

Source Attribution of AI-Generated Images in Data Scarcity Conditions via Training-Free Resynthesis

PIETRO BONGINI ¹, MEILING LI ², ANDREA COSTANZO ¹, BENEDETTA TONDI ¹ (Senior Member, IEEE),
AND MAURO BARNI ¹ (Fellow, IEEE)

¹Department of Information Engineering and Mathematics, University of Siena, 53100 Siena, Italy

²College of Computer Science and Artificial Intelligence, Fudan University, Shanghai 200438, China

CORRESPONDING AUTHOR: PIETRO BONGINI (e-mail: pietro.bongini@unisi.it).

This work was supported in part by Project SERICS under Grant PE00000014 through the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU -, in part by FOSTERER Project, funded by the Italian Ministry through PRIN 2022 Program under Contract 202289RHHP, in part by Remedy Project - New-Frontiers under Grant PSR 2024, funded by MUR, and in part by China Scholarship Council (CSC) under Grant 202406100170.

ABSTRACT Attributing synthetic images to the source that generated them is a difficult problem, particularly in data-scarcity conditions requiring the adoption of few-shot or zero-shot learning strategies. In this paper, we tackle this problem by introducing a training-free attribution method based on image resynthesis. Our method works by first automatically generating a textual prompt which describes the target image, and then using it to resynthesize the image with each candidate generator. Attribution is obtained by matching the image to the model whose resynthesis is most similar to the original one within an appropriate feature space - chosen based on a preliminary experimental analysis - among several possibilities offered by modern deep learning image analysis tools. Experiments using state-of-the-art few-shot models and other baselines show that our resynthesis-based method outperforms existing techniques when only a limited number of images are available for training or fine-tuning. Particular attention is given to assessing the robustness of the proposed method against common image processing operators and its adversarial robustness. In particular, the experimental results show that our method is robust to post-processing and that its adversarial robustness is significantly higher than that of baselines.

INDEX TERMS Image resynthesis, training-free, robust AI-generated image detection, text-to-image generators, synthetic image attribution.

I. INTRODUCTION

Generative AI has recently undergone very rapid development, with the evolution of image generative models being driven by several major novel contributions, including the introduction of diffusion models and the consequent proliferation of text-to-image generators [1]. On the one hand, this process has brought enormous benefits in terms of quality, availability, realism, and personalization. On the other hand, it has raised issues at multiple levels [2]. In particular, concerns about the misuse of generative AI have increased at the same fast pace [3]. As a consequence, efforts for the detection of AI-generated images and the identification of their source have increased accordingly [4].

Source attribution aims at identifying which generator produced a given synthetic image [5], [6]. This task can be addressed either jointly with detection or as a standalone problem. Our work focuses on the latter setting: we assume that the image is synthetic and aim at identifying the model that generated it.

The large number of generators available online, including open-source, free, and commercial solutions, makes Synthetic Image source Attribution (SIA) a very challenging task, especially in the wild. Moreover, the proliferation of new generators makes the problem even more difficult to handle [7]. As a matter of fact, it is virtually impossible to keep the pace by re-training a new version of an attribution model

every time a new generator becomes available. In addition, acquiring large amounts of data from commercial generators to fully train an attribution model on them is very expensive. As a consequence, few-shot source attribution methods, where only a small number of examples per source are used for training, are attracting increasing interest [8]. This scenario requires classifiers to be trained on a very small number of examples per class [9]. As an alternative, modular approaches can perform attribution by exploiting clustering, distance, or similarity techniques [10].

In this paper, we introduce a training-free method for SIA based on resynthesis, which works in a one-shot setting under closed set conditions.¹ The basic idea behind our scheme is that the generator that produced the image under analysis is likely to be able to resynthesize a copy of the image better than other models. Given that we do not know the caption that was originally used to generate the image to be attributed, we first use an LLM to generate a caption describing the image under analysis. The caption is subsequently used to *resynthesize* a copy of the original image with each candidate source model. The input image is attributed to the generator, which re-synthesized the image closest to the original one [11]. Since measuring pixel-wise distances between images is not useful to assess their similarity, we make use of a mixture of high-level semantic features and low-level signatures [12], extracted with a pretrained CLIP (Contrastive Language Image Pretraining) model [13], [14]. The distance is then measured in the feature space using a simple distance function.

Several SIA benchmark datasets have been introduced in recent years [15], [16], [17]. However, most of them include only a limited number of generators, and those generators are mostly, if not exclusively, open-source models. These two limits hinder their ability to reflect real-world scenarios where a large and diverse set of generators, including commercial ones, is available. Moreover, existing benchmarks rarely account for resynthesis, making it difficult to train and evaluate distance-based or similarity-based approaches [18]. To address these limitations, we introduce a new dataset that explicitly incorporates a resynthesis mechanism. The dataset focuses on head-and-shoulders photo-portrait images generated by text-to-image models and features two key characteristics: i) it includes images produced by 14 different generators, seven of which are commercial; and ii) it employs secondary textual descriptions to generate resynthesized versions of each image, enabling the evaluation of resynthesis-based methodologies. We use this dataset to assess the performance of the proposed source attribution method and to compare it against relevant baselines.

A key advantage of the proposed method is that, by relying on features extracted from a pretrained model and avoiding any training on data generated by the candidate generators, it inherently depends on more general (potentially semantic)

features for source attribution. As a result, the method is expected to be more robust to the application of common image processing operators and to adversarial attacks [19].

In summary, the main contributions of our work are:

- a training-free, one-shot source image attribution method based on image resynthesis;
- a dataset for few-shot SIA, incorporating a resynthesis mechanism based on text image description and generation of *resyntheses*;
- a set of experiments including comparisons with state-of-the-art few-shot source attribution methods, to assess the performance of our method and validate its superiority in a data scarcity regime, also setting baselines for future research on the new dataset;
- a thorough evaluation of the adversarial robustness of our method and the baselines, showing the superior performance of the new resynthesis-based method when operating in adversarial conditions.

This paper extends a previous work of ours presented at WIFS 2025 [20], by providing a substantially broader and more thorough evaluation of few-shot source attribution under post-processing conditions. In particular, we expanded the experimental setting by introducing six additional post-processing operators — namely CMYK and HSV color-space conversions, median filtering, print-and-scan simulation [21], social media compression via X/Twitter upload, and white noise addition - and by systematically measuring robust attribution performance across varying numbers of shots and classes. We also evaluated several feature spaces and distances, ranging from geometric distances calculated on general-purpose representations such as CLIP [13], DINO [22], and ViT [23], to perceptual similarity metrics such as DISTS [24] and LPIPS [25]. We also considered face recognition features extracted with FaceNet [26], and GhostFaceNet [27]. Our experiments showed that the best performance is consistently achieved when using CLIP-based features. Finally, we carried out a new set of experiments to assess the adversarial robustness of the proposed method and of competing baselines, in both full-training and few-shot configurations. Since conventional logit-based attacks [28], [29] cannot be directly applied to our resynthesis-based approach, we evaluated robustness by using the Layerwise Origin-Target Synthesis (LOTS) attack [30]. The results demonstrate that the proposed method exhibits very good adversarial robustness, requiring significantly stronger distortions to succeed compared to attacks against baseline methods.

II. RELATED WORK

Current synthetic image attribution methods primarily aim at capturing the distinctive fingerprint characteristics of different generative models by training a classifier. Based on the training paradigm, these methods can be broadly categorized into fully-supervised and few-shot approaches. Fully-supervised methods rely on training dedicated classifiers using large-scale collections of generated images. In contrast, few-shot methods leverage the representation capabilities of pretrained

¹ SIA is carried out within a closed set of known generators always including the one that actually generated the to-be-attributed image.

models to perform attribution with only a limited number of training samples. Another class of methods is represented by inversion-based approaches, which exploit access to the generative models to invert the image generation process. Methods from the various categories are reviewed in the following subsections.

A. FULLY-SUPERVISED SOURCE ATTRIBUTION

Fully supervised source attribution methods train classification models on labeled data to learn generator-specific fingerprints, leveraging the availability of training datasets. Over time, these approaches have evolved from exploiting explicit low-level artifacts to learning more abstract, content-invariant representations, paralleling the shift from early GAN-based synthesis to diffusion and large-scale text-to-image models. Early studies focus on stable artifacts embedded in generated images. Yu et al. [5] show that convolutional neural networks can implicitly capture generator fingerprints in the pixel domain. Frank et al. [16] identify grid-like spectral patterns introduced by GAN upsampling and demonstrate that frequency-domain cues improve robustness under post-processing. While effective, these methods primarily rely on low-level artifact signals and are mainly evaluated in GAN-based settings. Subsequent works emphasize learning content-irrelevant and architecture-consistent representations. GFD-Net [31] disentangles generator traces from semantic content through multi-task learning. DNA-Det [32] employs patch-wise contrastive learning to capture architecture-consistent fingerprints. RepMix [33] incorporates representation mixing and regularization to encourage semantics-invariant features and improve robustness to compression and resolution variations. Compared with earlier artifact-driven methods, these approaches introduce explicit mechanisms, such as disentanglement and contrastive consistency, to reduce content dependency in fingerprint learning and improve cross-domain generalization.

Recent research moves toward model-agnostic formulations and the reuse of pretrained backbones. ManiFPT [34] models fingerprints as residual artifacts obtained by projecting synthetic images onto a real-image manifold, framing attribution as deviation modeling. Pretrained multimodal and vision backbones are also increasingly adopted. DE-FAKE [9] utilizes BLIP-generated captions [35] together with CLIP-based image and text features [13] for source classification, and Xu et al. [36] leverage pretrained vision encoders to model RGB, stylistic, and mid-level cues. Diffusion-based approaches such as MAID [37] and MADE [38] reinterpret pretrained diffusion models as feature extractors, using internal activations or inverse-diffusion representations for attribution. Collectively, these works demonstrate an increasing reliance on transferable representations derived from large pretrained models.

Overall, fully supervised attribution progresses along three directions: from explicit artifacts to abstract representations, from GAN-centered analysis to framework-agnostic modeling, and from task-specific architectures to pretrained backbone reuse. Despite strong closed-set performance, these

methods require retraining or classifier expansion when new generators emerge. Techniques like incremental learning [39], metric learning [40], lazy learning [14], and LoRA-expandable networks [41] aim at improving scalability; however, they still depend on supervised updates and predefined label spaces. This reliance on labeled retraining limits adaptability in a rapidly evolving generative ecosystem.

B. FEW-SHOT SOURCE ATTRIBUTION

Recent works have explored the few-shot learning paradigm to address attribution under data scarcity conditions. Bindi et al. [10] proposed a lightweight framework based on tiny autoencoders, where a shared decoder is pretrained on natural images and model-specific encoders are fine-tuned using only a few generated samples. Reconstruction errors are then used as discriminative features and classified via a linear SVM, enabling incremental integration of new generators at a relatively low cost. Lee and Yi [42] addressed few-shot attribution from a class-incremental learning perspective, introducing an adaptive integration module that fuses multi-level CLIP-ViT features to capture generator-specific fingerprints. Wu et al. [43] and Uruena et al. [44] instead rely on contrastive representation learning to obtain generator-discriminative embeddings, which are then used with metric-based or non-parametric classifiers for attribution with limited samples.

In contrast to these supervised approaches, Bongini et al. [20] proposed a training-free attribution strategy. Instead of learning generator-specific representations, their method extracts a caption from the input image and re-generates images using each candidate model. The source is inferred by measuring similarity between the re-generated outputs and the original image, thus avoiding task-specific supervision.

Overall, few-shot attribution methods can be broadly divided into supervised representation learning and training-free resynthesis paradigms. Supervised approaches require labeled samples for each generator and rely on feature adaptation or metric learning. They generally provide strong discriminability within a predefined generator set, but extending them to unseen models necessitates fine-tuning or representation updates, limiting scalability when new generators emerge. Training-free methods, instead, eliminate supervision costs and simplify integration of new models. On the downside, it requires black-box access to the candidate generators and incurs computational overhead due to repeated regeneration. These distinctions reflect a fundamental trade-off between supervision requirements, scalability, and deployment efficiency in few-shot source attribution.

C. INVERSION-BASED SOURCE ATTRIBUTION

The methods discussed above are fingerprint-based, as they rely on learning discriminative representations that capture generator-specific artifacts. In contrast, inversion-based methods exploit the latent reconstruction capability of generative models [11], [45], [46]. These methods generally follow a two-step paradigm: for a given probe image, latent inversion

is performed, typically via gradient descent, independently for each candidate generator to obtain the latent image that best reconstructs the image; then, the reconstruction errors of all candidate generators are compared, and the generator yielding the lowest error is identified as the source. The underlying assumption is that an image can be most faithfully reconstructed by the generator that originally produced it. While sharing this principle, existing methods differ mainly in their optimization objectives and reconstruction metrics. Despite being conceptually straightforward and effective, such optimization is computationally intensive and highly sensitive to the non-convex structure of latent spaces, which may lead to suboptimal convergence and occasional wrong attribution.

III. METHOD

Resynthesis is used for image classification in a variety of different scenarios [47]. In a real-world case, the prompt used to generate a synthetic image is usually unavailable. Given an image to attribute, our approach relies on a secondary description of the *original* image to build *resyntheses* (resynthesized versions of the image) with all the candidate sources. The description can be obtained with any captioning method.² The secondary description generation is fully automated through the use of the Chat-GPT API.³ In this panel of *resyntheses*, the *resynthesis* which has the closest resemblance to the *original* image should be the one generated by the same model. Consequently, attribution can be obtained by distance calculation: the *original* image is attributed to the generator that produced the closest *resynthesis*. The method relies on the intuition that each generator has its own style, reflecting a high-level, distinctive manner of rendering image content. This, combined with the fact that the content is very similar to that of the original image - the two images originating from similar prompts -, provides the rationale for identifying the original source as the one that produces the closest resynthesis. Formally, given a set of n candidate text-to-image sources $S = \{s_1, \dots, s_n\}$, and a distance function d , an image I is attributed to the source s_j by extracting a prompt $p(I)$ describing the content of I , then obtaining a set of *resyntheses* $\{r_j = s_j(p(I)) \mid \forall s_j \in S\}$, and finally calculating the distances $\{d_j = d(r_j, I) \mid \forall s_j \in S\}$. Defining a distance function $d(\cdot, \cdot)$, the image I is attributed to the source s_{j^*} achieving the minimum distance, as described in (1).

$$s_{j^*} : j^* = \arg \min_j (d(s_j(p(I)), I)). \quad (1)$$

Moreover, the distance measurement can be performed in the pixel space or in any feature space of choice. To measure the distance in the pixel space, due to the infeasibility of using a geometric distance function [12], we employed two widely-used perceptual similarity metrics: DISTs [24] and LPIPS [25]. Alternatively, pretrained feature extractors can be used to extract a feature vector for every image. In this

scenario, the distance measurement happens in the model's feature space. The distances between the feature vector of the *original* image and the feature vector of each *resynthesis* are measured. The *original* image is then attributed to the model which produced the *resynthesis* with the smallest distance from the *original* itself. In the experimentation, we used three different pretrained feature extractors: CLIP [13], DINOv2 [22] (referred to as DINO in the rest of the paper), and Google's ViT [23] (referred to as ViT in the rest of the paper). Defining the feature extraction as an operator $E(\cdot)$, we can rewrite the equation as in (2).

$$s_{j^*} : j^* = \arg \min_j (d(E(s_j(p(I))), E(I))). \quad (2)$$

We tested several distances d , including Euclidean, Manhattan, and Correlation distances in CLIP's features space [13], DINO's feature space [22], and ViT's feature space [23]. In summary, we experimented with the following variants of our resynthesis method:

- *Res-CLIP*: A pretrained CLIP-Large (input size 336×336) model⁴ is used as a feature extractor and the distance is measured in the corresponding feature space, using (2). The distance can be Euclidean (Res-CLIP-E), Manhattan (Res-CLIP-M), or Correlation (Res-CLIP-C).
- *Res-DINO*: A pretrained DINOv2 [22] model is used as a feature extractor, subsequently measuring the Euclidean (Res-DINO-E), Manhattan (Res-DINO-M), and Correlation (Res-DINO-C) distances, using (2).
- *Res-ViT*: A pretrained ViT [23] model (input size 224×224) is used as a feature extractor, and the distance is measured in the corresponding feature space, using (2). The distance can be Euclidean (Res-ViT-E), Manhattan (Res-ViT-M), or Correlation (Res-ViT-C).
- *Res-DISTS*: DISTs [24] is a perceptual image similarity metric designed to capture both structural and textural differences between images. It has been widely used for image quality assessment tasks. In our experiments, we computed DISTs between each test image and candidate reference images, using the VGG16 backbone [48] with pretrained weights. The images were resized to 256×256 and attributed using (1).
- *Res-LPIPS*: LPIPS [25] (Learned Perceptual Image Patch Similarity) is a deep perceptual similarity measure based on feature embeddings from pretrained CNNs. It is commonly used to evaluate perceptual closeness between images. In our setup, LPIPS was computed using the VGG16 backbone [48] with pretrained weights, resizing all images to 256×256 and attributed using (1).
- *Res-Face*: facial recognition features are the most natural choice when analyzing a dataset of face images, as demonstrated by the DeepFace framework [49]. In particular, we used two deep neural networks for face recognition to extract informative features usable for resynthesis: FaceNet [26] (Res-Face-F) and Ghost-FaceNet [27] (Res-Face-G).

²<https://openai.com/chatgpt>

³<https://openai.com/it-IT/api/>

⁴<https://huggingface.co/openai/clip-vit-large-patch14-336>

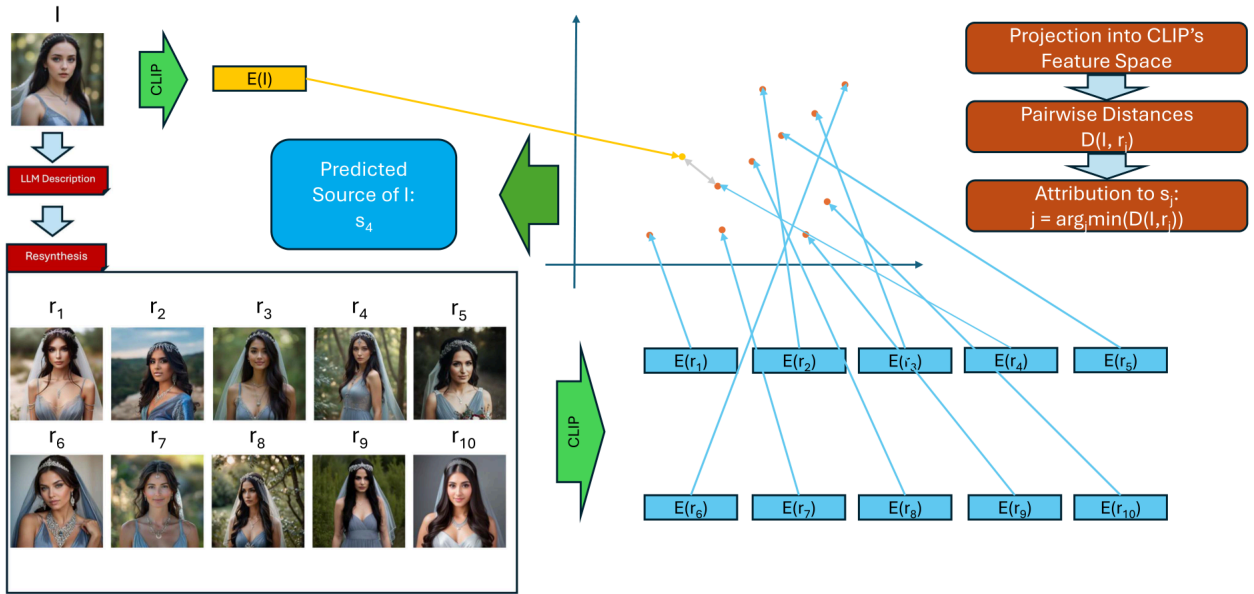


FIGURE 1. Workflow of the proposed source attribution method based on resynthesis. The *original* image I in this example was generated by Freepik. From left to right, the *resyntheses* r_j were produced by: Bing, Firefly, Flux-dev, Freepik, Imagen3, Leonardo AI, Midjourney, Nightcafe, Stable Diffusion 3, Starry AI. The features of I and all the r_j are extracted with CLIP. The distances between the projections $E(I)$ and each $E(r_j)$ in CLIP's feature space are measured, and I is attributed to the source s_{j^*} with the minimum distance $d(E(I), E(r_{j^*}))$. In this example, the image is correctly attributed to s_4 , which corresponds to Freepik.

The feature extraction models, the perceptual similarity models, and the facial recognition networks were not fine-tuned, and they were used only with frozen weights. Therefore, our model is training-free and can be considered a one-shot method. In fact, the attribution of an image consists in the extraction of the secondary description with a generic image-to-text language model (Chat-GPT in our case), the generation of one *resynthesis* for each candidate source, and finally the distance minimization using either a perceptual similarity metric or a distance function in the chosen feature space. The exact computational cost of the resynthesis approach is difficult to quantify precisely. The method requires producing one shot for every candidate generator. The more expensive operations are captioning and image generation, which are carried out through web servers. Both the extraction of the caption through Chat-GPT API and the resynthesized image generation depend on the hardware used by service providers, usage of the service at the time of request, and the internet connection. Based on our experiments, the average response time is in the order of a handful of seconds. Feature extraction is instead carried out locally on an Nvidia A100 GPU, and takes around 0.12 s per image for CLIP (and a slightly shorter but comparable time for ViT and DINO). Distance calculation and selection of the best candidate is carried out on the CPU, and its cost is negligible. A summary of our methodology is shown in Fig. 1.

IV. DATASET

To evaluate the attribution performance of our method, also in comparison with relevant few-shot baselines, we built a new

dataset⁵. The main novelty points introduced by our dataset are: i) images generated by 14 generators, ii) 7 commercial generators, iii) integration of the resynthesis mechanism with the inclusion of *original* images and their corresponding *resyntheses*. The dataset is composed of a total of 12,000 head-and-shoulder images of novel characters, produced with 14 text-to-image generators described in Section IV-A. The dataset building methodology is described in Section IV-B.

A. GENERATORS

To create the dataset, we used 7 open-source text-to-image generative models (4 of them on a subset of the 100 characters) and 7 commercial text-to-image generative models. All these sources are listed in the following, with the (*) symbol identifying commercial generators, and the (e) symbol identifying the generators used only on a subset of characters:

- Bing (DALL·E 3)⁶(*)
- Firefly⁷(*)
- Flux.1.dev⁸(*)
- Freepik⁹(*)
- Imagen3¹⁰(*)
- Leonardo AI¹¹(*)

⁵The dataset is publicly available on Kaggle at the following URL: <https://www.kaggle.com/datasets/pietrob92/resynthesis-dataset>

⁶<https://openai.com/it-IT/index/dall-e-3/>

⁷<https://firefly.adobe.com/>

⁸<https://github.com/black-forest-labs/flux>

⁹<https://docs.freepik.com/api-reference/mystic/post-mystic>

¹⁰<https://deepmind.google/models/imagen/>

¹¹<https://leonardo.ai>

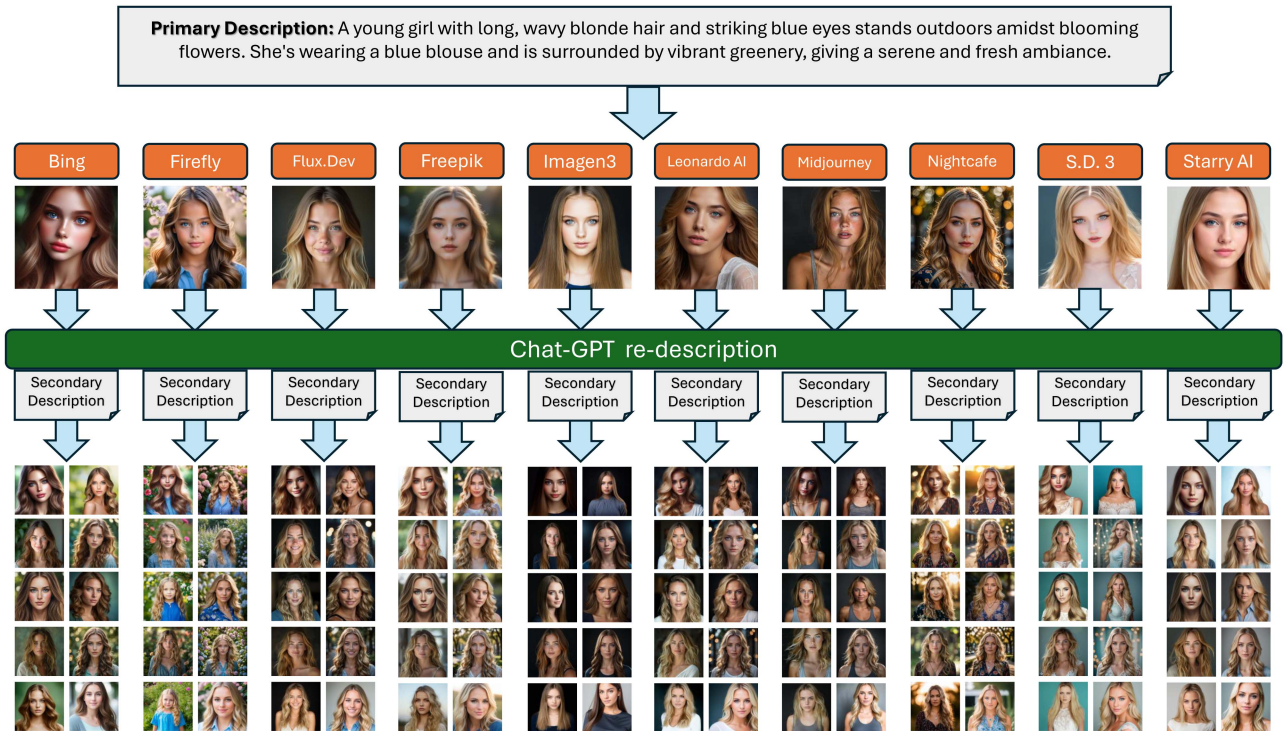


FIGURE 2. Dataset building procedure. This example corresponds to *character #10: Juliet (Romeo and Juliet)*. Starting from the top and going down: The primary description is used as a prompt for the 10 core generative models introduced in Section IV-A, which produce the images in the first row. The models are in alphabetical order, from left to right. Each image is then passed to Chat-GPT to obtain a secondary description. The latter is used to generate the *resyntheses* in the bottom double-column groups, using all the 10 core generators again. In each group of *resyntheses*, from left to right and from top to bottom, the generators are again in alphabetical order.

- Midjourney¹² (*)
- Nightcafe¹³ (*)
- Stable Diffusion 3¹⁴ (*)
- Starry AI¹⁵ (*)
- AuraFlow¹⁶ (e)
- Pixart¹⁷ (e)
- Playground v2.5¹⁸ (e)
- Tencent Hunyuan¹⁹ (e)

B. DATASET CONSTRUCTION

We used 100 descriptions of novel characters, summarized with Chat-GPT, to get prompts of uniform length (with a maximum limit of 200 text characters). We built an index on the *character* axis and used it to split the dataset. We split the dataset into training, validation, and test sets of 80, 10, and 10 characters, respectively. This prevents images originating from the same prompt from ending up in different sets and guarantees that the classes are always balanced. We then generated, respectively, 10 *original* images for every such

prompt belonging to the training or validation set, and 14 *original* images for every such prompt belonging to the test set (one with each of the text-to-image generators described in Section IV-A). Consequently, we obtained 1,040 *original* images, 100 for every core source and 10 for every other source. These *original* images are linked into groups of images produced from the same description by the *character* index. To apply the resynthesis methodology, we extracted a *secondary* description of each *original* image. This *secondary* description, intended as an input prompt for text-to-image generation, was produced by Chat-GPT. The input command for Chat-GPT was: “Provide a detailed description of the character represented in this image in less than 200 characters”. *Original* images of the same character from different sources have different *secondary* descriptions, based on the semantical elements introduced by the models and on low-level signatures that can influence language model style. For each *secondary* description, we generated 10 or 14 *resyntheses* (again, one with each model) respectively for training/validation characters and test characters. The *resyntheses* of each *original* image come from the same secondary description and are linked together. They are also linked to the other *original* images from the same *primary* description and to all their *resyntheses*, constituting groups of 100 or 140 *resyntheses* originating from the same prompt. Consequently, the training set is composed of 800 *original* images (OrTr) and their 8,000 *resyntheses* (ResTr), while the validation set is composed

¹²<https://www.midjourney.com>

¹³<https://creator.nightcafe.studio>

¹⁴<https://huggingface.co/stabilityai/stable-diffusion-3-medium-diffusers>

¹⁵<https://starryai.com/>

¹⁶<https://huggingface.co/fal/AuraFlow>

¹⁷<https://huggingface.co/PixArt-alpha/PixArt-Sigma-XL-2-1024-MS>

¹⁸<https://huggingface.co/playgroundai/playground-v2.5-1024px-aesthetic>

¹⁹<https://huggingface.co/Tencent-Hunyuan/HunyuanDiT-Diffusers>

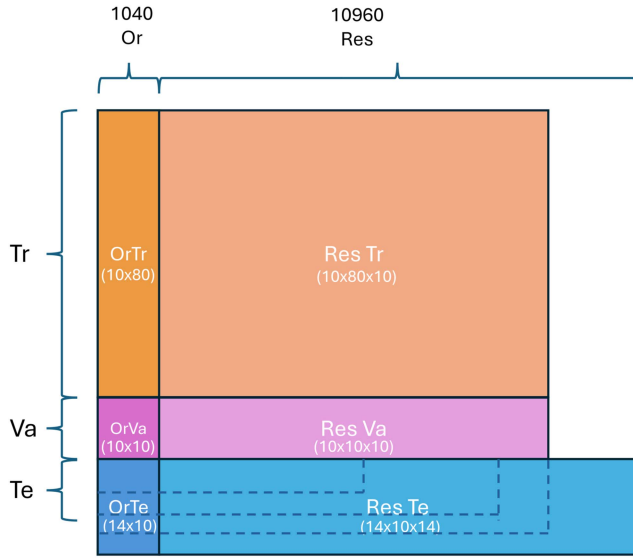


FIGURE 3. Dataset composition and split. The dataset is split along the *character* index, allocating 80 characters to the training set, 10 to the validation set, and 10 to the test set. The characters in the training and validation sets are generated with the 10 core sources, obtaining, respectively, 800 *originals* (OrTr) and 100 *originals* (OrVa). The characters of the test set are generated with all the 14 sources, obtaining 140 *originals* (OrTe). Then, the secondary descriptions of all the *originals* are extracted with Chat-GPT and used to generate 10 *resyntheses* for each training and validation character, and 14 *resyntheses* for each test character. Therefore, this second generation round produces a set of 8000 *resyntheses* of training set characters (ResTr), a set of 1000 *resyntheses* of validation set characters (ResVa), and a set of 1960 *resyntheses* of test set characters (ResTe). The character-based split ensures that all the 10 or 14 *original* images of each character and their 100 or 196 *resyntheses* belong to the same set. Each subset is represented in the figure according to its size, reported in parentheses: the first dimension corresponds to the generators used, and the second dimension is the number of characters. For subsets of *resyntheses*, we have a third dimension, which is the number of generators used for resynthesizing each *original*. The test set contains some sub-splits related to the experiments with variable numbers of classes (5, 8, and 10 classes - the dashed lines account for this). When 14 classes are used, the whole test set is employed.

of 100 *original* images (OrVa) and their 1,000 *resyntheses* (ResVa). The test set is composed of 140 *original* images (OrTe) and their 1960 *resyntheses* (ResTe). The procedure is shown in Fig. 2, while a scheme of the results is shown in Fig. 3.

For a detailed explanation of how the dataset is used throughout the experiments, please see the Appendix A1.

V. EXPERIMENTS

A. BASELINES

1) CLIP FEATURE CLASSIFIERS

CLIP-Large (input size 336×336) is widely used as a feature extractor for many downstream tasks. We use it to get image features, which are then sent to a small classifier: either a Multi-Layer Perceptron (CLIP+MLP) or a Support Vector Machine (CLIP+SVM). The CLIP parameters are frozen and not fine-tuned. The hyperparameters of both heads were optimized on the validation set. For CLIP+MLP, the architecture

has two hidden layers with 512 and 32 units, ReLU activation, and a softmax output layer. In few-shot scenarios, the architecture is reduced to a single hidden layer of 512 units to reduce overfitting. The MLP was trained using Adam [50], with initial learning rate 10^{-3} , and $l_{2\alpha}$ set to 10^{-4} , for a maximum of 1,000 epochs. Early stopping was applied on the validation loss with a patience of 10 epochs. CLIP+SVM employed a Radial Basis Function (RBF) kernel with regularization $C = 1.0$, tolerance 10^{-3} , and no constraints on the number of iterations.

2) DINO FEATURE CLASSIFIERS

We used DINOv2 [22] as a feature extractor for feeding simple classifiers in a similar way to the procedure we already described for CLIP. Also in this case, we focus on an MLP and an SVM classifier. For DINO+MLP, the architecture is identical to that of CLIP+MLP (Two hidden layers with 512 and 32 units, ReLU activation, and a softmax output layer). Also in this case, the MLP is trained with the Adam optimizer [50], with an initial learning rate of 10^{-3} , $l_{2\alpha}$ equal to 10^{-4} , a maximum of 1,000 epochs, and an early stopping mechanism applied to the validation loss with a patience of 10 epochs. Also in this case, in few-shot scenarios, the architecture is reduced to a single hidden layer of 512 units to reduce overfitting. DINO+SVM, similarly to CLIP+SVM, uses a Radial Basis Function (RBF) kernel, with regularization coefficient $C = 1.0$, tolerance 10^{-3} , and no constraints on the number of iterations.

3) ViT FEATURE CLASSIFIERS

We also used Google's ViT [23] as a feature extractor for feeding simple classifiers, as already described for CLIP and DINO. In this case, we had to resort to a more complex MLP architecture, composed of three hidden layers of 1024, 512, and 32 units, respectively. All the hidden layers have ReLU activation, and the output layer has softmax activation. The Adam optimizer [50] is used for training the MLP with an initial learning rate of 10^{-3} , $l_{2\alpha}$ equal to 10^{-4} , a maximum of 1,000 epochs, and an early stopping mechanism applied to the validation loss with a patience of 10 epochs. In few-shot experiments, the architecture is shrunk to a single hidden layer of 1024 units to reduce overfitting. ViT+SVM, similarly to the other two feature classifiers based on SVMs, uses a Radial Basis Function (RBF) kernel, with regularization coefficient $C = 1.0$, tolerance 10^{-3} , and no constraints on the number of iterations.

4) DE-FAKE

De-Fake [9] is a model used both for detection and attribution that works very well in few-shot scenarios. The architecture comprises a detector and an attributor. For SIA, only the attributor model is used. This network looks for slight artifacts and patterns in the image, which are typical of a certain generator. De-Fake experiments were carried out with the default parameters indicated in the paper [9].

5) CLIP-LoRA

CLIP-LoRA [51] is a Low Rank Adaptation (LoRA) module for CLIP models. The LoRA layers are applied to the model and trained, without modifying the CLIP weights. This procedure allows us to fine-tune the model for few-shot tasks. We apply the LoRA adapter to CLIP Large. The classification head is an MLP with two hidden layers of 768 and 1024 units, respectively, both with ReLU activation. The model is fine-tuned for 12 epochs (10 in few-shot experiments) using the Adam optimizer [50] with an initial learning rate of 10^{-3} , and a batch size of 16.

6) EFFICIENTNETB4

Convolutional Neural Networks (CNNs) are used with great success for image classification in a huge variety of scenarios. The model, which usually shows the best performance in SIA [52], is EfficientNetB4 [53]. It is difficult to fine-tune EfficientNetB4 without a large number of samples per class. As a consequence, our dataset is challenging for this model. We fine-tuned the network for 12 epochs (10 in few-shot experiments), using Adam [50], an initial learning rate of 10^{-4} , and a batch size of 16.

7) TINY AUTOENCODERS

Tiny Autoencoders [10] is a modular approach for SIA. For each plausible generator, a small CNN autoencoder is trained on a small number of images generated by that source. The image to attribute is given as input to each of these small networks. The autoencoder reproducing the image with the smallest reconstruction error should be the one trained on the real source of the image. A dense layer is responsible for aggregating the reconstruction errors of all the autoencoders and formulating a class probability prediction. We used the standard parameters introduced in [10].

B. TASKS AND SETTINGS

In our experimentation, the *resyntheses* are used for attributing the *originals* using our methodology. The baselines described in Section V-A are trained on a variable number of samples and compared to the various resynthesis methodology variants. The methods are evaluated using the Accuracy metric. Tests were carried out on the following tasks:

- T1 - Few-Shot SIA;
- T2 - Few-Shot SIA in the presence of post-processing;
- T3 - SIA in an adversarial setting.

Each task has a dedicated subsection in the following part of this Section. For each of them, a description of the experimental setup is followed by the presentation of the corresponding experimental results and their discussion.

C. T1 - FEW-SHOT SIA

1) T1 - EXPERIMENTAL SETUP

The goal of task T1 is to test the attribution performance of our method in comparison with state-of-the-art few-shot

baselines. In this scenario, a handful of examples are available to train the baselines. The test set always corresponds to a partition of the OrTe dataset section (see Section IV). This partition corresponds to the test examples contained in OrTe that belong to the classes selected for the experiment at hand. Since our method does not require training, we just measured the training-free performance of our method on the test set on different numbers of classes (5,8,10,14). The baseline experiments were carried out with different numbers of shots (1,2,3,5,7,10,15,20) and different numbers of classes (5,8,10,14). For every experiment, we averaged the results on 5 repetitions, each with a different seed for weight initialization and with different training examples (the same for all the models). For the baselines, the training samples are limited to the number of shots. For more details about the splitting procedure, see Appendix I. Our resynthesis approach does not use the training samples, attributing the *original* images by measuring the distance from their *resyntheses*.

2) T1 - RESULTS AND DISCUSSION

Table 1 displays Accuracy on task T1, when the baselines are trained with a fixed number of shots per class (10 and 7): these values were fixed based on the observations reported in Fig. 4, which allow to determine that the turning point where baselines start to be competitive with our method is around 10 shots. The resynthesis methods yield identical results in both settings. The number of classes is fixed at 10. In this 10-shot setup, CLIP+MLP emerges as the top-performing approach, closely followed by CLIP LoRA and the three CLIP-based resynthesis variants. The advantage of this group of methods can be attributed to the highly informative potential of CLIP features on both low-level and high-level characteristics. Among the resynthesis variants, Res-CLIP-E achieves the best results. This suggests that, without post-processing, the Euclidean distance guarantees a small advantage with respect to other distance measures. Moreover, the MLPs and SVMs processing CLIP features, respectively, show better performance than the MLPs and SVMs processing DINO or ViT features. This is likely due to the fact that DINO and ViT features are less informative than CLIP's for the task at hand. This difference is even more evident when looking at the non-outstanding results of resynthesis variants calculated in the feature spaces of DINO and ViT, regardless of the distance function of choice. Additionally, the two resynthesis variants based on perceptual similarity (Res-DISTS and Res-LPIPS) perform far worse than the resynthesis variants based on feature extraction. Interestingly, even features extracted with facial recognition networks (Res-Face-F, Res-Face-G) - which were the most natural choice on our dataset of facial images - cannot compete with CLIP features for resynthesis, underlying the highly informative potential of CLIP features for our objective. This advantage with respect to features extracted with other methods is likely given by characteristics of the model architecture and the data used to pretrain it, which make CLIP capture both local patterns and global semantics [54].

TABLE 1. Plain Source Attribution Accuracy on the Core Dataset (10 Generators) Under 7-Shot and 10-Shot Settings. Baselines are Trained With 7 or 10 Samples Per Class, While Our Resynthesis-Based Methods are Training-Free and Independent of the Shot Setting. For Res-CLIP/DINO/ViT, Distance Metrics Include Euclidean (E), Manhattan (M), and Correlation (C). For Res-Face, Features are Extracted Using FaceNet (F) and GhostFaceNet (G).

Shots	Baselines										Our Resynthesis-based Method												
	CLIP		DINO		ViT		Tiny AEs	CLIP LoRA	De-Fake	Eff.Net B4	Res-CLIP			Res-DINO			Res-ViT			Res-DISTS	Res-LPIPS	Res-Face	
	MLP	SVM	MLP	SVM	MLP	SVM					E	M	C	E	M	C	E	M	C			F	G
7	0.61	0.04	0.39	0.03	0.39	0.03	0.15	0.54	0.30	0.45	0.66	0.65	0.65	0.34	0.34	0.34	0.36	0.38	0.36	0.20	0.22	0.28	0.40
10	0.69	0.17	0.51	0.04	0.52	0.04	0.16	0.67	0.32	0.53													

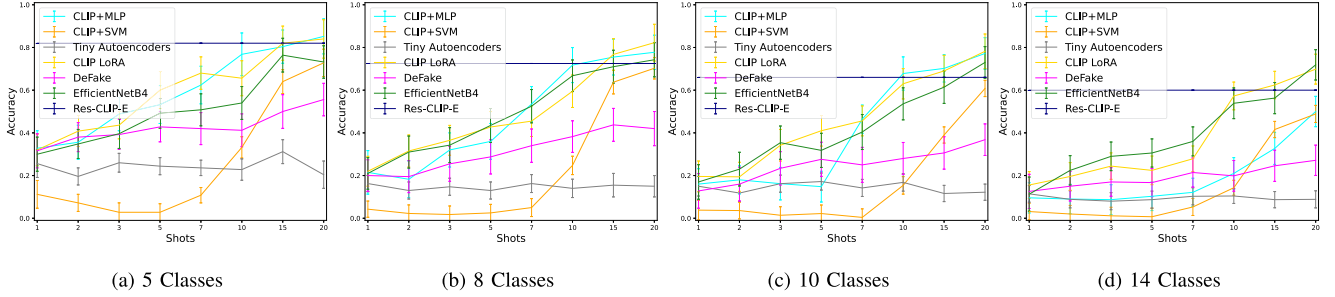


FIGURE 4. Accuracy in a few-shot scenario with different numbers of classes (T1). The Accuracy is plotted with respect to the number of shots available for training. The performance of our Res-CLIP-E method is represented by a navy-blue line. Since our method is training-free, the performance does not depend on the number of shots and is therefore constant. The intersection between the line representing our method's accuracy and a baseline's accuracy line corresponds to the number of shots needed by that baseline to outperform our training-free (one-shot equivalent) method. The baseline accuracy reported is obtained as the average accuracy over 5 runs, with errorbars representing the standard deviation.

The results of task T1 with variable numbers of shots and classes are reported in Fig. 4. For the sake of readability, we report only the best resynthesis methodology (Res-CLIP-E) and the best baselines (excluding ViT and DINO feature classifiers). Our resynthesis method outperforms the baselines when few examples are available, across all numbers of classes. Res-CLIP-E is the best method when 10 or fewer shots per class are available. This is shown by the navy-blue line representing the performance of Res-CLIP-E: line crossing always occurs at 10 or 15 shots. When more shots are available, the best methodologies are CLIP LoRA and CLIP+MLP (the latter shows bad performance on 14 classes, though). Concerning the baselines, since the results of DINO+MLP and ViT+MLP are significantly worse than those of CLIP+MLP, we report only the results of this latter model (see Table 1 for a comparison on task T2 on plain images). Moreover, we did not report results for DINO+SVM and ViT+SVM as these two baselines did not converge and their accuracy is seldom better than random guessing (see Table 1).

D. T2 - FEW-SHOT SIA IN PRESENCE OF POST-PROCESSING

1) T2 - EXPERIMENTAL SETUP

Based on the findings reported in Section V-C2, in all the following experiments, we considered only the Res-CLIP variants and excluded ViT and DINO feature classifiers, because their performance is not competitive with the other methods.

Task T2 evaluates the SIA performance of the resynthesis variants and the few-shot baseline methods on the OrTe

partition of the dataset, when the images are post-processed. We carried out 16 tests, using a different post-processing operator each time: Blur; Brightness (strengths from 1.2 to 2.4); CMYK conversion; Contrast (strengths from 1.2 to 2.4); Crop (central, 0.5 to 0.9 the original area); Greyscale; JPEG ($QF \in [50, 99]$); HSV Conversion; Median Filter (size of 3, 5, or 7 pixels); Print&Scan simulation [55]; Resize (0.4 to 2.0 the original size); Rotation (-5° to $+5^\circ$, aspect preserving); Social-Instagram (simulating an upload on Instagram); Social-Twitter (simulating an upload on Twitter); WEBP ($QF \in [50, 99]$); WhiteNoise, that is, addition of gaussian noise ($\mu = 0$, $\sigma \in [0.0625 - 0.25]$). The exact values of the parameters of the transformations is randomly chosen for every image, and the same value is used for all the methods.

A deeper investigation, involving multiple values of classes and shots, was carried out on the few-shot performance of our method and baselines in the presence of two quite disruptive post-processing operators, such as JPEG compression and Contrast enhancement. None of the baseline methods uses data augmentation with post-processing operations during fine-tuning. Since Res-CLIP-M is the most robust resynthesis method, we report the results of this variant here.

2) T2 - RESULTS AND DISCUSSION

We report the results obtained with 10 shots and with 7 shots in Table 2. Tiny AutoEncoders shows a very low performance, and CLIP+SVM performs well only on 15 or more shots (see Fig. 5 for reference). We compare the robustness of Res-CLIP-E, Res-CLIP-C, and Res-CLIP-M with that of the following baselines: CLIP+MLP, CLIP LoRA, De-Fake, and EfficientNetB4. On post-processed images, with 10-shots,

TABLE 2. Accuracy on Robust Source Attribution Under 10-Shot and 7-Shot Settings on the Core Dataset (10 Generators). Baseline Methods are Shown in the Upper Block of Each Setting, While Our Resynthesis-Based Variants are Shown in the Lower Block. Each Column Corresponds to a Different Post-Processing Operator Applied to Test Set Images. The Best Result of Each Column is Highlighted in Bold. The Operator Codes Correspond To: Bl = Blur, Br = Brightness, Cm = CMYK, Co = Contrast, Cr = Greyscale, H = HSV, J = JPEG, M = Median Filter, P = Print&scan, Re = Resize, Ro = Rotation, SI = Social(Instagram), ST = Social(Twitter), We = WEBP, WN = White Noise.

Shots	Method	Plain	Bl	Br	Cm	Co	Cr	G	H	J	M	P	Re	Ro	SI	ST	We	WN
10	CLIP+MLP	0.69	0.63	0.63	0.70	0.61	0.64	0.59	0.70	0.60	0.66	0.10	0.70	0.66	0.59	0.59	0.57	0.10
	CLIP-LoRA	0.67	0.60	0.57	0.63	0.59	0.55	0.56	0.63	0.58	0.59	0.11	0.57	0.60	0.61	0.59	0.65	0.10
	De-Fake	0.32	0.30	0.29	0.32	0.29	0.32	0.26	0.30	0.30	0.29	0.10	0.31	0.31	0.27	0.31	0.30	0.09
	Eff.NetB4	0.53	0.39	0.39	0.51	0.42	0.36	0.39	0.50	0.49	0.42	0.09	0.50	0.47	0.46	0.47	0.47	0.12
	Res-CLIP-E	0.66	0.62	0.62	0.66	0.52	0.50	0.54	0.66	0.53	0.66	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	Res-CLIP-M	0.65	0.63	0.61	0.65	0.56	0.54	0.57	0.65	0.55	0.64	0.55	0.65	0.63	0.59	0.60	0.50	0.54
7	CLIP+MLP	0.61	0.51	0.53	0.61	0.52	0.52	0.50	0.61	0.50	0.55	0.10	0.61	0.53	0.49	0.50	0.49	0.10
	CLIP-LoRA	0.54	0.49	0.45	0.51	0.45	0.44	0.47	0.51	0.53	0.49	0.10	0.52	0.51	0.54	0.55	0.50	0.10
	De-Fake	0.30	0.26	0.27	0.25	0.29	0.25	0.25	0.26	0.26	0.24	0.09	0.25	0.24	0.24	0.24	0.26	0.10
	Eff.NetB4	0.45	0.33	0.32	0.44	0.36	0.33	0.36	0.43	0.42	0.34	0.09	0.47	0.42	0.40	0.44	0.39	0.10
	Res-CLIP-E	0.66	0.62	0.62	0.66	0.52	0.50	0.54	0.66	0.53	0.66	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	Res-CLIP-M	0.65	0.63	0.61	0.65	0.56	0.54	0.57	0.65	0.55	0.64	0.55	0.65	0.63	0.59	0.60	0.50	0.54
7	CLIP+MLP	0.65	0.63	0.61	0.66	0.52	0.51	0.54	0.66	0.53	0.67	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	CLIP-LoRA	0.65	0.63	0.61	0.66	0.52	0.51	0.54	0.66	0.53	0.67	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	De-Fake	0.65	0.63	0.61	0.66	0.52	0.51	0.54	0.66	0.53	0.67	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	Eff.NetB4	0.65	0.63	0.61	0.66	0.52	0.51	0.54	0.66	0.53	0.67	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	Res-CLIP-E	0.65	0.63	0.61	0.66	0.52	0.51	0.54	0.66	0.53	0.67	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	Res-CLIP-M	0.65	0.63	0.61	0.66	0.52	0.51	0.54	0.66	0.53	0.67	0.53	0.67	0.62	0.58	0.59	0.47	0.49

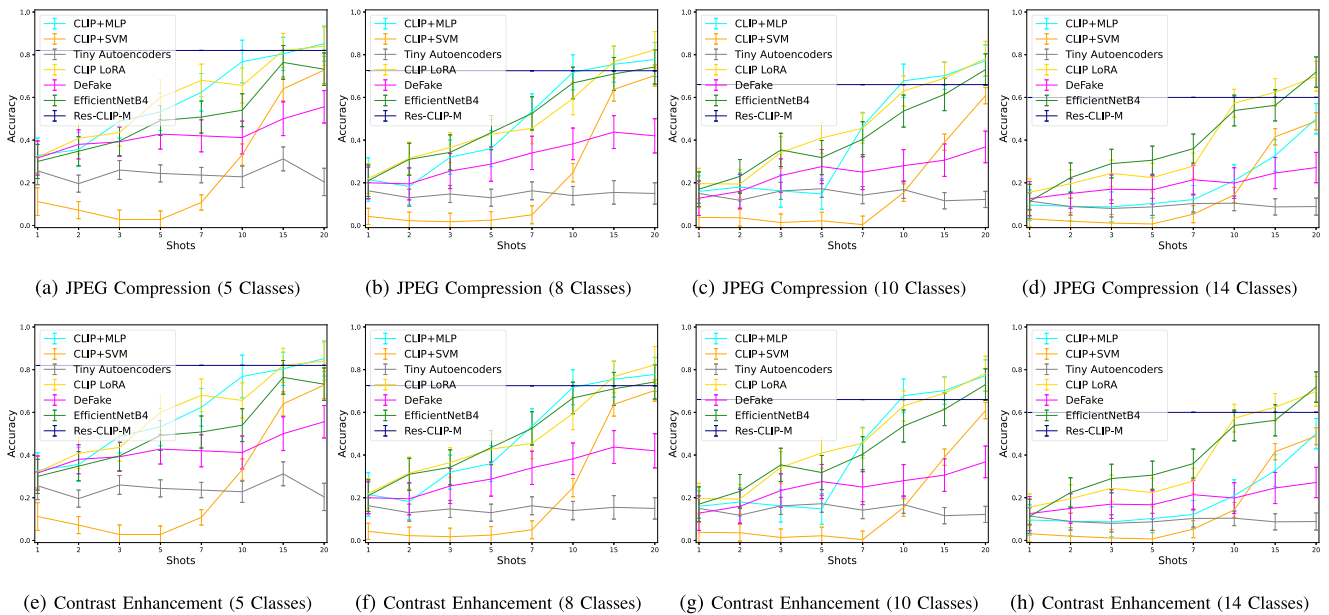


FIGURE 5. Few-shot accuracy in the presence of JPEG compression (top: (a)–(d)) and contrast enhancement (bottom: (e)–(h)) with different numbers of classes (T2). The Accuracy is plotted with respect to the number of shots available for training. The performance of our Res-CLIP-M method is represented by a navy-blue line. Since our method is training-free, the performance does not depend on the number of shots and is therefore constant. The intersection between the line representing our method's accuracy and a baseline's accuracy line corresponds to the number of shots needed by that baseline to outperform our training-free (one-shot equivalent) method. The baseline accuracy reported is obtained as the average accuracy over 5 runs, with errorbars representing the standard deviation.

the best-performing resynthesis variant is Res-CLIP-M. This suggests that the Manhattan distance gives a small robustness advantage with respect to other measures. Res-CLIP-M outperforms all the baselines trained with 10 shots on images post-processed with Blur, Twitter, and more evidently with Print&Scan and WhiteNoise. The latter two operators are the most disruptive when looking at the performance of all the other methods. Moreover, resynthesis variants based on CLIP (especially Res-CLIP-M) always follow closely the baseline method that obtained the best performance on all the other

operators, which is generally CLIP+MLP. Only on Instagram and WEBP compression, CLIP LoRA outperformed both CLIP+MLP and the resynthesis method. When the number of shots is fixed at 7 instead, resynthesis variants are always the best methods (with only a tie with CLIP LoRA on WEBP compressed images). In particular, Res-CLIP-M is the best overall methodology.

Fig. 5 shows the results of Task T2 on test images post-processed with JPEG compression and contrast enhancement, respectively. The results confirm the previous findings, with

Res-CLIP-M holding a very good performance, yet with a smaller advantage with respect to CLIP+MLP and CLIP LoRA, when compared to plain images (task T1). In particular, CLIP LoRA shows a very good robustness, both to JPEG compression and contrast enhancement, which allows it to outperform CLIP+MLP when the number of shots available for training is not too small. Moreover, the good performance of EfficientNetB4 and CLIP+SVM when the number of shots is not too small is confirmed. Since resynthesis is training-free and computationally equivalent to a one-shot method, our method (in particular, the Res-CLIP-M variant) is still the best attribution technique when 7 or fewer shots are available (in some cases even when 10 or fewer shots are available). These results, combined with the results reported in Table 2, allow us to assess the robustness of our method to image post-processing, and show that it maintains state-of-the-art performance in data scarcity conditions even with certain image distortions.

E. T3 - SIA IN AN ADVERSARIAL SETTING

1) T3 - EXPERIMENTAL SETUP

In task T3, we carried out an adversarial attack against our resynthesis method to evaluate its adversarial robustness and compare it with the adversarial robustness that can be achieved with the baselines. To attack our method, we resort to the Layerwise Origin-Target Synthesis (LOTS) attack [30]. This is a white-box attack that modifies the input image in order to move its internal deep feature representation to a target representation. When applied to CLIP, this means that the modification in the pixel space targets a change in the extracted CLIP feature vector. The final goal of the attack is to make the model misclassify the image as coming from a different source with respect to its real source. To do so, a vector in the target feature space must be chosen. The attack will iteratively modify the image to move its deep representation towards this target feature vector. The modification is optimized using backpropagation, using the feature-space distance from the chosen target as the loss function, and the image in pixel-space as the only optimizable tensor. In our case, we can choose the target among the different *resyntheses* of the image, trying to move the *original's* feature vector towards one of the wrong *resyntheses*. Specifically, we can choose to move it towards: a random wrong *resynthesis*, the wrong *resynthesis* which is the farthest from the correct *resynthesis* or from the *original* image (they often correspond), or the wrong *resynthesis* which is the closest to the correct *resynthesis* or the *original* image. After some preliminary experiments, we chose to target the wrong *resynthesis* which is closest to the *original* image, as this allowed to maximize the attack effectiveness.

Formally, let t be the chosen target feature vector, and I_a the adversarial image. We optimize I_a by initializing it with the same pixel values as those of the original image $I_a = I$ and minimizing its distance from the target in the feature space. To do so, we calculate the gradient of the feature space distance

between $E(I_a)$ and t with respect to I_a :

$$\eta(I_a, t) = \nabla_{I_a} (\|t - E(I_a)\|^2) \quad (3)$$

The attack is then iteratively refined by moving I_a in the direction opposite to the gradient direction. In order to keep the image distortion small, the learning rate of the optimizer should be low, and the attack is stopped as soon as I_a is misclassified. For a more detailed mathematical formulation, we make reference to the LOTS paper [30]. In particular, we backpropagate the loss using the Adam optimizer [50], with an initial learning rate $ILR = 10^{-3}$, and a maximum of 10,000 steps (never reached during the attacks). To the best of our knowledge, LOTS is the only white-box attack that can be adapted to attack the resynthesis methodology. Contrary to other methods that modify the input image to attack the classification logits, which are not defined in our case, LOTS attacks the image features before classification, making it possible to attack the resynthesis mechanism. To attack the baselines, gradient-based score-level attacks are considered, e.g., Carlini&Wagner (CW) attack [29] and the Projected Gradient Descent (PGD) attack [28]. As said, these algorithms do not apply to our case because they imply calculating and backpropagating an error function on the classification logits or final scores, which are not defined in our case. For the sake of simplicity, we attacked only the baselines that showed the most promising results in the previous tasks (T1 and T2): De-Fake, EfficientNetB4, CLIP LoRA, and CLIP+MLP. To decide the best way to attack the baselines, we carried out a set of preliminary experiments, described in the following.

The evaluation of attack success and consequently of adversarial robustness was based on three metrics: Attack Success Rate (ASR), Adversarial Accuracy (AdvAcc), and Mean Squared Error (MSE). The ASR is defined as the fraction of images that have been successfully attacked.

Since the images that are naturally misclassified by the model are not attacked, they are excluded from the ASR computation. This means that ASR measures the attack success without taking into account the base accuracy of the model. This information is exploited in the definition of the Adversarial Accuracy (AdvAcc) metric, as it measures the accuracy of the model after the attack:

$$\text{AdvAcc} = \text{Accuracy} * (1 - \text{ASR}) \quad (4)$$

Both ASR and AdvAcc can be defined as a function of the attack strength (the stronger the attack, the closer the ASR will be to 1.0 and AdvAcc to 0.0). Of course, a growing attack strength will also imply larger image distortion. To evaluate the image distortion introduced by the attacks, we measure the Mean Squared Error (MSE) as a pixel-space distance between the *original* image and its *attacked* version. Finally, to evaluate the adversarial robustness of our method and the baselines, we evaluate the MSE as a function of the ASR, allowing us to evaluate how disruptive the attack is at different levels of success. Since the attack should be as non-perceivable as possible, a growing MSE means that its effectiveness gets compromised, and its success rate (high ASR, low AdvAcc) is

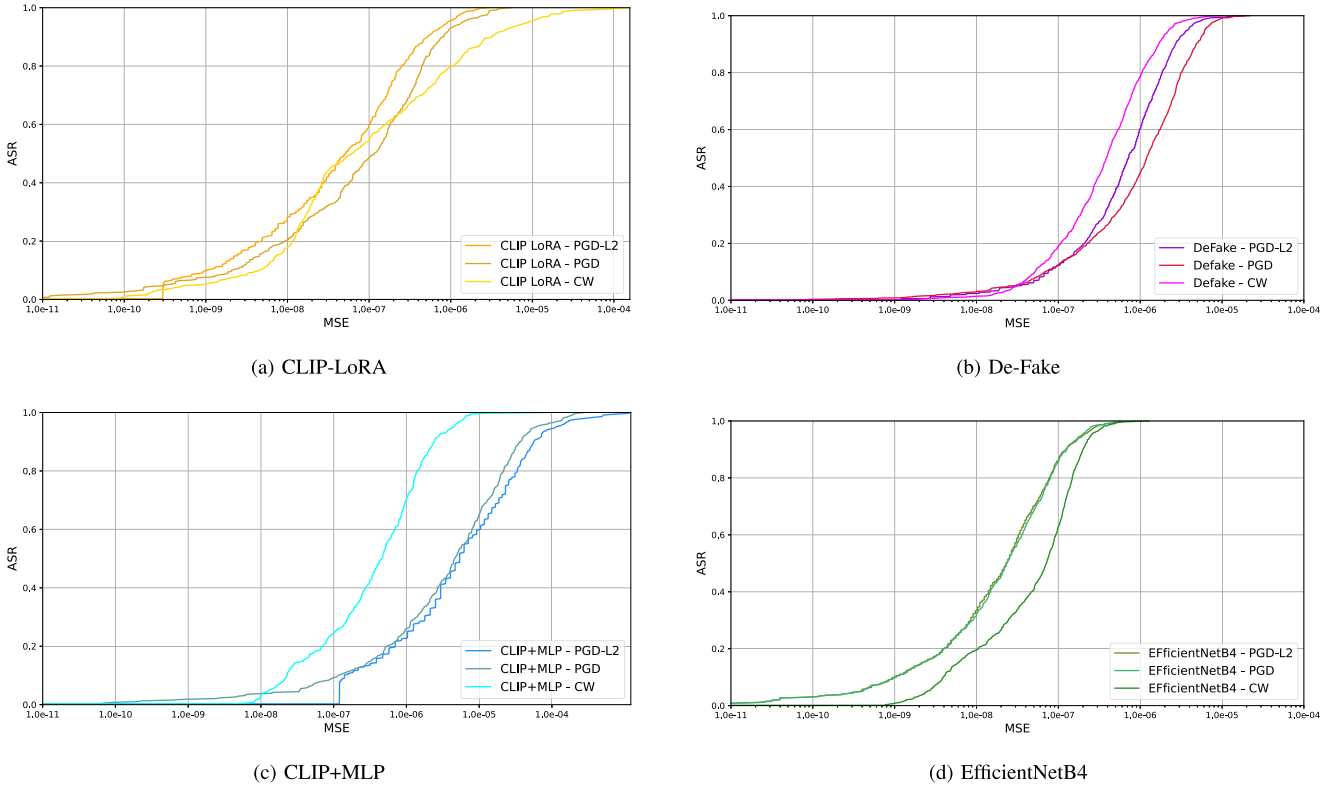


FIGURE 6. Attack Success Rate (ASR) of baselines in the presence of CW, PGD, or PGDL2. ASR is plotted as a function of the attack MSE. The best attack method for each baseline is, respectively, PGDL2 for CLIP-LoRA (a), CW for De-Fake (b), and CLIP+MLP (c), PGD for EfficientNetB4 (d).

less exploitable. Adversarial robustness is achieved when the attack success (high ASR, low AdvAcc) implies large image disruption (high MSE). In the final experiments (different from the preliminary ones), the metrics were measured on the attacks carried out on the test set images (OrTe).

2) T3 - PRELIMINARY ATTACKS

To decide the best way to attack the baselines, we carried out a preliminary round of experimentation, measuring the performance of various attack methodologies: CW [29], PGD [28], and the L2-norm version of PGD (PGDL2). In particular, to attack CLIP+MLP and De-Fake with CW, we used a learning rate of 10^{-3} , a weighting factor $C = 10^{-3}$, and confidence $K = 0.1$. To attack CLIP LoRA and EfficientNetB4 with CW, we used a learning rate of 10^{-2} , $C = 10^{-2}$, $K = 0.1$. The maximum number of steps was set to 1000 for all the models, which is never reached during the attacks. To attack the baselines with PGD, we used the following parameters: perturbation budget $\epsilon = 0.5$, step size $\alpha = 10^{-3}$, and a maximum of 1000 iterations. We used the same parameter values also for PGDL2, except $\alpha = 10^{-5}$.

To avoid biasing the attack towards the method that works best on test set images, we used the training set and the validation set (OrTr+OrVa) for these preliminary experiments. The results of this preliminary round of attacks are summarized in Fig. 6. As can be observed, the most convenient attack method (i.e., the attack method that

reaches ASR=1.0 or AdvAcc=0.0 with the lowest MSE) for each baseline is, respectively, PGDL2 for CLIP-LoRA, CW for De-Fake and CLIP+MLP, PGD for EfficientNetB4. In the following experiments, described in Section V-E3, we used these best performing attack methods for the baselines.

3) T3 - RESULTS AND DISCUSSION

Based on the findings of the preliminary experiments described above, the baselines were attacked using, respectively, PGDL2 (CLIP-LoRA), CW (De-Fake, CLIP+MLP), and PGD (EfficientNetB4). Our method (that cannot be attacked with CW, PGD, or similar methods) was attacked with LOTS using the procedure described in Section V-E1.

For the sake of readability, we report only the results of the most robust resynthesis methodology (Res-CLIP-M). The results of such attacks are summarized in Fig. 7. As can be seen, the attack always succeeded (eventually reaching ASR = 1.0 and AdvAcc = 0.0) on every image and on every model. The MSE required to do so can, though, vary much from one model to another. The adversarial robustness of baselines is much worse than that of the resynthesis method, as highlighted by the small MSE values to reach a near-zero AdvAcc or a near-one ASR, which is in the order of 10^{-6} . For our method, instead, a larger MSE is necessary to reach a similar near-zero AdvAcc or near-one ASR, which is about 7.0×10^{-5} .

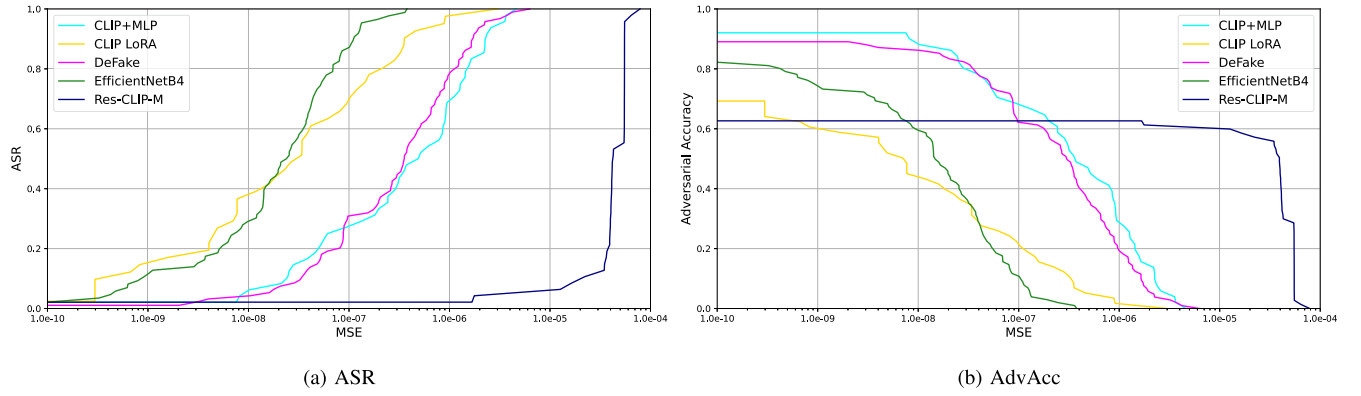


FIGURE 7. Attack Success Rate (a) and Adversarial Accuracy (b) of baselines and resynthesis in function of the MSE, when an adversarial attack is applied to the input images. De-Fake and CLIP+MLP are attacked with CW, EfficientNetB4 is attacked with PGD, CLIP-LoRA is attacked with PGDL2, while Res-CLIP-M is attacked with LOTS. The MSE needed to successfully attack resynthesis is significantly larger to those needed to successfully attack the baselines.

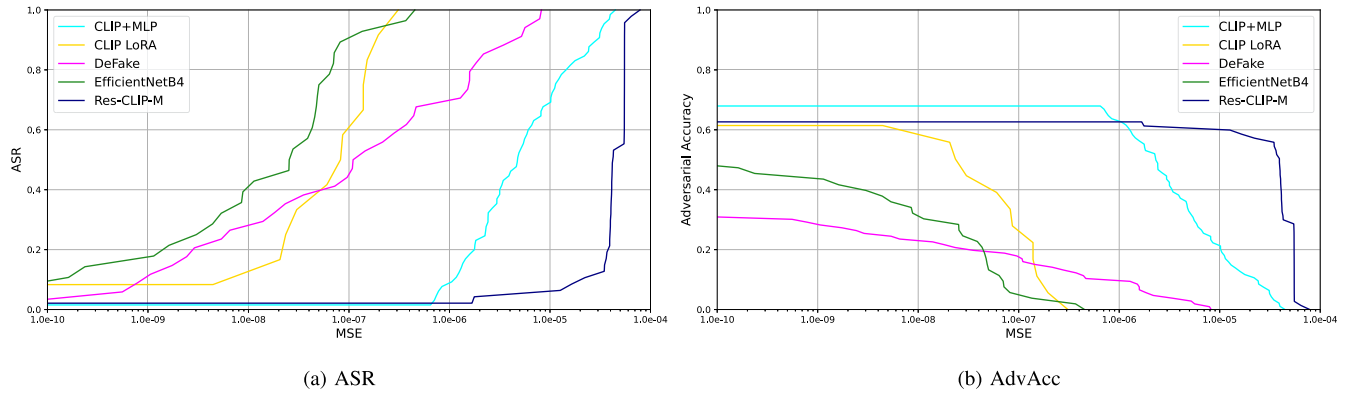


FIGURE 8. Attack Success Rate (a) and Adversarial Accuracy (b) of few-shot baselines and resynthesis in function of the MSE, when an adversarial attack is applied to the input images. In particular, in this case, baselines were trained on 10 shots. De-Fake and CLIP+MLP are attacked with CW, EfficientNetB4 is attacked with PGD, CLIP-LoRA is attacked with PGDL2, while Res-CLIP-M is attacked with LOTS. In this case, the gap in the MSE needed to successfully attack resynthesis is (expectedly) clearly larger than that of full-training models. The only exception is represented by CLIP+MLP, which is more robust to adversarial attacks (but way less accurate) in its few-shot version.

Moreover, the same experiments were also repeated on few-shot models to understand how much the adversarial robustness of baselines varies in this scenario. The same attack with the same parameters is used for each baseline. The results of this experiment are summarized in Fig. 8. The results show that the adversarial robustness gap between Res-CLIP-M and every baseline (except CLIP+MLP) is even larger in the few-shot case. In summary, experiments carried out in task T3 highlight a significantly higher adversarial robustness of the resynthesis method with respect to all of the baselines. Moreover, CLIP LoRA, which was the most robust baseline against post-processing, is not as robust in an adversarial setting. In this latter scenario, De-Fake and CLIP+MLP are more robust, yet neither of these baselines is as robust as Res-CLIP methods.

VI. CONCLUSION

We presented a new training-free method based on resynthesis for the attribution of the source of synthetic images, and a

dataset for few-shot source attribution that provides secondary descriptions and image *resyntheses*. We investigated different feature spaces and distances in order to develop the best resynthesis methodology, including vision transformers like CLIP, ViT, DINO, perceptual similarity metrics like DISTS and LPIPS, and facial recognition CNN features like FaceNet and GhostFaceNet. Our findings show that CLIP features are by far the most informative and robust for the method and task at hand, outperforming all the other extractors. The experimental results show that our method has very good performance with respect to state-of-the-art baselines, outperforming them consistently when 10 or fewer shots are available for training. Since our algorithm needs to generate just one *resynthesis* per class, it is equivalent to a one-shot method, and hence constitutes the best attribution method when the number of shots is between 0 and 10. This was demonstrated with different numbers of sources, ranging from 5 to 14. Moreover, when post-processing is applied to the images, our method shows very good robustness, outperforming the baselines in the

few-shot scenario. Adversarial attacks were carried out against our method and the baselines (both few-shot and full-training versions), showing that our method has a significantly higher adversarial robustness with respect to the baselines. Future work will focus on assessing the most suitable captioning model for the generation of secondary descriptions, including an investigation into different scenarios where the original prompt is produced with methods different from Chat-GPT. The possibility of using different captioning models to produce the secondary descriptions to be used as prompts for the resyntheses will also be explored. Moreover, it would be interesting to carry out experiments on real faces, synthesized and then resynthesized with generative models, by conducting them in full compliance with privacy requirements. In addition, new image categories will be incorporated into the dataset, and the performance of the resynthesis method will be evaluated on these expanded categories. In order to make our method applicable to real-world scenarios, as a further research direction, we plan to introduce a mechanism for open set rejection, to reject images coming from unknown sources or even real images.

APPENDIX

In the following, two appendices can be found that report additional details on the experimentation. In particular, Appendix I explains in detail how the dataset was used for resynthesis experiments, while Appendix II describes the results obtained with baselines when the full training set is available to train each model.

A. APPENDIX I: DATASET USAGE

For experiments involving regular full-training, like those described in Section V-E, the dataset is used following the split described in Section IV-B in training, validation and testing. For few-shot experiments, instead, only a small number of shots is considered for training the baselines (of course, this does not affect our method, which is training-free). Let k be the number of shots. For the cases with 5, 8, or 10 classes, the samples used for training are randomly drawn among the resyntheses of k characters (the same characters for each source) belonging to the training set, while the set used for testing is represented by the full test set. For experiments on 14 classes, where we only have the test set characters available in our dataset, we operate a partial leave- k -out split. Specifically, we randomly draw k *originals* among the 140 total characters. These *originals* are used as a training set and therefore never seen by the models. In particular, the training samples are their *resyntheses* (for every generator). The test samples are instead the other $140 - k$ *originals* of the test set, for experiments on 14 classes (in this case, the *originals* themselves are used as test examples, not their *resyntheses*). Moreover, to minimize possible biases, the *originals* to be used as a training set are always drawn in a way that does not allow more than one *original* from the same generator to be drawn when the required training shots are 14 or less. When

TABLE 3. Set of Generators Used in Attribution Setups With Different Numbers of Classes

Classes	Generators
5	Bing, Firefly, Flux.1.Dev, Freepik, Imagen3
8	Bing, Firefly, Flux.1.Dev, Freepik, Imagen3, Leonardo AI, Midjourney, Nightcafe
10	Bing, Firefly, Flux.1.Dev, Freepik, Imagen3, Leonardo AI, Midjourney, Nightcafe, Stable Diffusion 3, Starry AI
14	Bing, Firefly, Flux.1.Dev, Freepik, Imagen3, Leonardo AI, Midjourney, Nightcafe, Stable Diffusion 3, Starry AI, AuraFlow, Pixart, Playground, Tencent Hunyuan

the number of shots k is larger, the random drawing procedure ensures that at least one and at most two *originals* from each generator are drawn. Since each *original* carries one *resynthesis* for every generator, and the *resyntheses* are used as training examples, this procedure ensures that we have exactly k shots for every source to train the baseline models, and that these correspond to *originals* which are excluded from the test set. Since this procedure introduces a certain random component, few-shot experiments are always repeated, and an average value of each metric is calculated throughout the repetitions.

For the sake of simplicity, we also fix in advance the sets of sources to be used in experiments with 5, 8, 10, 14 classes, each of which is a successor set of the previous. We decided to simply select them in alphabetical order for the 10 core sources, and then again to select in alphabetical order the sources for which we have only generated test characters. In Table 3 we list these four sets of sources.

B. APPENDIX II: RESULTS OF FULLY TRAINED BASELINES

For completeness, we report the results obtained in plain and robust source attribution by training the baselines on the full training set. We use the core 10 generators of the dataset. Baseline training can be performed on the 800 examples of OrTr and/or on the 8,000 examples of ResTr. For each baseline, we trained three versions: one on OrTr only, one on ResTr only, and one on both subsets. We then selected the best version by measuring its accuracy on the validation set. In particular, for De-Fake, the best version was the one trained on OrTr only, while for all the other baselines, the best version was the one trained on ResTr only. In the following, we only report the results of the best version for each model. The results are summarized in Table 4. As we can observe, CLIP+MLP has a large advantage in terms of accuracy on plain data. As a consequence, and taking into account also its robustness, CLIP+MLP is the best method on every post-processing operator. Nevertheless, there is a tie with De-Fake on contrast enhancement, and another with Res-CLIP-M on white noise. Moreover, our method, and in particular the Res-CLIP-M variant, shows a solid robustness performance, especially on the most disruptive post-processing operators, which allows it to beat some of the baselines, notwithstanding

TABLE 4. Accuracy on Plain and Robust Source Attribution on the Core Dataset (10 Generators) When the Baselines are Trained on the Full Training Set of the Dataset. Baseline Methods are Shown in the Upper Block of Each Setting. The Performance of Our Resynthesis-Based Variants are Reported in the Lower Block. Each Column Corresponds to a Different Post-Processing Operator Applied to Test Set Images. The Best Result of Each Column is Highlighted in Bold. The Operator Codes Correspond To: Bl = Blur, Br = Brightness, Cm = CMYK, Co = Contrast, Cr = Crop, G = Greyscale, H = HSV, J = JPEG, M = Median Filter, P = Print&scan, Re = Resize, Ro = Rotation, SI = Social(Instagram), ST = Social(Twitter), We = WEBP, WN = White Noise.

Shots	Method	Plain	Bl	Br	Cm	Co	Cr	G	H	J	M	P	Re	Ro	SI	ST	We	WN
ALL	CLIP+MLP	0.96	0.86	0.89	0.96	0.68	0.84	0.71	0.96	0.78	0.95	0.60	0.96	0.89	0.88	0.89	0.75	0.54
	CLIP-LoRA	0.78	0.77	0.62	0.76	0.63	0.50	0.62	0.76	0.70	0.76	0.11	0.79	0.77	0.75	0.74	0.74	0.10
	De-Fake	0.72	0.65	0.67	0.71	0.68	0.72	0.64	0.71	0.72	0.70	0.15	0.72	0.70	0.71	0.70	0.72	0.16
	Eff.NetB4	0.61	0.63	0.57	0.61	0.60	0.37	0.64	0.61	0.59	0.60	0.10	0.62	0.61	0.60	0.59	0.60	0.16
	Res-CLIP-E	0.66	0.62	0.62	0.66	0.52	0.50	0.54	0.66	0.53	0.66	0.53	0.67	0.62	0.58	0.59	0.47	0.49
	Res-CLIP-M	0.65	0.63	0.61	0.65	0.56	0.54	0.57	0.65	0.55	0.64	0.55	0.65	0.63	0.59	0.60	0.50	0.54
	Res-CLIP-C	0.65	0.63	0.61	0.66	0.52	0.51	0.54	0.66	0.53	0.67	0.53	0.67	0.62	0.58	0.59	0.47	0.49

the fact that these models were trained on the full training set.

ACKNOWLEDGMENT

None of the authors has a conflict of interest to disclose.

REFERENCES

- [1] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, "Text-to-image diffusion models in generative AI: A survey," 2023, *arXiv:2303.07909*.
- [2] H. H. Jiang et al., "AI art and its impact on artists," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, 2023, pp. 363–374.
- [3] N. Marchal, R. Xu, R. Elasmari, I. Gabriel, B. Goldberg, and W. Isaac, "Generative AI misuse: A taxonomy of tactics and insights from real-world data," 2024, *arXiv:2406.13843*.
- [4] D. Tariat, R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, "Synthetic image verification in the era of generative artificial intelligence: What works and what isn't there yet," *IEEE Secur. Privacy*, vol. 22, no. 3, pp. 37–49, May/Jun. 2024.
- [5] N. Yu, L. S. Davis, and M. Fritz, "Attributing fake images to GANs: Learning and analyzing GAN fingerprints," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7555–7565.
- [6] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do GANs leave artificial fingerprints?" in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval*, 2019, pp. 506–511.
- [7] N. Zhang and H. Tang, "Text-to-image synthesis: A decade survey," 2024, *arXiv:2411.16164*.
- [8] Y. Sheng, W. Si, and J.-Y. Zhu, "Few-shot image attribution for AI-generated art," in *Proc. NeurIPS 2022 Workshop Synthetic Data Empowering ML Res.*, 2022.
- [9] Z. Sha, Z. Li, N. Yu, and Y. Zhang, "DE-FAKE: Detection and attribution of fake images generated by text-to-image generation models," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2023, pp. 3418–3432.
- [10] L. Bindini, G. Bertazzini, D. Baracchi, D. Shullani, P. Frasconi, and A. Piva, "Tiny autoencoders are effective few-shot generative model detectors," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2024, pp. 1–6.
- [11] M. Albright, S. McCloskey, and A. Honeywell, "Source generator attribution via inversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, vol. 8, pp. 3–12.
- [12] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do neural networks dream of deepfakes? Analyzing synthetic image detectors," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2019, pp. 1–6.
- [13] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
- [14] D. Cioni, C. Tzelepis, L. Seidenari, and I. Patras, "Are clip features all you need for universal synthetic image origin attribution?" in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 363–382.
- [15] R. Goebel, N. ElSayed, B. Fröhlich, and A. Kappeler, "Synthetic image attribution: A benchmark," in *Proc. Eur. Conf. Comput. Vis.*, 2022.
- [16] J. Frank, T. Eisenhofer, A. Rahmati, A. Fischer, D. Kolossa, and T. Holz, "Leveraging the attribution signal for training robust GAN detectors," in *Proc. Int. Conf. Learn. Representations*, 2020.
- [17] D. Gragnaniello, F. Marra, and L. Verdoliva, "Detection of GAN-generated fake images over social networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2020, pp. 1602–1607.
- [18] N. Carlini, H. Farid, I. Goodfellow, N. Papernot, C. Raffel, and R. Shokri, "On the difficulty of detecting GAN-generated images," in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 2100–2114.
- [19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–11.
- [20] P. Bongini, V. Molinari, A. Costanzo, B. Tondi, and M. Barni, "Training-free source attribution of AI-generated images via resynthesis," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Perth, Australia, 2025, pp. 114–119, doi: [10.1109/WIFS66636.2025.00028](https://doi.org/10.1109/WIFS66636.2025.00028).
- [21] K. Solanki, U. Madhoo, B. S. Manjunath, S. Chandrasekaran, and I. El-Khalil, "Print and scan resilient data hiding in images," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 4, pp. 464–478, Dec. 2006.
- [22] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res. J.*, vol. 1, pp. 1–31, 2024.
- [23] B. Wu et al., "Visual transformers: Token-based image representation and processing for computer vision," 2020, *arXiv:2006.03677*.
- [24] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, May 2022.
- [25] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
- [26] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.
- [27] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. Zweiri, and N. Werghi, "GhostFaceNets: Lightweight face recognition model from cheap operations," *IEEE Access*, vol. 11, pp. 35429–35446, 2023.
- [28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–28.
- [29] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
- [30] A. Rozsa, M. Günther, and T. E. Boulton, "LOTS about attacking deep features," in *Proc. IEEE Int. Joint Conf. Biometrics*, 2017, pp. 168–176.
- [31] T. Yang et al., "Learning to disentangle GAN fingerprint for fake image attribution," 2021, *arXiv:2106.08749*.
- [32] T. Yang, Z. Huang, J. Cao, L. Li, and X. Li, "Deepfake network architecture attribution," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 4, pp. 4662–4670.
- [33] T. Bui, N. Yu, and J. Collomosse, "RepMix: Representation mixing for robust attribution of synthesized images," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 146–163.
- [34] H. J. Song, M. Khayatkhoei, and W. AbdAlmageed, "Man-iPT: Defining and analyzing fingerprints of generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 10971–10981.
- [35] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 12888–12900.

- [36] K. Xu, L. Zhang, and J. Shi, "Detecting origin attribution for text-to-image diffusion models," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2025, pp. 8775–8785.
- [37] L. Zhu et al., "MAID: Model attribution via inverse diffusion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2025, pp. 1–5.
- [38] S. Bonechi, P. Andreini, and B. T. Corradini, "Who made this? fake detection and source attribution with diffusion features," 2025, *arXiv:2510.27602*.
- [39] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental learning for the detection and classification of GAN-generated images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2019, pp. 1–6.
- [40] X. Xuan, B. Peng, W. Wang, and J. Dong, "Scalable fine-grained generated image classification based on deep metric learning," 2019, *arXiv:1912.11082*.
- [41] D. Sullivan-Pao, N. Tian, and P. Khorrami, "LoRAX: LoRA expandable networks for continual synthetic image attribution," 2025, *arXiv:2504.08149*.
- [42] H. Lee and J. Yi, "Few-shot class-incremental model attribution using learnable representation from CLIP-ViT features," 2025, *arXiv:2503.08148*.
- [43] S. Wu, S. Li, J. Li, J. Liu, and Y. Wang, "OmniDFA: A unified framework for open set synthesis image detection and few-shot attribution," 2025, *arXiv:2509.25682*.
- [44] J. Á. Urueña, D. Camacho, and J. H. Tato, "Supervised contrastive learning for few-shot AI-generated image detection and attribution," 2025, *arXiv:2511.16541*.
- [45] B. Zhang, J. P. Zhou, I. Shumailov, and N. Papernot, "On attribution of deepfakes," 2020, *arXiv:2008.09194*.
- [46] S. Hirofumi, K. Fukuchi, Y. Akimoto, and J. Sakuma, "Did you use my GAN to generate fake? Post-hoc attribution of GAN generated images via latent recovery," in *Proc. Int. Joint Conf. Neural Netw.*, 2022, pp. 1–8.
- [47] M. Dong, F. Li, Z. Li, and X. Liu, "PRSN: Prototype resynthesis network with cross-image semantic alignment for few-shot image classification," *Pattern Recognit.*, vol. 159, 2025, Art. no. 111122.
- [48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–14.
- [49] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.
- [50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [51] M. Zanella and I. Ben Ayed, "Low-rank few-shot adaptation of vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 1593–1603.
- [52] S. Mandelli, P. Bestagini, and S. Tubaro, "When synthetic traces hide real content: Analysis of stable diffusion image laundering," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2024, pp. 1–6.
- [53] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 6105–6114.
- [54] M. Lewis et al., "Does clip bind concepts? probing compositionality in large image models," in *Proc. Findings Assoc. Comput. Linguist.: EACL 2024*, 2024, pp. 1487–1500.
- [55] N. Purnekar, L. Abady, B. Tondi, and M. Barni, "Improving the robustness of synthetic images detection by means of print and scan augmentation," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2024, pp. 65–73.