When Anonymity Breaks: Identifying Models Behind Text-to-Image Leaderboards

Supplementary Material

9. Automated Distinguishability Score Analysis

To better understand trends in distinguishability across visual concepts, we construct a large-scale pipeline using ≈ 2.5M prompts collected from Midjourney [33]. We perform basic filtering based on length (25–300 characters), language (English), and quality (e.g., requiring >75% alphanumeric characters, no excessive repetition, and reasonable word lengths). MinHash deduplication is then applied to remove near-duplicate prompts, resulting in ≈ 1.4M prompts. We use datatrove [38] to implement these filters.

We embed all prompts using the `Qwen3-Embedding-4B` model [62] to obtain text representations, and cluster them into 920 semantic groups. For each cluster, we sample 15 prompts and generate 15 images per prompt from 11 representative models (selected from the full set of 22), producing over two million generated samples in total. We then compute distinguishability scores for all prompts and analyze trends across clusters. To interpret clusters, we automatically extract keywords using KeyBERT [44], selecting the top-10 keywords in the 1–3 word range across prompts within each cluster. Each cluster is then assigned a concise label (≤ 4 words) using Llama 3.1 8B Instruct [9], based on the extracted keywords and 10 sampled prompts for context. These LLM-generated labels are propagated to all prompts in the corresponding clusters.

Prompts from stylistically rich or visually constrained categories (*e.g.* "oil painting," "anime portrait") exhibit high distinguishability, whereas generic or ambiguous prompts (*e.g.* "city street," "landscape," "logo") yield lower separability. We visualize the most and least distinguishable concepts in Fig. 7, and provide representative examples of high- and low-distinguishability prompts, along with their generations from two models, in Figs. 8 and 9.

10. Details on Our Countermeasure

Setup. We randomly sampled 100 images to evaluate the effectiveness of our adversarial post-processing defense. Following prior work [13], we adopt the same optimization hyperparameters: a learning rate of 0.1 and a contrastive-loss temperature of $\tau = 0.1$. For the visual encoder, we use an ensemble of surrogate models listed in Table 8. To test transferability across model families, the target encoder is set to `google/siglip2-large-patch16-512`, which is distinct from all surrogate models.

Algorithm 1 Adversarial Post-Processing against Deanonymizition

Input: T2I models $\mathcal{C} = \{M_1, \ldots, M_n\}$ on the leaderboard; prompt p; visual encoder E_v; similarity $S(\cdot,\cdot)$; temperature τ; perturbation budget ε; iterations T; learning rate η; target image I^* generated by M_i using p
Output: Defended image $\tilde{I}^*$
$S_i \leftarrow \mathcal{C} \setminus \{M_i\}$ ▷ all other models
$e^* \leftarrow E_v(I^*)$
for each $M_j \in S_i$ **do** ▷ same-prompt candidates
  $I_j \leftarrow M_j(p)$
  $e_j \leftarrow E_v(I_j)$
end for
$j^* \leftarrow \arg\min_{M_j \in S_i} S(e^*, e_j)$ ▷ farthest = least similar
$x^+ \leftarrow I_{j^\star}, \quad x^- \leftarrow I^*$
$\delta^{(0)} \leftarrow 0$
for $t = 0$ to $T-1$ **do**
  $\tilde{I}^{(t)} \leftarrow I^* + \delta^{(t)}$
  $z^{(t)} \leftarrow E_v(\tilde{I}^{(t)})$
  $s^+ \leftarrow S\big(z^{(t)}, E_v(x^+)\big), \quad s^- \leftarrow S\big(z^{(t)}, E_v(x^-)\big)$
  $p^+ \leftarrow \dfrac{\exp(s^+/\tau)}{\exp(s^+/\tau) + \exp(s^-/\tau)}$
  $p^- \leftarrow \dfrac{\exp(s^-/\tau)}{\exp(s^+/\tau) + \exp(s^-/\tau)}$
  $\mathcal{L}^{(t)} \leftarrow -\log p^+ + \log p^-$ ▷ contrastive loss
  $\delta^{(t+1)} \leftarrow \Pi_\varepsilon\big(I^*,\ \delta^{(t)} - \eta \nabla_\delta \mathcal{L}^{(t)}\big)$ ▷ $\Pi_\varepsilon(I^*, \cdot)$ projects onto the ℓ_∞-ball of radius ε around I^*.
end for
$\tilde{I}^* \leftarrow I^* + \delta^{(T)}$

Undoing Post-Processing. An adversary aware of post-processing by leaderboard maintainers may try to remove the effect of these perturbations. Following Hönig et al. [12], we evaluate two strategies: Gaussian noising, which adds random noise, and noisy upscaling, which uses a diffusion model but incurs higher cost. At ε=2, our defense yields a Top-1 accuracy of 0.75; Gaussian noising reduces it to 0.69 and noisy upscaling to 0.72. These results show that post-hoc operations do not significantly weaken our defense, indicating that standard adversarial-example mitigation methods may be ineffective against embedding-space displacement.

Adversarial examples. Fig. 10 shows the original image and adversarial examples under different perturbation budgets $\varepsilon \in \{2, 4, 8\}$. Larger ε yields more visible perturbations but stronger defense performance (see Table 3).

Table 4. Full list of T2I models used in our experiments, along with their provider, image resolution, and the number of inference steps. Where inference-step counts are available on the ArtificialAnalysis methodology page, we adopt those values directly. For models not mentioned there, we use the default values documented on their respective Hugging Face model pages. For OpenAI and Midjourney models we did not explicitly set the number of inference steps and used their internal default generation settings.

Model	**Company / Provider**	**Resolution (W×H)**	**Inference Steps**
DALL·E 3 HD [2]	OpenAI	1024×1024	–
FLUX.1-dev [17]	Black Forest Labs	1024×1024	28
FLUX.1-schnell [17]	Black Forest Labs	1024×1024	4
FLUX.1-Krea-dev [17]	Black Forest Labs	1024×1024	28
FLUX-1.1-pro	Black Forest Labs	1024×1024	28
flux.1-kontext-pro	Black Forest Labs	1024×1024	28
Stable Diffusion v1.5 [43]	Stability AI	512×512	50
Stable Diffusion 2.1 [43]	Stability AI	768×768	50
Stable Diffusion XL [39]	Stability AI	1024×1024	30
SDXL Turbo [39]	Stability AI	1024×1024	4
Stable Diffusion 3.5 Large Turbo	Stability AI	1024×1024	4
Stable Diffusion 3.5 Large	Stability AI	1024×1024	35
Stable Diffusion 3 Medium [8]	Stability AI	1024×1024	30
Stable Diffusion 3.5 Medium	Stability AI	1024×1024	40
GPT-Image-1	OpenAI	1024×1024	–
Midjourney v6 [30]	Midjourney	1024×1024	–
Lumina 2 [40]	Alpha-VLLM	1024×1024	50
HiDream [3]	HiDream.ai	1024×1024	50
Playground v2.5 [20]	Playground AI	1024×1024	50
Playground v2 [20]	Playground AI	1024×1024	50
Playground v1	Playground AI	512×512	–
Qwen-Image [54]	Alibaba	1024×1024	50

Table 5. Mean deanonymization accuracy for each target model. Values are averaged over all prompts and reported as percentages.

Target Model	**Mean Accuracy (%)**
DALL·E 3 HD	99.9
SDXL Turbo	99.9
GPT-Image-1	99.8
Playground v2	99.7
HiDream	99.7
Playground v2.5	99.6
FLUX.1-Krea-dev	99.6
Lumina 2	99.5
flux.1-kontext-pro	99.5
Playground v1	99.4
Stable Diffusion 3.5 Medium	99.3
Stable Diffusion 3 Medium	99.2
Stable Diffusion 3.5 Large Turbo	99.2
Stable Diffusion XL	99.1
Midjourney v6	99.0
FLUX.1-schnell	98.9
Stable Diffusion 3.5 Large	98.8
Qwen-Image	98.7
Stable Diffusion 2.1	98.6
Stable Diffusion v1.5	98.3
FLUX-1.1-pro	97.9
FLUX.1-dev	97.8

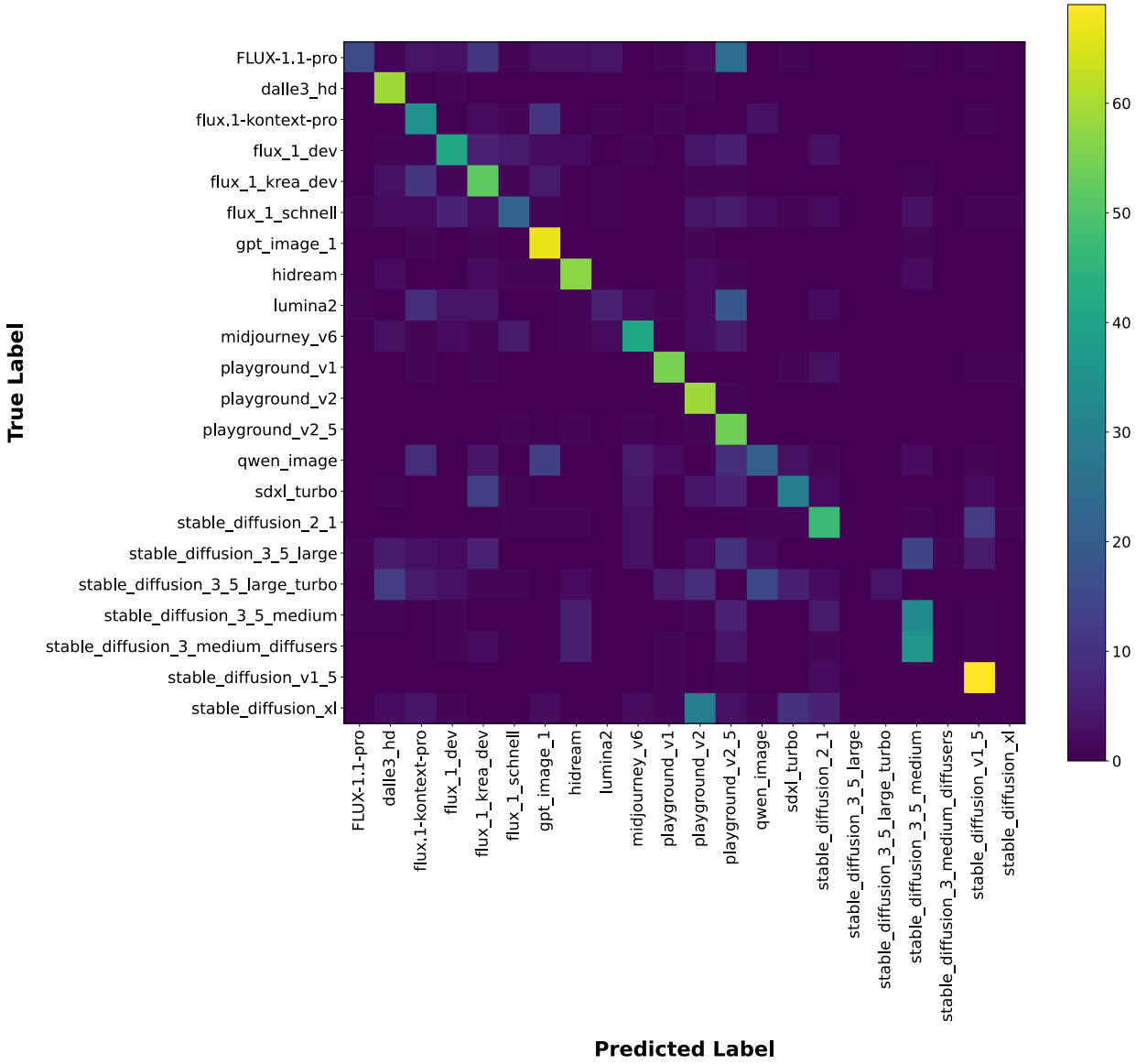


Figure 6. Confusion matrix of the image-only classifier aggregated over five runs, showing per-model prediction patterns and overall variability across models.

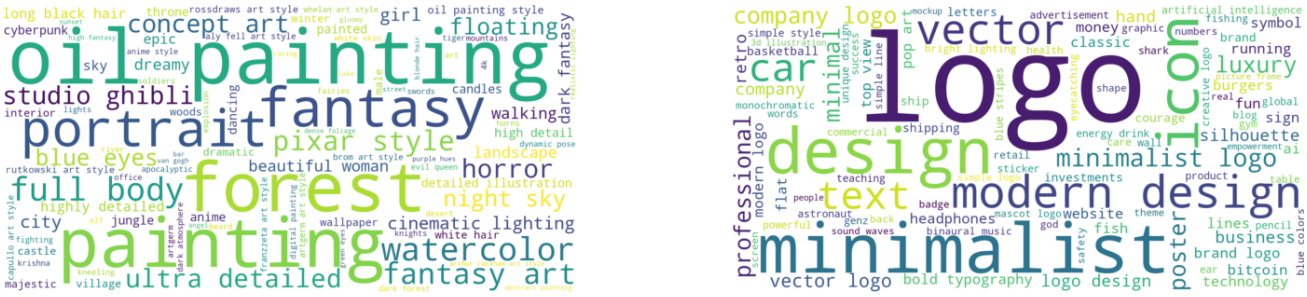


Figure 7. Word-cloud for keywords corresponding to **Left:** prompts with high distinguishability scores and **Right:** low distinguishability.

Table 6. Per-model deanonymization performance in the one-vs-rest setting, where the adversary has access only to its target model. Models are sorted in decreasing order of accuracy.

Target Model	**Accuracy**	**FPR**	**FNR**	**TPR @1%**	**TPR @5%**	**ROC-AUC**
playground_v2_5	0.986	0.003	0.292	0.871	0.988	0.993
sdxl_turbo	0.985	0.000	0.318	0.971	0.983	0.997
playground_v2	0.984	0.003	0.258	0.902	0.968	0.995
gpt_image_1	0.984	0.000	0.345	0.917	1.000	0.997
hidream	0.979	0.008	0.271	0.750	0.976	0.989
stable_diffusion_3_5_large_turbo	0.969	0.017	0.305	0.544	0.864	0.974
dalle3_hd	0.969	0.025	0.191	0.520	0.883	0.979
flux.1-kontext-pro	0.954	0.035	0.304	0.534	0.851	0.961
stable_diffusion_3_5_medium	0.954	0.032	0.404	0.331	0.702	0.951
stable_diffusion_3_5_large	0.954	0.037	0.252	0.549	0.852	0.968
stable_diffusion_3_medium_diffusers	0.953	0.037	0.307	0.300	0.730	0.957
lumina2	0.944	0.044	0.381	0.178	0.651	0.922
flux_1_krea_dev	0.943	0.042	0.291	0.340	0.810	0.950
flux_1_dev	0.932	0.051	0.399	0.213	0.623	0.930
midjourney_v6	0.931	0.059	0.276	0.229	0.707	0.939
playground_v1	0.926	0.061	0.345	0.154	0.598	0.919
flux_1_schnell	0.921	0.071	0.250	0.274	0.682	0.919
stable_diffusion_xl	0.908	0.079	0.367	0.179	0.510	0.923
FLUX-1.1-pro	0.896	0.092	0.318	0.260	0.557	0.915
stable_diffusion_2_1	0.876	0.116	0.272	0.177	0.466	0.891
stable_diffusion_v1_5	0.850	0.142	0.255	0.117	0.398	0.869
qwen_image	0.694	0.306	0.304	0.087	0.235	0.717

Table 7. Deanonymization performance for our method across different image encoders.

Image Encoder	**Accuracy** (%)		
	Top-1	Top-2	Top-3
laion/CLIP-ViT-L-14-DataComp.XL [15]	87.86	94.71	96.29
google/siglip2-large-patch16-512 [51]	90.36	95.79	97.29
openai/clip-vit-large-patch14 [41]	84.07	92.71	95.64
laion/CLIP-ViT-bigG-14-laion2B [15]	**90.86**	**96.14**	**97.50**

Prompt: "Design a LOGO of instant tea with white background."

Model	Generated Images (five different seeds)
Stable Diffusion 3.5 Medium	
Stable Diffusion 3.5 Large Turbo	

Figure 8. Example of a low-distinguishability prompt with high intra-model variation, showing generations from two models across five seeds.

Prompt: "create a painting of a lake surrounded by forest in the winter, foggy weather, snow falling, andre kohn style painting."

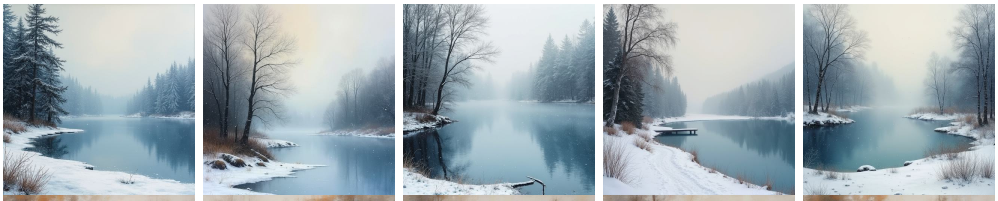
Model	Generated Images (five different seeds)
FLUX.1-dev	
SDXL Turbo	

Figure 9. Example of a high-distinguishability prompt with low intra-model variation, showing generations from two models across five seeds.

Table 8. Local encoder ensemble used for adversarial optimization.

Model	Training Dataset
ViT-H-14-378-quickgelu	dfn5b
ViT-H-14-quickgelu	dfn5b
ViT-SO400M-14-SigLIP-384 [60]	webli
ViT-SO400M-14-SigLIP [60]	webli
ViT-L-16-SigLIP-384 [60]	webli
ViT-bigG-14 [15]	laion2b_s39b_b160k
ViT-H-14-CLIPA-336 [22]	datacomp1b
ViT-H-14-quickgelu	metaclip_fullcc

(a) Original (b) $\varepsilon = 2$ (c) $\varepsilon = 4$ (d) $\varepsilon = 8$

Figure 10. Original image and adversarial examples generated under different perturbation budgets ε. As expected, visual artifacts are more apparent for larger perturbation budgets.