PRADA: Probability-Ratio-Based Attribution and Detection of Autoregressive-Generated Images

Supplementary Material

A. Implementation Details

A.1. Models

HMAR [43] We set up HMAR according to the instructions in the official repository[3] and use the checkpoints `hmar-d20.pth` and `hmard30.pth`. We sample images at 256×256 pixels using the default configuration.

VAR [72] We set up VAR according to the instructions in the official repository[4] and use the checkpoints `vard20.pth` and `vard30.pth`. We sample images at 256×256 pixels using the default configuration.

LlamaGen [70] We set up LlamaGen according to the instructions in the official repository[5]. For LlamaGen-B and LlamaGen-L, we use the corresponding AR checkpoints `c2i_B_256.pt` and `c21_L_256.pt` with the VQ-VAE checkpoint `vq_ds16_c2i.pt` and sample images at 256×256 pixels using the default configuration. For LlamaGen-XL, we use the AR checkpoint `t2i_XL_stage2_512.pt` with the VQ-VAE checkpoint `vq_ds16_t2i.pt` and sample images at 512×512 pixels using the default configuration.

RAR [84] We set up RAR according to the instructions in the official repository[6] and use the checkpoints `rar_l.bin` and `rar_xxl.bin`. We sample images at 256×256 pixels using the default configuration.

Infinity [32] We set up Infinity according to the instructions in the official repository[7] and use the AR checkpoint `infinity_2b_reg.pth` with the VQ-VAE checkpoint `infinity_vae_d32reg.pth`. In contrast to the other models, Infinity does not return the (conditional and unconditional) likelihoods for each codebook entry, but for each bit of the token. We therefore compute a token's likelihood as the product of all individual bits' likelihoods.

Janus-Pro [7] We set up Janus-Pro according to the instructions in the official repository[8] and use the checkpoint `Janus-Pro-1B`. We sample images at 384×384 pixels using the default configuration.

Switti [75] We set up Switti according to the instructions in the official repository[9] and use the checkpoint `Switti-1024`. We sample images at 1024×1024 pixels using the default configuration.

A.2. Baselines

Corvi [13] We set up Corvi according to the instructions in the official repository[10] and use the provided checkpoint `Grag2021_latent/model_epoch_best.pth`, which was trained on LDM-generated images. The architecture is based on ResNet-50 [33] but avoids early downsampling to capture high-frequency features.

DRCT [6] We set up DRCT according to the instructions in the official repository[11] and use the provided checkpoint `DRCT-2M/sdv2/clip-ViT-L-14_224_drct_amp_crop/last_acc0.9112.pth`. DRCT fine-tunes a pre-trained foundation model (CLIP [57]) on pairs of real images and visually similar fake images, thereby enabling the model to focus on generative artifacts instead of semantic biases. Fake images are created by conditioning a diffusion model on real images.

[3] https://github.com/NVlabs/HMAR
[4] https://github.com/FoundationVision/VAR
[5] https://github.com/foundationvision/llamagen
[6] https://github.com/bytedance/1d-tokenizer
[7] https://github.com/FoundationVision/Infinity
[8] https://github.com/deepseek-ai/Janus
[9] https://github.com/yandex-research/switti
[10] https://github.com/grip-unina/DMimageDetection
[11] https://github.com/beibuwandeluori/DRCT

RINE [41] We set up RINE according to the instructions in the official repository[12] and use the provided checkpoint `model_ldm_trainable.pth`. Similar to other approaches, it used CLIP [57] for feature extraction. However, instead of relying exclusively on the output features, RINE uses representations from intermediate layers, which are weighted using a learned projection network.

RIGID [34] We set up RIGID according to the instructions in the official repository[13]. RIGID is a training-free method based on similarities in the feature space of a vision foundation models. We use the default foundation model (`dinov2_vitl14`) and leave the noise intensity (0.05) unchanged.

B-Free [30] We set up B-Free according to the instructions in the official repository[14] and use the provided checkpoint `BFREE_dino2reg4`. Similar to DRCT [6], the idea is to remove dataset biases by generating aligned real–fake image pairs before training a classification network. Unlike DRCT, B-Free uses DINOv2 [56] and inpainting-based augmentation.

D^3QE [87] We set up D^3QE according to the instructions in the official repository[15] and use the provided checkpoint `model_epoch_best.pth`. For a fair comparison, we adapt the dataset-pipeline to use our test sets, and we extend the validation script to extract the AUROC metric. The scores are then obtained from the official `eval.sh` script.

AEROBLADE* Similar to AEROBLADE [62], we obtain an image's reconstruction by passing it through the VQ-VAE's encoder and decoder. We then compute the reconstruction error as the spatial average over the second LPIPS layer, as proposed by the authors. Since the correct model should achieve a low reconstruction error, the final classification score is the largest negative error over all candidate models.

Quantization Error For next-token prediction models, the quantization error corresponds to the mean squared error (MSE) between continuous and quantized tokens. For next-scale prediction models, we leverage the multi-scale VQ-VAE's training objective and compute the quantization error as the MSE between the original feature map and upsampled reconstruction over all scales.

B. Example Images and PRADA Scores

In Figures 7 to 9 we show examples of real and corresponding generated images for all tested AR image generators. For text-to-image models, we also compare the ground-truth prompts (used for generation) with the prompts extracted by BLIP2 [45] that serve as semantic conditioning during likelihood extraction (see Figure 1). Moreover, under each image we report the PRADA scores for each model, illustrating how to perform attribution. An image is attributed to the model with the highest positive score or, if all scores are negative, predicted to be real or from an unknown model.

In Figure 6, we additionally provide visual examples for the perturbations (JPEG compression, center crop, blur, and noise) applied in our robustness analysis in Section 5.3.

[12] https://github.com/mever-team/rine
[13] https://github.com/IBM/RIGID
[14] https://github.com/grip-unina/B-Free
[15] https://github.com/Zhangyr2022/D3QE

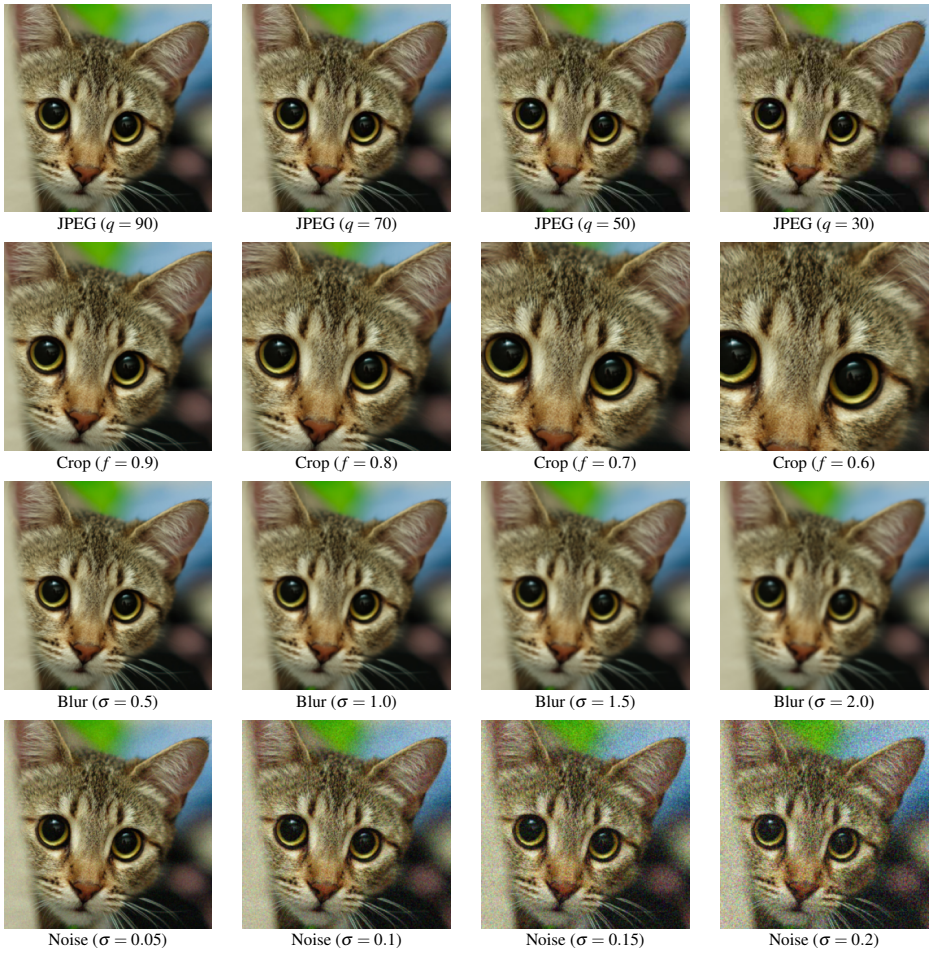

Figure 6. **Visualization of image perturbations used to analyze the robustness of the proposed method.** From top to bottom: JPEG compression, center crop, Gaussian blur and Gaussian noise at different strengths.

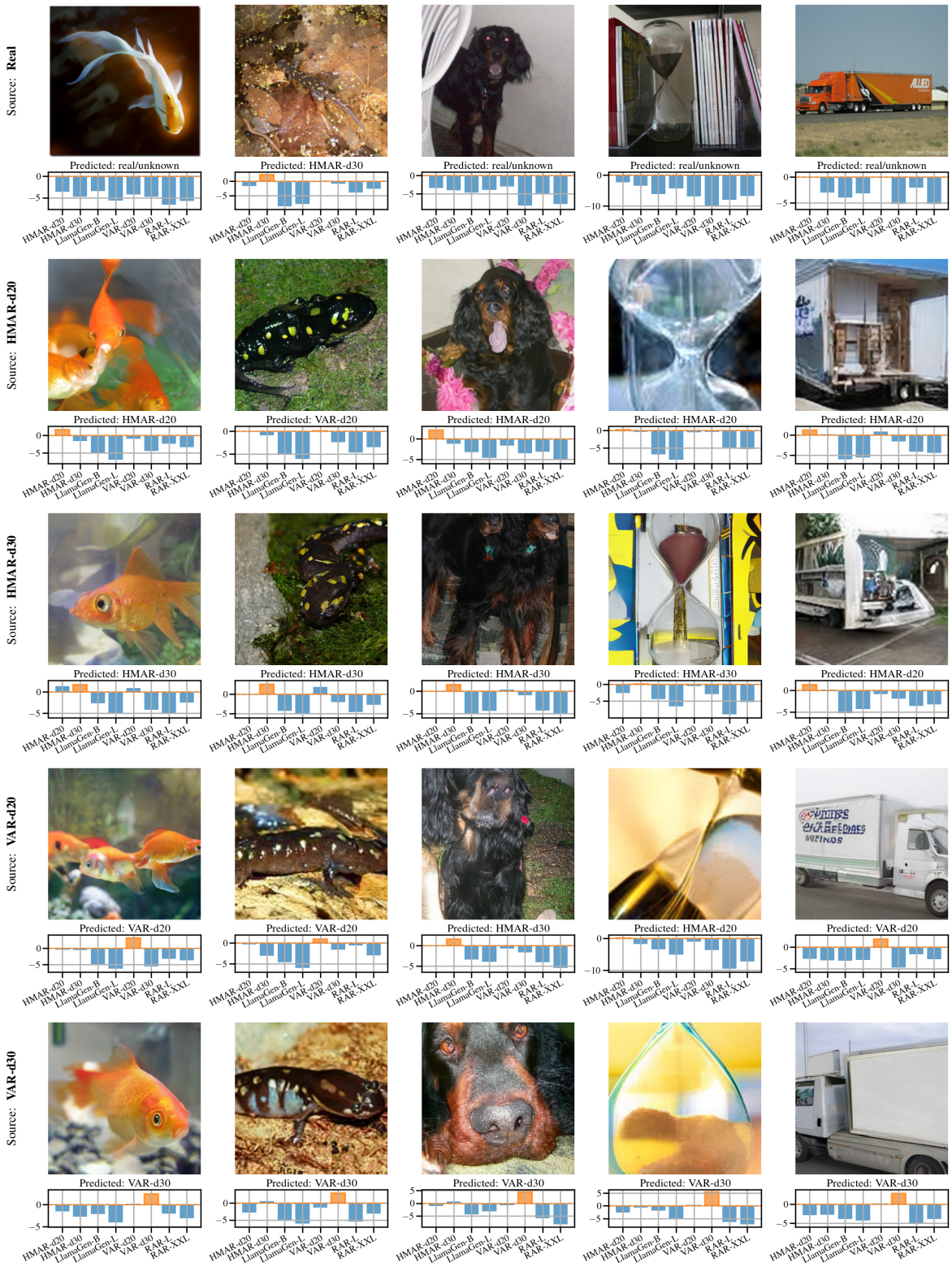

Figure 7. **Example images and PRADA scores for class-to-image models.** ImageNet class labels (from left to right): 1 (goldfish), 28 (spotted salamander), 214 (Gordon setter), 604 (hourglass), 675 (moving van).

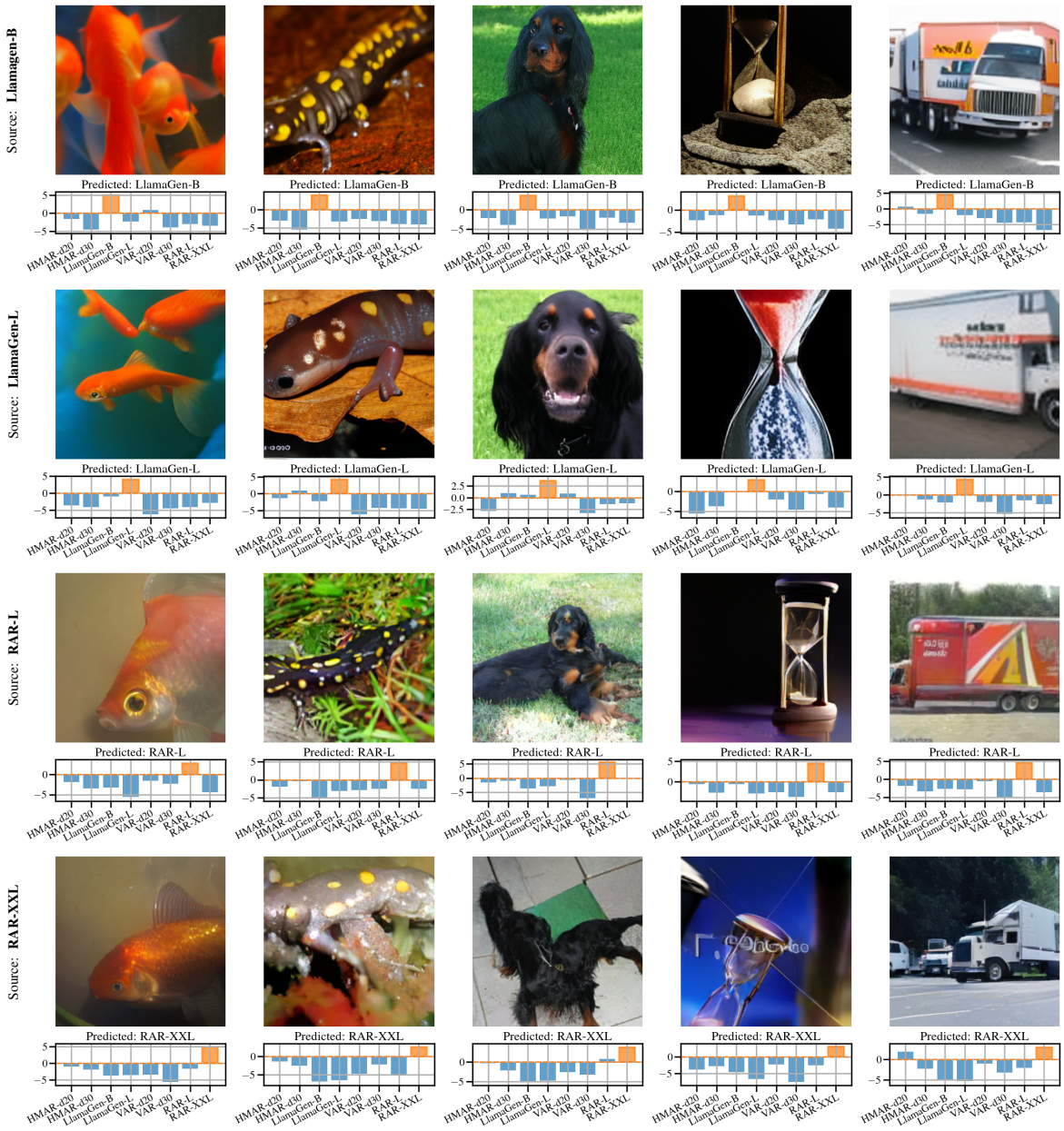

Figure 8. **Example images and PRADA scores for class-to-image models (continued).**

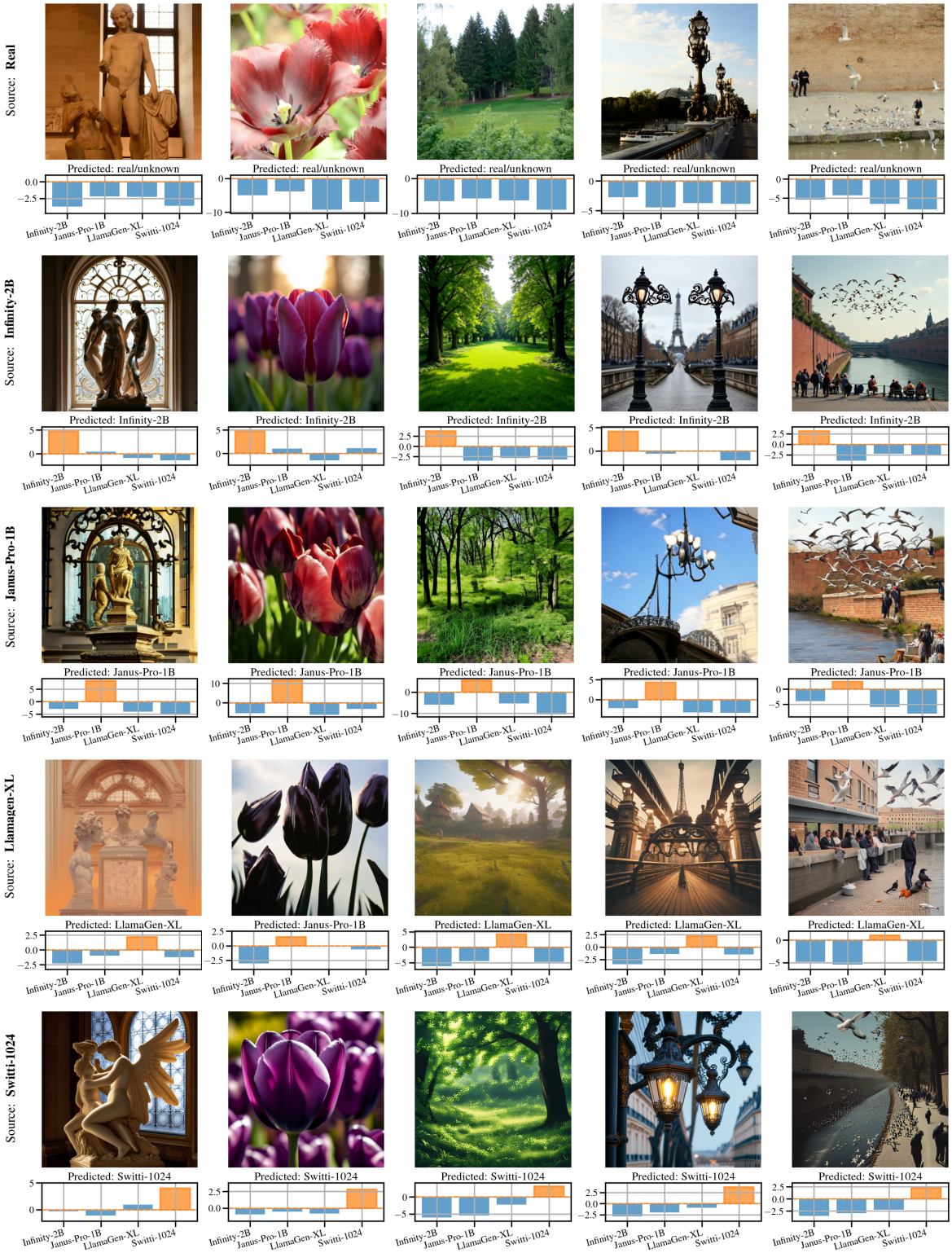

Figure 9. **Example images and PRADA scores for text-to-image models.** Generation prompts from Synthbuster [2] (from left to right): "statue of 'Eros et Psyché' in front of an ornamented glass window, in the style of museum, warm white balance", "close-up of dark purple tulips with large blooms high in the sun, soft-focus, 62mm f/4.8", "a green grass glade surrounded by trees with lots of foliage", "wrought iron bridge lamps in paris", "Flock of birds flying over a river next to a brick wall, pedestrians, editorial, seagull, people enjoying the show, at midday, people are eating, dressed in a worn". We provide the prompts extracted with BLIP2 in Table 3.

Table 3. **Ground truth and extracted prompts for real and generated images, corresponding to Figure 9.** The first row contains the prompts that were used to generate the images in Figure 9. They are part of the Synthbuster dataset [2] and were optimized to reproduce the corresponding real images from RAISE-1k [17]. The remaining rows show the prompts extracted using BLIP2 [45] that were used as conditioning c when computing the likelihoods for the PRADA score (see Figure 1).

Ground Truth	statue of 'Eros et Psyché' in front of an ornamented glass window, in the style of museum, warm white balance	close-up of dark purple tulips with large blooms high in the sun, soft-focus, 62mm f/4.8	a green grass glade surrounded by trees with lots of foliage	wrought iron bridge lamps in paris	Flock of birds flying over a river next to a brick wall, pedestrians, editorial, seagull, people enjoying the show, at midday, people are eating, dressed in a worn.
Real	a statue of a man and a woman	a close up of some red tulips	a man is riding a horse in a field	a bridge with a railing	a group of people are taking pictures of seagulls
Infinity-2B	three statues in front of a window	purple tulips in the sun	a green field with trees in the background	two street lamps in front of the eiffel tower	people sitting on benches near a river with birds flying over them
Janus-Pro-1B	a statue of a woman and a child in front of a window	tulips in bloom with sunlight shining on them	a green forest with trees and grass	a chandelier is hanging in the middle of a building	a group of people are sitting on a boat with birds flying over them
LlamaGen-XL	a statue of a man in front of a building	a group of black tulips in the sun	a grassy field with trees and houses	a woman is walking on a bridge with a clock tower	a group of people
Switti-1024	a statue of a cupid and a woman in front of a window	purple tulips in the garden	a green forest with trees and flowers	two street lamps on a bridge	a group of people walking along a river

C. Visualizations of PRADA Score Functions

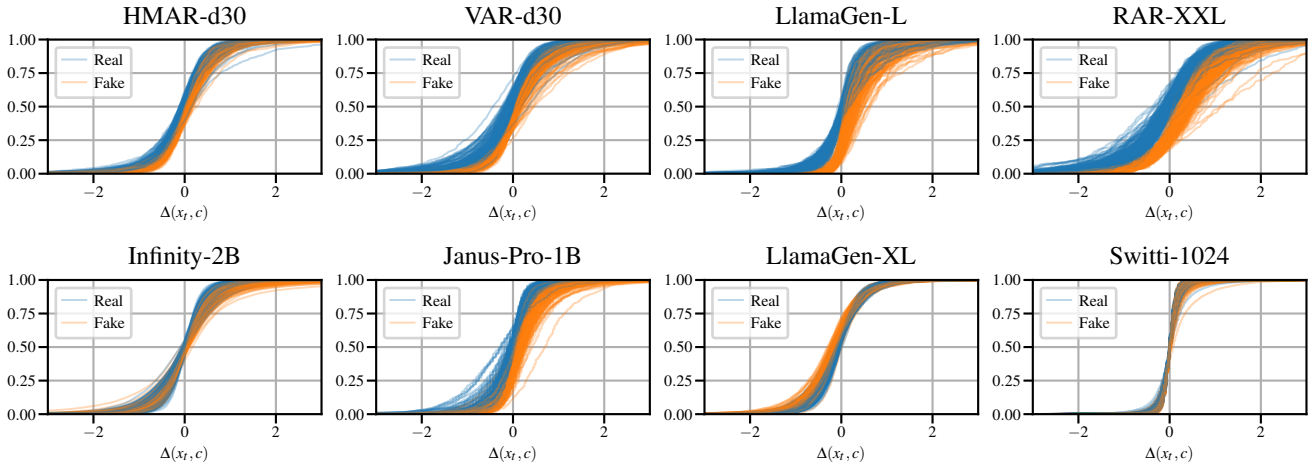

Figure 10. **Cumulative distributions of the log-probability ratio $\Delta(x_t, c)$ over tokens for 200 random images and selected models.** For all models, the log-probability ratio shows differences between real and generated images, but their differences are not uniform over different models.

In this section, we visualize PRADA's score function for different AR image generators and show how the learned parameters can help to understand the model's probability distribution.

We begin by reviewing why a uniform score function, such as ICAS, is not suitable for all models. Figure 10 shows the cumulative distributions for probability ratios $\Delta(x_t, c)$ over tokens (for 100 images). For HMAR-d30, VAR-d30, and LlamaGen-L, we see that real samples can be distinguished from generated samples primarily by using low values for $\Delta(x_t, c)$. This agrees with the motivation for the ICAS score, Equation (2), introduced in [82], which is designed to amplify those deviations: ICAS assigns large negative score values to negative values of $\Delta(x_t, c)$ (see Figure 11). At the same time it attenuates large positive $\Delta(x_t, c)$, because such are observed for both real and generated images (or non-members and members in MIAs) and are thus not helpful for distinguishing them. Infinity-2B and LlamaGen-XL, however, show a completely different picture. For these models, large negative $\Delta(x_t, c)$ are mostly observed for generated images. Real samples can still be distinguished from generated ones, as their probability ratios cluster around zero. To tell apart real and fake images, it requires a vastly different token-wise scoring.

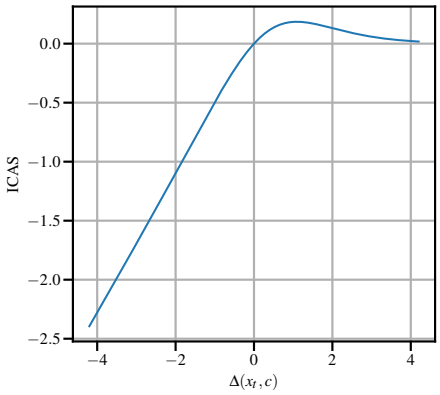

Figure 11. **Token-wise scoring used by ICAS.** In several scenarios, PRADA recovers a scoring function resembling the ICAS scoring function.

This observation is the key motivation to *learn* a suitable score function from a small training set. By inspecting the token-wise scoring $f_\theta : \mathbb{R} \to \mathbb{R}$, the ratio balancing parameter $\alpha \in \mathbb{R}$, and the scale weights $w \in \mathbb{R}^S$ for a learned PRADA score function, we can also interpret the characteristics of the likelihood distribution for that particular model. Figures 12 to 14 visualize the learned PRADA score functions for class-to-image and text-to-image models, respectively.

The **first column** shows the cumulative distributions for $\Delta^\alpha(x_t, c)$. Comparing them to the cumulative distributions for $\Delta(x_t, c)$ in Figure 10, we observe that if $\alpha \not\approx 1$, real and generated images are separated more clearly with α-balancing applied.

The **second column** shows the token-wise scoring f_θ for different values of $\Delta^\alpha(x_t, c)$. If $\alpha \approx 1$, this can be compared to ICAS's score function (see Figure 11). Each scoring allows us to interpret how PRADA assigns score values to α-balanced probability ratios for tokens x_t. For example, Figure 13 shows that the calibrated f_θ reproduces a score function similar to ICAS for VAR-d20, VAR-d30, RAR-L and RAR-XXL. In contrast, for Janus-Pro-1B in Figure 14, generated images can be distinguished from real ones by large values of $\Delta^\alpha(x_t, c)$, leading to a score function that assigns large scores to large

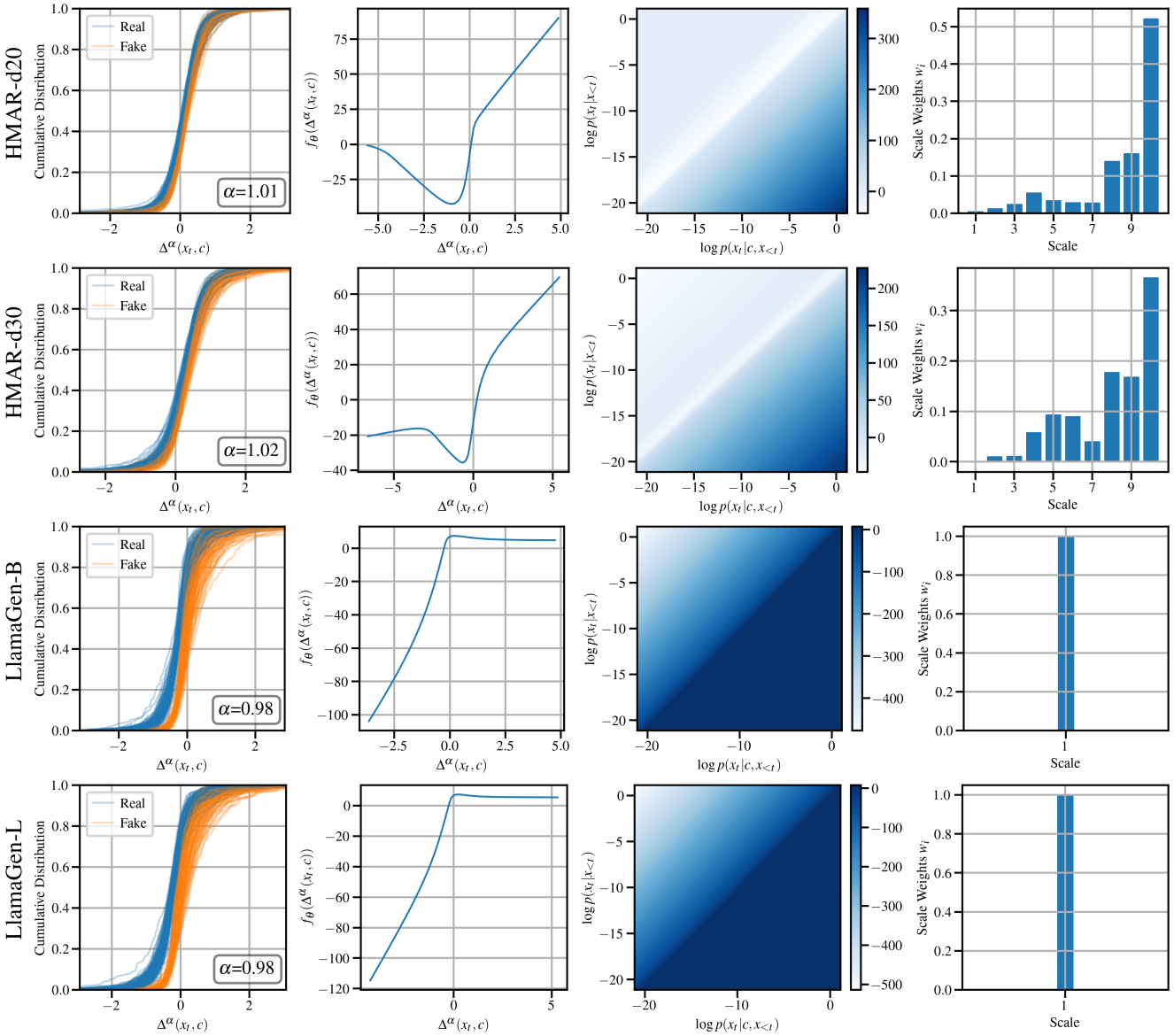

Figure 12. **PRADA scoring for class-to-image models.** The first column visualizes the differences in the cumulative distributions of α-balanced probability ratios $\Delta^\alpha(x_t, c)$ over tokens for 100 random images, which are exploited by the scoring $f_\theta(x_t, c)$ in the second column. The third column visualizes $f_\theta(x_t, c)$ as function of conditional and unconditional likelihoods. The last column visualizes scale weights, showing how average scores of individual scales are linearly combined to form the PRADA score of the model.

$\Delta^\alpha(x_t, c)$ instead of a score close to zero. Further, for LLamaGen-XL, the cumulative distributions even reverse the roles of real and generated images, which leads to a decreasing score function, assigning low negative scores to large positive $\Delta^\alpha(x_t, c)$.

The **third column** visualizes the scoring as a function of conditional and unconditional probabilities. This is in particular interesting if $\alpha \neq 1$. For Infinity-2B, we can observe how the score has a larger dependence on the conditional probability (with $\alpha < 1$) than on the unconditional. The resulting scoring is once again similar to the ICAS scoring. A similar balancing parameter α and scoring is learned for Switti-1024, just that the positive scores are located around $\Delta^\alpha(x_t, c) \approx -2$, which resonates well with the higher occurrence of real tokens in that range (see bottom right plot of Figure 16 in the next section).

Finally, the **fourth column** visualizes the scale weightings (for next-scale prediction AR image generators). Overall, we observe that the later scales receive higher weights. This is expected as later scales contain far more tokens, yielding a

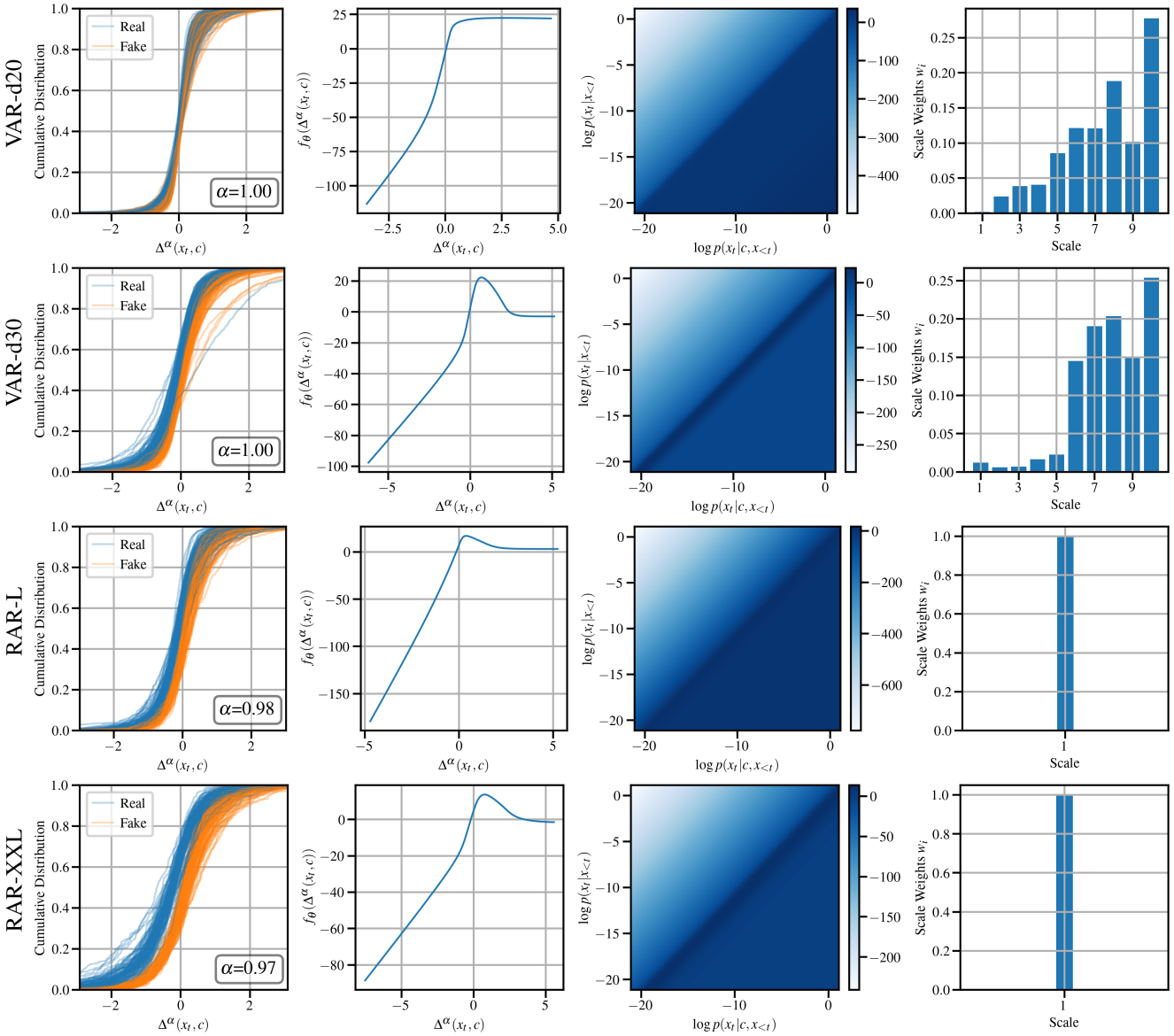

Figure 13. **PRADA scoring for class-to-image models (continued).**

stronger and more stable signal, and therefore a higher signal-to-noise ratio, than the earlier, coarser scales. As weight is still distributed across many scales, we also conclude that a single learnable function f_θ can indeed work well across different scales.

It remains to analyze a notable exception, namely Infinity-2B. Here, the finest scales receive the largest weights, but the second-to-last scale is assigned a *negative* optimal weight. This implies that, at that scale, real and generated images partially swap their likelihood-ratio behavior: generated samples tend to produce lower values than real ones. As seen in Figure 16 (in the next section, lower left plot), this inversion occurs mainly for the early tokens of the second-to-last scale in Infinity-2B. Further analysis of the final scales reveals that some real-image tokens cluster around $\Delta^\alpha(x_t, c) \approx 0$, even though values around $[-3, -10]$ would be expected. Figure 15 shows this distribution for the early tokens of the last two scales. Interestingly, only some real images tend to show this anomaly, generated images do not. As this clustering $\Delta^\alpha(x_t, c) \approx 0$ occurs jointly, for scale 12 *and* 13, a negative scale weight can be used to 'cancel' the negative effect of unusually high likelihood ratios produced by those real tokens. This correction does not degrade overall performance, but improves it (and is thus beneficial).

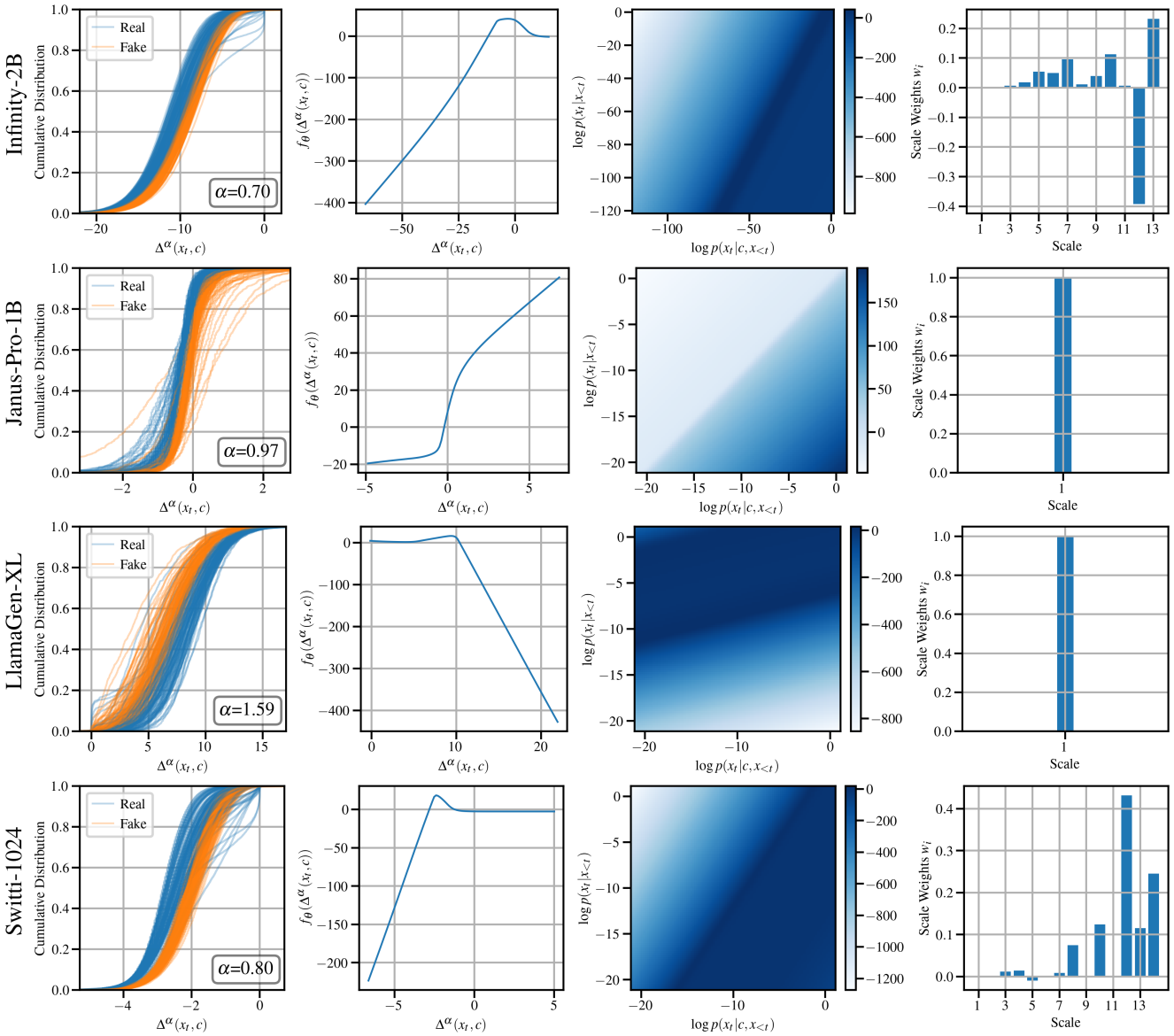

Figure 14. **PRADA scoring for text-to-image models.** For additional context see the caption of Figure 12.

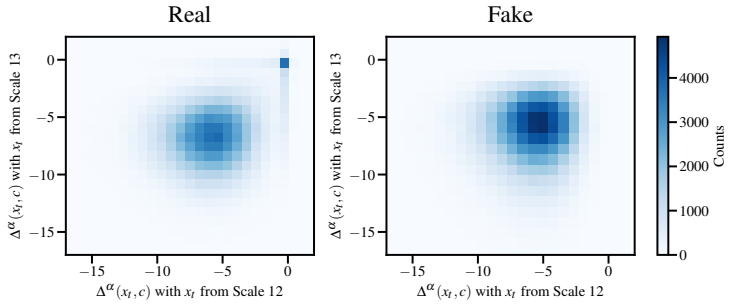

Figure 15. **Cancellation effect observed in the last two scales of Infinity-2B.** We show heatmaps of $\Delta^\alpha(x_t, c)$ for scale 12 vs. scale 13, for real and generates images (first 250 tokens per scale, over 1000 real and 1000 fake images). We observe that some tokens from real images cluster around $\Delta^\alpha(x_t, c) \approx 0$ (left plot, top right corner). The negative scale weight w_{12} effectively eliminates the detrimental impacts caused by this anomaly (usually, tokens from real images tend to have slightly *lower* scores compared to tokens from generated images). This anomaly seems to happen only for some real images, not for generated ones.

D. Scale Behavior

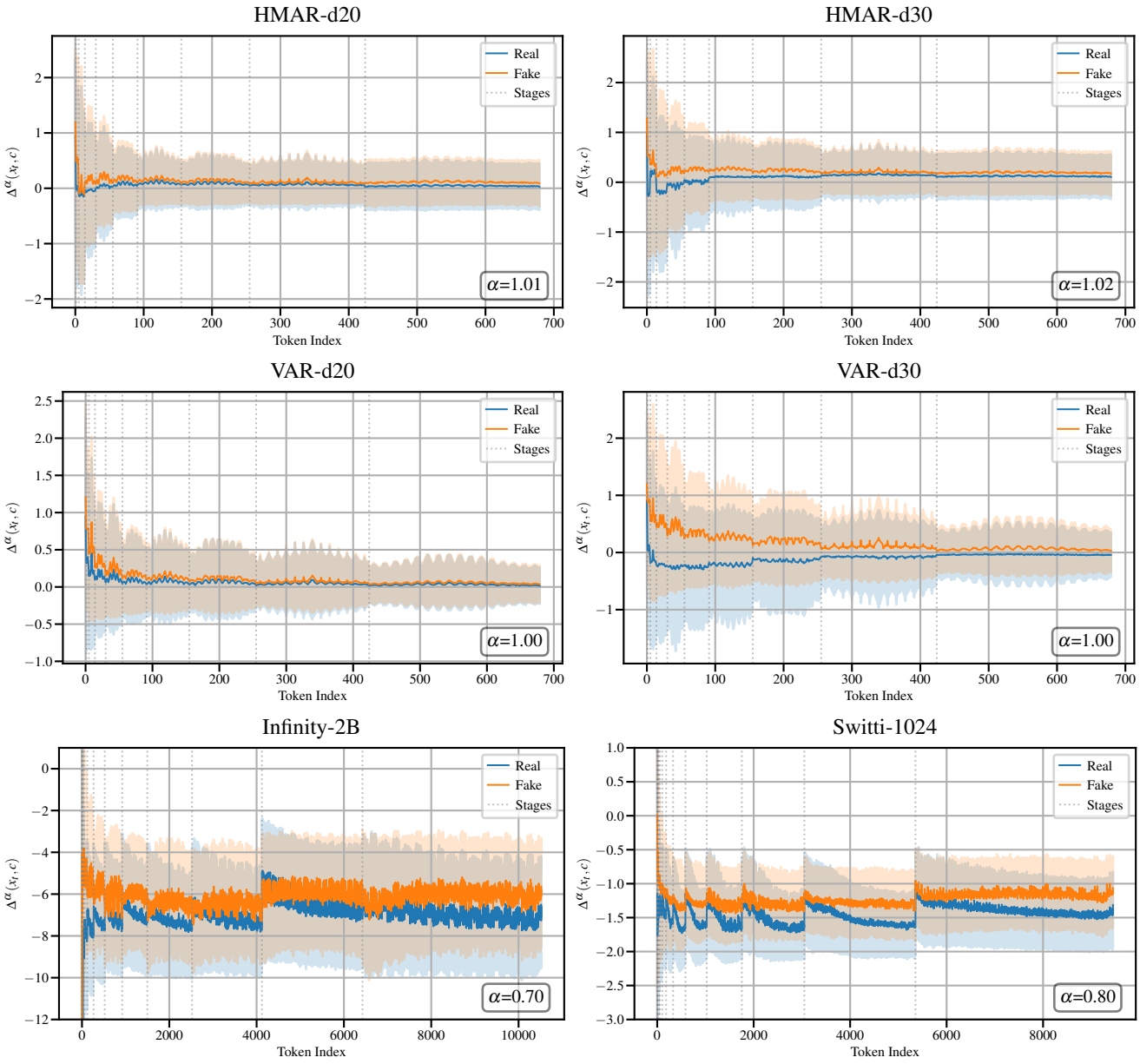

Figure 16. **Token-wise mean and standard deviation of $\Delta^\alpha(x_t, c)$.** Tokens are ordered from the coarsest to the finest scale and averaged over 10 000 real and 10 000 fake images for class-to-image models, and 1000 real and 1000 fake images for text-to-image models. We observe different behavior across scales, with coarser scales showing larger differences in the mean values, but also a larger standard deviation. This leads to a worse signal-to-noise ratio for coarser scales where only very few tokens are available for each image.

In this section we show how the α-balanced probability ratios $\Delta^\alpha(x_t, c)$ differ between scales, justifying why it is beneficial to learn weights to linearly combine scale-wise score averages. Figure 16 depicts the token-wise means and standard deviations of α-balanced probability ratios $\Delta^\alpha(x_t, c)$, ordered from the coarsest to the finest scale, computed over all real and all generated images for the respective model. We observe an almost consistently larger mean value for generated images over all scales, but the difference decreases toward finer scales. Intuitively, for early scales, there is a strong dependence on the semantic conditioning. For later scales of high-resolution models, there is a natural extension from coarser to finer scales, and hence the conditional and unconditional predictions agree. In all cases, the variance is fairly large, in particular for the coarser

scales. We also observe that average scores for later scales strongly benefit from their larger amount of tokens, leading to a more stable signal across images. PRADA's calibration mechanics support this observation by automatically putting more weight on finer scales, see Figures 12 to 14.

Finally, the token-wise inspection of mean and standard deviation visually supports our decision to introduce the balancing parameter α. For Infinity-2B and Switti-1024, where the balancing parameter α differs significantly from 1, Figure 17 shows the same token-wise plot, but for the unbalanced probability ratio $\Delta(x_t, c)$. In contrast to Figure 16, the mean differences are much smaller, in particular for finer scales (which are the more stable ones due to their high number of tokens). This demonstrates the benefit of deviating from the standard probability ratio when the unbalanced probability ratios for real and generated samples both converge to zero in the finest scales.

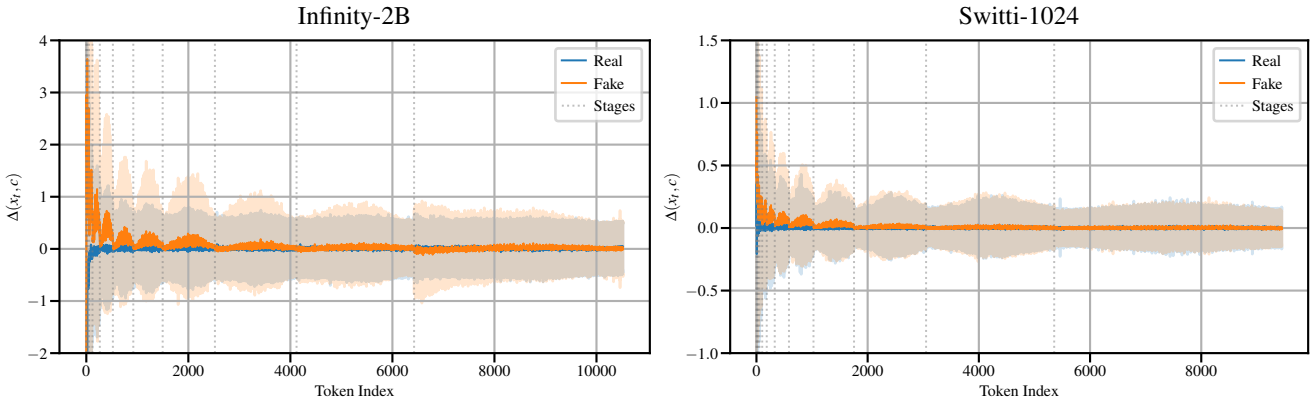

Figure 17. **Mean and standard deviation of $\Delta(x_t, c)$ across scales.** Comparing these to Figure 16 shows how differences between real and fake samples are visibly amplified with a learned balancing parameter α.

E. Detailed Ablation Study Results

In this section we provide the detailed results for our ablation study in Section 5.4. Similar to Table 1, we report the detection performance in AUROC for each AR image generator.

Score Function Components Tables 4 and 5 show the detailed results for different variants of the PRADA score function for class-to-image and text-to-image models, respectively. Considering the variant with fixed $\alpha = 1$ and $w_i = \frac{1}{S}$ as a baseline, we observe that learning α is especially effective for text-to-image models ($+11.2\%$). For class-to-image models, learning α only yields an improvement of $+0.3\%$. Learning only the scale weights w_i improves the performance for both model classes.

Number of Training Samples In Tables 6 and 7 we report the detection performance depending on the number of samples used for learning the score function. Note that n_{train} refers to the number of samples per class, e.g., real and generated images. While, as expected, more samples lead to better results, we observe that there is only a small improvement when increasing n_{train} from 125 to 250. Notably, training with only 50 real and 50 generated samples yields an average AUROC of more than 90% for both model classes.

Number of Hidden Neurons in f_θ Tables 8 and 9 show how the number of hidden neurons in each layer of f_θ influences the detection performance. We also report the total number of trainable parameters of f_θ. While we observe a more pronounced drop in performance when using only 4 hidden neurons for text-to-image models, PRADA is overall insensitive to the choice of n_{hidden}. Using more neurons does not appear to cause overfitting. Based on these results, we selected $n_{\text{hidden}} = 16$ as our default to keep the number of parameters small.

Table 4. **Detection performance of different score function variants for class-to-image models, measured in AUROC (%)**. We report the mean and standard deviation over five calibration runs.

Variant	HMAR-d20	HMAR-d30	LlamaGen-B	LlamaGen-L	VAR-d20	VAR-d30	RAR-L	RAR-XXL	**Avg.**
fixed α, fixed w	72.0 $\pm$ 0.6	79.8 $\pm$ 0.2	99.0 $\pm$ 0.2	99.0 $\pm$ 0.2	74.0 $\pm$ 0.6	91.1 $\pm$ 0.3	97.8 $\pm$ 0.3	98.6 $\pm$ 0.2	88.9 $\pm$ 0.3
learnable α, fixed w	72.3 $\pm$ 0.6	80.1 $\pm$ 0.6	99.5 $\pm$ 0.1	99.5 $\pm$ 0.2	74.1 $\pm$ 0.9	91.1 $\pm$ 0.5	98.0 $\pm$ 0.1	98.8 $\pm$ 0.0	89.2 $\pm$ 0.4
fixed α, learnable w	85.5 $\pm$ 0.7	90.4 $\pm$ 0.5	98.2 $\pm$ 0.1	98.2 $\pm$ 0.1	83.4 $\pm$ 0.3	96.4 $\pm$ 0.5	97.0 $\pm$ 0.2	97.8 $\pm$ 0.2	93.3 $\pm$ 0.3
$f_\theta : \mathbb{R}^2 \to \mathbb{R}$, fixed w	68.1 $\pm$ 1.3	76.8 $\pm$ 1.3	99.8 $\pm$ 0.1	99.9 $\pm$ 0.0	72.4 $\pm$ 2.0	90.7 $\pm$ 0.8	99.3 $\pm$ 0.1	98.8 $\pm$ 0.2	88.2 $\pm$ 0.7
$f_\theta : \mathbb{R}^2 \to \mathbb{R}$, learnable w	81.5 $\pm$ 1.6	86.4 $\pm$ 2.5	99.5 $\pm$ 0.3	99.6 $\pm$ 0.1	83.6 $\pm$ 3.1	96.1 $\pm$ 0.6	98.6 $\pm$ 0.3	97.3 $\pm$ 0.6	92.8 $\pm$ 1.1
PRADA (default)	85.9 $\pm$ 0.8	90.5 $\pm$ 0.6	98.7 $\pm$ 0.2	98.8 $\pm$ 0.1	83.6 $\pm$ 0.3	96.6 $\pm$ 0.5	97.4 $\pm$ 0.1	98.1 $\pm$ 0.2	93.7 $\pm$ 0.3

Table 5. **Detection performance of different score function variants for text-to-image models, measured in AUROC (%)**. We report the mean and standard deviation over five calibration runs.

Variant	Infinity-2B	Janus-Pro-1B	LlamaGen-XL	Switti-1024	**Avg.**
fixed α, fixed w	92.4 $\pm$ 0.7	94.6 $\pm$ 0.7	78.6 $\pm$ 1.0	82.7 $\pm$ 0.4	87.1 $\pm$ 0.7
learnable α, fixed w	98.7 $\pm$ 0.2	98.0 $\pm$ 0.3	98.7 $\pm$ 0.3	97.6 $\pm$ 0.2	98.3 $\pm$ 0.2
fixed α, learnable w	94.6 $\pm$ 0.5	95.3 $\pm$ 0.6	82.1 $\pm$ 0.6	91.4 $\pm$ 0.5	90.9 $\pm$ 0.6
$f_\theta : \mathbb{R}^2 \to \mathbb{R}$, fixed w	99.4 $\pm$ 0.1	97.9 $\pm$ 0.4	98.9 $\pm$ 0.4	98.6 $\pm$ 0.1	98.7 $\pm$ 0.2
$f_\theta : \mathbb{R}^2 \to \mathbb{R}$, learnable w	99.9 $\pm$ 0.1	98.6 $\pm$ 0.4	99.4 $\pm$ 0.2	99.9 $\pm$ 0.0	99.5 $\pm$ 0.2
PRADA (default)	99.7 $\pm$ 0.1	98.8 $\pm$ 0.3	99.3 $\pm$ 0.1	99.8 $\pm$ 0.0	99.4 $\pm$ 0.1

Table 6. **Detection performance with different numbers of training samples for class-to-image models, measured in AUROC (%)**. We report the mean and standard deviation over five calibration runs.

n_{train}	HMAR-d20	HMAR-d30	LlamaGen-B	LlamaGen-L	VAR-d20	VAR-d30	RAR-L	RAR-XXL	**Avg.**
10	69.4 $\pm$ 7.1	77.0 $\pm$ 4.4	78.0 $\pm$ 5.5	78.3 $\pm$ 4.7	66.3 $\pm$ 4.4	77.6 $\pm$ 4.7	78.1 $\pm$ 5.9	76.1 $\pm$ 6.1	75.1 $\pm$ 5.3
25	77.1 $\pm$ 3.9	83.1 $\pm$ 2.3	89.5 $\pm$ 1.9	89.7 $\pm$ 1.9	72.0 $\pm$ 4.0	88.0 $\pm$ 0.8	87.2 $\pm$ 1.4	87.3 $\pm$ 2.5	84.2 $\pm$ 2.3
50	82.7 $\pm$ 0.9	87.3 $\pm$ 1.3	95.7 $\pm$ 2.0	95.9 $\pm$ 2.1	77.7 $\pm$ 2.0	93.7 $\pm$ 2.5	93.7 $\pm$ 2.4	94.3 $\pm$ 2.9	90.1 $\pm$ 2.0
125	84.9 $\pm$ 1.2	90.1 $\pm$ 0.5	98.4 $\pm$ 0.2	98.4 $\pm$ 0.2	82.7 $\pm$ 0.9	96.2 $\pm$ 0.8	97.1 $\pm$ 0.2	97.8 $\pm$ 0.4	93.2 $\pm$ 0.6
250	85.9 $\pm$ 0.8	90.5 $\pm$ 0.6	98.7 $\pm$ 0.2	98.8 $\pm$ 0.1	83.6 $\pm$ 0.3	96.6 $\pm$ 0.5	97.4 $\pm$ 0.1	98.1 $\pm$ 0.2	93.7 $\pm$ 0.3

Table 7. **Detection performance with different numbers of training samples for text-to-image models, measured in AUROC (%)**. We report the mean and standard deviation over five calibration runs.

n_{train}	Infinity-2B	Janus-Pro-1B	LlamaGen-XL	Switti-1024	**Avg.**
10	94.2 $\pm$ 3.1	93.8 $\pm$ 2.4	93.1 $\pm$ 1.1	93.0 $\pm$ 2.1	93.5 $\pm$ 2.2
25	98.4 $\pm$ 1.0	97.5 $\pm$ 1.2	96.7 $\pm$ 3.3	98.2 $\pm$ 0.8	97.7 $\pm$ 1.6
50	99.1 $\pm$ 0.5	98.0 $\pm$ 1.1	97.4 $\pm$ 2.9	99.2 $\pm$ 0.4	98.4 $\pm$ 1.2
125	99.6 $\pm$ 0.2	98.7 $\pm$ 0.3	99.1 $\pm$ 0.1	99.7 $\pm$ 0.1	99.3 $\pm$ 0.2
250	99.7 $\pm$ 0.1	98.8 $\pm$ 0.3	99.3 $\pm$ 0.1	99.8 $\pm$ 0.0	99.4 $\pm$ 0.1

Table 8. **Detection performance with different numbers of hidden neurons for class-to-image models, measured in AUROC (%)**. We report the mean and standard deviation over five calibration runs.

n_{hidden}	#Params	HMAR-d20	HMAR-d30	LlamaGen-B	LlamaGen-L	VAR-d20	VAR-d30	RAR-L	RAR-XXL	**Avg.**
4	33	85.5 $\pm$ 0.6	90.2 $\pm$ 0.3	98.5 $\pm$ 0.2	98.6 $\pm$ 0.2	83.3 $\pm$ 0.2	96.2 $\pm$ 0.7	96.8 $\pm$ 0.7	98.0 $\pm$ 0.0	93.4 $\pm$ 0.4
8	97	85.5 $\pm$ 0.8	90.3 $\pm$ 0.7	98.7 $\pm$ 0.3	98.8 $\pm$ 0.2	83.5 $\pm$ 0.3	96.5 $\pm$ 0.6	97.2 $\pm$ 0.3	98.0 $\pm$ 0.2	93.6 $\pm$ 0.4
16	321	85.9 $\pm$ 0.8	90.5 $\pm$ 0.6	98.7 $\pm$ 0.2	98.8 $\pm$ 0.1	83.6 $\pm$ 0.3	96.6 $\pm$ 0.5	97.4 $\pm$ 0.1	98.1 $\pm$ 0.2	93.7 $\pm$ 0.3
32	1153	86.0 $\pm$ 0.6	90.7 $\pm$ 0.4	98.7 $\pm$ 0.3	98.8 $\pm$ 0.2	83.9 $\pm$ 0.4	96.7 $\pm$ 0.4	97.3 $\pm$ 0.2	98.1 $\pm$ 0.2	93.8 $\pm$ 0.3
64	4353	85.9 $\pm$ 0.7	90.8 $\pm$ 0.4	98.7 $\pm$ 0.3	98.8 $\pm$ 0.2	84.0 $\pm$ 0.1	96.7 $\pm$ 0.5	97.3 $\pm$ 0.2	98.1 $\pm$ 0.3	93.8 $\pm$ 0.3

Table 9. **Detection performance with different numbers of hidden neurons for text-to-image models, measured in AUROC (%)**. We report the mean and standard deviation over five calibration runs.

n_{hidden}	#Params	Infinity-2B	Janus-Pro-1B	LlamaGen-XL	Switti-1024	**Avg.**
4	33	99.0 $\pm$ 0.5	96.1 $\pm$ 3.2	96.2 $\pm$ 2.7	98.8 $\pm$ 0.4	97.5 $\pm$ 1.7
8	97	99.4 $\pm$ 0.3	98.4 $\pm$ 0.6	98.5 $\pm$ 1.0	99.4 $\pm$ 0.3	98.9 $\pm$ 0.6
16	321	99.7 $\pm$ 0.1	98.8 $\pm$ 0.3	99.3 $\pm$ 0.1	99.8 $\pm$ 0.0	99.4 $\pm$ 0.1
32	1153	99.7 $\pm$ 0.1	98.9 $\pm$ 0.3	99.3 $\pm$ 0.1	99.8 $\pm$ 0.0	99.4 $\pm$ 0.1
64	4353	99.7 $\pm$ 0.1	99.0 $\pm$ 0.2	99.3 $\pm$ 0.1	99.8 $\pm$ 0.0	99.4 $\pm$ 0.1