Rethinking Artistic Copyright Infringements in the Era of Text-to-Image Generative Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

The advent of text-to-image generative models has led artists to worry that their individual styles may be improperly copied. Copying a style is more complex than replicating a single image, as style is comprised by a set of elements (or *signature*) that frequently co-occurs across a body of work, where each individual work may vary significantly. Thus, we reformulate the problem of "artistic copyright infringement" from probing image-wise similarities to a classification problem over image sets. We then introduce `ArtSavant`, a practical (i.e., efficient and easy to understand) tool to (i) determine the unique style of an artist by comparing it to a reference corpus of works from hundreds of artists, and (ii) recognize if the identified style reappears in generated images. We leverage two complementary methods to perform artistic style classification over image sets, including TagMatch, which is a novel inherently interpretable and attributable method, making it more suitable for broader use by non-technical stake holders (artists, lawyers, judges, etc). We then further validate `ArtSavant`by applying it in an empirical study to quantify the prevalence of artistic style copying across 3 popular text-to-image generative models, finding that under simple prompting, 20% of 372 prolific artists studied appear to have their styles be at risk of copying by today's generative models.

1 Introduction

The impressive capabilities of text-to-image generative models such as Stable Diffusion, Imagen, Mid-Journey, and DeepFloyd [27, 28, 2, 23] trained on massive web-scraped datasets [29] have captured widespread attention and at times concern, for they may make infringing copyrighted material far easier. While previous studies [5, 31, 32] have shown that direct copying of individual training images is generally rare in diffusion models, the degree to which image generative models can replicate art *styles* as opposed to art works remains unclear.

This issue has human and material consequences (potentially unfairly undermining the value of original art), and is fundamentally interdisciplinary, engaging artistic and legal communities. There are currently no laws to identify and protect an artist's style - mainly due to challenges in definition and a previous lack of necessity. However, at least one major actor has proposed such legislation [3], raising the issues of how well individual artistic style can be defined, and how much artists should be worried that their style can be effectively mimicked. To this end, we seek to tackle the problem of defining and identifying artistic styles, as well as building a practical tool to detect instances of style infringement. Our tool, `ArtSavant`, prioritizes accessibility and transparency so that it is useful to a broad audience: we make it simple and fast enough for an end-user (e.g., artist or lawyer) to run, and interpretable enough so that the user can understand and convey the results to another party (e.g., judge or jury).

We frame artistic style as characterized by a set of elements that co-occur frequently across an artist's *body of work*, which makes it challenging to determine style by inspecting individual works (a la

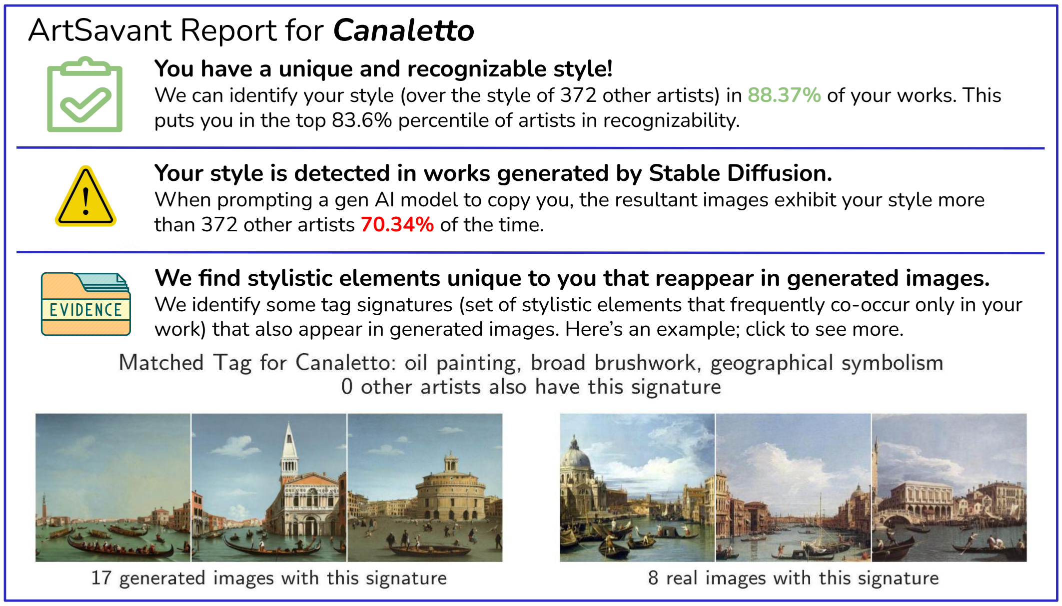

Figure 1: Our primary contribution is an accessible framework for arguing style infringement from the perspective of classification. Given artworks by Canaletto, `ArtSavant` identifies a unique style and recognizes said style in generated art, and produces an easy to understand yet quantitative report.

previous image-wise copy studies). For e.g., Vincent Van Gogh's style comprised of expressive wavy lines, bright unblended coloring, post-impressionism, choppy textured brushwork, etc. In Figure 3, we illustrate that while generative models seldom reproduce Van Gogh's artworks exactly, they frequently capture and replicate elements of his style. While describing his (or any style) can be challenging, and making a case for distinctiveness between two styles is even more so, as artists draw inspiration from each other, we can still recognize Van Gogh's style. Building on this intuition, our approach to proving the uniqueness of a style is to show that from a collection of artworks, one can identify the artist who created them. That is, if an artist's work can consistently be attributed to its creator, this entails a uniqueness to that artist's style. Therefore, the task of showing the existence and distinctiveness of artistic styles can be reduced to **classification** over image *sets*. To empirically study style copying in generative models and to build a corpus of artistic styles, we collect a dataset of works from 372 artists, and develop two complementary methods to classify artistic style over a body of works, strongly motivated by notions of of 'holistic' and 'analytic' comparisons from the copyright legal literature [13, 20].

The first method – **DeepMatch** – is a neural network that classifies artwork to artists. DeepMatch implicitly maps each artist to a vector (via the classification head) during training, which can be interpreted as a *neural signature* representing an artist. Aggregating its predictions over a set of artworks via majority voting, we find that DeepMatch achieves 89.3% test accuracy, indicating that *unique artistic styles do indeed exist for a large fraction of artists*. Since deep features are not very interpretable, DeepMatch is not suited for articulating the elements that comprise each artistic style. Thus, we complement DeepMatch with a novel inherently *interpretable* and *attributable* method called **TagMatch**.

TagMatch first tags individual artworks using a novel method, validated with an MTurk study, based on *zero-shot, selective, multilabel classification* with CLIP [24], resulting in tags spanning diverse aspects of artistic style. Individual tags are common across artists and thus cannot define unique styles alone, but, by efficiently searching the space of tag combinations, we surface *tag signatures*, where a set of tags frequently co-occur only over the set of works from a single artist. To map a set of unseen works to an artist, we employ a look-up scheme, where we predict the artist who's works share the most unique tag composition with the test set of works. We find tag signatures for *all* artists in our dataset, and observe them to be reliable enough to detect the style of the artists in our dataset (on a held out set) with 61.6% top-1 and 82.5% top-5 accuracy. Crucially, TagMatch articulates the stylistic elements that were uniquely present in the test set of images and the matched reference set, and offers as attribution, by way of the subset of images from both sets that contain the matched tag signature.

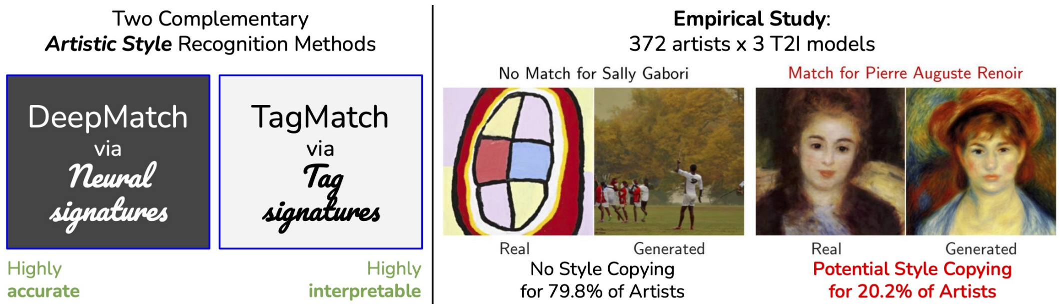

Figure 2: We define artistic style as a set of elements (or signature) that appear frequently over a body of work, and reduce the problem of style copy detection to classification of *sets* of images to artists. (**left**) We offer proof-of-concept via two ways to recognize artistic styles over image *set*, including a novel inherently interpretable and attributable tag-based method. (**right**) In an empirical study of 372 prolific artists, we find generative models potentially copy artistic styles for 20.2% of these artists.

Given a set of works by a concerned artist, `ArtSavant` applies DeepMatch and TagMatch to generate report like Figure 1 in minutes, offering quantitative evidence (if present) of the existence of the artist's unique style and copying by a generative model. To better understand style copying at scale, we employ `ArtSavant` on images generated in the style of artists in our dataset via simple prompting of 3 popular text-to-image models. We find 20% of the artists we study to be at risk of style copying, though this number may rise as models and prompting schemes grow in sophistication. We hope `ArtSavant` can continue to offer quantitative insight on the prevalence of style copying, while also being accessible and practically useful to the broad range of relevant stakeholders. In summary, we make the following contributions:

- We reformulate the copyright infringement of artistic styles through the lens of classification over image sets, rather than a single image.
- We introduce `ArtSavant`, a practical tool consisting of a reference dataset of artworks from 372 prolific artists, and two complementary methods (including a novel, highly interpretable and attributable one) which effectively can detect unique artistic styles.
- With `ArtSavant`, we perform a large-scale empirical study to measure style copying across 3 popular text-to-image generative models, finding that generated images (using simple prompting) from *only* 20% of the artists examined appear to be at high risk of style copying.

2 Related Works

The rapid advance of image generative models has made the possibility of mimicking artists' personal styles a topic of discussion in the literature [25]. Some works describe ways to either detect direct image copying in generated images, or to foil any future copying attempts by imperceptibly altering the artists' works to prevent effective training by the generative models. These include techniques like adding imperceptible watermarks to copyrighted artworks [36, 9, 10], and crafting "un-learnable" examples on which models struggle to learn the style-relevant information [30, 37, 39]. Others have suggested methods to mitigate this issue from the model owner's perspective - to either de-duplicate the dataset before training [5, 31, 32], or to remove concepts from the model after training ("unlearning") [18, 11, 4]. Methods like [5, 31, 32] are also more focused on analyzing direct image copying from the training data, and thus may not be applicable to preventing style copying.

None of these works tackle the problem of *detecting* potentially copied art *styles* in generated art, especially in a manner which may be relevant to legal standards of copyright infringement. According to current US legal standards [1], an artwork has to meet the "substantial similarity" test for it to be infringing on copyright. This similarity has to be established on *analytic* and *holistic* terms [20, 13]. Analytic here refers to explaining an artwork by breaking it down into its constituents using a concrete and objective technical vocabulary, while holistic refers to the overall "look and feel" of the artwork. So to be relevant to the legal community (who ultimately decides on alleged cases of style copying), we design our tool to reflect this dichotomy in its working, while also emphasizing ease of use and

Figure 3: Example generations from Stable Diffusion 2 when prompted to produce specific paintings by Vincent Van Gogh, along with the histogram of similarities between the generated image and corresponding real image. Even for a famous artist like Van Gogh, generative models rarely produce near-exact duplicates. However, Van Gogh's *style* appears consistently, even when similarity is low.

interpretability, to make our tool practically useful for a concerned artist hoping to protect themselves. These priorities manifest in our reformulation of detecting style copying as classification in §4. But first, we discuss limitations in applying the typical copy detection approach to artistic styles.

3 Motivation: Image-wise similarity may be limited for Style Copying

A prevailing approach to investigating copying involves representing images in a deep embedding space via models like SSCD [22] or DINO [6], and computing image-to-image similarities across generated and real images. Such an approach has been employed by [31, 32, 5] to show that generative models can (though rarely do) create exact replicas of training images. Inspired by these results and the consequent concerns from artists, we first explore if generative models can recreate famous artworks, e.g., by Vincent Van Gogh. Specifically, we generate images by prompting "*{artwork title}* by Vincent Van Gogh" for 1500 Van Gogh works, and compute the DINO similarity between pairs of a real and corresponding generated image. Figure 3 visualizes the distribution of similarities, as well as examples at each similarity level. We find that the vast majority of similarities are lower than 0.75, which amounts to pairs that are far from duplicates. However, even when the generated image differs significantly from the source real image, certain stylistic elements associated with Van Gogh seem to appear consistently in the generated works. Thus, while instance-wise copying of artwork appears rare for even the ultra famous Van Gogh, style copying may require going beyond image-to-image comparisons, as artists may still have their personal styles, developed over a long career/many artworks and at significant personal cost, infringed upon in ways that searching for exact replicas would miss. A concurrent work finetunes embeddings so that cosine similarity better proxies style similarity [33], though even in this case, the utility of such a tool in court is limited by its lack of interpretability.

4 Reformulating Artistic Style Copying as Classification over Image Sets

Having established that style is comprised over a body of work (instead of a single image) and that copy detection must be interpretable to hold weight in court, we now present an alternate framework for arguing style infringement, with the following intuition: if an artist's work can consistently be distinguished from that of other artists, then there must exist something unique that is present across that artist's portfolio. Thus, we can use classification over image sets to demonstrate a unique style exists given an artist. Then, style infringement can be argued by showing the copied artist can again be predicted (over many others) given a set of generated works. We now detail DeepMatch and TagMatch, two complementary methods (w.r.t. accuracy and interpretability) that classify artistic styles over image sets, in holistic and analytic manners respectively.

A necessary preliminary: WikiArt Dataset. To distinguish one artist's style from that of others, we need a corpus of artistic styles (i.e. portfolios from many artists) to compare against. To this end, we curate a dataset $\mathcal{D}$ consisting of artworks from WikiArt [1] (like others [34, 16]) to serve as (i) a reference set of artistic styles, (ii) a validation set of real art to show (most) artists have unique styles and our methods can recognize them on held-out sets of their works, and (iii) a test-bed to explore if text-to-image models replicate the styles of the artists in our dataset in their generated images. We include $\sim$91k artworks from 372 artists $\mathcal{A}$ spanning diverse eras and art movements, including any artist with at least 100 works on WikiArt. Each work is labeled with its genre (e.g., *landscape*) and style (e.g., *Impressionism*), though we primarily use the artist and title labels. We provide an easy-to-execute script to enable others to scrape newer versions of this dataset if desired. We now detail DeepMatch and TagMatch, which each compare a test set of images to our reference corpus.

4.1 DeepMatch: Black-Box Detector

DeepMatch consists of a light-weight artist classifier[2] (on images) and a majority voting aggregation scheme to obatin one prediction for a *set* of images. Majority voting requires that at least half the images in a test set $\hat{D_a}$ are predicted to a for DeepMatch to predict a, allowing for abstention in case no specific style is recognized with sufficient confidence. For our classifier, we train a two layer MLP on top of embeddings from a frozen CLIP ViT-B\16 vision encoder [24], using a train split containing 80% of our dataset. We employ weighted sampling to account for class imbalance. Since we utilize frozen embeddings, training takes only a few minutes on one RTX2080 GPU. Thus, a new artist could easily retrain a detector to include their works (and thus encode their artistic style).

Validation of the Detector. We apply DeepMatch on the held-out test split of our dataset and observe that the image-wise classifier attains 72.8% accuracy per image over 372 artists. When aggregating image-wise predictions via majority vote, 89.3% of artists are matched, validating our method, and offering strong evidence towards the existence of unique artistic styles. Specifically, neural classifiers capture unique and frequently co-occurring characteristics of the artists in their embedding space, which can be thought of as 'holistic' *neural signatures*. Figure 4 shows the distributions of image-wise accuracies per artist, shading correctly matched artists (green). We also present an image from one of the few artists who's style is not matched by DeepMatch, along with an image from a similar artist. Notice that the style of two artists can be extremely similar (see Appendix B.1), making the existence of unique artistic styles for the vast majority of artists considered (by way of neural signatures) a non-trivial observation.

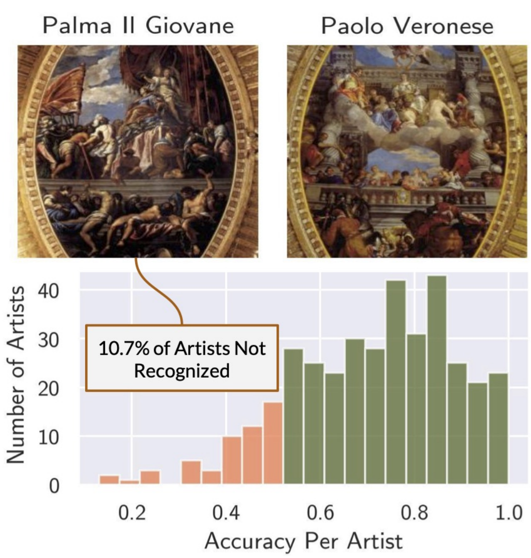

Figure 4: DeepMatch on held-out real art: 89.3% of artists can be recognized. The remaining 10.7% of artists have very similar styles to other artists: e.g., Palma Il Giovane's work differs marginally from other Italian renaissance painters.

4.2 Interpretable Artistic Signatures

Now we provide an analytic complement to DeepMatch's holistic approach. Namely, we seek to articulate the elements that comprise an artist's unique style. We do so by tagging images with descriptors (called atomic tags) drawn from a vocabulary of stylistic elements. Then,

[1]https://www.wikiart.org/; note that we only include Public domain or fair use images.

[2]Others have trained art classifiers [16, 15, 35], but they do not operationalize them for style infringement.

we *compose* tags efficiently to go from atomic tags that are common across artists to longer tag compositions that are unique to each artist (i.e. *tag signatures*). We detail these steps now, before explaining how tag signatures can be used to classify an image set to an artist in the following section.

"Guitar, Sheet music and Wine glass" by Pablo Picasso

Tags:
abstract expressionism style
collage
repetitive composition
musical instruments
simple colors
Contemporary influences
abstract subject matter

"Leda Atomica" by Salvador Dali

Tags:
magic realism style
heaven
Contemporary influences
surrealistic
sharp angles

Figure 5: Example atomic tags assigned via our proposed CLIP-based zero-shot method. We perform selective multilabel classification along various aspects of art (e.g. medium, colors, shapes, etc), so that atomic tags span diverse categories. Details in section 4.2.

Zero-shot Art Tagging We utilize the zero-shot open-vocabulary recognition abilities of CLIP to tag images with descriptors of stylistic elements. First, we construct a concept vocabulary $\mathcal{V}$ with help from LLMs. Namely, we prompt Vicuna-13b and ChatGPT to generate a dictionary of concepts along various aspects of art. We manually consolidate and amend the concept dictionary, resulting in a vocabulary of 260 concepts over 16 aspects (see Appendix E.1).

To assign concepts to images, we a design a novel scheme that consists of selective multilabel classification per-aspect. Namely, for an image, we compute CLIP similarities to all concepts, and normalize similarities *within each aspect*. Then, we only assign a concept its normalized similarity (i.e. z-score) exceeds a threshold of 1.75. This means that a concept is only assigned for an aspect if the image is substantially more similar to this concept than other concepts describing the same aspect. Classifying per-aspect allows for a diversity of descriptors to emerge, as global thresholding results in a biased tag description, as concepts for certain aspects (e.g. subject matter) consistently have higher CLIP similarity than those for more nuanced aspects (e.g. brushwork). We call the assigned concepts *atomic tags*; figure 5 shows atomic tags assigned for a few examples.

Validation of Quality of Tags Using Human-Study. We validate the effectiveness of our tagging via a human-study involving MTurk workers. In particular, given an image of an artwork and an assigned atomic tag $v_{predict}$ from the vocabulary $\mathcal{V}$ – MTurk workers are asked "*Does the term $v_{predict}$ match (i.e. the concept $v_{predict}$ present) the artwork below?* ". The workers are then asked to select between $\{$Yes, No, Unsure$\}$. We collect responses for 1000 images with 3 annotators each. We find that in only 17% cases, a majority of workers disagree with the provided tag, suggesting our tagging results in a low false positive rate. We also observe all three annotators agree in only 51% of cases, reflecting that describing artistic style can be subjective. While our tagging is not perfect, it is a deterministic and automatic method of articulating artistic style elements, and that our tagging method will improve as underlying VLMs improve too. See the appendix for more details and discussion on the human study.

Tag Composition for Artists. Using the atomic tags in the artwork specific vocabulary $\mathcal{V}$, in this section we design a simple and easy-to-understand iterative algorithm to obtain a set of *tag signatures* $\mathcal{S}_a$ for each artist $a \in \mathcal{A}$. These signatures are a composition of a subset of tags in $\mathcal{V}$. In particular, our algorithm efficiently searches the space of tag compositions to go from atomic tags to composition of tags which become more unique as the length of the tag composition grows. For e.g., while 40% of the artists may use simple colors, *only* 15% may use both simple colors and impressionism style.

To efficiently search the space of tag compositions per artist $a \in \mathcal{A}$, we first assign a set of tags to each of their images $x \in \mathcal{D}_a$ via the zero-shot *selective multi-label classification* method described above. For each image x, let $\text{tag}(x)$ denote the set of predicted atomic tags. To get atomic tags *for an artist*, we aggregate all atomic tags over images, and keep only the tags occurring in at least 3 works. We denote this aggregate set of atomic tags as the "Common Atomic Tags Per Artist" and denote it as $\mathcal{C}_a$. Then, we iterate through all the images $x \in \mathcal{D}_a$ for a given artist a, to find the intersection $I(x) = \text{tag}(x) \cap \mathcal{C}_a$. We then compute a powerset $\mathcal{P}(I(x))$ of the tags occurring in the intersection $I(x)$ and increment the count of each occurrence of the tag composition from the powerset in $\mathcal{S}_a$.

Note that the size of $I(x)$ is much smaller than that of $\mathcal{C}_a$, and thus, iterating through $\mathcal{P}(I(x))$ for each image x is much, much faster than iterating through $\mathcal{P}(\mathcal{C}_a)$. Finally, we again filter the tag compositions in $\mathcal{S}_a$, only including those that occur in at least 3 works. We provide the details of this tag composition algorithm in 1 and Appendix E.3.

Do Unique Signatures Exist for Artists? Using our tag composition method on the curated dataset from WikiArt, we find that *artistic signatures* in the form of an unique tag composition exists per artist. In Figure 6, we show that our tag composition algorithm is able to select unique tag compositions such that *only* a very few artists exhibit such compositions in their paintings as the tag length increases. This shows that artists exhibit *unique style* which can effectively be captured by our iterative algorithm. Leveraging these observations, in the next section, we describe TagMatch, which can classify a set of artworks to an artist by uniquely matching such tags (or tag signatures).

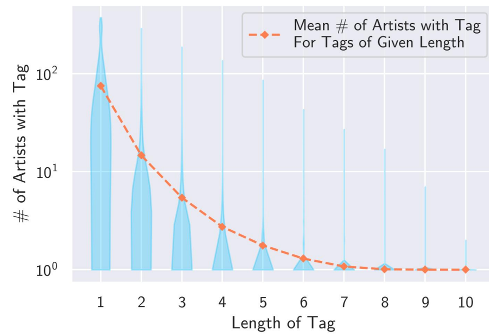

Figure 6: Composing atomic tags results in more unique tags, towards artistic *tag signatures*.

4.3 TagMatch: Interpretable and Attributable Style Detection

In 4.1, we outlined a holistic approach to accurately detect artistic styles. While DeepMatch obtains high accuracy (recognizing styles for 89.3% of artists), the neural signatures it relies upon lack interpretability. For a copyright detection tool to be useful in practice (e.g., to be used as assistive technologies), providing explanations of the classification decisions can tremendously benefit the end-user. To this end, we leverage our efficient tag composition algorithm as defined in 4.2 to develop TagMatch - an interpretable classification and attribution method which can effectively classify a set of artworks to an artist, as well provide reasoning behind the classification and example images from both sets that present the matched tag signature. TagMatch follows the intuition of matching a test portfolio to a reference artist who's portfolio shares the most unique tag signatures. Given a set of N test images $\mathcal{T} = \{x_i\}_{i=1}^{N}$, we first obtain a number of tag compositions for them using our iterative algorithm in 4.2. These tag compositions are then compared with the tag compositions of the artists in the reference corpus in order of uniqueness (i.e. we first consider tag signatures present in the test portfolio that occur for the fewest number of reference artists). We can then rank reference artists by how unique the shared tags are with the test portfolio. Detailed steps of the algorithm is in Appendix E.3. Also, TagMatch is fast, taking only about a minute, after caching embeddings of all images.

Validation of TagMatch. We again utilize the test split of our WikiArt Dataset to validate the proposed style detection method. TagMatch predicts the correct artist with top-1 accuracy of 61.6%, with top-5 and top-10 accuracies rising to 82.5% and 88.4% respectively. While less accurate than DeepMatch, the *tag signatures* provided by TagMatch allow for analytic arguments to be made regarding style copying, as the exact tag signatures used in matching can be inspected. Moreover, the subset of images in both the test portfolio and matched reference portfolio can be easily retrieved, offering direct attribution of the method; examples can be seen in the next section, where we match generated images to our reference artists. Overall, we hope TagMatch and DeepMatch can serve as automatic and objective tools to navigate the subtle problem of identifying artistic styles, towards detecting style copying and helping artists argue their case (i.e. in a court of law) in such instances.

5 `ArtSavant`: A Practical Tool for Concerned Artists

We package DeepMatch and TagMatch into `ArtSavant`, a practical tool designed with a concerned artist in mind. Given a set of works by the concerned artist, `ArtSavant` would create an easy-to-understand report characterizing the degree to which generative models copy the styles of the artist. As shown in Figure 7, the artist can present a set of generated images, or we can generate them by prompting text-to-image models with prompts of the form "{title of work} by {name of artist}". The provided works are then combined with our existing art repository and split into train/test sets. Using the train split, we (a) train a classifier over the $372 + 1$ artists, and (b) tag all images, compose tags

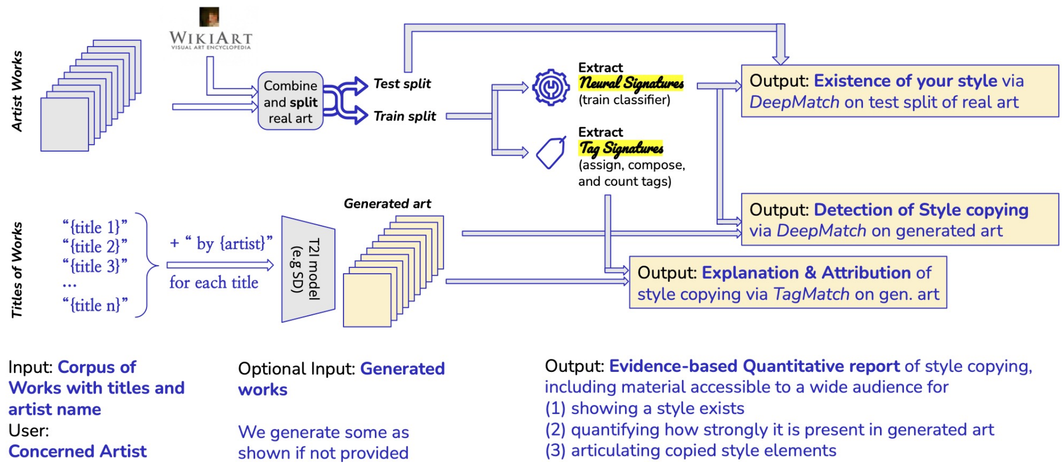

Figure 7: `ArtSavant` flow. We design our tool with a concerned artist in mind, who wishes to quickly investigate the degree to which they may be at risk of style copying by generative models.

within artists, and store extracted tag compositions per artist, resulting in neural and tag signatures. With these, we can apply DeepMatch and TagMatch respectively. Applying DeepMatch to the held-out art provides a measure of recognizability, establishing that the artist has an identifiable style to begin with. Then, running DeepMatch on generated images provides a quantitative manner to understand if (and to what degree) the artist's style appears consistently in generated works. Finally, running TagMatch on the generated images helps articulate the particular style signatures that are copied, enabling an analytic way to argue infringement, while also surfacing stylistically similar examples.

Figure 1 shows an example report outputted by `ArtSavant`when presented with art from an artist named Canaletto, who we observed was at risk of style infringement. We design the report to be easy to read and understand, as well as being evidence-based. Moreover, the report can be generated very quickly. Because all steps operate on embeddings from a frozen CLIP encoder, the process takes about 1-2 minutes, as we can simply compute embeddings once (and offline for the WikiArt corpus).

5.1 Analysis with `ArtSavant`: Quantifying Style Copying of 372 Prolific Artists

While enough anecdotal instances of style mimicry have been observed to raise concern [30, 25], the prevalence and nature of such instances remains nebulous. To shed quantitative insight on style copying, we now leverage `ArtSavant`on the 372 artists from our WikiArt dataset, generating images with three popular text-to-image models: (i) Stable-Diffusion-v1.4; (ii) Stable-Diffusion-v2.0; and (iii) OpenJourney from PromptHero. Following figure 7, we employ a simple prompting strategy of augmenting painting titles with the name of the artist; we explore alternate prompts in D.

We first apply **DeepMatch** to see what fraction of artists' styles can be recognized consistently over generated images. Namely, each generated image is classified to one of 372 artists, and per artist, predictions are aggregated via majority voting. Figure 8 shows the 'accuracy' on generated images per artist, where accuracy is now interpreted as the rate which images generated to copy an artist are classified as that artist. In red, the fraction of artists who see accuracies of at least 50% (i.e. so that the

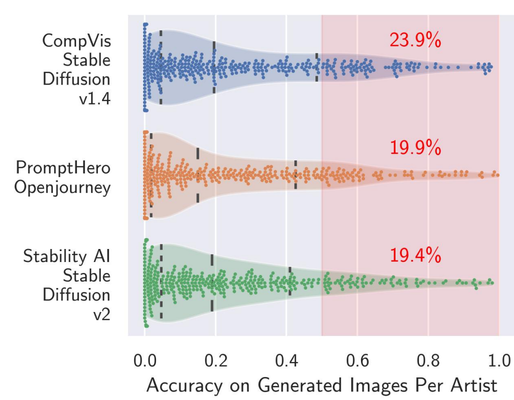

Figure 8: DeepMatch on generated art. In red: the fraction of artists with their styles recognized in at least half of their respective generated images.

Figure 9: Examples of applying TagMatch to generated images. TagMatch is inherently interpretable with respect to tags, as each inference comes with the exact set of tags that are (i) shared between the sets of test art and art from the predicted artist, and (ii) used to predict the artist.

generated image *set* is classified to the original artist) are denoted per model, which we call the match rate. We observe an average match rate of 20.2%, indicating that for the vast majority of artists in our study, *simple prompting of generative models does not reproduce their styles* in a way recognizable to DeepMatch, which has an 89% match rate on real art. For all three models, over half the artists see accuracies below 20%, with 26% of artists seeing an average accuracy below 5% for generated images. On the other hand, a handful of artists' styles are matched with high confidence: 16 artists see average accuracies over 75%. These include ultra famous artists like Van Gogh, Claude Monet, Renoir, which we'd expect generative models to do well in emulating. However, a few relatively lesser known artists are also present, like Jacek Yerka, who are still alive, and thus could be negatively affected by generative models reproducing their styles.

With **TagMatch**, in addition to predicting an artistic style, we can also articulate the specific tag signature shared between the test set of images and the reference set of images for the predicted style. Thus, we can inspect the shared signature, as well as instances from both sets where the signature is present, providing direct evidence of the potential style infringement a broader audience to independently verify. Inspecting some examples in figure 9 (more in fig. 15), we observe that while pixel level differences are common across retrieved image subsets, stylistic elements are consistent in both sets with the labeled tags, echoing our motivating claim that style copying goes beyond image or pixel-wise similarity. Lastly, TagMatch also allows for understanding image distributions from the perspective of interpretable tags. We explore this direction in appendix E.2, finding differences in the uniqueness of the tags present in generated art vs real art.

6 Conclusion

In our paper, we rethink the problem of copyright infringement in the context of artistic styles. We first argue that image-similarity approaches to copy detection may not fully capture the nuance of artistic style copying. After reformulating the task to a classification problem over image sets, we develop a novel tool – `ArtSavant`, a tool to reliably and interpretably (via a novel attributable method) extract and detect artistic style *signatures* in a way a broader audience can understand. We find evidence of the existence of artistic styles, and in an empirical study, quantify the degree to which styles are potentially infringed, validating our framework. We hope `ArtSavant` can be of use to the broader community who this problem affects, and serve as an accessible framework to quantitatively examine the nuanced issue of artistic style infringements.

References

[1] Generative artificial intelligence and copyright law, Sep 2023. URL https://crsreports.congress.gov/product/pdf/LSB/LSB10922.

[2] Deepfloyd, Apr 2023. URL https://github.com/deep-floyd/IF.

[3] Adobe. Fair act to protect artists in age of ai, September 12 2023. URL https://blog.adobe.com/en/publish/2023/09/12/fair-act-to-protect-artists-in-age-of-ai#:~:text=The%20right%20requires%20intent%20to,independent%20creation%20is%20a%20defense. Accessed: 2024-05-22.

[4] Samyadeep Basu, Nanxuan Zhao, Vlad Morariu, Soheil Feizi, and Varun Manjunatha. Localizing and editing knowledge in text-to-image generative models, 2023.

[5] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. URL https://arxiv.org/abs/2104.14294.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[8] Stephen Casper, Zifan Guo, Shreya Mogulothu, Zachary Marinov, Chinmay Deshpande, Rui-Jie Yew, Zheng Dai, and Dylan Hadfield-Menell. Measuring the success of diffusion models at imitating human artists, 2023.

[9] Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, Yue Xing, and Jiliang Tang. Diffusionshield: A watermark for copyright protection against generative diffusion models, 2023.

[10] Yingqian Cui, Jie Ren, Yuping Lin, Han Xu, Pengfei He, Yue Xing, Wenqi Fan, Hui Liu, and Jiliang Tang. FT-SHIELD: A watermark against unauthorized fine-tuning in text-to-image diffusion models, 2024. URL https://openreview.net/forum?id=OQccFglTb5.

[11] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models, 2023.

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[13] Paul Goldstein. *Goldstein on Copyright, 3rd edition*. Wolters Kluwer Legal & Regulatory U.S., 2014.

[14] Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. Open-set image tagging with multi-grained text supervision. *arXiv e-prints*, pages arXiv–2310, 2023.

[15] C. Richard Johnson, Ella Hendriks, Igor J. Berezhnoy, Eugene Brevdo, Shannon M. Hughes, Ingrid Daubechies, Jia Li, Eric Postma, and James Z. Wang. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4):37–48, 2008. doi: 10.1109/MSP.2008.923513.

[16] Sergey Karayev, Aaron Hertzmann, Matthew Trentacoste, Helen Han, Holger Winnemoeller, Aseem Agarwala, and Trevor Darrell. Recognizing image style. In *Proceedings of the British Machine Vision Conference 2014*, BMVC 2014. British Machine Vision Association, 2014. doi: 10.5244/c.28.122. URL http://dx.doi.org/10.5244/C.28.122.

[17] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models, 2020.

[18] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models, 2023.

[19] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment, 2023.

[20] Oakes, Calebrisi, and Sotomayor. Tufenkian import export ventures inc v. einstein moomjy inc, 2003. URL https://caselaw.findlaw.com/court/us-2nd-circuit/1455682.html.

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[22] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection, 2022.

[23] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[25] Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Hui Liu, Yi Chang, and Jiliang Tang. Copyright protection in generative ai: A technical perspective, 2024.

[26] Keivan Rezaei, Mehrdad Saberi, Mazda Moayeri, and Soheil Feizi. Prime: Prioritizing interpretability in failure mode extraction, 2023.

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

[28] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[29] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=M3Y74vmsMcY.

[30] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by text-to-image models, 2023.

[31] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models, 2022.

[32] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 47783–47803. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9521b6e7f33e039e7d92e23f5e37bbf4-Paper-Conference.pdf.

[33] Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models, 2024.

[34] Wei Ren Tan, Chee Seng Chan, Hernán E. Aguirre, and Kiyoshi Tanaka. Artgan: Artwork synthesis with conditional categorial gans. *CoRR*, abs/1702.03410, 2017. URL http://arxiv.org/abs/1702.03410.

[35] Nanne van Noord, Ella Hendriks, and Eric O. Postma. Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, 32:46–54, 2015. URL https://api.semanticscholar.org/CorpusID:15774940.

[36] Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris N. Metaxas, and Shiqing Ma. DIAGNOSIS: Detecting unauthorized data usages in text-to-image diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=f8S3aLm0Vp`.

[37] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion based mimicry through score distillation, 2024.

[38] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models, 2023.

[39] Zhengyue Zhao, Jinhao Duan, Kaidi Xu, Chenan Wang, Rui Zhangp Zidong Dup Qi Guo, and Xing Hu. Can protective perturbation safeguard personal data from being exploited by stable diffusion?, 2023.

A Limitations

Our work tackles a novel problem of artistic *style* infringements. Style, however, is qualitative. We merely put forward one definition for artistic style, along with two implementations for demonstrating the existence of a style given example works from an artist and recognizing the identified style in other works.

Importantly, we argue that an artist's style is unique if we can consistently distinguish their work from that of other artists. However, we can only proxy the entire space of artists. We construct a dataset consisting of works from 372 artists spanning diverse schools of art and time periods in attempt to represent the space of existing artists, though of course we will always fall short in capturing all kinds of art. We provide tools to allow for this dataset to grow with time, and we caution that if only one artist for some broader artistic style is not present in our reference set, the uniqueness of that artist's style may be overestimated, and as such, generated images may be matched to this artist with an overestimated confidence. However, if only one out of 372 artists exhibits some style, than one could argue that that alone reflects a notable uniqueness of that artist. To employ a stricter criterion for alleging style copying, we'd recommend augmenting the reference set to include more artists with very similar styles to the artist in question. Nonetheless, we believe our reference dataset does well in representing all art, to where analysis based on this reference set is still informative.

We also note that our atomic tagging leverages an existing foundation model (CLIP) with no additional training. While we verify the precision of our tags, CLIP is known to have issues with complex concepts. Further, we do not claim our tags achieve perfect recall (most image taggers do not). We advise users to interpret the assignment of a tag to indicate a strong presence of that concept, relative to similar concepts (i.e. from the same aspect of artistic style). While our tagger is not perfect, it is objective and automatic, enabling interpretable style articulation and detection. Also, we note that the field of image tagging in general has seen rapid improvement in the past year [14], and an improved tagger could easily be swapped into our pipeline.

Lastly, we only analyze generated images using off-the-shelf text-to-image models. It is possible that particularly determined and AI-adept style thiefs fine-tune a model to more closely replicate specific artistic styles. This is a much more threatening scenario, though requires greater effort and ability by the style thief. We elect to demonstrate the feasability of our approach in the more broadly accessible setting of using models off-the-shelf, and note that our method can flexibly accept generated images produced in a different way (or perhaps discovered on the internet); notice generated images are an optional input in figure 7. We look forward to explorations of more threatening scenarios in future work, and hope both our formulation and methods for measuring style copying prove to be of use.

B A nuance in artistic style infringements: Existing Artists can have very similar styles

A crucial step in arguing that an artist's style has been infringed is to first demonstrate the existence of the given artist's *unique* style. We note that doing so objectively is non-trivial, as a style may not have a clear definition, and thus, it can be challenging to systematically compare to all other artistic styles, so to show uniqueness. In our work, we utilized classification, claiming that if an artist's works can consistently be mapped (i.e. at least half the time) to that artist (over a large set of other artists), than that artist must have some underlying unique style (parameterized by a neural signature).

In doing so, we found that 89.3% of artists could be recognized based of a set of (at least 20 of) their works (held-out in training the classifier). What about the remaining 10.7% of artists? We now take a closer look at these artists, and also introduce a second, stricter style copying criterion. Namely, we consider the notion that it may be unfair to claim a generative model is copying the style of an artist, if another existing artist seems to also be copying that artist. That is, we propose a way to verify that the generative model not only shows a substantial similarity to the copied artist, but also an *unprecedented* similarity.

B.1 Artists who's styles were not recognized

First, we inspect more examples from artists who were not recognized using our majority voting threshold in DeepMatch. That is, less than half of their held-out works were predicted to them. Figure

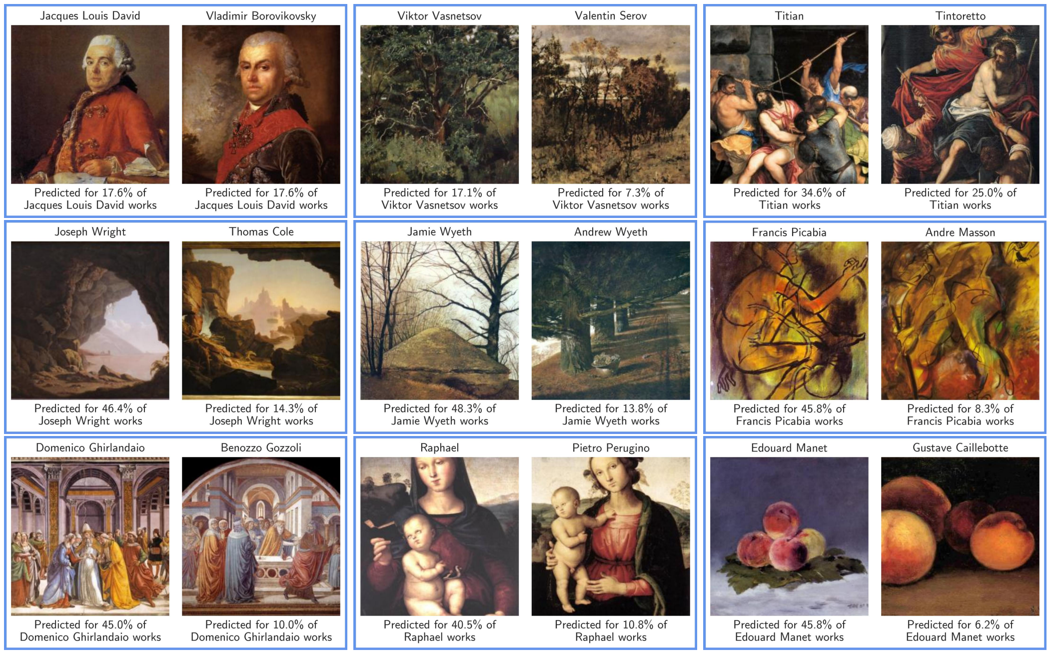

Figure 10: Examples of artists who's styles were not recognized by DeepMatch (i.e. less than half of their held-out works were predicted to the artist). Each panel shows an example work from (left) the unrecognized artist and (right) the artist that is incorrectly predicted most frequently over works from the unrecognized artist. We see that artists can use very similar, at times arguably indistinguishable, styles.

10 shows a number of examples, from which we can make some qualitative observations. First, the styles of artists who operate in the same broader genre (e.g. portraiture, landscapes, narrative scenes in renaissance styles, etc) can be extremely similar. We even see an instance where an artist's son's style is indistinguishable from his father's (Jamie and Andrew Wyeth). Lastly, we note that in most cases, the artists only marginally fall short of our recognition threshold (i.e. accuracy for their held-out works is only a bit below 50%). We utilize majority voting because (i) it is intuitive, (ii) it requires *consistent* appearance of the neural signature across works, and (iii) it allows for abstention when no particular style is strongly present. However, the exact threshold of 50% can be altered as desired. In summary, as in Figure 4, we see artistic styles can be very similar, making the existence of unique artistic styles for the vast majority of artists a non-trivial observation.

If an artist's style cannot be recognized over their own held-out works, arguing that a generative model copies that style is strenuous, as the style itself is ill-defined. Notably, in these cases, the classifier had an option to predict the correct artist. However, in applying DeepMatch to generated images, there is no direct option for the classifier to abstain from predicting anyone, under that generated art comes from a "new artist", which takes inspiration from existing artists. Note that abstention is still possible (due to the majority voting in DeepMatch), and occurs when a match confidence falls below 50%. To make comparisons fairer to generative models, we now discuss a stricter criterion of *unprecedented similarity*.

B.2 *Unprecedented Similarity*: Do generative models copy styles more than existing artists already do?

A nuance that requires consideration when studying artistic style copying is that it is possible for two artists to have very similar styles. Thus, it may be unfair to allege that a generative model is copying an artist a if there exists another artist b who's style is just as or in fact even more similar to artist a. Towards this end, we introduce *unprecedented similarity*, which requires that the similarity between works of a generative model A' and works of the artist inteded to be copied A is higher than the similarity of any existing artist with A. That is, $sim(A, A') \geq sim(A, B)$ for works B from all other existing artists b.

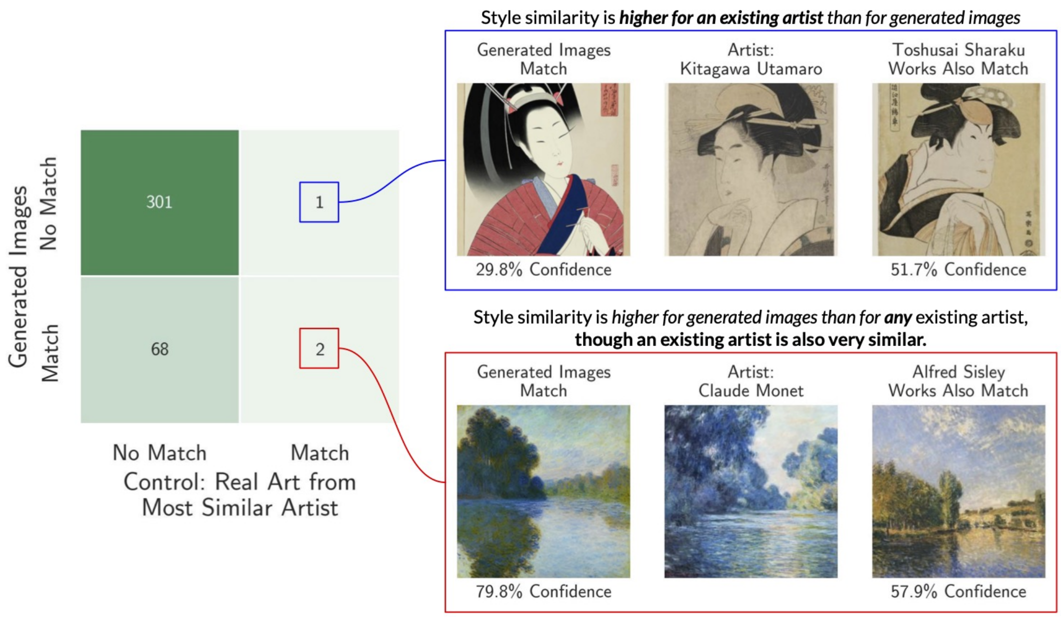

Figure 11: We verify the stricter criterion of *unprecedented similarity* by holding out the real artist with highest similarity to a given artist, and checking if the held-out real artist's works are flagged as potential style copying by DeepMatch. (**left**) We observe only three artists where the most similar held-out artist has their work flagged as a style match, and in all cases, when generated images are flagged, the match confidence of the generated images exceeds that of the held-out real artist's works (i.e., **the generated images flagged by our method reflect *unprecedented similarity* to the given artist's style**). (**right**) Inspecting the flagged held-out artists further show that style copying is very nuanced, as artists take inspiration from one another, and as such, they may already have very similar styles. While we always observe unprecedented similarity, a potential solution to style copying may be for generative models to ensure that they do not copy any more than what already exists; that is, they may exhibit some copying, but no more than for which precedent already exists.

Note that this is a stricter criterion than our previous threshold. In DeepMatch, we required that at least half of the works in a given set of test images were predicted to a single artist in order for us to flag the test images as a potential style infringmenet. In other words, that threshold required that $sim(A, A') \geq 0.5$, which in turn implies that $sim(A, A') \geq sim(A', B)$ for all B (with room to spare; here we use match confidence to denote similarity).

Now, however, instead of just comparing A' to all B, we must also compare all B to A. Instead of comparing all other artists, we inspect the most similar artist b^* to a, identified by taking the artist b with the highest rate of false positive predictions to artist a. Then, we hold out b, and train a new classifier on the remaining 371 artists. Finally, we check for style matches of for the set of generated images A' and the works B^* from the most similar artist b^*.

Figure 11 summarizes our result for OpenJourney (all three models studied show consistent results). We find that only in three cases do we see a held-out artist's work flagged as potential style copying. Notably, in all instances where generated work is flagged as potential style copying, the corresponding held-out artist's work is either not flagged or is flagged with lower confidence, indicating that the instances of style copying of generative models that we observe always also satisfy the criterion of unprecedented similarity.

Taking a closer look at instances where held-out art is flagged for style copying (or perhaps style emulation?), we again see just how similar the works of different artists can be. Namely, we see that some artists works seem to fall into a broader genre of art that many artists utilize (e.g. ukiyo-e or impressionism). In summary, while generative models can very closely resemble the style of a given artist, contextualizing copying by generative models with respect to copying (or perhaps, 'style

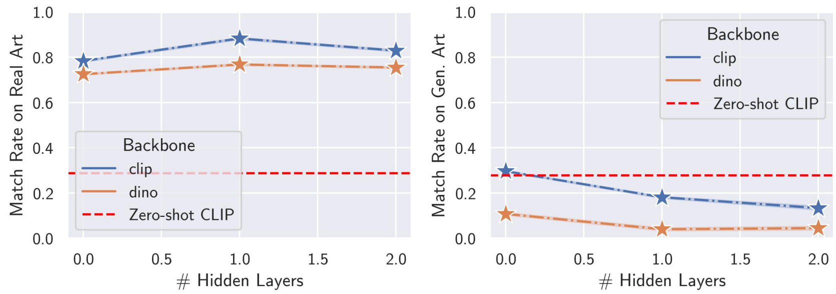

Figure 12: Alternate implementations of DeepMatch, using DINOv2 and CLIP backbones, and varying the number of hidden layers. We also present performance of zero-shot CLIP. Numbers are averaged over five trials, except for zero-shot CLIP, which is deterministic.

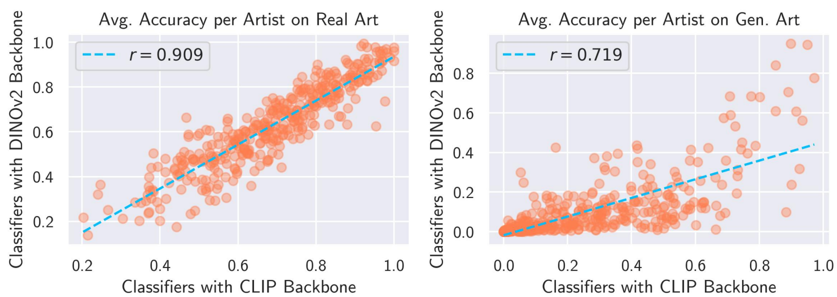

Figure 13: Per-artist accuracy for classifiers using CLIP and DINO backbones are highly correlated. While each classifier may yield different overall accuracy, the *relative* notions of (i) how recognizable the artist's real art is and (ii) how much so the artist's style appears in generated works appear to be classifier agnostic.

emulation') already done by existing artists is crucial in order to afford the same artistic liberties to generative models as have been provided to other artists in the past.

C Baselines

We now present some alternate implementations to the methods we present, so to serve as baselines. We note that a key contribution of our work is reformulating the problem of detecting style infringements from computing image-wise similarity to performing classification over image sets, and building a tool around this idea. Thus, it is rather challenging to perform apples-to-apples comparisons to prior copy detection works, as our methods implement a different task. We include substantial qualitative discussion comparing our approach to image-similarity techniques (and thus motivating our framework) in section 3, and we add to that discussion here.

We further stress that there is not a singular numerical objective that we can use as a way to compare methods. For example, we report the accuracy of matching artists (i.e. aggregating classification predictions with majority voting), but since it is not necessarily true that all artists are distinguishable, it would be imprudent to strictly prefer a higher accuracy, as there is no strict groundtruth; that is, there is no completely definitive way to say if an artist has a unique style or not, due to the subjective/qualitative nature of style. Nonetheless, for lack of other quantitative metrics, we inspect accuracy on real and generated images for a few lightweight approaches to artist classifications, and compare them below.

C.1 DeepMatch

Figure 12 shows the performance of different classifiers, where we vary the frozen backbone and the number of hidden layers. We find that classifiers trained on CLIP yield higher match-rates for both

	CBM	CBM + sparsity	Ours
Accuracy on real art	62.8%	58.7%	61.5%

Table 1: Baselines for TagMatch

real and generated art than classifiers train on DINOv2 [21] embeddings. Interestingly, zero-shot CLIP does poorly on real art, but well on generated art, perhaps because many generative models optimize using CLIP-score, which applies the same mechanism as zero-shot CLIP classification, perhaps explaining the assertion that generative models are highly capable of imitating humans found in this brief work [8]. The number of hidden layers does not have a very strong affect on recognizing real art, but it does appear inversely related to the ability of the model to recognize generated art. It is possible that having two many hidden layers can overfit the model to the distribution of real images, creating a distribution shift when applied on the generated images.

While exact numbers seem to vary, we note that relative trends (i.e. between artists) appear agostic to the underlying classifier. Figure 13 shows accuracy per artist for classifiers trained on CLIP vs DINOv2 embeddings. For both real and generated art, the per-artist accuracies are strongly correlated, which could motivate using relative metrics in addition to absolute values dependent on exact accuracy values; note that we include relative numbers in our `ArtSavant`report (see Figure 1; e.g., 'percentile of recognizability').

We ultimately choose something in the middle of the round: a 1-hidden layer MLP on CLIP embeddings, which has the strongest performance recognizing real art, and appears to have some ability to recognize generated art. We note the majority aggregation that we apply is just one way to summarize the classification output across an image set. We opt for it because it is intuitive and it provides a natural avenue for abstention, though this threshold can be modified as desired, and inspecting relative accuracies could be most informative. We again stress that our current implementation serves as a proof of concept of our framework, which is our primary contribution.

C.2 TagMatch

We now present baselines for TagMatch. Like above, and indeed more so, accuracy is not exactly an objective to maximize. In fact, what is most important with TagMatch is interpretability, and ease with which the output of TagMatch can be used in arguments to a broader, non-technical audience. Thus, we consider a popular framework from the interpretable classification literature: concept bottleneck models (CBM) [17]. Namely, we train a linear layer atop concept predictions extracted from CLIP, so to create a CBM without direct concept supervision, as in [19, 38].

As shown in 1, accuracy values are roughly similar. We note the interpretability provided by the methods are markedly different. CBM allows one to inspect the final linear layer to discern which concepts are important to which class, but this results in requiring users to inspect a coefficient for every concept. Adding sparsity by way of an ℓ_1 penalty can help, but the problem persists. Our version of TagMatch, on the other hand, affords concise articulations of tag signatures, as well as a number of how many other artists share a given signature. Perhaps most crucially, our implementation also yields faithful attribution, which can be critical in gathering evidence to present to a judge or jury.

C.3 Stability

We also explore the stability of our method to using different data splits. We perform five different random train / test splits, and inspect the accuracy of our implementations of DeepMatch and TagMatch. DeepMatch per-image accuracies are very stable, with a standard deviation of 0.1%. TagMatch is also stable, though less so, with a standard deviation 1.1%.

D On alternate prompts

We briefly explore using alternate prompts to generate images. Namely, we create 120 prompts of the form "{an object} in {location} in the style of {artist}" (e.g. "A bottle in forest in the style of

Jeff Koons", which are by nature no longer artist-specific (like the titles we originally use). Using DeepMatch, average match rate drops considerably in this less specific case, from 20% to 8%. This is in line with existing wisdom that prompting can significantly affect the behavior of a model, and also echoes our overall empirical observation that current style copying does not appear to be very prevalent. We hope that our framework can be useful in examining which prompts induce greatest copying going forward, especially as prompt and model sophistication grows.

E Details on TagMatch

We now provide greater details regarding the implementation of TagMatch, a central technical contribution of our work. TagMatch is a method to classify a set of images to a class; specifically, we map a set of artworks to one artist, selected over 372 choices. TagMatch is not as accurate as DeepMatch, as it maps held-out works of each artist in our WikiArt dataset to the correct artist about 61% of the time (compared to 89% top-1 accuracy for DeepMatch). However, top-5 accuracy is more reasonabe, achieving above 80%. Most notably, **TagMatch is inherently interpretable and attributable**. It consists of three steps: (i) assigning atomic tags to images, (ii) efficiently composing tags to obtain more unique *tag signatures*, and (iii) matching a test set of images to a reference artist based on the uniqueness of the tags shared between the test set and works from the predicted reference artist.

Our method is fast and flexible: after caching image embeddings, the whole thing only takes minutes, and it is easy to modify the concept vocabulary as desired, as the tagging is done in a zero-shot manner. Through MTurk studies, we verify that the atomic tags we assign our mostly precise, though we recognize that these descriptors can be subjective. Thus, while we do not claim perfect tagging, we stress that our method is easy to understand, and crucially, is deterministic per image. Therefore, ideally our tagging may be more reliable biased than human judgements, particularly when the humans involved may be biased (e.g. an artist alleging copying and a lawyer defending a generative model would have strong and opposing stakes).

Below, we provide details for image tagging (§E.1), artist tagging (§E.2), artistic style inference via tag matching (§E.3), effect of hyperparameters (§E.4), details on efficiency (§E.5), and a review of validation (§E.6).

E.1 Image Tagging

As explained in §4.2, we utilize CLIP to attain a diverse set of atomic tags per image in a zero-shot manner. Specifically, we first define a vocabulary of descriptors along various aspects of artistic style. Then, given an image, we do selective multi-label zero-shot classification *for each aspect*. Performing zero-shot classification per aspect proves to be critical in order to achieve a diversity of tags and a similar number of tags per image. We find that some descriptors always lead to higher CLIP similarities than others. Specifically, descriptors for simple aspects, like colors and shapes, yield higher similarities than more complex aspects like brushwork and style. Thus, using a global threshold across descriptors would lead to a less diverse descriptor set. Moreover, we observe some images have higher similarities across the board than others, which again would lead global thresholding to result in a disparate number of tags per image. Our per-aspect scheme requires that the descriptors within each aspect are mostly mutually exclusive; we prioritize this in the construction of the concept vocabulary, via the prompt we present the LLM assistants and our manual verification.

Namely, we prompt both Vicuna-33b and ChatGPT with "*I want to build a vocabulary of tags to be able to describe art. First, consider different aspects of art, and then for each aspect, list about 20 distinct descriptors that could describe that aspect of art. Please return your answer in the form of a python dictionary.* ". We then perform a filtering step with a human in the loop, where we manually remove tags that are difficult to recognize or redundant. After this filtering step, we add in a few new aspects. First, we incorporate the 20 *styles* (e.g., "impressionism") and *genres* (e.g., "portrait") that are most common amongst works in our WikiArt dataset; note that all WikiArt images also contain metadata for these categories. Finally, we add some easy to understand tags such as *color* and *shape* which can be important characteristics describing a given painting. The concept vocabulary we use is contains shown below:

- **Style**, caption template: *{} style*. Descriptors:

- *realism, impressionism, romanticism, expressionism, post impressionism, art nouveau modern, baroque, symbolism, surrealism, neoclassicism, naïve art primitivism, northern renaissance, rococo, cubism, ukiyo e, abstract expressionism, mannerism late renaissance, high renaissance, magic realism, neo impressionism*

- **Genre**, caption template: *the genre of {}*. Descriptors:
  - *portrait, landscape, genre painting, religious painting, cityscape, sketch and study, illustration, abstract art, figurative, nude painting, design, still life, symbolic painting, marina, mythological painting, flower painting, self portrait, animal painting, photo, history painting, digital art*
- **Colors**, caption template: *{} colors*. Descriptors:
  - *pale red, pale blue, pale green, pale brown, pale yellow, pale purple, pale gray, black and white, dark red, dark blue, dark green, dark brown, dark yellow, dark purple, dark gray*
- **Shapes**, caption template: *{}*. Descriptors:
  - *circles, squares, straight lines, rectangles, triangles, curves, sharp angles, curved angles, cubes, spheres, cylinders, diagonal lines, spirals, swirling lines, radial symmetry, grid patterns*
- **Common Objects**, caption template: *{}*. Descriptors:
  - *male figures, female figures, children, farm animals, pet animals, wild animals, geometric shapes, fruit, vegetables, intsruments, flowers, boats, waves, roads, household items, the moon, the sun, saints, angels, demons*
- **Backgrounds**, caption template: *{} in the background*. Descriptors:
  - *fields, blue sky, night sky, sunset or sunrise, forest, rolling hills, simple colors, beach, port, river, starry night, clouds, shadows, living room, bedroom, trees, buildings, chapels, heaven, hell, houses, streets*
- **Color Palette**, caption template: *{} color palette*. Descriptors:
  - *vibrant, muted, monochromatic, complementary, pastel, bright, dull, earthy, bold, subdued, rich, simple, complex, varying, minimal, contrasting*
- **Medium**, caption template: *the medium of {}*. Descriptors:
  - *oil painting, watercolor, acrylic, ink, pencil, charcoal, etching, screen printing, relief, intaglio, collage, montage, photography, sculpture, ceramics, glass*
- **Cultural Influence**, caption template: *{} influences*. Descriptors:
  - *Indigenous, European, American, East Asian, Indian, Middle Eastern, Hispanic, Aztec, Contemporary, Greek, Roman, Byzantine, Russian, African, Egyptian, Tahitian, Polynesian, Dutch*
- **Texture**, caption template: *{} texture*. Descriptors:
  - *rough, smooth, bumpy, glossy, matte, roughened, polished, textured, smoothed, brushstroked, layered, scraped, glazed, streaked, blended, uneven, smudged*
- **Other Elements**, caption template: *{}*. Descriptors:
  - *stippled brushwork, chiaroscuro lighting, pointillist brushwork, multimedia composition, impasto technique, repetitive, pop culture references, written words, chinese characters, japanese characters*

Now, we detail the implementation of our modified zero-shot classification. Recall that in zero-shot classification, one computes a text embedding per class, which amounts to the classification head, and computes an image embedding for the test input, so that the prediction is the class who's text embedding has the highest cosine similarity to the test image embedding. In computing the text embeddings, we take each descriptor (e.g. *Dutch*) and place it an aspect-specific caption template (e.g. *Dutch* $\rightarrow$ *Dutch influences*), and then average embedddings over multiple prompts (e.g. "artwork containing *Dutch influences*", "a piece of art with *Dutch influences*", etc), as done in [24]. We modify standard zero-shot classification to allow for the fact that more than one descriptor (or perhaps none) from a given aspect may be present. Namely, instead of assigning the most similar descriptor

Algorithm 1 Iterative Algorithm to Obtain Tag Composition Per Artist $a \in \mathcal{A}$

Require: $\mathcal{D}_a$ (Images for artist a), $\mathcal{C}_a$ (Common tags for artist a)
  $\mathcal{S}_a = \{\}$ ▷ Stores the tag compositions with their associated counts
  for $x \in \mathcal{D}_a$ **do**
    $I(x) = \text{tag}(x) \cap \mathcal{C}_a$ ▷ Compute the intersection with common atomic tags
    $\mathcal{P}(I(x)) = \text{ComputePowerSet}(I(x))$ ▷ Compute power-set of the tags
    UpdateCount($\mathcal{S}_a, \mathcal{P}(I(x))$) ▷ Update the count of each tag composition
  end for
  Filter(S_a) ▷ Keep tag compositions which occur above a count threshold of 3

per-aspect, we assign an atomic tag for any descriptor who's similarity is significantly higher than other descriptors for that aspect. We achieve this via z-score thresholding: per-aspect, we convert similarities to z-scores by subtracting away the mean and dividing by the standard deviation, and then admit atomic tags who's z-score is at least 1.5.

The template prompts we utilize for embedding each concept caption are as follows:

- art with
- a painting with
- an image of art with
- artwork containing
- a piece of art with
- artwork that has
- a work of art with
- famous art that has
- a cropped image of art with

E.2 From Image Tags to *unique* Artist Tags

Recall that we define styles not per-image, but over a set of images. Namely, we seek to surface tags that occur frequently. The best way to do so is to simply count the occurrences of each tag, and discard the ones that rarely appear. However, each atomic tag is not particularly unique with respect to artists. We utilized *efficient composition* of atomic tags to arrive at more unique tag signatures, as shown in figure 6 and detailed in algorithm 1. Importantly, we utilize a threshold here to differentiate what a common tag is; we require a tag to appear in at least three works for an artist in order for the tag to count as a frequently used tag by the artist. We note that tag composition can be done efficiently because we have a relatively low number of tags per image: on average, there are 6.2 atomic tags per image. Moreover, because the number of occurrences for a composed tag is bound belo by the number of occurrences of each atomic tag in the composition, we can ignore all non-frequent atomic tags. Thus, we can iterate over the powerset of common atomic tags per image without it taking exorbitantly long. We include one fail safe, which is that in the rare instance where an image has a very high number of common atomic tags, we truncate the tag list to include only 25 tags. Over the $91k$ images that we encounter, this happens only once. We highlight that our tag composition takes inspiration from [26].

E.3 Predicting Artistic Styles based on Matched Tags

Once we have converted tags per image to tags per artist, we can then utilize these artist tags to perform inference over a set of images. Namely, given a test set of images, we extract common tags (including tag compositions) for the test set and compare them to tags extracted for each artist in our reference corpus. Then, we predict the reference artist who shares the most unique tags with the test set.

Figure 14 best explains our method, as it shows the documented code. We note that all code will be released upon acceptance. We'll now explain it step by step. First, for each artist and for the test set of images, we find common tags via (i) assigning atomic tags to each image, (ii) finding the commonly

```python
def tag_match(self, test_img_paths: List[str], test_artist: str):
    dset = BasicDsetFromImgPaths(test_img_paths, self.vlm.transform, dsetname=test_artist)

    tags_by_path = self.tag_images(dset)
    common_tags = self.find_common_tags(tags_by_path)
    composed_tags_w_counts = self.compose_tags(common_tags, tags_by_path)

    # Now we cross-reference the found tags w/ tags for reference artist
    counts_over_ref_artists_by_tag = dict({
            t:len(self.ref_artists_by_tag[t])
            for t in composed_tags_w_counts if t in self.ref_artists_by_tag
        })
    # We sort the tags by uniqueness: we first inspect tags that occur for the lowest number of reference artists
    counts_over_ref_artists_by_tag = dict(sorted(counts_over_ref_artists_by_tag.items(), key=lambda x:x[1]))

    # We will return a score per artist to resemble the typical output of a classifier
    scores_by_artist = dict({artist: [] for artist in self.ref_dset.artists})
    # We will also keep track of the tags used in computing the score per artist -- this provides faithful interpretations
    matched_tags_by_artist = dict({artist: [] for artist in self.ref_dset.artists})
    # Now we loop through each tag that also occurs for reference artists
    for t, num_ref_artists_w_tag in counts_over_ref_artists_by_tag.items():
        # For each tag, we loop through all matches (i.e. any reference artist that also has the tag)
        for ref_artist in self.ref_artists_by_tag[t]:
            # We only consider the top k most unique matched tags per artist (k = self.matches_per_artist_to_consider)
            if len(scores_by_artist[ref_artist]) < self.matches_per_artist_to_consider:
                # Compute frequency of matched tag over works from the reference artist
                num_works_of_ref_artist_w_tag = self.ref_tags_w_counts_by_artist[ref_artist][t]
                freq_for_ref_artist = num_works_of_ref_artist_w_tag / self.num_works_by_ref_artist[ref_artist]
                # Compute frequency of matched tag over works from the test artist
                freq_for_test_artist = composed_tags_w_counts[t] / len(tags_by_path)
                # Our score is the uniqueness of the matched tag + |diff in frequencies of tag for ref artist and test artist|
                scores_by_artist[ref_artist].append(num_ref_artists_w_tag + np.abs(freq_for_ref_artist - freq_for_test_artist))
                matched_tags_by_artist[ref_artist].append(t)

    # We set the score to inf for any artists that did not have enough matched tags
    scores = np.array([np.mean(scores_by_artist[artist][:self.matches_per_artist_to_consider])
        if len(scores_by_artist[artist]) >= self.matches_per_artist_to_consider else np.inf for artist in self.ref_dset.artists])

    # Finally, we return scores along with explanations for each artist
    return scores, matched_tags_by_artist
```

Figure 14: Code for predicting artistic styles via matched tags.

occurring atomic tags, (iii) counting compositions of the commonly occurring atomic tags, and (iv) discarding tags (including compositions) that do not occur frequently enough. The code shows this done for the test set of images; we perform this per reference artist when the `TagMatcher` object (for which `tag_match` is function) is initialized; notice fields like `self.ref_tags_w_counts_by_artist`, which contain useful information about the reference artists, computed once and re-used for each inference.

Then, we loop through the set of 'matched' tags (i.e. those that occur for both the test set of images and at least one reference artist), starting with the most unique ones. Here, uniqueness refers to the number of reference artists that frequently use a tag. For each tag, we loop through all artists that also use that tag. For the first k (denoted by `self.matches_per_artist_to_consider` in the code) matched tags per artist, we add a score to a list of scores for the artist, which ultimately are averaged. The score contains an integer and a decimal component. The integer component is the number of reference artists that share the matched tag. The decimal component is the absolute value of the difference in frequency with which the tag appears, over the reference artist's works and the test set of images; note that this is always less than one. This way, when comparing two matched tags, a lower score is assigned to a more unique one, and one there is a tie in uniqueness, we break the tie based on how similar the frequency of the matched tag is for the test artist and reference artist.

Finally, we average the list of scores per artist to get a single score per reference artist, analogous to a logit. We assign a score of `inf` for any artist with less than `self.matches_per_artist_to_consider` (which we set to 10) matched tags. This hyperparameter makes our tag matching less sensitive to individual matched tags, and empirically results in a substantial improvement in top-1 accuracy on held-out art from WikiArt artists (see next section).

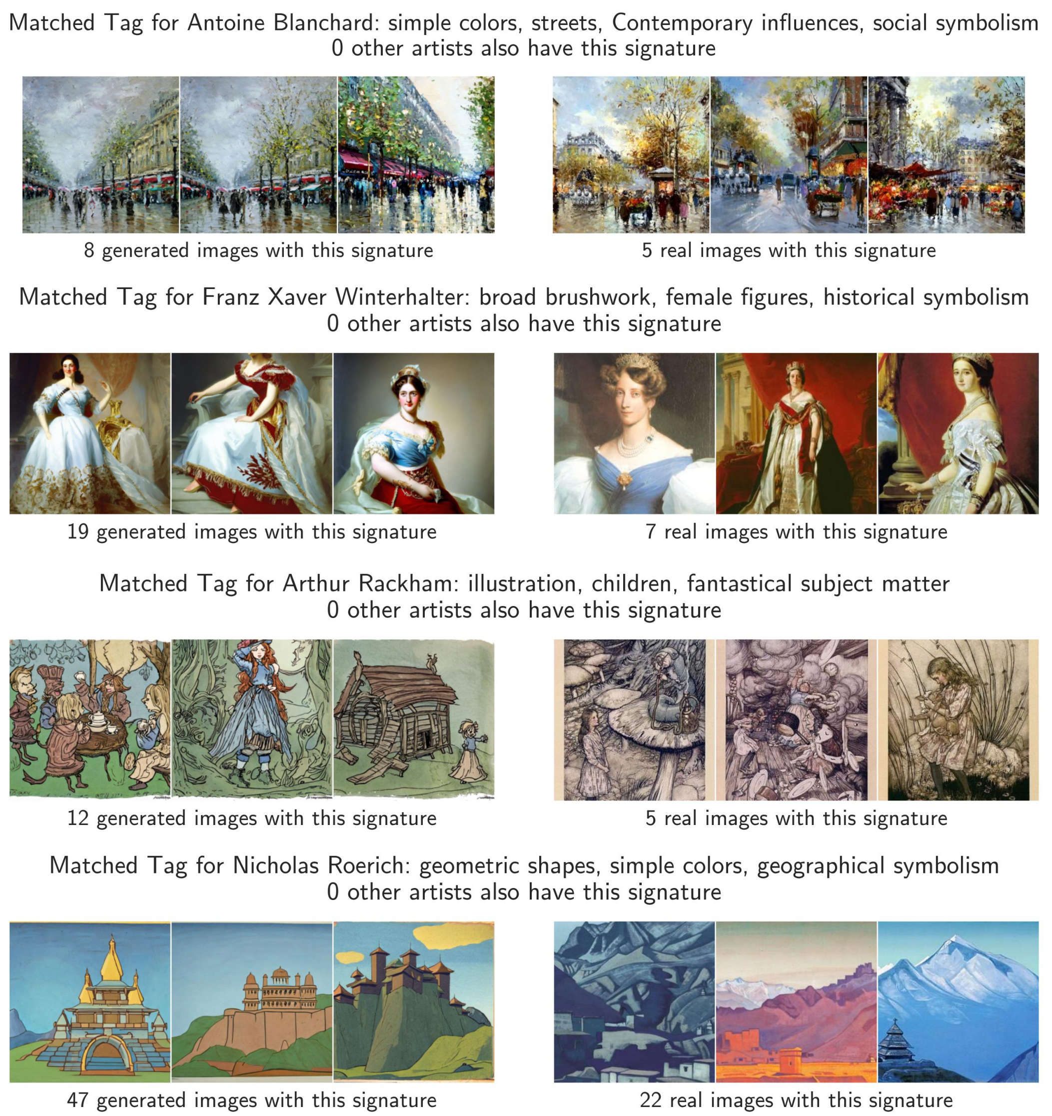

Figure 15: Additional examples of applying TagMatch to generated images.

E.4 Choosing Hyperparameters

Overall, there are three hyperparameters to our method: the z-score threshold, the tag count threshold, and the number of matches to consider per artist. Here is quick refresher on what they each do:

- The z-score threshold determines how much more similar a descriptor needs to be to an image compared to other descriptors for the same aspect in order for the descriptor to be assigned as an atomic tag of the image. The value we use is 1.75.
- The tag count threshold is the minimum number of an artist's works that a tag needs to be present in order for a the tag to be deemed common for the artist. The value we use is 3.
- The number of matches to consider per artist pertains to how many matched tags are considered when computing the final score per artist in tag match. That is, the final score for an artist is the average of the top-k most unique tags that the artist shares with the test set of images, where k corresponds to this hyperparameter. The value we use is 10.

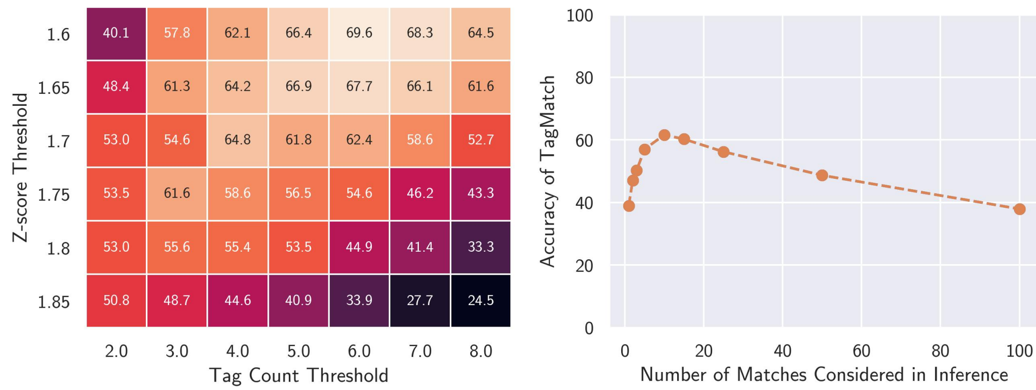

Figure 16: Sweep of hyperparameters asssociated with TagMatch. **(left)** We jointly sweep the z-score threshold and the tag count threshold. **(right)** Having fixed the first two parameters, we sweep the last one: the number of matches considered in inference. See detailed discussion in §E.4.

Now that the role of each hyperparameter is clear, let's discuss how hyperparameters can be adjusted towards particular ends, along with the potential consequence of each action:

- To increase the number of atomic tags, lower the z-score threshold. Risk: atomic tags may be less precise, and the method will take longer to run, as there will atomic tags and composed tags.
- To get more tags per artist, lower the tag count threshold. Risk: some tags will become less unique. Other tags will be introduced, and may be very unique, which could skew tag matching. Also, the method may take longer to run, as there will be more tags.
- To make inference less sensitive to a low number of matched tags, increase the number of matches to consider per artist. Risk: when you consider more matches, interpretation is a little more difficult, as you have more reasons for each inference, and it will take longer to view them all.

To choose hyperparameters, we selected a small range of reasonable values and swept each hyperparameter individually. While a combined search would likely yield better accuracy numbers, we opt out of hyper-tuning TagMatch for accuracy, as its main objective is to provide and interpretable and attributable complement to DeepMatch. We find the (relatively strong, considering the high number of artists considered) accuracy numbers encouraging, but do not find it a priority, as DeepMatch arguably provides a stronger and easier to understand signal of *if* style copying is happening. TagMatch, on the other hand, tells us *how* and *where* it is happening (if observed with DeepMatch).

We also include a hyperparameter sweep, of the z-score threshold and tag count threshold jointly, and of the number of matches to consider separatedly afterwards. Figure 16 visualizes the results. Choosing a lower z-score threshold results in higher TagMatch accuracies. However, a lower z-score threshold would admit a greater number of false positive tags, and also incurs a longer time of computation, as there are more tags to compose (we empirically observe an increase of about 50% in run time using our 372 artist reference corpus). Increasing the tag count threshold can reduce the time of computation and also increase sensitivity to false positive tags (on individual images), resulting in higher TagMatch accuracies. Interestingly, considering more matches improves accuracy considerably, but eventually saturates and reduces accuracy. Essentially, by considering more matches per artist, inference becomes less sensitive to the most unique matched tag between the artist and the test set. The smoothed predictions are more accurate up to a point (i.e. 10 matches), but then hinder accuracy. Also, choosing too high a number here can make faithful interpretation more cumbersome, as there are more matches to inspect afterwards.

We reiterate that the main goal of TagMatch is not to be super accurate, but to complement DeepMatch with interpretations (via matched tag signatures) and attributions (via works from the test set and from the reference artist that present the matched tags). We ultimately first choose a high z-score threshold of 1.75, as a preliminary check revealed this threshold to have considerably higher precision in its atomic tags (which we validate with a human study), and since it speeds up the analysis. Then,

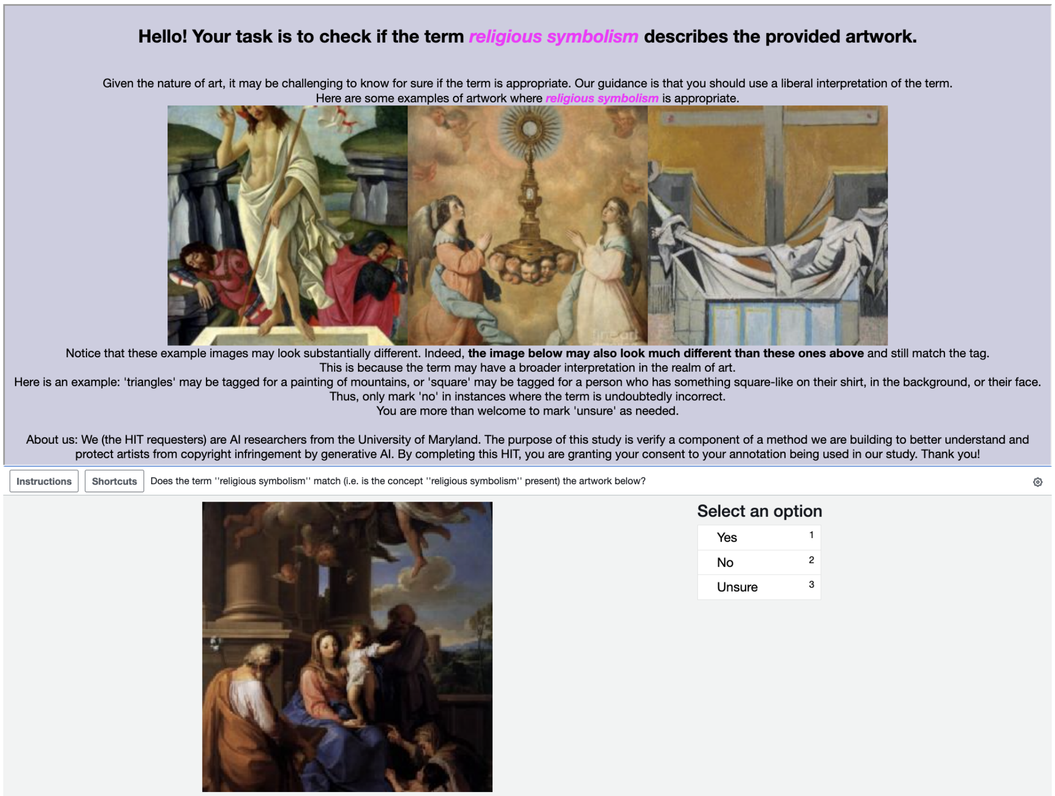

Figure 17: Instructions showed to MTurk workers to validate atomic tags.

we choose the best tag count threshold (3) and number of matches to consider (10), in that order. We hope our discussion of the impact of each hyperparameter can enable practitioners to modify these choices as they please. Furthermore, as base VLMs and tagging methods improve, our framework can modularly swap out our zero-shot tagging (and thus also the z-score threshold) for a stronger method, while retaining the other structure of TagMatch.

E.5 Efficiency of TagMatch: Runs in roughly 1 minute

TagMatch is surprisingly fast. The longest step by far is computing CLIP embeddings for the reference artworks. This takes us about 5 minutes using one rtx2080 GPU with four CPU cores to embed the $73k$ training split images using a CLIP ViT-B$\backslash$16 model. Importantly, this step is done only once, and in practice, is done offline. The other steps and approximate time needed for each are as follows: embedding concepts (5 seconds), extracting common atomic tags and composing them (45 seconds), reorganizing tags and removing non-common tags (3 seconds). Then, inference for a test set of $100-200$ works takes about 10 to 15 seconds. Again, we will release all code upon acceptance, as we truly hope our tool can be of use to artists who are concerned by generative models potential infringing upon their unique styles.

E.6 Validation

Because tag match has multiple steps, we perform multiple validations. First, for image tagging, we utilize an MTurk study. We collect 3000 separate human judgements on instances of assigned atomic tags. Namely, we show 1000 randomly selected (tag, image) pairs to three annotators each. Figure 17 shows an example of the form presented to MTurk workers. MTurkers provide consent and are awarded $\$0.15$ per task, resulting in an estimated hourly pay of $\$12-\$18$. For each task, they answer 'yes', 'no', or 'unsure' to the question 'does the term {atomic tag} match the artwork below?' They are also shown example artworks for each term which were manually verified to be

		Top 1	Top 5	Top 10
Generated Art	CompVis Stable Diffusion v1.4	10.10	35.49	49.74
	Stability AI Stable Diffusion v2	12.95	37.82	52.59
	PromptHero Openjourney	6.99	31.87	45.08
	Average	10.02	35.06	49.14
Real Art (held out)		61.56	82.53	88.44

Table 2: Match rates using TagMatch for three generative models, as well as on real held out art.

correct. Response rates were as follows: 69.89% yes, 8.99% unsure, 21.12% no. In investigating inter-annotator agreement, we find that at least 2 annotators agree 92.1% of the time, but all 3 agree only 51.52% of the time. This reflects the subjectivity associated with assigning artistic tags, and partially motivates the need for a deterministic automated alternative, in order to objectively tag images at scale. All three annotators said no only 5.16% of the time, and at least two said no 17.11% of the time, suggesting that our zero-shot tagging mechanism achieves reasonable precision.

To validate the value of tag composition, we refer to figure 6, which shows how tags become more unique as they get longer (i.e. consist of more atomic tags). Moreover, our time analyses show that the added benefit of composing tags to find unique tag signatures does not come at the cost of the efficiency of our method. Finally, the non-trivial top-1 matching accuracy and strong top-5 matching accuracy shows that the extracted tag signatures do indeed capture some unique properties of artistic style. Figure 15 reflects a few more examples of successful inference, interpretation, and attribution for the task of detecting style copying by generative models.

F A Sim2Real Gap in Tag Distributions

An added advantage of ascribing tags to images is that we can better compare image distributions from an interpretable basis (the tags). We briefly explore this direction now.

First, we provide complete results from applying TagMatch to generated images from each of the three text-to-image models in our study, presented in table 2. Consistent with our DeepMatch results, we observe substantially lower matching accuracy for generated images than for real held-out artwork. While the primary takeaway is that for many artists, generative models struggle to replicate their styles, we can also hypothesize that generative models may output images that follow a different distribution than the distribution of real artworks.

Motivated by this hypothesis, we now compare the distribution of real to generated artworks from the perspective of tags. Because we consider composed tags, the total space of tags is vast and hard to reason over. However, we can look at properties of each tags. Namely, we can inspect the uniqueness of tags. That is, for each tag present in generated images, we inspect the number of reference artists that also present that tag; we do the same for real art as well (subtracting one so to not double count the artist for which a given a tag is being considered). Figure 18 shows a kernel density estimation plot of the distributions of tag commonality, where a tag commonality of 5 means that for each tag assigned to a set of images (either from a real artist or from a generative model emulating an artist), 5 other artists also commonly use that tag. We see tags tend to be rather unique (due to our tag composition), and notably, tags for generated images are more unique.

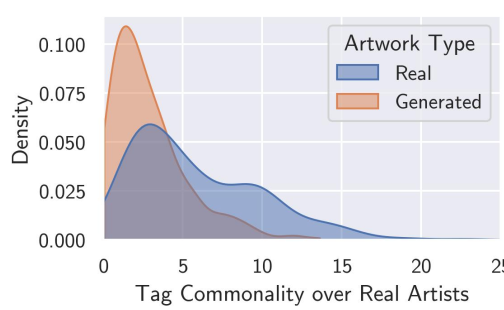

Figure 18: The tags for generated images are less common compared to tags in real art.

G Patch Match: Generating Additional Visual Evidence of Copying

Detecting artistic style copying in a given art requires analyzing local stylistic elements that manifest across an artist's body of work. To address this, we employ a patch-based approach that compares small image regions between a given art and original artworks, enabling a fine-grained analysis of stylistic and semantic (e.g. objects) similarities at a local level. We consider three patch matching methods: CLIP-based, DINO-based, and Gram matrix-based.

Gram Matrix-based Patch Matching [12]: The Gram matrix is a measure of style similarity introduced in the context of neural style transfer. It captures the correlations between the activations of different feature maps in a convolutional neural network, representing the style of an image. For patch matching, the Gram matrices of patches from the given art and original arts can be computed and compared using a suitable distance metric (e.g., Frobenius norm). The Gram matrix is specifically designed to capture stylistic elements, making it well-suited for detecting style copying.

CLIP-based Patch Matching [24]: CLIP (Contrastive Language-Image Pre-training) is a powerful model that can effectively capture the semantic similarity between text and images. In the context of patch matching, CLIP embeddings can be used to measure the similarity between a patch from a given art and patches from original artworks. The patches can be encoded using the CLIP image encoder, and the cosine similarity between their embeddings can be computed to find the closest matches. CLIP may not be as sensitive to low-level stylistic elements, such as brushstrokes, textures, and color palettes, however it focuses more on higher-level semantic concepts, which can be useful to find if the given art pictured the same objects as the selected original patch.

DINO-based Patch Matching [7]: DINO is a self-supervised vision transformer that learns robust visual representations by solving a self-distillation task. DINO embeddings can be used for patch matching by computing the cosine similarity between the embeddings of patches from the given art and original artworks. We use DINO to capture higher semantical similarities, and check whether the given art pictured similar subjects of interest and high-level visual features as selected original artworks.

G.1 Experimental setting

For our experiments, we aim to identify the most similar artwork from a pool of $10,000$ original artworks in the WikiArt dataset given a reference image. The reference image is first resized to a resolution of $512 * 512$ pixels and normalized. From this normalized image, we select a patch size of $128 * 128$ pixels. This process is repeated for all original artworks in the dataset, resulting in a total of $40,000$ patches from original artworks for comparison with the reference patch. We then use three methods, namely Gram matrix, CLIP, and DINO, to find the most similar patches.

Figure 19 showcases the patches that are deemed most similar to the image being referenced. These matches are determined using Gram-matrix, CLIP, and DINO methods.

We then select an artist and find patches from our original image dataset that closely match this artist's style. In Figure 20, we utilize the Gram-matrix method to identify the most similar patches to three chosen artworks by Van Gogh. Our dataset includes all paintings by Van Gogh as well as works by nine other artists. Gram-matrix selects original artworks that closely resemble the style of the reference image, all of which are from Van Gogh. Essentially, this means that Gram-matrix predominantly selects Van Gogh's artworks because they are the most stylistically similar to the referenced paintings compared to the works of the other nine artists.

G.2 Discussion and limitations

Patch matching methods like Gram-matrix, CLIP, and DINO are effective in detecting similarities between artworks by examining their local stylistic and semantic elements. Gram-matrix focuses on capturing stylistic correlations, CLIP evaluates semantic similarity, and DINO concentrates on higher-level features. However, these methods have limitations. They primarily focus on local aspects of artworks and may overlook broader artistic characteristics such as texture, composition, and brushwork that are crucial to detect copyright infringements. Moreover, the process of finding the most similar patches for each given art takes approximately fifteen minutes when considering $10,000$ original artworks, and if we opt to include more original artworks, the duration of the process would inevitably increase. Therefore, patch-matching methods are computationally expensive, which restricts their practical application. Despite these limitations, patch matching is valuable

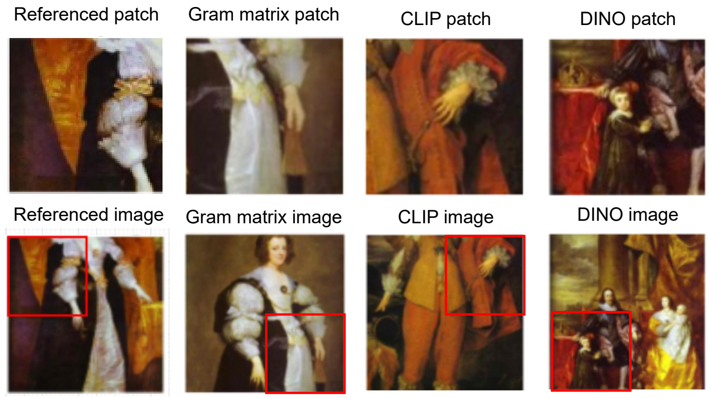

Figure 19: The most similar patches to a referenced patch in an image using Gram-matrix, CLIP, and DINO.

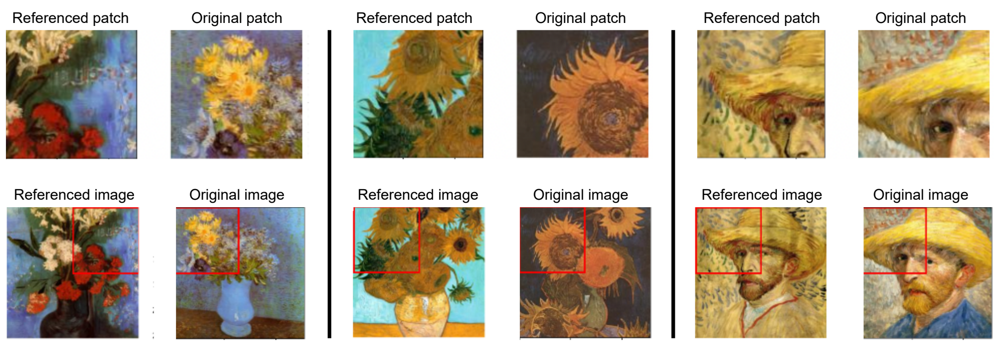

Figure 20: Comparison of patches using the Gram-matrix method, highlighting the closest matches to three selected artworks by Van Gogh. The selected original arts, all from Van Gogh, closely resemble the style of the referenced paintings.

for identifying instances of direct copying in artworks and they aid in the detection of plagiarized content.

H Details on WikiArt Scraping

WikiArt is a free project intended to collect art from various institutions, like museums and universities, to make them readily accessible to a broader audience. We design a scraper to collect a corpus of reference artists, with which we can define a test artist's style in contrast to the other artists, and to provide a testbed to empirically study copying behavior of generative models. Some important landing pages to perform scraping are (i) the works by artist page (`https://www.wikiart.org/en/Alphabet/j/text-list`; url shows all artists starting with the letter 'j', and we loop through all letters), (ii) the page containing information on allowed usage (`https://www.wikiart.org/en/terms-of-use`), (iii) an example artist landing page (`https://www.wikiart.org/en/vincent-van-gogh`), and (iv) an example painting landing page (`https://www.wikiart.org/en/vincent-van-gogh/the-starry-night-1889`). As you can see, many pages have standard formats, making scraping particularly feasible. We will provide our scraping code, along with all other code, to facilitate easy updating of our dataset as time goes by.

We obtain artworks only from artists with at least 100 works on WikiArt, so to focus on somewhat famous artists who are arguably more likely to be copied. For every work, we also scrape the licensing information, and annotation for styles, genres, and title. In total, our dataset has 90,960 artworks over 372 artists. There are 81 styles with at least 100 works, with the most popular styles being *realism, impressionism, romanticism,* and *expressionism*. There were 37 genres with at least 100 works, with the most popular being *portrait, landscape, religious painting, sketch and study*, and *cityscape*. We note that we only include images who's license is either public domain or fair use, with the vast majority of works being public domain. Nonetheless, we strongly advise against using this dataset for commercial purposes, and especially for the purpose of copying artists.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the abstract accurately summarizes the paper's claims, contributions, and scope. We do indeed release a tool consisting of two complementary components, including a highly interpretable one, and we utilize this tool to conduct an empirical study who's results are as stated in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a detailed discussion of limitations as the first section in our Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No theoretical resutls.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We explain all methods and experiments in detail, with lots of additional detail provided in the appendix. We also provide code in a zip file, and will fully open source all code and data if the paper is accepted.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Documented code is attached in a zip file, and lots of details are included in the appendix, including a code block. We include code to scrape the dataset as well, but provide cached embeddings so that experiments can be run without scraping the dataset.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Again, we provide extensive details in the appendix for both of our methods. These details can also be found in the attached code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In the appendix, we perform stability analyses where we conduct multiple trials in instances where randomness may be at play, and even try different splitting of our data to confirm that our hyperparameters are not overfit to our test set.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We conduct all experiments using a single RTX2080 GPU with four cpu workers. We discuss the time to run our method as well. In general, this method is efficient and does not require much compute.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We adhere to the ethical guidelines and discuss the societal implications of our work at length.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This paper is designed to answer a pressing legal and material question around how AI ultimately affects people. We attempt to be objective in our analysis, while building a tool that will help artists with stylistic infringments, even if they are not being infringed upon yet. This tool can also help producers of generative models defend themselves, as they now have a way to say that they aren't producing infringing upon unique artistic styles (when that is the case).

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss the potential risks and safeguards associated with our dataset in the Appendix.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We mention the licenses of the data we use, and include these licenses in the metadata of our dataset for others to see later.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Included in the appendix, with workers receiving pay between $12 and $18 an hour (USD).

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We confirm with IRB that our crowdsourced validation does not require IRB review.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.