Bi-Level Optimization for Self-Supervised AI-Generated Face Detection

Supplementary Material

A. Computation of Bi-level Optimization

In solving the bi-level optimization Problem (1), parameters $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are updated alternately:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \sum_{\boldsymbol{x} \in \mathcal{B}_{\text{tr}}} \sum_{i=1}^{K} \lambda_i \ell_i(\boldsymbol{x}; \boldsymbol{\theta}) \tag{14a}$$

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \beta \nabla_{\boldsymbol{\lambda}} \sum_{\boldsymbol{x} \in \mathcal{B}_{\text{val}}} \ell_1\left(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda})\right) \tag{14b}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \sum_{\boldsymbol{x} \in \mathcal{B}_{\text{tr}}} \sum_{i=1}^{K} \lambda_i \ell_i(\boldsymbol{x}; \boldsymbol{\theta}), \tag{14c}$$

where α and β are the inner- and outer-loop learning rates, respectively. To compute $\nabla_{\boldsymbol{\lambda}} \ell_1(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda}))$, we apply the chain rule, yielding

$$\nabla_{\boldsymbol{\lambda}} \ell_1\left(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda})\right) = \nabla_{\boldsymbol{\lambda}} \boldsymbol{\theta}'(\boldsymbol{\lambda}) \nabla_{\boldsymbol{\theta}'} \ell_1\left(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda})\right). \tag{15}$$

Making use of Eq. (14a) and defining $\ell_{\text{pre}}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \sum_{\boldsymbol{x} \in \mathcal{B}_{\text{tr}}} \sum_{i=1}^{K} \lambda_i \ell_i(\boldsymbol{x}; \boldsymbol{\theta})$, we have the Jacobian:

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}} \boldsymbol{\theta}'(\boldsymbol{\lambda}) &= \nabla_{\boldsymbol{\lambda}} \left(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \ell_{\text{pre}}(\boldsymbol{\theta}, \boldsymbol{\lambda})\right) \\ &= -\alpha \nabla^2_{\boldsymbol{\theta}, \boldsymbol{\lambda}} \ell_{\text{pre}}(\boldsymbol{\theta}, \boldsymbol{\lambda})^{\intercal}. \end{aligned} \tag{16}$$

Substituting it into Eq. (15), we obtain the final expression:

$$\nabla_{\boldsymbol{\lambda}} \ell_1\left(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda})\right) = -\alpha \nabla^2_{\boldsymbol{\theta}, \boldsymbol{\lambda}} \ell_{\text{pre}}(\boldsymbol{\theta}, \boldsymbol{\lambda})^{\intercal} \nabla_{\boldsymbol{\theta}'} \ell_1\left(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda})\right). \tag{17}$$

Finite Difference Approximation. Direct computation of the mixed second-order derivative $\nabla^2_{\boldsymbol{\theta}, \boldsymbol{\lambda}} \ell_{\text{pre}}(\boldsymbol{\theta}, \boldsymbol{\lambda})$ is computationally intensive. To circumvent this, we employ a finite difference scheme [A1, A2] to approximate the Jacobian-vector product required in the outer-loop gradient. Let $\boldsymbol{v} = \nabla_{\boldsymbol{\theta}'} \ell_1(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda}))$ denotes the gradient of the outer-loop loss with respect to the updated parameters. Define the gradient of the inner-loop objective with respect to $\boldsymbol{\lambda}$ as a function of $\boldsymbol{\theta}$, *i.e.*, $\boldsymbol{F}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\lambda}} \ell_{\text{pre}}(\boldsymbol{\theta}, \boldsymbol{\lambda})$. We apply a second-order central difference approximation of the directional derivative of $\boldsymbol{F}$ along $\boldsymbol{v}$:

$$\begin{aligned} &\nabla^2_{\boldsymbol{\theta}, \boldsymbol{\lambda}} \ell_{\text{pre}}(\boldsymbol{\theta}, \boldsymbol{\lambda})^{\intercal} \nabla_{\boldsymbol{\theta}'} \ell_1(\boldsymbol{x}; \boldsymbol{\theta}'(\boldsymbol{\lambda})) \\ &= \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})^{\intercal} \boldsymbol{v} \\ &\approx \frac{\boldsymbol{F}(\boldsymbol{\theta} + \epsilon \boldsymbol{v}) - \boldsymbol{F}(\boldsymbol{\theta} - \epsilon \boldsymbol{v})}{2\epsilon} \\ &= \frac{\nabla_{\boldsymbol{\lambda}} \ell_{\text{pre}}(\boldsymbol{\theta}^+, \boldsymbol{\lambda}) - \nabla_{\boldsymbol{\lambda}} \ell_{\text{pre}}(\boldsymbol{\theta}^-, \boldsymbol{\lambda})}{2\epsilon}, \end{aligned} \tag{18}$$

where

$$\boldsymbol{\theta}^{\pm} = \boldsymbol{\theta} \pm \epsilon \boldsymbol{v}, \tag{19}$$

and ϵ is a small constant.

B. EXIF Tag Selection

This section details the procedure for selecting EXIF tags used in our self-supervised pretraining.

We begin by identifying all EXIF tags that appear in more than 50% of the collected photographic face images from the FDF dataset [62]. Table 8 lists their frequencies and representative values. To ensure that the selected tags offer meaningful supervisory signals, we apply three empirical filtering criteria:

- **Relevance to Digital Imaging:** The chosen tag must encode semantically or physically interpretable imaging attributes (*e.g.*, exposure settings and camera make).
- **Information Richness:** Tags dominated by unknown entries or a single category are excluded to avoid degenerate supervision.
- **Semantic Redundancy Removal:** Tags whose semantics largely overlap with others (*e.g.*, `F-number` with `aperture`) are removed for parsimony.

For example, tags such as `date/time`, `EXIF version`, `resolution unit`, and `scene capture type` are excluded due to irrelevance to image formation. Likewise, `flash` is excluded because approximately $\sim 75\%$ of its values are missing or unknown.

After refinement, we retain nine EXIF tags for pretraining: `aperture`, `exposure mode`, `exposure program`, `exposure time`, `focal length`, `ISO speed`, `makes`, `metering mode`, and `white balance mode`.

C. Competing Detectors

This section briefly summarizes all competing detectors used for comparison.

CNND [52] trains a ResNet-50 classifier with standard data augmentations such as JPEG compression, Gaussian blurring as a way of improving generalizability.

GramNet [34] identifies AI-generated faces by capturing global texture statistics.

RECCE [3] learns to reconstruct face photographs. It is originally designed for face forgery detection, but is extended here to detect AI-generated faces.

LNP [31] extracts noise patterns using a pretrained denoising network, and fits a one-class support vector machine [A3] to detect AI-generated faces as anomalies.

LGrad [47] feeds gradient maps from a pretrained network as input to a detector to capture generative artifacts.

DIRE [53] assumes that diffusion models reconstruct synthetic images more accurately. Detection is based on reconstruction errors as input to a ResNet-50.

EXIF tag	Example value	#Unique entries	Count
Aperture	F2.8, F4, F5.6, F3.5	152	198,448
Exposure Mode	Auto, Auto-bracketing, Program, Manual	7	194,666
Exposure Program	Manual Control, Normal Program, Portrait Mode	6	176,787
Exposure Time	1/60 sec, 1/125 sec, 1/250 sec	1,745	198,488
Focal Length	18.0 mm, 50.0 mm, 6.3 mm	858	198,488
ISO Speed	100, 200, 400, 800	269	198,488
Makes	Canon, Apple, Sony, Nikon	10	198,247
Metering Mode	Center-weighted, Average, Partial, Spot	8	196,340
White Balance Mode	Auto, Manual	2	193,279
Custom Rendered	Custom Process, Normal Process, Unknown	3	180,667
Date/Time	2013:03:28 04:20:46	96,568	196,639
Date/Time Digitized	2013:03:28 04:20:46	96,447	197,521
Date/Time Original	2013:03:28 04:20:46	96,827	198,227
EXIF Version	2.21, 2.20, 2.30	11	197,673
Flash	Unfired, Fired, Unknown, Fired Auto	4	198,317
F-Number	F2.8, F4, F5.6, F3.5	111	197,501
Resolution Unit	Inch, Cm, No Unit, Unknown	4	193,347
Scene Capture Type	Standard, Portrait, Nightscene, Landscape, Unknown	5	192,638
Shutter Speed	1/60 sec, 1/125 sec, 1/250 sec	1,195	196,970
X Resolution	72 dots per inch	107	193,173
Y Resolution	72 dots per inch	108	193,173

Table 8. Overview of EXIF tags appearing in more than 50% of the collected photographic face images. The upper section lists the nine tags retained for pretraining based on relevance, informativeness, and non-redundancy.

Ojha23 [38] employs CLIP's frozen visual encoder to extract features for binary classification of photographic vs. AI-generated images.

AEROBLADE [43] is a training-free approach that calculates LPIPS [A4] reconstruction errors of latent diffusion autoencoders, leveraging the similar observation that synthetic images are reconstructed more faithfully.

FatFormer [32] fine-tunes CLIP with a forgery-aware Transformer adapter, integrating spatial and frequency cues.

Zou25 [66] casts ordinal EXIF-tag ranking as a pretext task, and employs one-class anomaly detection for inference.

CLIP [41] is a vision-language model pretrained on large-scale image-text pairs. Its general-purpose features can be adapted for identifying AI-generated content.

FaRL [62] jointly learns signal-level and semantic-level face representations from photographic images via contrastive and masked modeling. Due to its transferable features, we tailor it for detecting AI-generated faces.

EAL [61] aligns EXIF metadata (as text prompts) with images, aiming to learn imaging-specific representations.

Hu21 [22] detects GAN-generated faces by analyzing inconsistencies in corneal specular highlights, which are typically stable in human eyes but erratic in synthetic imagery.

D. Visual Samples

To illustrate the diversity and practical relevance of our evaluation setups, we provide representative face images used during training and testing.

- **Training Set:** Fig. 5(a) displays face photographs drawn from the CelebA-HQ dataset [25].
- **Cross-Generator Evaluation:** Figs. 5(b)-(j) show synthetic face images generated by nine representative models: StyleGAN2 [27], VQGAN [13], LDM [44], DDIM [46], SDv2.1 [44], FreeDoM [58], HPS [55], Midjourney [1], and SDXL [39].
- **Cross-Dataset Evaluation:** Fig. 6 presents additional samples for domain transfer experiments. Specifically, the evaluation involves testing on FFHQ photographs and synthetic images generated by StyleGAN2, VQGAN, and LDM trained on FFHQ [26]. This setup assesses the robustness of the learned representations to variations in data distribution and image source.

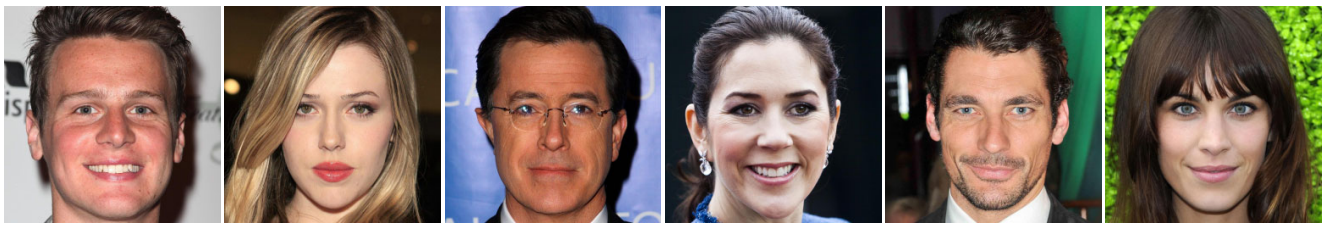
These samples qualitatively demonstrate the visual similarity between photographic and AI-generated faces, highlighting the challenges of reliable detection and the necessity of learning discriminative, generalizable features.

References

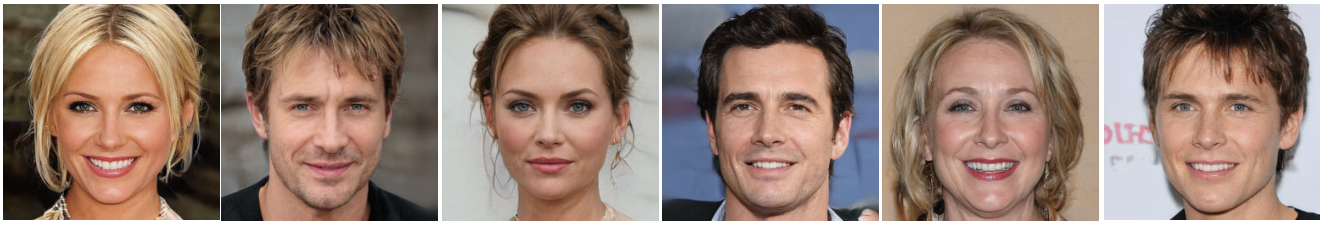
[A1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. in *ICML*, pages 1126–1135, 2017. 1

[A2] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 1

[A3] Bernhard Schölkopf, Robert C. Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support



(a) CelebA-HQ



(b) StyelGAN2

(c) VQGAN

(d) LDM

(e) DDIM

(f) SDv2.1

Figure 5. Representative face images used in cross-generator evaluation. (Part 1 of 2).

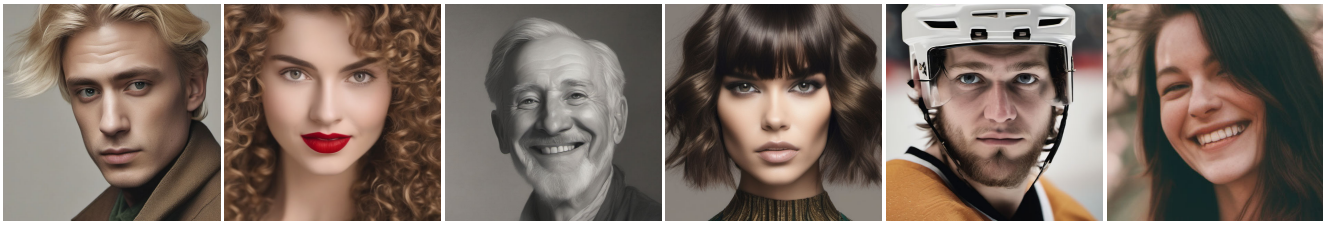
vector method for novelty detection. In *NeurIPS*, pages 582–588, 1999. 1

[A4] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable ef-

(g) FreeDoM

(h) HPS

(i) Midjourney



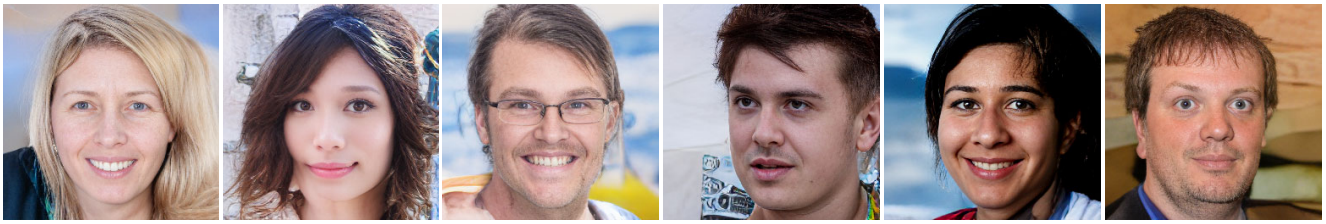
(j) SDXL

Figure 5. Representative face images used in cross-generator evaluation. (Part 2 of 2).

fectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 2

(a) FFHQ

(b) StyleGAN2



(c) VQGAN

(d) LDM

Figure 6. Representative face images used in cross-dataset evaluation.