of $\log_2 3$ bits which is greater than that in SPAR-RL. As aforementioned, this allows textured pixels to carry more information bits, making the embedding more texture-adaptive and secure.

IV. Experimental Results and Analysis

In this section, we examine the superiority of PICO-RL in payload independence, security and coding compatibility.

[1]We also fit the curve with cost maps defined by other cost functions (such as MiPOD [9]), and find that these fitted curves have similar performance.

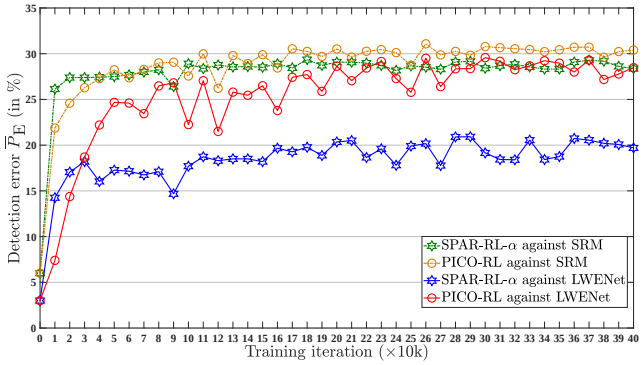


Fig. 3: Stability of SPAR-RL-α and PICO-RL at $\alpha = 0.4$ bpp.

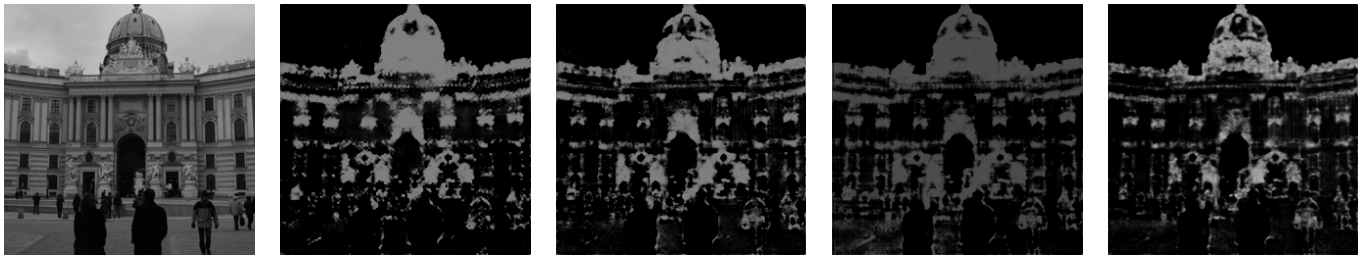
TABLE II: The security $\overline{P}_{\mathrm{E}}$ (in %) against the detections of SRM and LWENet. The data on the left and the right sides of the symbol "/" correspond to SRM and LWENet, respectively.

Method	0.05 bpp	0.2 bpp	0.4 bpp	0.6 bpp	0.8 bpp
SPAR-RL-0.05	47.93 / 48.32	35.97 / 27.82	15.82 / 7.23	7.92 / 3.43	4.63 / 2.10
SPAR-RL-0.2	47.87 / 48.25	39.49 / 33.20	26.20 / 16.97	12.05 / 5.45	6.06 / 2.98
SPAR-RL-0.4	47.14 / 45.15	39.40 / 32.95	28.89 / 20.35	19.21 / 12.10	10.21 / 5.70
SPAR-RL-0.6	47.26 / 46.57	38.89 / 32.17	28.68 / 19.65	21.00 / 12.42	13.90 / 8.05
SPAR-RL-0.8	43.72 / 41.55	36.49 / 30.80	27.63 / 20.30	20.37 / 12.35	14.21 / 8.90
PICO-RL(XuNet)	47.98 / 48.20	39.34 / 33.95	29.09 / 20.50	20.97 / 13.60	14.42 / 8.32
SPAR-RL(SRNet)-0.05	46.92 / 47.80	31.39 / 21.53	12.66 / 6.20	6.82 / 3.40	4.27 / 2.15
SPAR-RL(SRNet)-0.2	46.73 / 46.20	39.93 / 38.82	27.62 / 23.78	14.56 / 9.52	6.84 / 3.63
SPAR-RL(SRNet)-0.4	45.29 / 44.43	39.22 / 37.25	29.96 / 28.15	20.36 / 17.23	10.80 / 7.83
SPAR-RL(SRNet)-0.6	42.63 / 40.02	37.36 / 35.28	29.87 / 27.87	21.89 / 20.77	14.92 / 12.82
SPAR-RL(SRNet)-0.8	43.38 / 39.40	36.45 / 32.30	28.38 / 24.85	20.63 / 16.85	15.06 / 13.20
PICO-RL	**48.01 / 48.43**	**40.88 / 39.68**	**30.66 / 29.57**	**22.35 / 21.27**	**15.13 / 13.42**

A. Settings

The experiments are mainly conducted on SZUBase [11] and BOSSBase [21], where the images are resized to 256×256. SZUBase is used to train PICO-RL and SPAR-RL, while BOSSBase is for testing. We also include 10,000 images randomly selected from the ALASKA dataset [22] for testing. In PICO-RL, we set $a = 0.05$ and $b = 0.8$ to serve more payloads, follow the same $\beta = 10^7$ in SPAR-RL, and determine the optimal $\gamma = 100$ by traversal search from 10^{-5} to 10^5 with a step of 10 times. For a fair comparison, PICO-RL adopts the same training settings in SPAR-RL [13]. Both models are implemented using the TensorFlow framework, and trained and tested on a Centos7 system equipped with an Intel Xeon E5-2695 v4 CPU and a NVIDIA Tesla P100-PCIE-16GB GPU. The batch size is set to 25 and the Adam optimizer is employed with a learning rate of 0.0001.

We denote SPAR-RL trained with a specific payload α as SPAR-RL-α (SPAR-RL-v2 [13]) more specifically. The enhanced version of SPAR-RL-α by replacing the XuNet-based environment with SRNet is denoted as SPAR-RL(SRNet)-α accordingly. And the version of PICO-RL by replacing SRNet with the XuNet-based environment is denoted as PICO-RL(XuNet). We also include the state-of-the-art hand-crafted and statistical model-based cost functions, HILL [6] and MiPOD [9], for comparison. Unless otherwise specified, the embedding simulator [3] is employed to generate the stego images under $\alpha \in \{0.05, 0.2, 0.4, 0.6, 0.8\}$ bpp. Four steganalyzers, including two ensemble classifiers [23] using the hand-crafted features SRM [19] and maxSRMd2 [20], and two DL-based LWENet [17] and SRNet [16], are used to evaluate the security performance. Both tested datasets containing 10,000 images



(a) Cover image (b) SPAR-RL-0.4, $\pi_{\mathrm{m}} = 0.5$ (c) SPAR-RL(SRNet)-0.4, π_{m}=0.5 (d) SPAR-RL(SRNet)-0.8, π_{m}=0.31 (e) PICO-RL, $\pi_{\mathrm{m}} = 2/3$

Fig. 4: Probability map at 0.4 bpp of "1013.pgm" in BOSSBase. π_{m} denotes the maximum probability in each map.

TABLE III: The security $\overline{P}_{\mathrm{E}}$ (in %) against the detections of SRM, maxSRMd2, LWENet and SRNet on BOSSBase.

Steganalyzer	SRM					maxSRMd2					LWENet					SRNet				
Payload α	0.05	0.2	0.4	0.6	0.8	0.05	0.2	0.4	0.6	0.8	0.05	0.2	0.4	0.6	0.8	0.05	0.2	0.4	0.6	0.8
HILL	47.70	38.09	27.27	18.60	12.57	43.45	32.75	23.09	16.61	12.11	44.50	25.98	16.20	11.12	7.27	38.38	24.12	14.78	9.83	5.60
MiPOD	47.20	37.03	26.78	19.09	13.00	44.85	34.29	24.15	17.59	12.53	47.20	28.87	17.83	11.75	7.47	43.25	26.62	15.90	9.75	5.77
SPAR-RL-α	47.93	39.49	28.89	21.00	14.21	43.59	32.44	23.79	18.20	13.70	48.32	33.20	20.35	12.42	8.90	43.28	27.07	17.38	9.68	5.98
PICO-RL	**48.01**	**40.88**	**30.66**	**22.35**	**15.13**	**44.85**	**37.11**	**28.42**	**21.49**	**14.89**	**48.43**	**39.68**	**29.57**	**21.27**	**13.42**	**45.78**	**35.23**	**26.85**	**17.38**	**10.50**

are respectively divided with a proportion of 1:1 for training and testing SRM and maxSRMd2, and with 7:1:2 for training, validating and testing LWENet and SRNet. The steganographic security is measured by the average classification error rate $\overline{P}_{\mathrm{E}}$, and larger $\overline{P}_{\mathrm{E}}$ means higher security.

B. Performance on Payload Independence

We first evaluate the stability performance of PICO-RL and select the optimal model of the policy network. The security of PICO-RL and SPAR-RL-α changed with the training iteration is depicted in Fig. 3. Like SPAR-RL-α, PICO-RL is stable and converges rapidly though it employs a more complicated environment network. Besides, PICO-RL has higher security than SPAR-RL-α especially against LWENet. In the following experiments, we select the optimal model at the 200k-th iteration for SPAR-RL-α, and the 300k-th iteration for PICO-RL using the complex environment. Similarly, PICO-RL(XuNet) and SPAR-RL(SRNet)-α utilize the optimal models at the 200k-th and 300k-th iterations, respectively.

The performance on payload independence is shown in Table II, where both SPAR-RL-α and SPAR-RL(SRNet)-α are tested for all matched and mismatched payloads. Clearly, both of them achieve the optimal security under the matched payloads, and SPAR-RL(SRNet)-α is much more secure thanks to the more powerful environment. However, as expected, they do not perform well in the face of the mismatched payloads. For example, the security against LWENet at 0.4 bpp drops sharply from 20.35% of SPAR-RL-0.4 to 7.23% of SPAR-RL-0.05. Instead, PICO-RL(XuNet) and PICO-RL, which are trained with arbitrary payloads, can yield payload-independent costs. Note that, PICO-RL gains the best security under all payloads, and its performance advantage over SPAR-RL(SRNet)-α reveals the advantage of direct cost learning. In Fig. 4 we visualize the probability maps learned by different methods. Compared with the SPAR-RL methods, the probability map of PICO-RL shows stronger adaptability, i.e., the larger probabilities appear in the textured regions where the modifications are more undetectable. As mentioned before, the direct cost learning ensures that the probability derived by (9) can reach the maximum $\pi_{\mathrm{m}} = 2/3$. However, π_{m} of SPAR-RL is limited to 0.5 during training, and π_{m} of the mismatched model is further suppressed. Probabilities with larger value ranges correspond to more differentiated and superior costs in practice. In summary, a well-trained PICO-RL model can serve any payload, without the cumbersome need to train different models for each predefined payload.

TABLE IV: The security $\overline{P}_{\mathrm{E}}$ (in %) against the detections of SRM and LWENet on ALASKA.

Steganalyzer	SRM		LWENet	
Payload α	0.4 bpp	0.8 bpp	0.4 bpp	0.8 bpp
HILL	32.51	18.69	20.10	9.95
MiPOD	31.74	18.55	20.77	9.48
SPAR-RL-α	34.61	20.29	25.92	12.22
PICO-RL	**35.54**	**21.04**	**31.43**	**16.53**

C. Performance on Steganographic Security

We herein report the security comparison between PICO-RL and the state-of-the-art steganographic methods in Table III (on BOSSBase) and Table IV (on ALASKA). One can observe that PICO-RL outperforms HILL, MiPOD and SPAR-RL-α (in matched cases) by a large margin, especially in resisting the DL-based steganalyzers. For example, the security advantages of PICO-RL over SPAR-RL-α are respectively +9.22% and +9.47% against LWENet and SRNet at 0.4 bpp in Table III.

D. Performance on Coding Compatibility

Since the learned costs need to work with steganographic codes for actual message embedding, good coding compatibility is essential in practice. In Table V, we compare the coding compatibility of PICO-RL and SPAR-RL-α when using STC ($h = 10$) [4] and SPC ($L = 1$) [5] over the whole BOSSBase dataset. For SPAR-RL-α, a very small value (i.e., 10^{-6}) is added to the learned probability map to eliminate the probabilities with values of 0, ensuring the usability of the converted cost map before embedding. Additionally, a large value (i.e., wet cost [4], [5]) is assigned to the cost of a saturated pixel (i.e., 255 being modified by $+1$ or 0 being modified

TABLE V: The number of coding failures (n_{cf}) and the security $\overline{P}_{\mathrm{E}}$ (in %) against LWENet for the actual embedding. The values on the left and the right sides of the symbol "/" correspond to the results by using STC and SPC, respectively.

	Method	0.05 bpp	0.2 bpp	0.4 bpp	0.6 bpp	0.8 bpp
n_{cf}	SPAR-RL-α	0 / 0	0 / 1	35 / 1	19 / 1	2 / 1
	PICO-RL	**0 / 0**	**0 / 0**	**0 / 0**	**0 / 1**	2 / 1
$\overline{P}_{\mathrm{E}}$	SPAR-RL-α	47.50 / 46.85	31.93 / 32.53	18.90 / 19.87	11.50 / 11.62	7.85 / 8.30
	PICO-RL	**48.18 / 47.85**	**38.62 / 37.88**	**28.33 / 28.85**	**19.00 / 20.35**	**11.80 / 12.52**

by -1) to ensure message extraction for both methods. The superior coding compatibility of PICO-RL highlights its better applicability. Table V also demonstrates the significant security improvement of PICO-RL in the case of actual embedding, matching the case of simulated embedding in Table III.

V. Conclusions

In this paper, we proposed PICO-RL, the first direct cost learning framework that yields payload-independent costs for image steganography. With better reward feedback from a SRNet-based environment, a novel OPA module was designed to not only ensure learning efficiency but also make the learned costs for different embedding payloads similar. Experimental results demonstrated that a well-trained PICO-RL model can serve all payloads with superior steganographic security and better coding compatibility, showing its excellent applicability. The PICO-RL framework has good scalability, since a more effective reward feedback mechanism as well as some advanced architectures of the policy and environment networks can be incorporated in future.

References

[1] J. Huang and Y. Q. Shi, "Reliable information bit hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 10, pp. 916–920, 2002.

[2] J. Fridrich, *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[3] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 705–720, 2010.

[4] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, 2011.

[5] W. Li, W. Zhang, L. Li, H. Zhou, and N. Yu, "Designing near-optimal steganographic codes in practice based on polar codes," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 3948–3962, 2020.

[6] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *IEEE Int. Conf. Image Process.* IEEE, 2014, pp. 4206–4210.

[7] W. Zhang, Z. Zhang, L. Zhang, H. Li, and N. Yu, "Decomposing joint distortion for adaptive steganography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2274–2280, 2016.

[8] W. Li, K. Chen, W. Zhang, H. Zhou, Y. Wang, and N. Yu, "Jpeg steganography with estimated side-information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2288–2294, 2019.

[9] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 221–234, 2016.

[10] W. Su, J. Ni, X. Hu, and J. Fridrich, "Image steganography with symmetric embedding using gaussian markov random field model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1001–1015, 2020.

[11] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.

[12] J. Yang, D. Ruan, J. Huang, X. Kang, and Y.-Q. Shi, "An embedding cost learning framework using gan," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 839–851, 2019.

[13] W. Tang, B. Li, M. Barni, J. Li, and J. Huang, "An automatic cost learning framework for image steganography using deep reinforcement learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 952–967, 2021.

[14] W. Tang, B. Li, M. Barni, J. Li, and J. Huang, "Improving cost learning for jpeg steganography by exploiting jpeg domain knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 4081–4095, 2022.

[15] S. Bernard, P. Bas, J. Klein, and T. Pevny, "Explicit optimization of min max steganographic game," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 812–823, 2020.

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

[17] S. Weng, M. Chen, L. Yu, and S. Sun, "Lightweight and effective deep image steganalysis network," *IEEE Signal Processing Letters*, vol. 29, pp. 1888–1892, 2022.

[18] Q. Guan, K. Chen, H. Chen, W. Zhang, and N. Yu, "Detecting steganography in jpeg images recompressed with the same quantization matrix," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 6002–6016, 2022.

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

[20] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *IEEE Int. Workshop. Inf. Forensics Security*. IEEE, 2014, pp. 48–53.

[21] P. Bas, T. Filler, and T. Pevnỳ, ""break our steganographic system": The ins and outs of organizing boss," in *Information Hiding*. Springer, 2011, pp. 59–70.

[22] R. Cogranne, Q. Giboulot, and P. Bas, "The alaska steganalysis challenge: A first step towards steganalysis," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2019, pp. 125–137.

[23] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, 2012.