
MLEP: Multi-granularity Local Entropy Patterns for Generalized AI-generated Image Detection (Supplementary Materials)

Lin Yuan, Xiaowan Li, Yan Zhang*, Jiawei Zhang, Hongbo Li, Xinbo Gao*
Chongqing Key Laboratory of Image Cognition,
Chongqing University of Posts and Telecommunications, Chongqing 400065, China
yuanlin@cqupt.edu.cn, s230201063@stu.cqupt.edu.cn,
{yanzhang1991, zhangjw, lihongbo, gaodb}@cqupt.edu.cn

1 More Details about LEP Computation

The local entropy patterns (LEP) are computed across the entire image using a 2×2 sliding window. The resulted entropy can only take values from a finite set $\mathbb{V} = \{0, 0.8, 1.0, 1.5, 2.0\}$. We provide detailed calculation processes as follows:

Denoting a single 2×2 block as $X_{i,j}$. The probabilities distribution of the four pixels is denoted as \mathbb{P} . According to the standard definition of Shannon entropy [1], LEP calculation of a single block is given by:

$$\text{LEP}(X_{i,j}) = - \sum_{p \in \mathbb{P}} p \cdot \log_2(p). \quad (1)$$

Since a 2×2 image block contains only four pixels, the LEP computation for such a block can only take from the following five possible situations depending on their occurrences:

- All pixels are the same:

$$\begin{aligned} \mathbb{P} &= \{1\}, \\ \text{LEP}(X_{i,j}) &= -1 \cdot \log_2 1 = 0. \end{aligned}$$

- Three pixels are the same, one different:

$$\begin{aligned} \mathbb{P} &= \left\{\frac{1}{4}, \frac{3}{4}\right\}, \\ \text{LEP}(X_{i,j}) &= -\frac{1}{4} \cdot \log_2 \frac{1}{4} - \frac{3}{4} \cdot \log_2 \frac{3}{4} \approx 0.8. \end{aligned}$$

- Two pairs of pixels are the same:

$$\begin{aligned} \mathbb{P} &= \left\{\frac{1}{2}, \frac{1}{2}\right\}, \\ \text{LEP}(X_{i,j}) &= -2 \cdot \frac{1}{2} \cdot \log_2 \frac{1}{2} = 1.0. \end{aligned}$$

- Two pixels are the same, two different:

$$\begin{aligned} \mathbb{P} &= \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}, \\ \text{LEP}(X_{i,j}) &= -\frac{1}{2} \cdot \log_2 \frac{1}{2} - 2 \cdot \frac{1}{4} \cdot \log_2 \frac{1}{4} = 1.5. \end{aligned}$$

*Corresponding authors.

Table 1: Quantity of image samples in different datasets involved in the experiments.

Dataset			# True	# Fake
Training	ProGAN		72k	72k
Validation	ProGAN		800	800
Testing	GAN-Set	ProGAN	4k	4k
		StyleGAN	6k	6k
		StyleGAN2	8k	8k
		BigGAN	2k	2k
		CycleGAN	1.3k	1.3k
		StarGAN	2k	2k
		GauGAN	5k	5k
		AttGAN	2k	2k
		BEGAN	2k	2k
		CramerGAN	2k	2k
		InfoMaxGAN	2k	2k
		MMDGAN	2k	2k
		RelGAN	2k	2k
		S3GAN	2k	2k
		SNGAN	2k	2k
		STGAN	2k	2k
	Diffusion-Set	DDPM	1k	604
		IDDPM	1k	1k
		ADM	6k	6k
		LDM	1k	1k
		PNDM	1k	1k
		VQ-Diffusion	1k	1k
		SDv1	6k	11k
		SDv2	1k	1k
		DALL·E mini	1k	1k
		Glide-100-10	1k	1k
		Glide-100-27	1k	1k
		Glide-50-27	1k	1k
		LDM-200	1k	1k
		LDM-200-CFG	1k	1k
		Midjourney	2k	2k
		DALL·E 2	2k	2k

- All four pixels are different:

$$\mathbb{P} = \left\{ \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right\},$$

$$\text{LEP}(X_{i,j}) = -4 \cdot \frac{1}{4} \cdot \log_2 \frac{1}{4} = 2.0.$$

The computation of LEP is determined solely by whether the pixel values within the block are mutually identical or not, without considering the actual values of those pixels. Moreover, a block of four pixels results in only five situations of pixel occurrence. Therefore, we utilized a fast algorithm for LEP computation based on identifying from the above five situations for a given block using logical operations to check the equality of pixels, instead of executing the actual procession of Shannon entropy defined by Eq. 1. A Python script named `entropy.py`, which implements a function for fast LEP computation of an entire image without using a for loop, is provided as an attachment.

Table 2: Mean Acc. and A.P. over 8 additional generative models released recently.

Model	NPR[6]		ForgeLens[7]		D ³ [8]		VIBAIGC[9]		Ours	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
GPT-4o	100.0	<u>100.0</u>	47.9	97.2	<u>95.9</u>	99.6	54.9	83.4	100.0	100.0
Flux	94.2	98.9	53.2	91.5	<u>90.1</u>	<u>97.7</u>	47.8	43.3	84.8	96.7
firefly	93.4	<u>99.3</u>	90.9	99.7	78.9	94.2	86.2	94.5	<u>93.3</u>	97.9
SDXL	100.0	<u>100.0</u>	83.9	98.7	94.0	98.7	77.2	93.4	<u>99.8</u>	100.0
Midjourney-v5	99.4	<u>99.8</u>	64.6	89.7	95.8	99.4	52.0	52.4	<u>98.7</u>	100.0
SDv1.4	92.3	95.6	93.6	99.5	<u>97.4</u>	<u>99.8</u>	69.0	86.7	98.9	99.9
SDv1.5	92.0	95.5	93.0	99.2	<u>97.5</u>	<u>99.7</u>	68.7	86.3	98.7	99.8
Wukong	87.9	92.8	90.1	99.2	<u>97.1</u>	<u>99.6</u>	74.6	90.7	98.6	99.9
Mean	<u>94.9</u>	97.7	77.1	96.9	93.3	<u>98.6</u>	66.7	78.8	96.6	99.4

2 Details about Image Datasets

More details about the image datasets involved in our experiment (the main manuscript), specifically the number of positive and negative image samples in each subset are shown in Table 1.

3 Additional Experiments on Newly Released Datasets

We further evaluated our method on several recently released high-quality generative image datasets, including those produced by GPT-4o [2], Flux², SDv1.4, SDv1.5, Wukong, Firefly, SDXL, and Midjourney v5. Among them, SDv1.4, SDv1.5, and Wukong are sourced from GenImage [3], while Firefly, SDXL, and Midjourney v5 come from Synthbuster [4]. The results, summarized in Table 2, compare our method with several recent detection models using mean detection accuracy (Acc.) and average precision (AP), with each dataset containing balanced real and fake samples. As shown, our approach consistently achieves superior performance across most datasets (except for Flux), highlighting its strong generalization capability in detecting AI-generated content from previously unseen generators.

4 Additional Samples for Qualitative Analysis

More visualizations in the pixel, entropy and Fourier domains inspecting the difference between original and synthesized samples are shown in Fig. 1. More visualization samples for qualitative comparison between Zheng [5], NPR [6], and our method are shown in Fig. 2. More t-SNE visualizations of real vs. fake samples across extra generative models are shown in Fig. 3.

Moreover, Figs. 4, 5, and 6 visualize the MLEP feature maps for various types of images (real, GAN-generated, and Diffusion-generated) in comparison with their shuffled forms across different patch sizes. These visualizations highlight the semantic mitigation effect achieved by combining LEP features with patch shuffling, particularly when very small patch sizes are applied.

References

- [1] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [2] Z. Yan, J. Ye, W. Li, Z. Huang, S. Yuan, X. He, K. Lin, J. He, C. He, and L. Yuan, “GPT-Imgeval: A Comprehensive Benchmark for Diagnosing GPT4o in Image Generation,” *arXiv preprint arXiv:2504.02782*, 2025.
- [3] M. Zhu, H. Chen, Q. Yan, X. Huang, G. Lin, W. Li, Z. Tu, H. Hu, J. Hu, and Y. Wang, “GenImage: A Million-Scale Benchmark for Detecting AI-Generated Image,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 77 771–77 782, 2023.

²<https://huggingface.co/datasets/jackyhate/text-to-image-2M>

- [4] Q. Bammey, “Synthbuster: Towards Detection of Diffusion Model Generated Images,” *IEEE Open Journal of Signal Processing*, vol. 5, pp. 1–9, 2023.
- [5] C. Zheng, C. Lin, Z. Zhao, H. Wang, X. Guo, S. Liu, and C. Shen, “Breaking Semantic Artifacts for Generalized AI-generated Image Detection,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [6] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, “Rethinking the Up-Sampling Operations in CNN-Based Generative Network for Generalizable Deepfake Detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 130–28 139.
- [7] Y. Chen, L. Zhang, and Y. Niu, “Forgelens: Data-Efficient Forgery Focus for Generalizable Forgery Image Detection,” 2025.
- [8] Y. Yang, Z. Qian, Y. Zhu, O. Russakovsky, and Y. Wu, “D³: Scaling Up Deepfake Detection by Learning from Discrepancy,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 23 850–23 859.
- [9] H. Zhang, Q. He, X. Bi, W. Li, B. Liu, and B. Xiao, “Towards Universal AI-Generated Image Detection by Variational Information Bottleneck Network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 23 828–23 837.

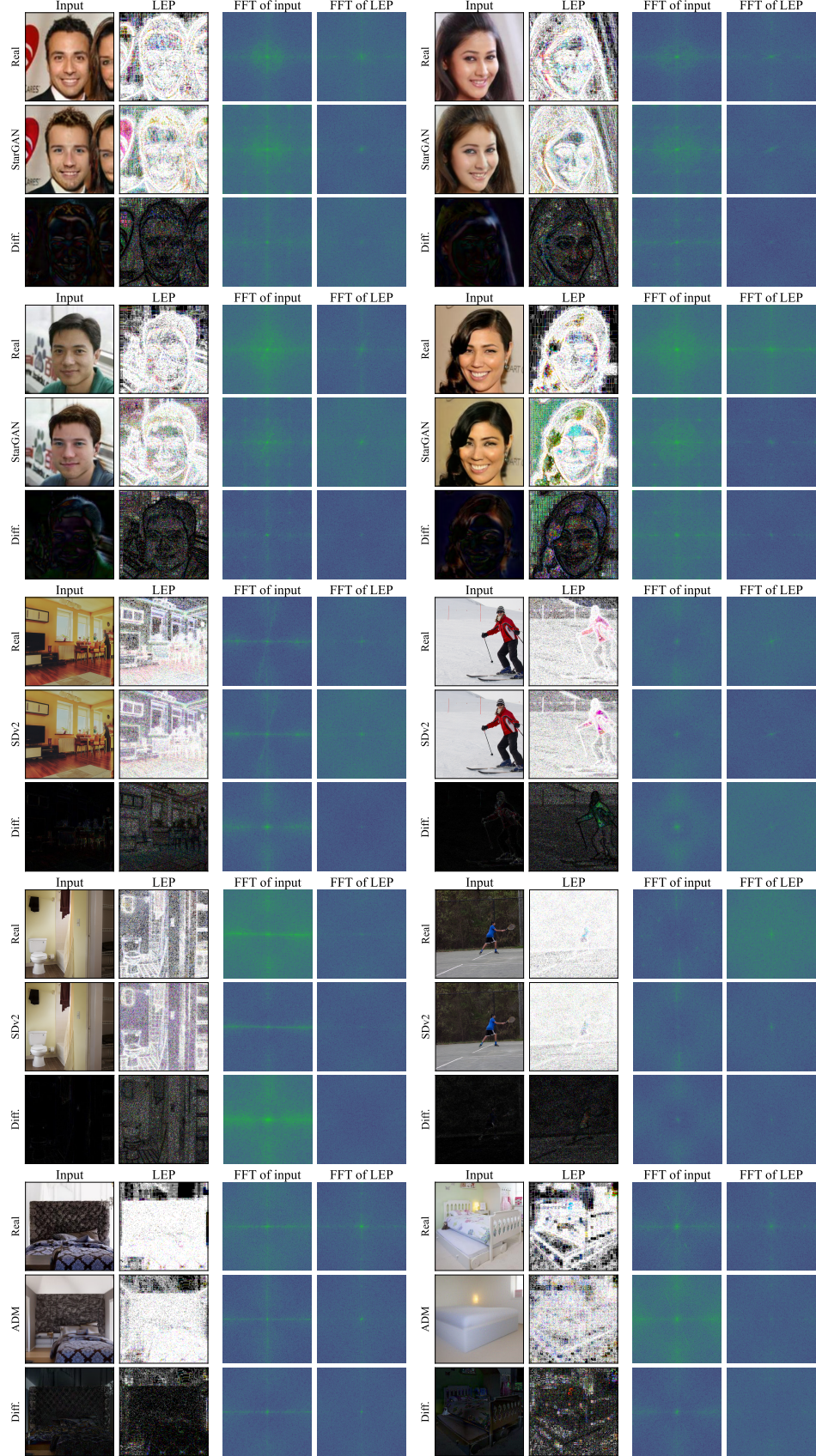


Figure 1: Visualization of local entropy patterns for additional real–fake image pairs, along with their differences in the pixel, entropy, and Fourier domains.

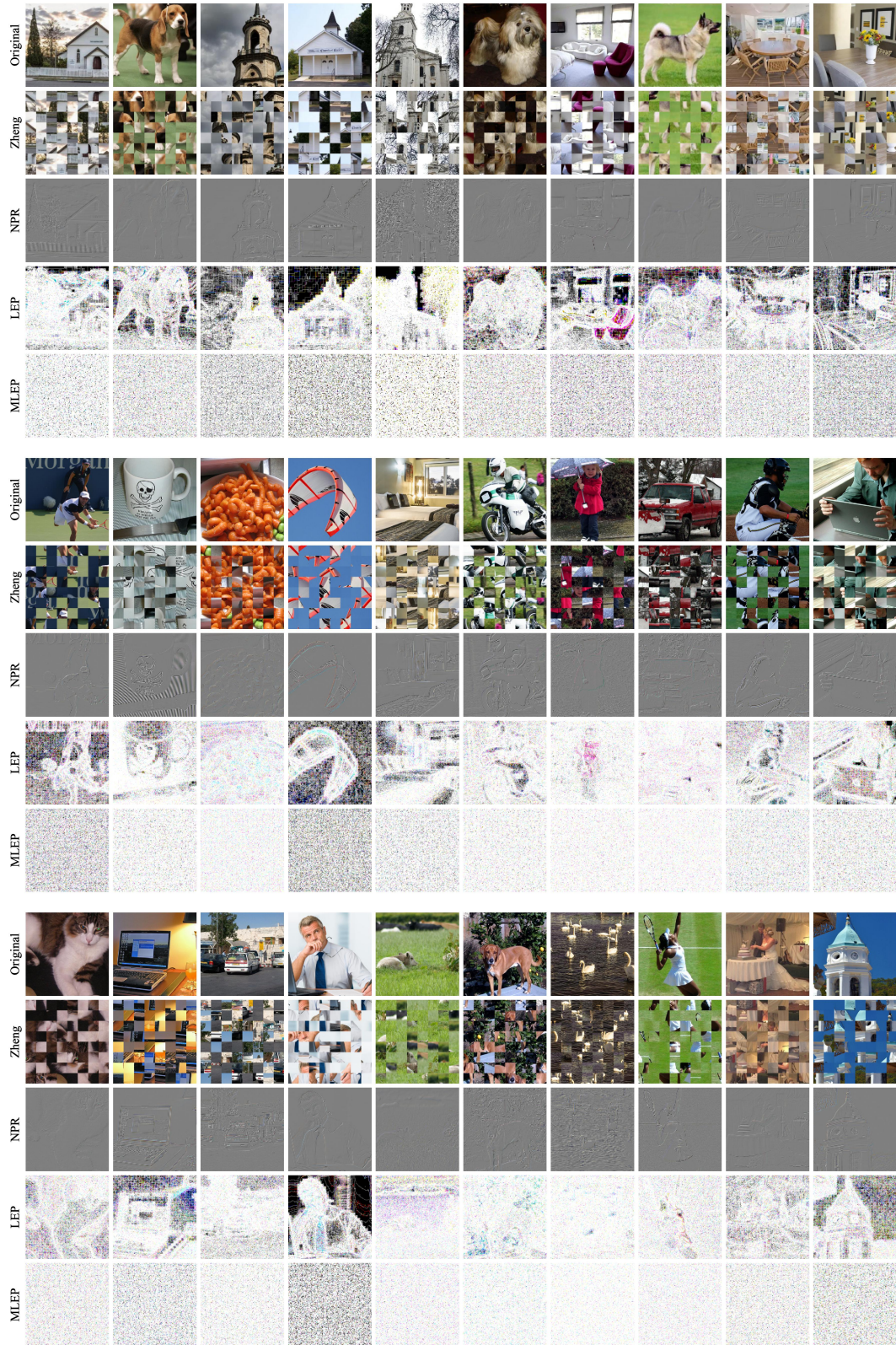


Figure 2: Qualitative comparison between Zheng [5], NPR [6], and our method. LEP preserves minimal visible semantics, while MLEP (without resampling) further suppresses semantic content.

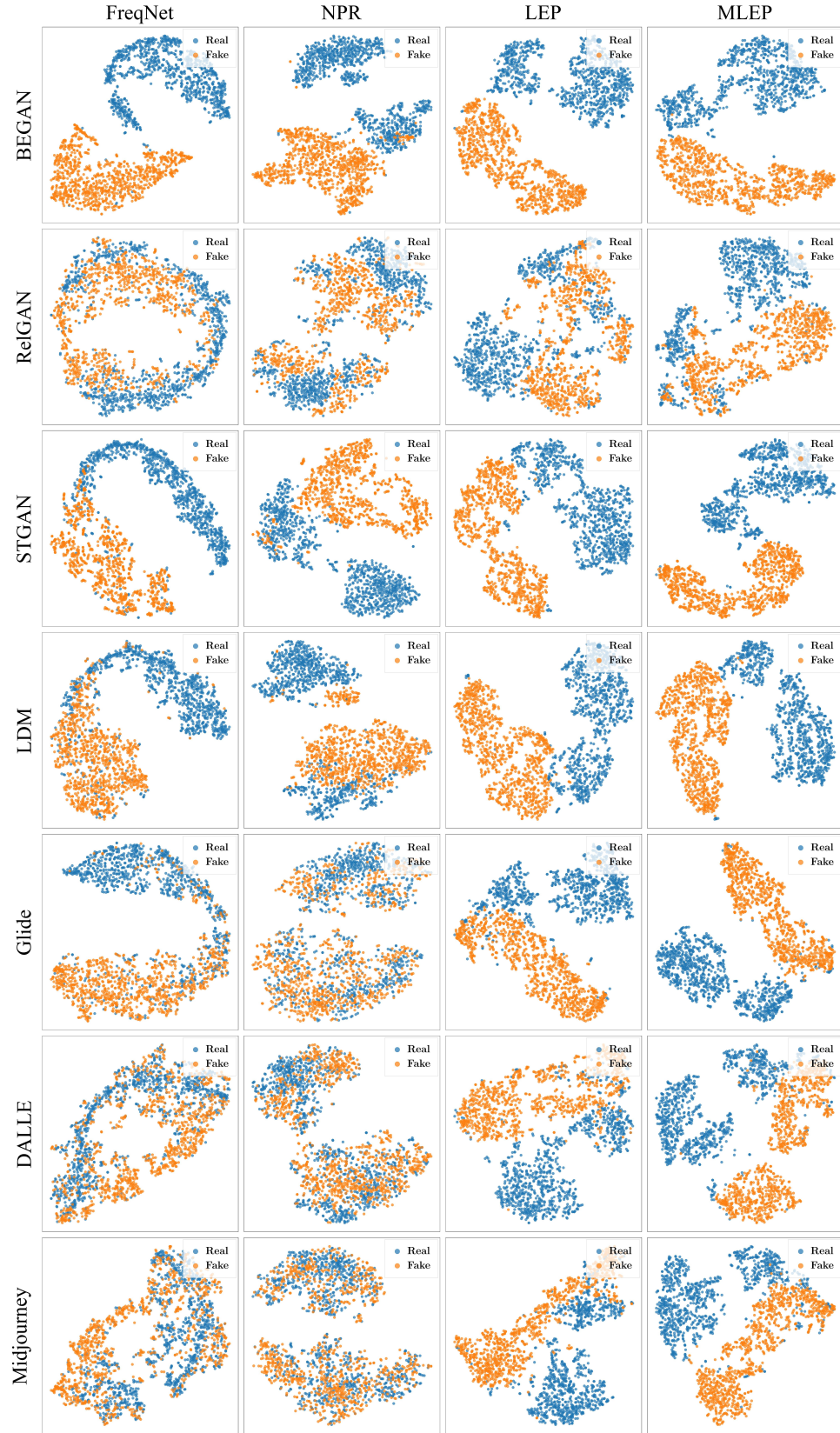


Figure 3: t-SNE visualization of real vs. fake samples across methods on extra generative models.

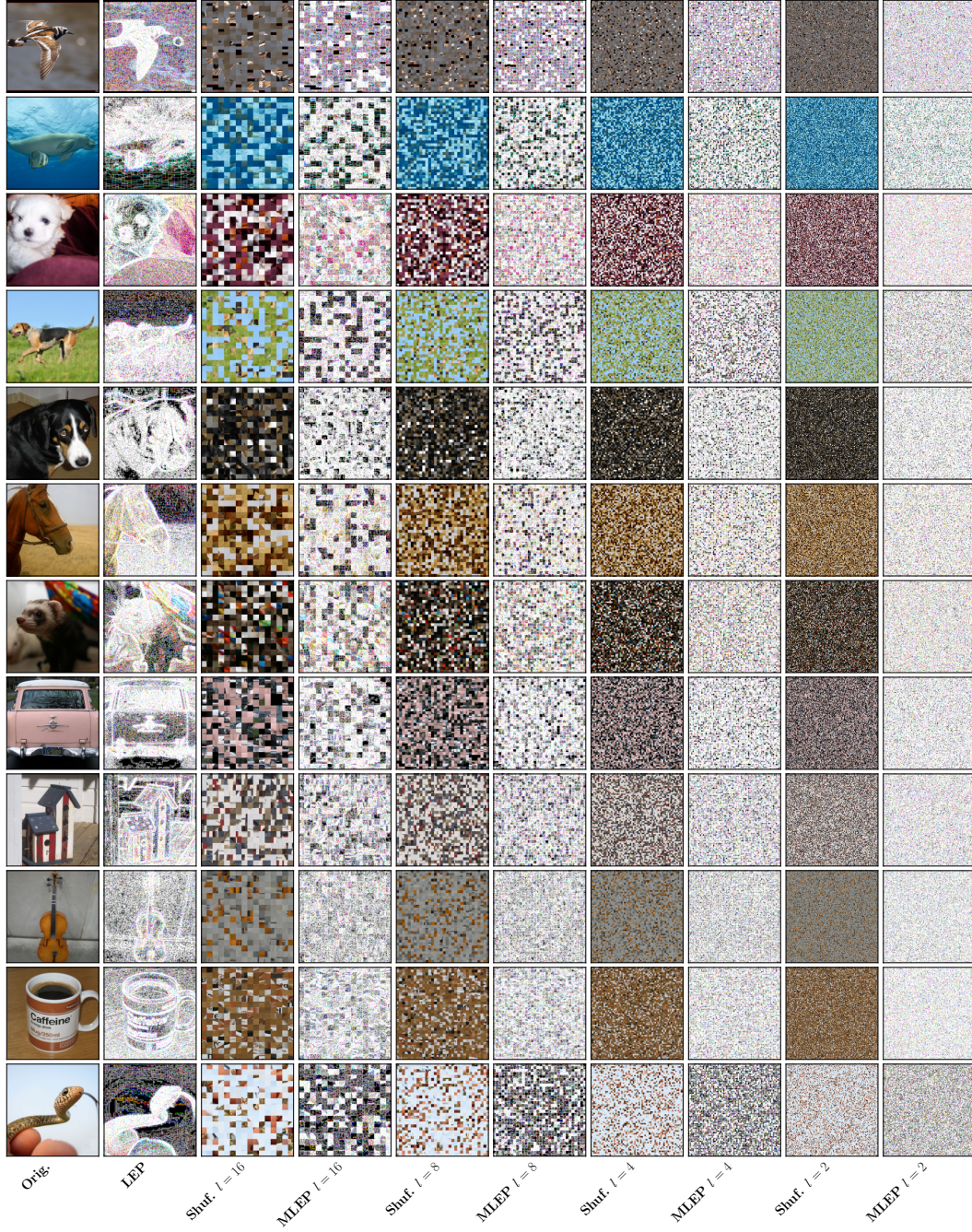


Figure 4: Visualization of MLEP for real images and their shuffled versions with varying patch sizes.

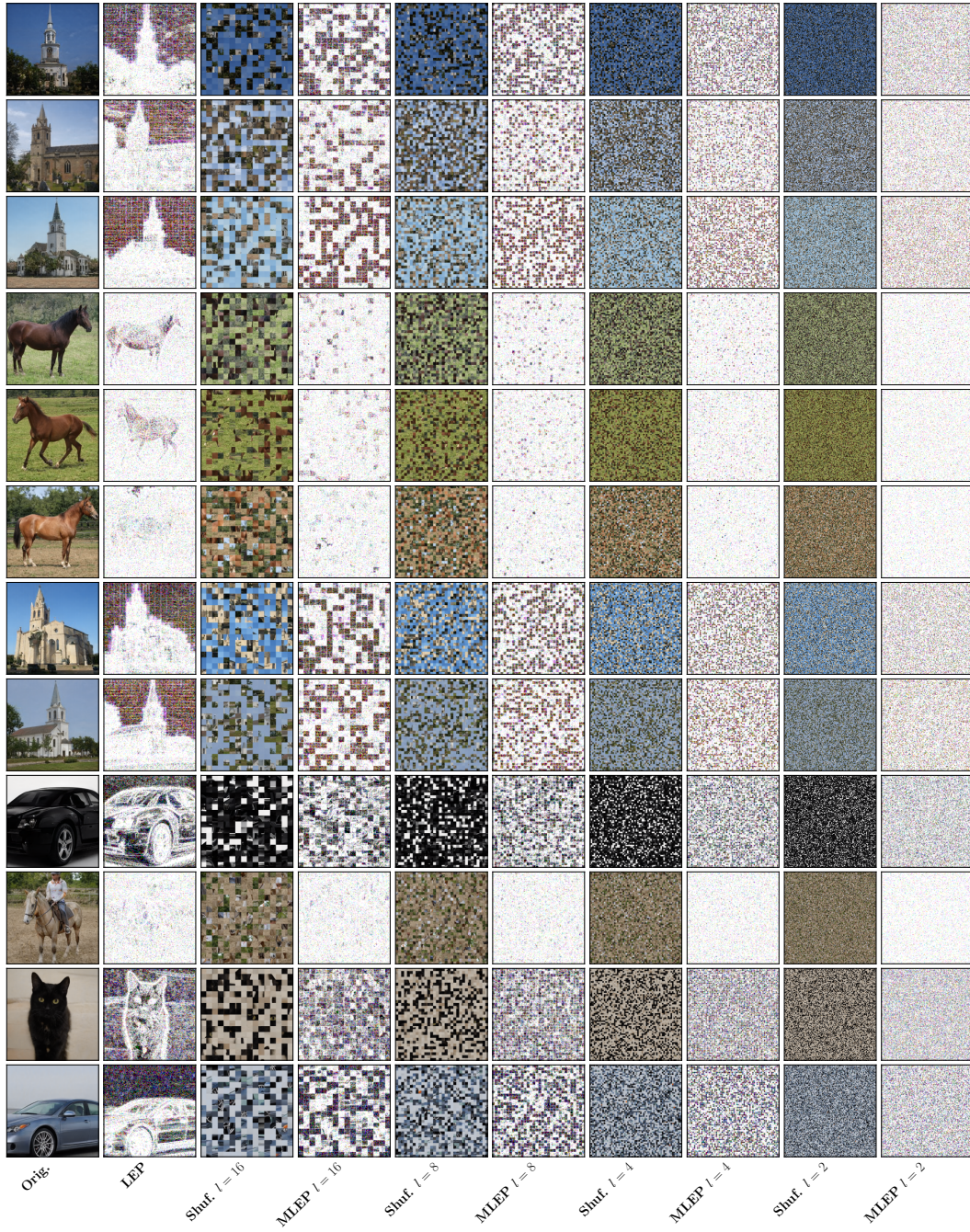


Figure 5: Visualization of MLEP for StyleGAN2 images and their shuffled versions with varying patch sizes.

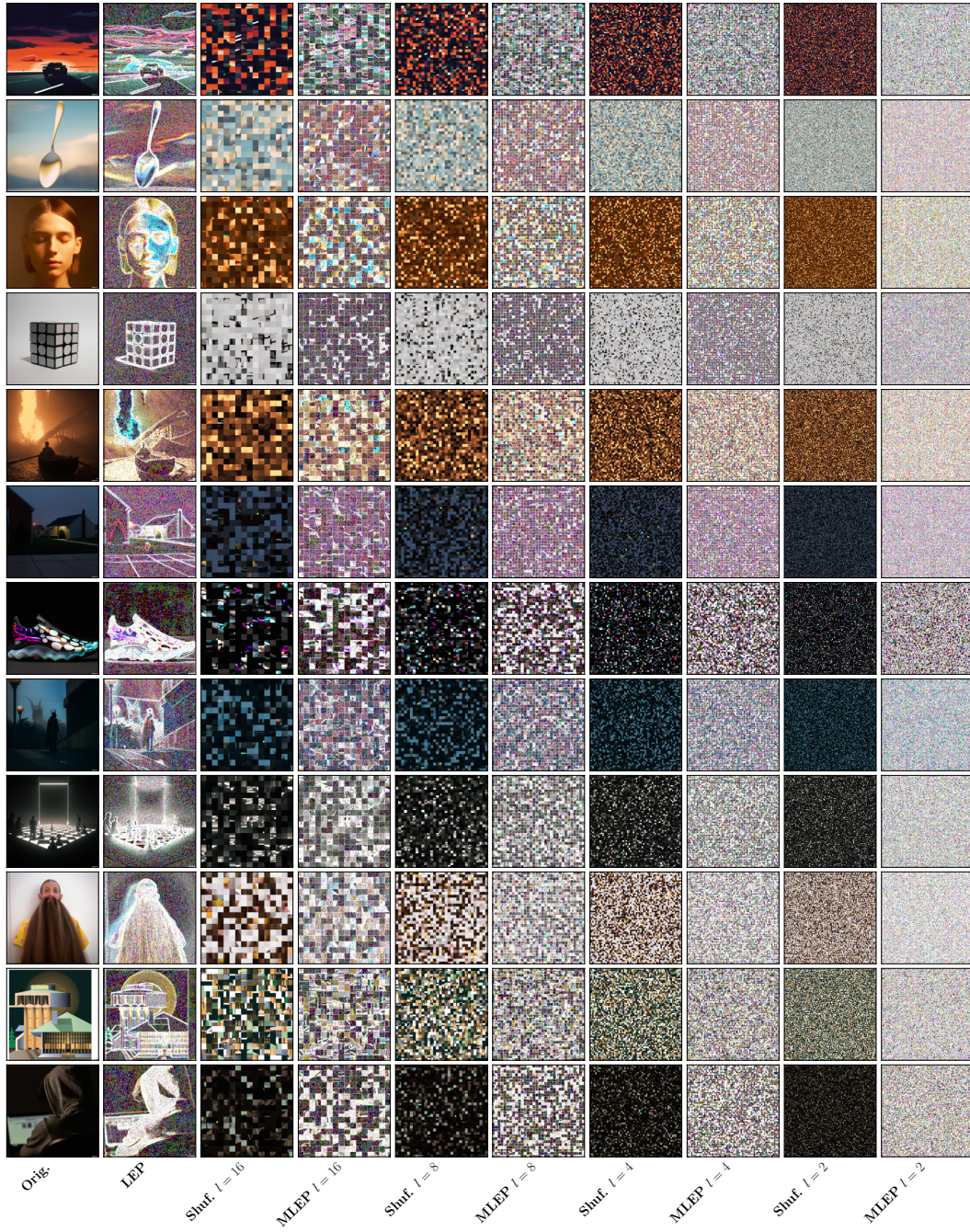


Figure 6: Visualization of MLEP for DALL-E 2 images and their shuffled versions with varying patch sizes.