A GRU OPERATIONS

$$\begin{aligned} \boldsymbol{z}_t &= \sigma\left(\mathrm{Conv}_{3\times 3}\left(\left[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\right], \boldsymbol{W}_z\right)\right) \\ \boldsymbol{r}_t &= \sigma\left(\mathrm{Conv}_{3\times 3}\left(\left[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\right], \boldsymbol{W}_r\right)\right) \\ \bar{\boldsymbol{h}}_{t-1} &= \tanh\left(\mathrm{Conv}_{3\times 3}\left(\left[\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}, \boldsymbol{x}_t\right], \boldsymbol{W}_h\right)\right) \\ \boldsymbol{h}_t &= (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \bar{\boldsymbol{h}}_{t-1} \end{aligned} \tag{3}$$

As described in the paper, the Gated Recurrent Unit (GRU) is an integral part of the LISO encoder. The operations of the GRU used in LISO are defined in Equation 3, where $\boldsymbol{x}_t$ is the input to the GRU in t^{th} iteration, $\boldsymbol{h}_t$ is the GRU's hidden state, and $\boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}_h$ are GRU's weight matrices. For LISO, $\boldsymbol{x}_t$ is the concatenation of features extracted from the input image, the gradient from the loss and the perturbation.

B LPIPS RESULTS

In Table 3 of the main paper, we used PSNR and SSIM to evaluate image similarity between the cover images and the corresponding steganographic images. In Table 6 we show the results of using (LPIPS) (Zhang et al., 2018), a newer perceptual image similarity metric, to evaluate image similarity betwenn cover and steganographic images. We observe similar trends as with PSNR and SSIM.

Dataset	Method	LPIPS ↓			
		1 bit	2 bits	3 bits	4 bits
Div2k	SteganoGAN	0.13	0.13	0.14	0.14
	LISO	0.04	0.05	0.06	0.08
	FNNS-D	0.01	0.02	0.08	0.07
	LISO+L-BFGS	0.04	0.05	0.07	0.08
CelebA	SteganoGAN	0.15	0.15	0.14	0.15
	LISO	0.09	0.08	0.12	0.15
	FNNS-D	0.07	0.08	0.15	0.15
	LISO+L-BFGS	0.06	0.06	0.11	0.14
MS COCO	SteganoGAN	0.12	0.13	0.13	0.13
	LISO	0.04	0.05	0.09	0.14
	FNNS-D	0.01	0.01	0.05	0.05
	LISO+L-BFGS	0.07	0.06	0.09	0.15

Table 6: Steganographic image quality measured with the LPIPS metric (Zhang et al., 2018). Lower numbers indicate better image quality.

C ERROR VS ITERATION

Figure 6 shows how the recovery error rate decreases every iteration. As we see, the error rate monotonically decreases and this implies that LISO learns a good descent direction. We also see that 15-20 iterations are enough for the error rate to converge.

D LOSS FOR DIFFERENT OPTIMIZATION METHODS

Figure 7 is analogous to figure 4 in the main paper. However, instead of seeing how the error rate decreases for different optimization methods, we see how the loss decreases. Note that the first data point we plot for each curve is after 1 iteration and we do so to improve the visualization. Similar to figure 4, we see that the loss for LISO optimization decreases much faster than any other method.

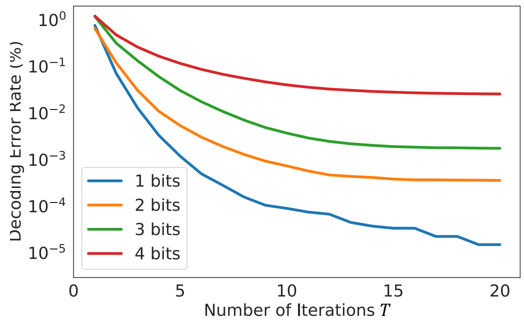

Figure 6: LISO recovery error rate with regard to number of iterations T under different bit rates, evaluated on Div2k's validation set. Y-axis is shown in log scale.

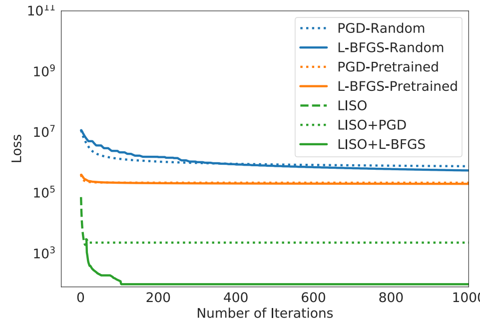

Figure 7: Loss-iteration curves for different optimization methods (at 4 bpp). The pre-trained network is SteganoGAN.

E Comparison with CHAT-GAN

Tan et al. (2021) present steganography results for hiding 1-4 bpp messages in images from the COCO dataset using CHAT-GAN. We train LISO using different λ weights to match the error rates of CHAT-GAN (as listed in their paper) as closely as possible and the results are presented in Table 7; we use $\lambda = 100$ for 1-2 bpp and $\lambda = 20$ for 3-4 bpp for training LISO. From the table we see that the error rates and SSIM values are comparable for both methods, but CHAT-GAN obtains better PSNR numbers. This is likely due to the fact that CHAT-GAN has a specially designed feature attention module to hide bits imperceptibly. However, as evidenced by both the SSIM values and the qualitative images below, steganographic images from LISO visually look indistinguishable from the cover images despite having lower PSNR values (PSNR is a local statistic and it measures the absolute mean squared difference in pixel values and two images can look similar despite have a low PSNR). Moreover, note that the contributions of our paper are orthogonal to the contributions of Tan et al. (2021). We can also combine our iterative LISO method with the CHAT-GAN encoder network architecture (Tan et al., 2021) to take advantage of both methods and to perform steganography with higher payloads and harder images.

Capacity	Method	Error (%) ↓	PSNR ↑	SSIM ↑
1 bpp	LISO	0.34	38.02	0.99
	CHAT-GAN	0.93	46.42	0.99
2 bpp	LISO	1.36	34.25	0.98
	CHAT-GAN	1.20	43.17	0.99
3 bpp	LISO	0.60	30.06	0.93
	CHAT-GAN	3.82	41.84	0.99
4 bpp	LISO	4.87	26.16	0.83
	CHAT-GAN	5.44	38.92	0.95

Table 7: Comparing the performance of LISO with CHAT-GAN

F Comparison of Different Steganography Methods

Table 8 compares LISO with both learned methods like SteganoGAN (Zhang et al., 2019) and CHAT-GAN (Tan et al., 2021), and optimization based methods methods like FNNS (Kishore et al., 2021). We see that LISO has a trained encoder like learned methods, but its learned encoder is iterative and emulates the optimization algorithm in optimization based steganography methods.

Method	Type	Encoder	Speed	Decoder
Learned Methods	Learned	Learned 1-step Network	Fast	Learned Network
FNNS-R (Kishore et al., 2021)	Optimization	L-BFGS / PGD	Slow	Random Network
FNNS-D (Kishore et al., 2021)	Optimization	L-BFGS / PGD	Slow	Fixed Pre-trained Network
LISO (ours)	Hybrid	Learned Iterative Network	Fast	Learned Network

Table 8: A comparison of how different learned and optimization based steganography methods differ from each other.

G LISO RESULTS WITH MSE LOSS WEIGHT $\lambda = 10$ FOR HIGHER IMAGE QUALITY

If higher image quality is desired, a higher mse loss weight λ can be used. Consequently, the image quality will be better but the error rate will be worse. We repeat the experiments in Table 1 of the main paper with mse loss weight $\lambda = 10$, and show the results in Table 9. As seen in the table, the error rate is slightly worse (it's still less than 1.5%), but the image quality is better. We can further increase the value of λ to obtain even better image quality.

Dataset	Method	Error Rate (%) ↓				PSNR ↑				SSIM ↑			
		1 bit	2 bits	3 bits	4 bits	1 bit	2 bits	3 bits	4 bits	1 bit	2 bits	3 bits	4 bits
Div2k	LISO	5.1E-4	1.0E-2	6.5E-2	3.8E-1	36.40	36.62	32.85	30.88	0.95	0.95	0.91	0.87
	LISO*	0	0	3.8E-6	5.8E-2	34.86	35.49	32.41	29.45	0.92	0.93	0.91	0.82
CelebA	LISO	1.5E-3	6.4E-3	4.5E-2	1.1E+0	35.81	36.80	35.19	32.94	0.90	0.92	0.89	0.84
	LISO*	0	0	1.3E-6	8.4E-1	35.13	35.91	34.35	32.89	0.89	0.91	0.88	0.84
MS COCO	LISO	1.8E-3	7.6E-3	1.6E-1	1.4E+0	34.32	36.43	31.46	30.14	0.91	0.95	0.86	0.81
	LISO*	0	0	1.2E-6	3.2E+0	33.20	35.23	30.35	29.57	0.90	0.93	0.83	0.78

Table 9: Image steganography results of LISO with loss weight $\lambda = 10$. Note that LISO* is LISO+L-BFGS and was shortened due to space constraints.

H ADDITIONAL STEGANALYSIS

In addition to SiaStegNet (You et al., 2020), we attempt to avoid detection from two other state-of-the-art steganalysis systems: SRNet (Boroumand et al., 2018) and XuNet (Xu et al., 2016). Results are reported in Table 10. For most cases, the results are similar to SiaStegNet. However, with XuNet we see a very high detection accuracy in the "w/o defense" scenario; this is because XuNet uses handcrafted kernels in addition to using a convolutional network to detect steganographic images. Despite this, adding the loss XuNet allows us to evade detection from it (as seen from the "w/ defense" scenario). We can also achieve lower detection accuracy in the "w/o defense" scenario if a method like XuNet (with hand-crafted kernels) is used as a discriminator during LISO training.

Method	Error Rate (%) ↓				PSNR ↑				Detection Accuracy Rate (%) ↓			
	1 bit	2 bits	3 bits	4 bits	1 bit	2 bits	3 bits	4 bits	1 bit	2 bits	3 bits	4 bits
SRNet (w/o defense)	1E-04	3E-04	5E-03	4E-02	33.83	34.15	30.08	24.58	51	40	33	74
SRNet (w/ defense)	6E-04	1E-04	1E-03	2E-01	33.43	32.74	28.51	24.89	0	0	0	1
XuNet (w/o defense)	1E-04	3E-04	5E-03	4E-02	33.83	34.15	30.08	24.58	100	98	100	100
XuNet (w/ defense)	2E-04	3E-03	1E-02	4E-02	34.16	32.98	28.40	25.31	2	2	42	100

Table 10: Steganalysis results with SRNet and XuNet. All experimental configurations follow Table 3 in main paper.

I EXPERIMENT RESULTS WITH HIGHER PAYLOADS

Extending the results from the main paper, in Table 11, we evaluate LISO with 5-6 bits encoded per pixel. We see that we are able to get low error rates ($< 3\%$) even for messages hiding 5-6bpp of information. However, the image quality is worse; as described in Appendix G we can get images with better image quality but slightly worse error rate if we increase the weight on the MSE loss term.

Dataset	Method	Error Rate (%) ↓		PSNR ↑		SSIM ↑	
		5 bits	6 bits	5 bits	6 bits	5 bits	6 bits
Div2k	SteganoGAN	31.44	35.35	20.05	20.34	0.79	0.80
	LISO	2.37	9.08	22.58	23.29	0.53	0.56
	FNNS-D	18.12	19.67	12.34	12.3	0.13	0.14
	LISO+L-BFGS	2.13	9.17	20.86	23.24	0.44	0.55
CelebA	SteganoGAN	32.15	31.16	19.51	21.82	0.74	0.79
	LISO	0.48	8.73	24.88	25.15	0.48	0.47
	FNNS-D	15.3	18.41	12.94	12.99	0.07	0.07
	LISO+L-BFGS	0.16	8.31	24.68	24.69	0.46	0.45
MS COCO	SteganoGAN	33.20	35.67	22.74	23.07	0.86	0.85
	LISO	1.09	8.66	21.32	22.63	0.36	0.45
	FNNS-D	16.41	17.44	15.63	15.78	0.19	0.20
	LISO+L-BFGS	1.52	8.50	21.08	22.55	0.35	0.45

Table 11: Image steganography results with 5-6 bits encoded per pixel.

J Additional Qualitative Results

A few qualitative examples of steganographic images were presented in the main paper. Additional steganographic images produced by LISO with 1-4bpp of hidden information are shown in Figure 8. Stenographic images obtained with an increased mse weight $\lambda = 10$ are shown in Figure 9.

As explained in the paper, we can use L-BFGS with LISO to further reduce the error rate obtained from LISO. Figure 10 shows steganographic images obtained from LISO+L-BFGS and we see that there is no difference in visual quality even after the additional optimization.

In the experiments section of the main paper we show that LISO can avoid being detected by SiaStegNet (You et al., 2020) by using the gradient from SiaStegNet in LISO's optimization network. In Figure 11, we show sample images from Div2k dataset that are generated in this way to avoid SiaStegNet detection. Again we see no noticeable difference in visual quality.

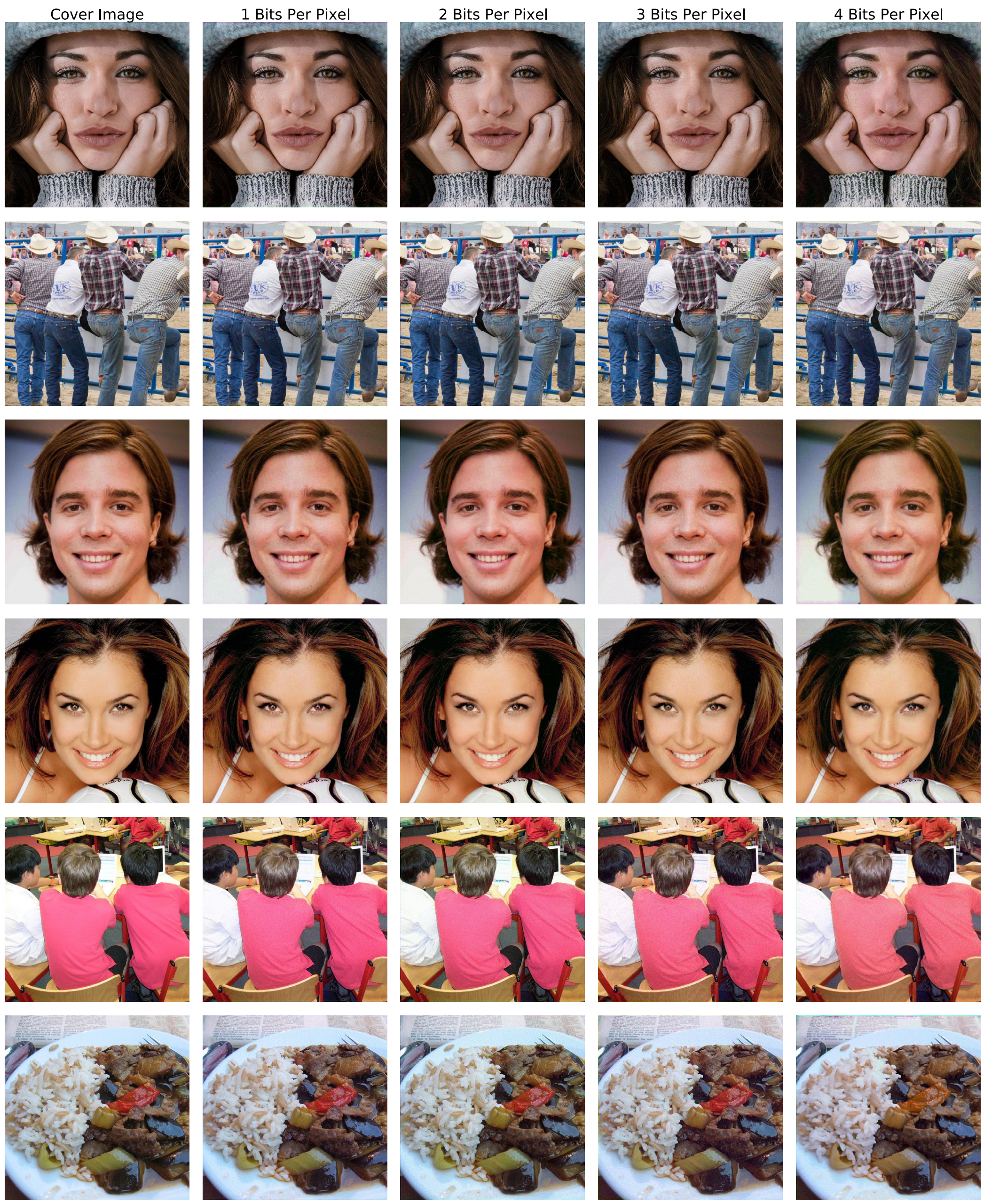

Figure 8: Cover images with corresponding steganographic images under different payloads. The first 2 images are from Div2k, the following 2 are from CelebA, and the last 2 are from MS-COCO.

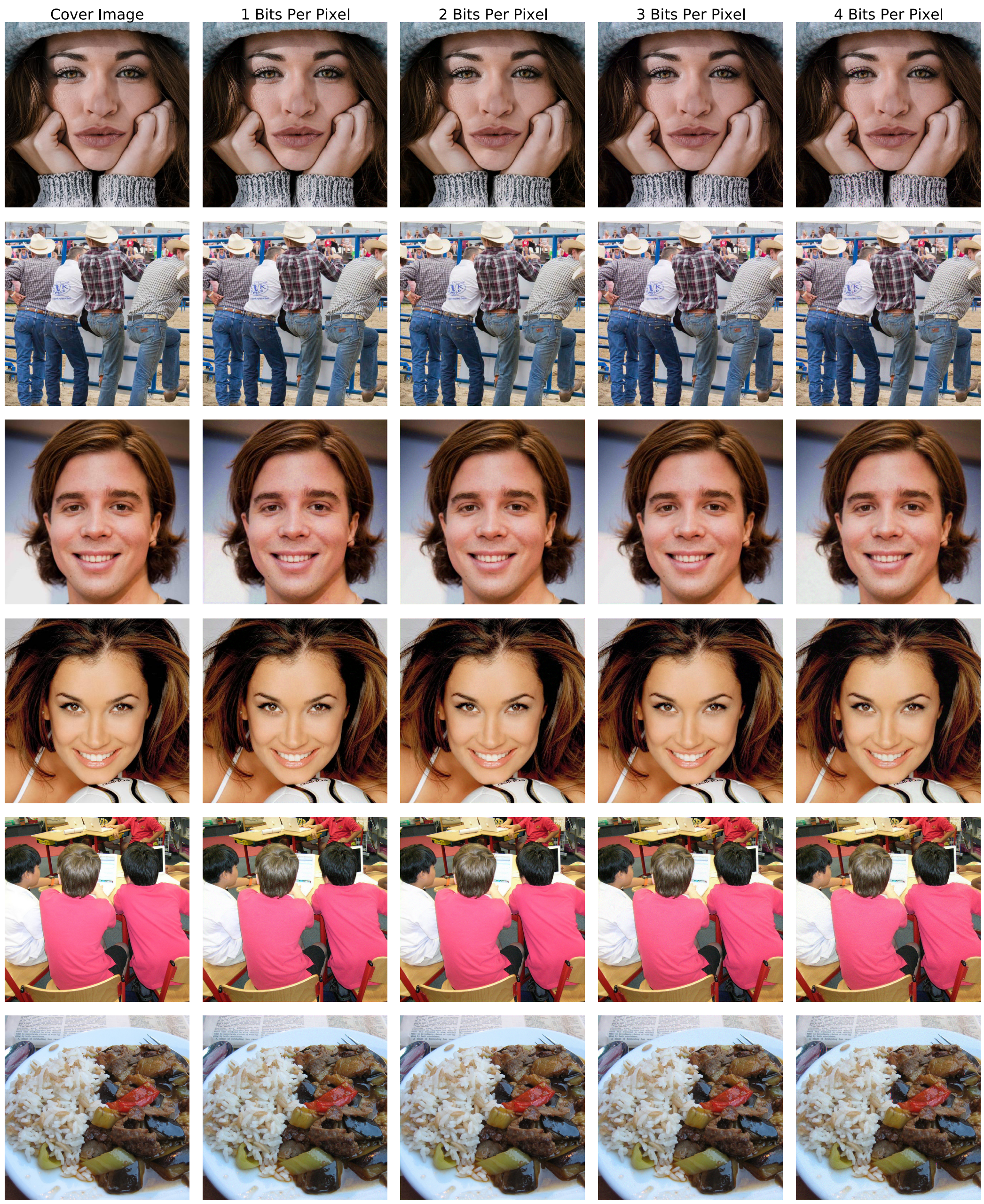

Figure 9: Cover images with corresponding steganographic images under different payloads trained with $\lambda = 10$.

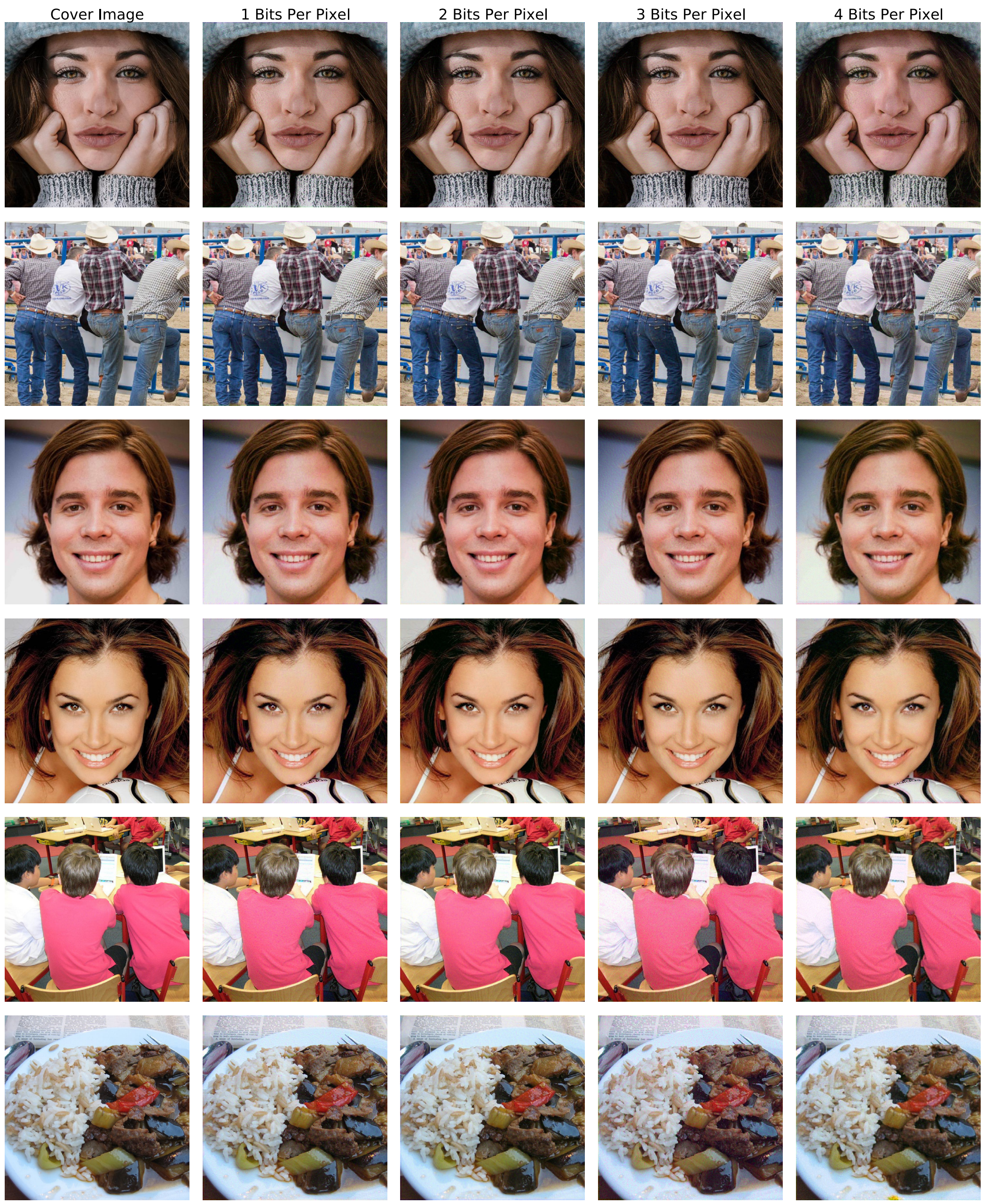

Figure 10: Cover images with corresponding steganographic images under different payloads using LISO+L-BFGS.

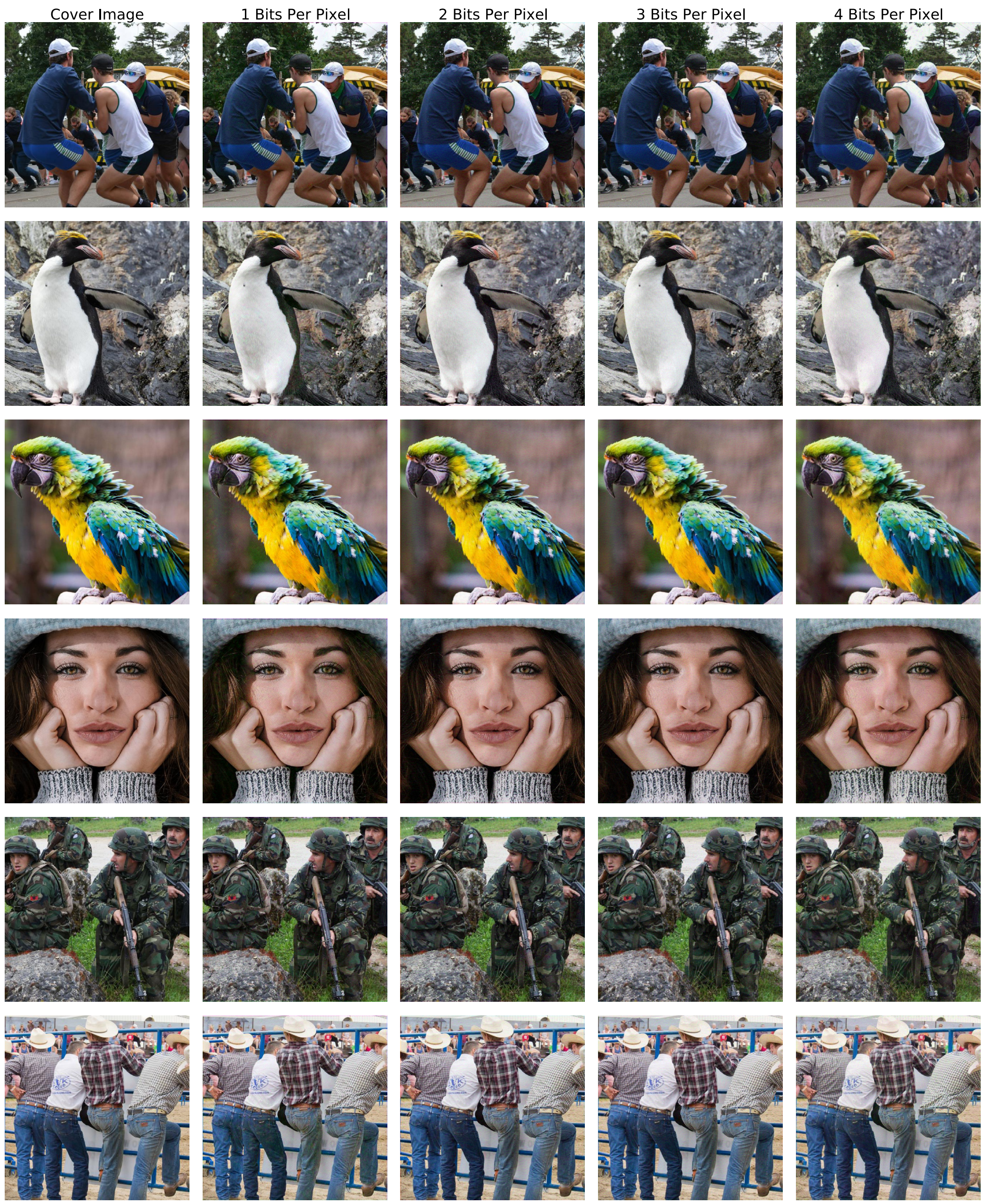

Figure 11: Steganographic produced by LISO with additional detection loss. These images can all avoid detection by SiaStegNet.