EKILA: Synthetic Media Provenance and Attribution for Generative Art Supplementary Material

Kar Balan[1], Shruti Agarwal[2], Simon Jenni[2], Andy Parsons[2], Andrew Gilbert[1], John Collomosse[1,2]
[1]University of Surrey, [2]Adobe Inc.
{k.balan, a.gilbert}@surrey.ac.uk, {shragarw, jenni, andyp, collomos}@adobe.com

1. Additional Attribution Examples

We present further visual attribution examples using queries sampled from the 10M subset of LAION-400M [3], forming our LAION experiments in the main paper. The examples in Figures 1, 2 show synthetic image queries comprising either whole images or patches at $\frac{1}{4} \times \frac{1}{4}$ scale to retrieve the closest matches from LAION to enable the attribution of credit to the contributors of those training images.

2. Style based attribution

Due to the flexible nature of the image matching in EKILA for attribution, we can also swap out the content or object-focused fingerprinting and match verification of the primary attribution method proposed by EKILA. Instead, a style-based attribution method can be applied to identify and retrieve images in the training corpus similar to a query image's artistic style. We propose using the ALADIN [2] extraction of an embedding within EKILA's framework to encode an image's style in a fine-grained manner. Figure 3 demonstrates a style-based retrieval example of visual attribution on the 10M subset of the LAION400M dataset, with synthetic query images used to retrieve the closest stylistic matches. This has the potential to address the concerns of creatives regarding the memorization of living artists' visual style by GenAI models.

Further addressing this concern, given that ALADIN can distinguish between fine-grained artistic styles, we can also query LAION with images generated that resemble the style of several well-known contemporary and modern artists (Figure 4). This raises further potential use cases around the attribution of credit to artists for the presence of their style within existing GenAI datasets.

3. Submission of the codebase

The accompanying zip file contains the codebase that was used to generate the results for the 3 use case experiments in our paper.

3.1. MNIST GAN Experment

In the MNIST_GAN_demo directory:

- **MNIST_train_GAN_sign_model_images.py**. Demonstrates training a GAN using the MNIST dataset. During training, a record of all images used is built. After training, the model file is saved, and a C2PA manifest is generated, which contains all the images as ingredients to the model file. Images are generated using the GAN, and C2PA files are built for each image, encoding the model file as an ingredient to them. This allows them to inherit all the manifests of the ingredient images from MNIST.
- **donate.py**. Given a C2PA signed generated image and a total sum to be donated, the script identifies all authors of the ingredient images and their respective DLT wallet addresses, splits the reward sum, and initiates transactions through the blockchain to transfer the reward to each of them.

3.2. C2PA Apportionment Experiment

In Dreambooth_demo directory:

- **dreambooth_donate.py**. Given a C2PA signed image generated using a specialized DreamBooth model, this script identifies all ingredient assets and their respective rights smart contract addresses. The user is prompted to decide which ingredient assets they want to donate for and how much currency for each. This is where apportionment based on content/style attribution can be implemented according to the user's needs. The script initiates donation transactions from the user's account and transfers the specified amount of currency into each contributor's rights smart contract.
- **exercise_right_to_train.py**. Given a local image asset with a C2PA manifest detailing its authorship and provenance, this script analyses the manifest and identifies the smart contract which manages its rights licenses. Given the unique ID of the right owned by the

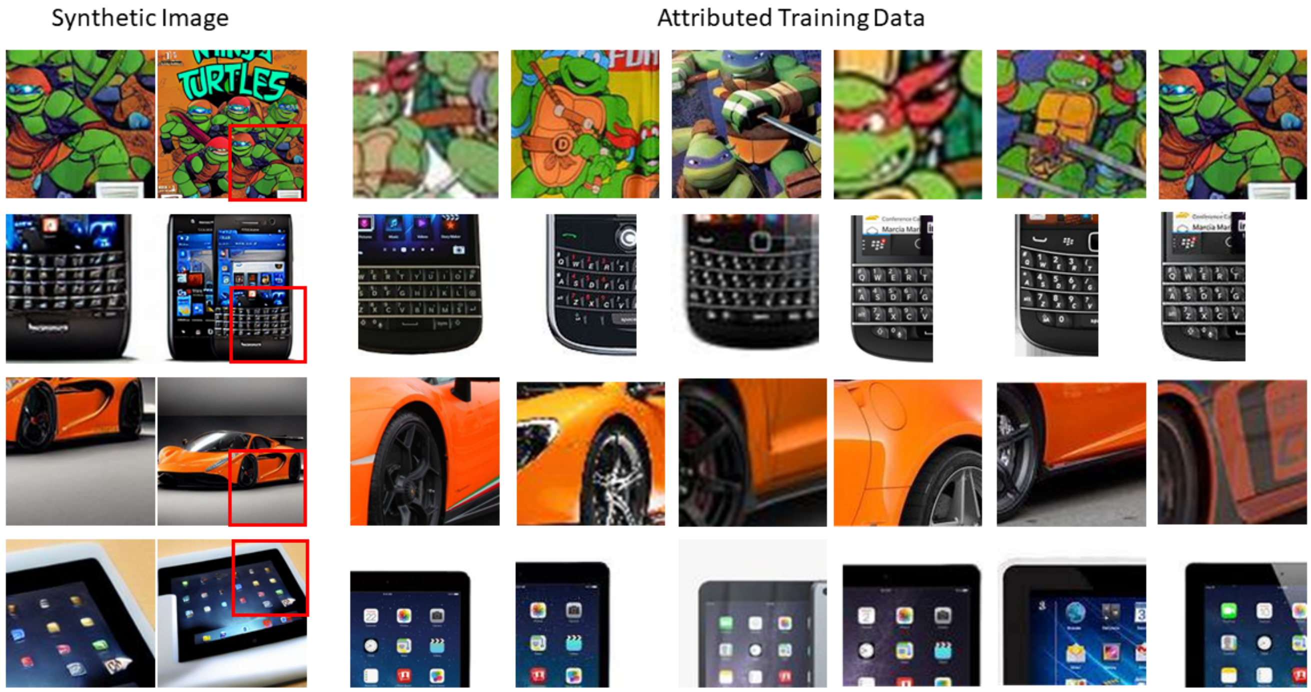

Figure 1. Attribution over the 10M subset of LAION400M dataset showing examples of object fragments attributed from four synthetic images, using the EKILA attribution method. Matching patches are outlined in red.

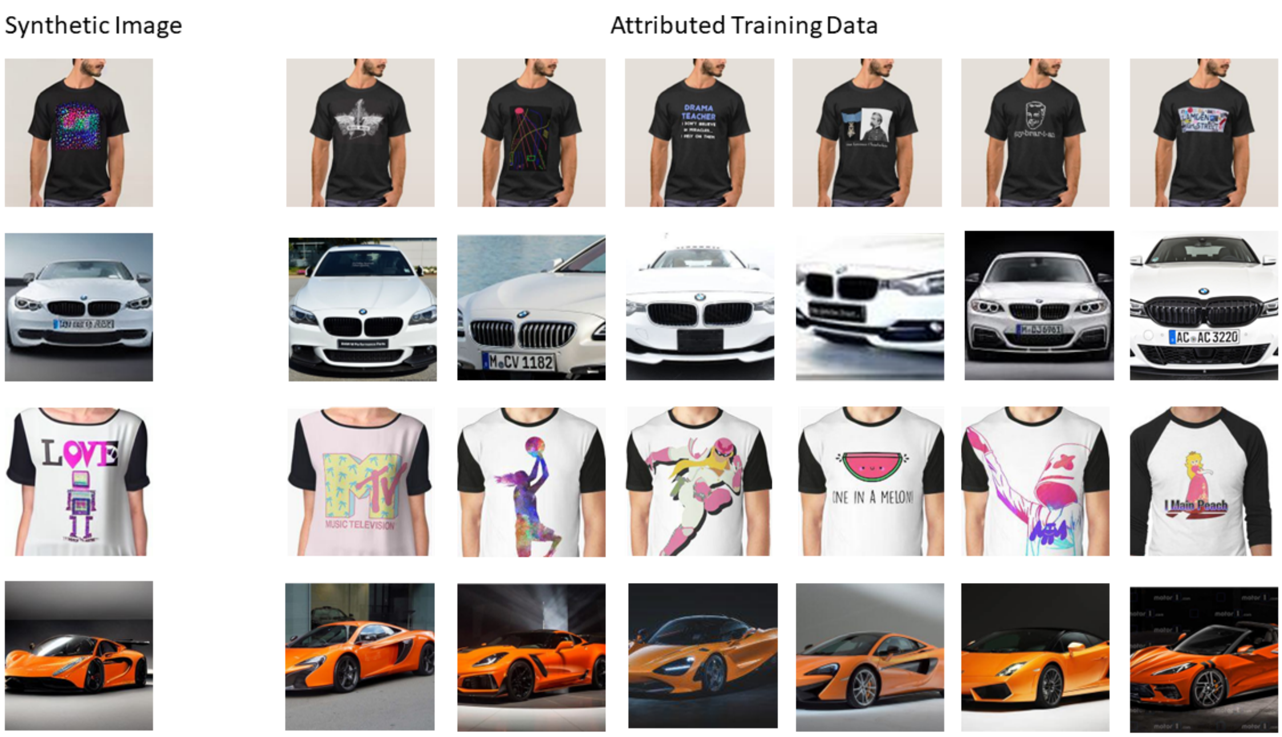

Figure 2. Attribution over the 10M subset of LAION400M dataset showing examples of object fragments attributed from four synthetic images, using the EKILA attribution method.

Figure 3. Attribution over the 10M subset of LAION400M dataset showing examples of similar-style images attributed from four synthetic query images, using the ALADIN model.

user, the script then queries the smart contract to identify the price of exercising that right - in this case, the right to train for AI. Once the price is determined, a transaction is initiated using the user's local balance within that smart contract to pay for exercising their right.

3.3. Rights Contract Framework

In rights_smart_contract directory:

- **Rights721.sol**. is the Rights Contract Framework code in the Solidity programming language. It contains all necessary functions and storage necessary to track, manage and sell rights licenses to any content NFT.
- **Rights721_abi.json**. This is the compiled ABI of the rights framework smart contract. It is needed to interact with the deployed smart contract.
- **user_creditsAcct_buysRights.py**. In this demo script, users credit their account on a specific rights smart contract. After crediting their account, the user buys two rights licenses - one to train and one to generate. Using the transaction logs, the user identifies the unique right IDs of their licenses and keeps a record of them.
- **withdraw_credit_artist.py**. In this demo, an artist initiates a call to the smart contract they own to identify their current balance and then initiates a transaction withdrawing their entire balance. The balance is then transferred to their wallet.

4. Video Dreambooth Demo

We provide a demo video showcasing the proposed EKILA pipeline, applied to a DreamBooth experiment. We visualize the provenance information of an image generated using DreamBooth within the Verify tool. We then identify the concept images used in specializing the model and find that several of them are available as NFTs within a rights framework smart contract. We run the Python script to identify the relevant smart contract addresses and donate to the authors of the DreamBooth model. We access the rights framework smart contract using Goerli Etherscan and show the creator's balance increasing after the donation.

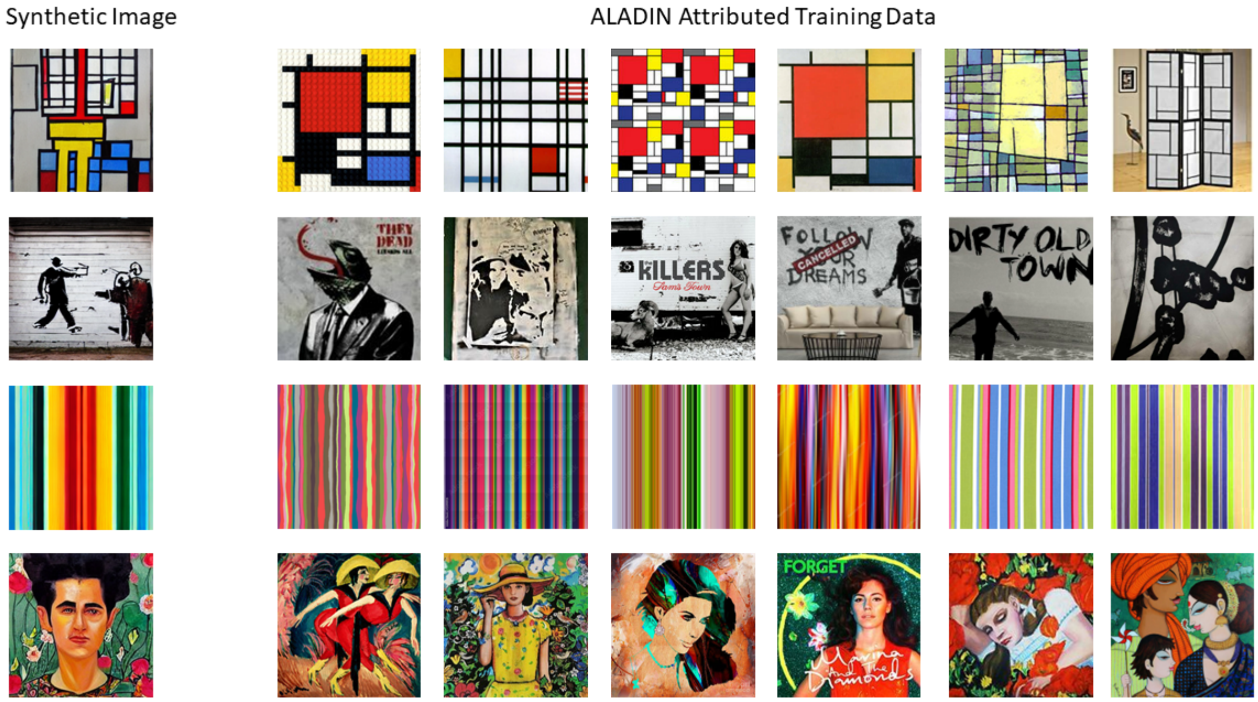

Figure 4. Style based ALADIN attribution over the entire LAION400M dataset showing examples of similar-style images attributed from four query images synthesized by a Latent Diffusion model [1], with the prompts being well-known artists' names.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[2] D. Ruta, S. Motiian, B. Faieta, Z. Lin, H. Jin, A. Filipkowski, A. Gilbert, and J. Collomosse. Aladin: All layer adaptive instance normalization for fine-grained style similarity. 2021.

[3] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021.