NeAT: Neural Artistic Tracing for high resolution Style Transfer

Dan Ruta[1] Andrew Gilbert[1] John Collomosse[1,2] Eli Shechtman[2] and Nicholas Kolkin[2]
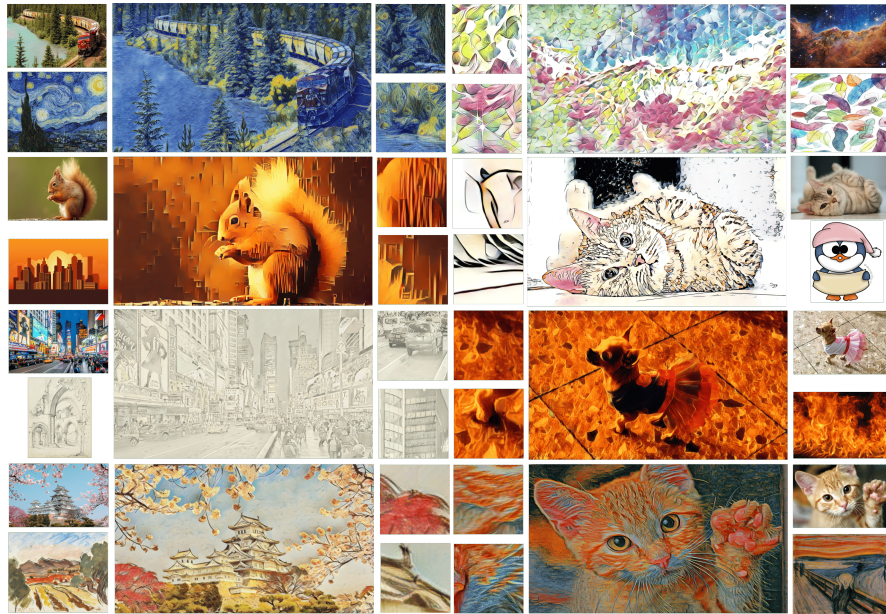
[1] University of Surrey
[2] Adobe Research

Figure 1. Style transfer results using NeAT, trained on BBST-4M. Zoomed in areas are shown in the middle columns.

Abstract. Style transfer is the task of reproducing the semantic contents of a source image in the artistic style of a second target image. In this paper, we present NeAT, a new state-of-the art feed-forward style transfer method. We re-formulate feed-forward style transfer as image editing, rather than image generation, resulting in a model which improves over the state-of-the-art in both preserving the source content and matching the target style. One component of our model's success is identifying and fixing "style halos", a commonly occurring artefact across many style transfer techniques. In addition to training and testing on standard datasets, we introduce the BBST-4M dataset, a new, large scale, high resolution dataset of 4M images. As a component of curating this data, we present a novel model able to classify if an image is stylistic. We use BBST-4M to improve and measure the generalization

of NeAT across a huge variety of styles. Not only does NeAT offer state-of-the-art quality and generalization, it is designed and trained for fast inference at high resolution.

1   Introduction

Since the introduction of techniques for neural style transfer (NST) by Gatys et al. [11], there has been an explosion of research driving style transfer with deep neural networks. Initially performed via an optimization based approach, interest has gradually shifted to parametric and feed-forward approaches. The primary drawback of optimization approaches are their lengthy run-times, limiting their practical use. The research field has thus focused on advancing the quality and style diversity of feed-forward approaches to match or exceed that of optimization models, whilst maintaining practical run-times. First, by adapting models to contain stylization capabilities for multiple styles [5,14], and eventually arbitrary styles [13,18].

We set out to achieve 3 main objectives: **A)** state-of-the-art generality to both modern and classical styles; **B)** efficient scaling to high resolution images; and **C)** state-of-the-art visual stylization quality. We achieve all three of these objectives at the same time. Our contributions are:

1. Reformulation of NST to modify pixels in a content image rather than to generate a brand new image (contrasting with prior work). *This achieves*: Better detail preservation from the content image, measured in Table 1 and via our user study, and visualized in Fig 7.

2. A new Sobel-based feature-complexity guided discriminator loss for a better matched stylization learning signal. *This achieves*: Solving a long-standing issue - style halos present in prior NST. We are first to do so (Figs. 4,5).

3. Big Beautiful Style Transfer (BBST-4M) - a novel dataset of 4 million very high resolution images for the NST community. WikiArt and MSCoco (~80k images) typically used in NST, limit techniques to low resolution and focus on traditional fine-art styles. *This achieves*: High resolution output, and improved style diversity/quality measured in Sec. **5.3** through a user study comparing training on WikiArt+MSCoco vs. BBST-4M. Users prefer BBST-4M, especially for modern styles. We filter the style/content subsets of this dataset with the assistance of a novel dual-branch ViT [8] and ALADIN [28] model trained to predict how artistic an image is. We release both this stylistic prediction model, and the BBST-4M dataset (time-stamped).

2   Related Work

Great stylization quality has been achieved in literature by focusing on the aligning the statistics of features extracted by frozen pretrained networks in the final output image and the target style image [11,13,16,17]. However, as with classical computer vision techniques, these approaches are limited by human design, and are likely sub-optimal, in quality and runtime.
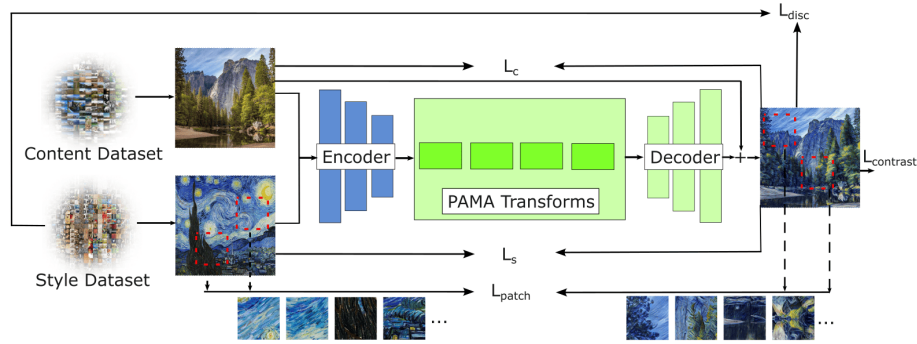
Fig. 2: Basic architecture diagram. The green modules represent the trainable parameters. For clarity, the discriminator modules and contrastive projection heads are not shown. The contrastive loss is computed 1) between the stylized output and the target style, and 2) between the stylized images only [3] - a group of contrastive losses anchoring on the same style regardless of content, and a group of contrastive losses anchoring on the same content regardless of style. We also leverage several common identity losses between the stylized image and the respective style/content images from the datasets - computed as a Gatys loss [11].

More recent works have started exploring the idea of shifting this design process to the actual model via transformations learned through attention mechanisms. SANet [22] introduced a style attentional module to perform style transfer in a feed-forward approach. Improving on local style information compared to Avatar-Net [33], they combine global and local style patterns extracted from different VGG layers, by normalizing the content/style feature maps, and computing attention between them via learnt 1x1 convolutions [37]. PAMA [20] introduces iterative alignment of the content with the target style across multiple attention modules, improving quality and reducing inconsistencies of style across similar areas in the content image.

ContraAST [3] expands SANet by injecting information about the entire distribution of training data via domain-level adversarial losses. This is implemented by a discriminator acting on the stylized image, aiming to classify these outputs as *fakes* and images from the style dataset as *reals*. ContraAST also introduces contrastive losses between the stylized images. One encourages several stylized images with the same target style to stylistically match. Another encourages several stylizations of the same source content image but different styles to share the same content. These losses shift the burden of designing feature spaces and comparing them from human design into the learning process.

CAST [40] improves on the contrastive loss from ContraAST by including ground truth images in the contrastive pairs, expanding the loss to compare the generated images to images in the dataset.

This work also leverages several important ideas from the style transfer literature. In image translation literature, Swapping Auto-encoders [23] introduce

a powerful discriminator based on patch co-occurrence. While it was originally used for photo-realistic style transfer, we find it equally powerful for artistic stylization. Another important concept is increasing the effective batch size used in contrastive loss; as it has been shown that the value of contrastive loss scales well with increasing numbers of negative examples [4]. However, in non-trivial models optimally high batch sizes are difficult to fit into VRAM. Logit accumulation, introduced in ALADIN [28] enables compatibility of the gradient accumulation technique to models trained using contrastive loss. This works by iteratively accumulating the model's output logits without model gradients, performing the large batch-wise contrastive loss, then iteratively propagating the sample-wise gradients back through the model in sub-batches, this time using gradients from re-forwarding the respective sub-batch images through the model. Other similar accumulation approaches have been used in works such as Listwise loss [25], and ArtFlow [2].

Recently, diffusion based generative models [9, 24, 26] have attracted much attention in the research community. These models exhibit impressive quality in novel image generation. Through extensions such as textual inversion [10] and Dreambooth [27] existing models can be finetuned to generate specific concepts or styles. SDEdit [21] offers a mechanism to alter an existing image, although without much finegrained control over content preservation. Prompt2Prompt [12] offers a mechanism to improve content preservation while editing, but has unpredictable failure cases and works best on generated images rather than real photos. Imagic [15], and Unitune [35], finetune the diffusion model for editing a specific image, making them extremely time and compute intensive. While these methods are adjacent to the field of style transfer, and show great promise, we are not aware of diffusion-based methods which simultaneously offers the content-preservation, exemplar based style control, and feed forward inference speed of NST methods.

3   BBST-4M Dataset

We compile a large scale dataset of content and style images, using images from Behance.net for the style subset, and images from Flickr for the content subset. We filter based on the criteria that images must be high resolution (at least 1024px on the smallest side), and they must be stylistic for the style subset, and not stylistic for the content subset.

To filter based on the stylistic properties of the image, we first build a model to predict if an image is stylistic. We use ViT [8] for a content feature branch, and ALADIN-ViT [29] for a style feature branch. We merge these two pre-trained branches with a small MLP for a final score. The model was trained iteratively, human-in-the-loop, on data from the StyleBabel dataset [29]. We manually annotated images as being stylistic or not, with assistance from the model to help guide the process, with iterative re-training of the model with each pass of further dataset expansion through annotation. Our stylistic prediction model achieves 96% accuracy in determining if an image is artistic or not.

Our criteria to judge if an image is stylistic or not is subjective, but our general rule was that if it made sense to use an image as a target style, it was deemed stylistic. Images such as ordinary photographs or interface screenshots would be labelled as negative examples; while images such as oil paintings or digital art would be positive examples.

We used this stylistic detection model to filter out stylistic images from the Flickr data, and filter out non-stylistic images from the Behance data. This is a similar post-processing step to the aesthetic predictor model used to curate LAION-5B [31] into LAION-aesthetic, except here we focus on how stylistic an image is, or is not. The largest source of content/style separation comes from the data sources - with content images from Flickr photographs, and style images from Behance artworks. The final dataset size is 2.2 million stylistic images from Behance, and 2 million content images from Flickr.

| Content | No colour adjustments | With colour adjustments | Style |

Fig. 3: Visualization of the content colour bleeding problem, in content priors use

4 Design decisions

Works in the hypernetworks literature [1, 30] find that reformulating a model's task as the modification of the target domain can be more effective than generating completely new values. In the hypernetworks, this is done by generating weight deltas, to modify existing trained weights of a target model. This idea of generating deltas can similarly be applied to other tasks, such as style transfer. In our work, we propose this by designing our model to generate RGB deltas, over the existing content prior image. Generating a completely new image is unnecessary, as we already know what the structure of the image should be. For neural stylization, we are only interested in modifying the existing content image. So far, content preservation has been an area of focus for balancing design decisions in previous literature, but we propose this approach to alleviate this problem, as the content image already has all the needed information. An ablation visualizing the benefit of our deltas-based approach over priors can be seen in Fig 7, and calculated in Table 1.

There are nevertheless some challenges to overcome with this approach. First, a balance must be found in deciding how much information from the content image should be propagated. A stylized image should preserve the content of the reference image, but not necessarily retain all the fine details. For example,

a stylized landscape image should rarely replicate details as fine as grass blades. We therefore first apply some light gaussian blurring on the content image, and apply bilateral filtering to further simplify the source image while preserving sharp edges. Finally, we first apply a weighting (0.5) to the content prior, to force the model to not rely purely on all the content prior's details (similar to how an artist traces over a faded version of the reference image). During inference, the blur strength can be adjusted/disabled, to allow slightly more or less original details to come through into the final image.

The second issue is colour information from the content image bleeding into the stylized image. We are only interested in the structure information, not the colours. There are several ways this issue can be mitigated, such as only using grayscale content priors, or using only priors for *structure* channels in a different colour space such as HSL or OKLAB. Still, we find the best approach to be to re-colour the content prior with the style image colours. We adjust the mean and covariance, as per the work in *Artistic Radiance Fields* [38]. Figure 3 shows a visualization of the colour bleed from the content image when a content image is used as a prior, and how our pre-recolouring pipeline remedies it.

4.1   Architecture and losses

Most inspired by ContraAST and PAMA, we first extract image features from the ground truth images using a pre-trained VGG [34] model. Next, we use an attention based mechanism to induce the style transfer in feature space. We use 4 PAMA [20] blocks, between VGG features of *style* and *content* images. Finally, we decode these features using a decoder module. We use a domain-level adversarial loss [3] to support realism in the generated images, and identity losses to ensure strong information propagation. During the stylization process, we use contrastive losses amongst the generated stylized images to ensure consistency between the style and content. We further perform contrastive losses between the stylized images and the ground truth style images [40]. We use two crops of ground truth style for this, to raise the number of comparisons made to the style image whose style we replicate.

Finally, we use a patch co-occurrence discriminator [23] to provide an additional learning signal for style, via a patch based comparison. This further helps with small details in textures such as paint strokes, highly visible when generating high resolution images. This improves local style information, in addition to the global style information already supported by the other losses. We use a custom implementation of a patch co-occurrence discriminator, where we perform a more principled sampling of patches via edge maps, to solve *style halos*.

4.2   Style Halos

An issue we commonly observe in literature is the appearance of *halos* in the generated style images. Visualized in Fig 5, these are outlines around objects which reduce the overall quality of the image. We notice that they mostly occur in low frequency areas adjacent to high frequency areas. The style halos
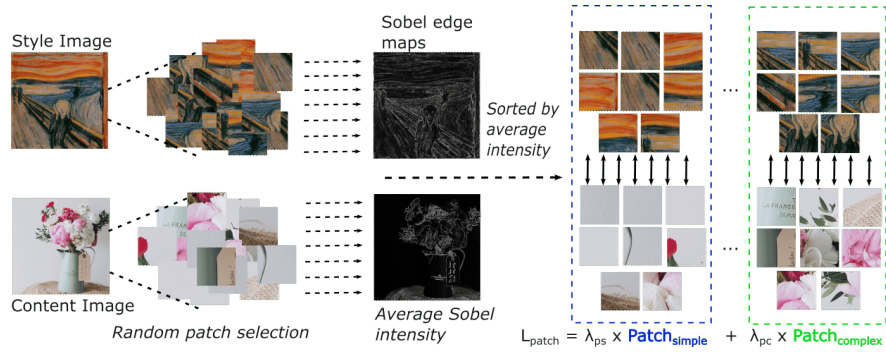
Fig. 4: A visualization of the patch co-occurrence loss, specifically showing the Sobel guided selection process (Sec **Style Halos**). Patches are randomly selected from both the stylized image, and the style image. Sobel edge maps of the content image and style image are used to compute average intensity scores for all patches, which are then sorted by this intensity score. Two patch co-occurrence losses are computed separately, for the simple patches, and the complex patches.
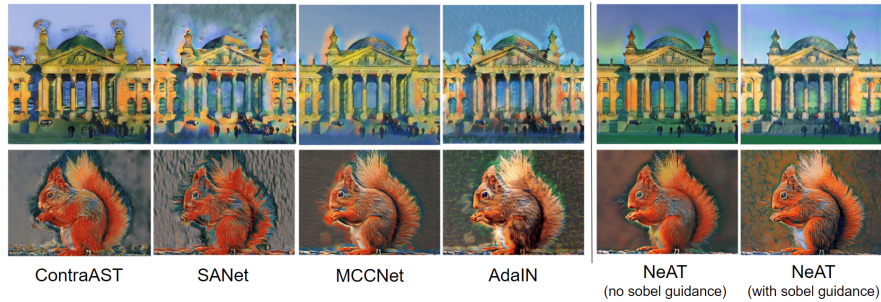

Fig. 5: Visualization of the style halos intermittently present in many models. Style halos can be seen in the top row mostly around the two towers, in the sky. In the bottom row, they can be seen around the edges of the squirrel.

themselves contain low frequency style textures, which are appropriate for a low frequency area. The problem, therefore, is not with the areas inside the halos, but rather in the remaining low frequency region, further away from the edge. While previous work can sufficiently stylize transitions from high-frequency to low-frequency regions, they fail in entirely low-frequency regions. When relative edge information is not present locally, there is no guidance for how complex the texture/style needs to be. As a result, style complexity increases in low-frequency areas further from the boundaries, creating these *halo*-like artifacts.

We propose a solution to this problem based on an informed selection of patches used in a patch co-occurrence discriminator, to provide a localized, non-global style signal. The vanilla discriminator selects 8 patches at random from

the style image, and 8 at random from the generated image. Informed by our observations regarding *style halos*, we improve this by splitting the loss into two separate patch co-occurrence discriminators, one acting on low frequency areas, and one acting on high frequency areas (Figure 4).

We use Sobel edge maps to determine regions' frequency intensities for randomly sampled patches from style and content images. We then sample these regions from the style and stylized images and sort them by average intensity respective to their Sobel versions. The high average intensity patches are used in the high-frequency discriminator, and the low average intensity patches are used in the low-frequency discriminator. By forcing the discriminators to focus on high/low complexity patches separately, we provide a local signal that low-frequency regions of the content image should remain lower frequency *relative* to the rest of the image when stylized. Thus, low-frequency regions far from boundaries are not stylized with overly complex texture for lack of a local relative signal, alleviating the halo artifacts. The two losses are weighted by λ_{ps} for the simple patches' loss and λ_{pc} for the complex patches' loss.

Eq 1 details $\mathcal{L}_s$, the style loss computed amongst style and stylized image features, where ϕ_i represents VGG-19 layer index, with μ and σ representing mean and standard deviation of extracted feature maps. I_s represents style image from the style dataset S, I_c represents content image from the content dataset C, and I_sc represents the final stylized image, after applying the generated RGB deltas over the original I_c.

$$\mathcal{L}_s := \sum_{i=1}^{L} \|\mu\left(\phi_i\left(I_{sc}\right)\right) - \mu\left(\phi_i\left(I_s\right)\right)\|_2 + \|\sigma\left(\phi_i\left(I_{sc}\right)\right) - \sigma\left(\phi_i\left(I_s\right)\right)\|_2 \quad (1)$$

Eq 2 represents the domain-level adversarial loss, as per [3], learning to discriminate between generated stylized images, and real artworks. Here, a discriminator $\mathcal{D}$ operates over the stylized image, following encoder E, transformation T, and decoder D modules.

$$\mathcal{L}_{adv} := \underset{I_s \sim S}{\mathbb{E}}\left[\log\left(\mathcal{D}\left(I_s\right)\right)\right] + \underset{I_c \sim C, I_s \sim S}{\mathbb{E}}\left[\log\left(1 - \mathcal{D}\left(D\left(T\left(E\left(I_c\right), E\left(I_s\right)\right)\right)\right)\right)\right] \quad (2)$$

Eq 3 details standard perceptual loss, and Eq 4 details an identity loss, to preserve the same image when the content and style images are the same.

$$\mathcal{L}_c := \left\|\phi_{\text{conv4\_2}}\left(I_{sc}\right) - \phi_{\text{conv4\_2}}\left(I_c\right)\right\|_2 \quad (3)$$

$$\text{L}_{\text{identity}} := \lambda_{\text{identity 1}}\left(\|I_{cc} - I_c\|_2 + \|I_{ss} - I_s\|_2\right) + \lambda_{\text{identity 2}}\sum_{i=1}^{L}\left(\|\phi_i\left(I_{cc}\right) - \phi_i\left(I_c\right)\|_2 + \|\phi_i\left(I_{ss}\right) - \phi_i\left(I_s\right)\|_2\right) (4)$$

Eqs 5 and 6 show contrastive losses as detailed in Sec 4.1, similar to [3] and [40], where h is a projection head l_s and l_c are style/content embedding extraction respectively, and τ is the temperature hyper-parameter.

$$\text{L}_{s-\text{contra}} := -\log\left(\frac{\exp\left(l_s(s_i c_j)^T l_s(s_i c_x)/\tau\right)}{\exp\left(l_s(s_i c_j)^T l_s(s_i c_x)/\tau\right) + \sum \exp\left(l_s(s_i c_j)^T l_s(s_m c_n)/\tau\right)}\right) (5)$$

$$\text{L}_{c-\text{contra}} := -\log\left(\frac{\exp\left(l_c(s_i c_j)^T l_c(s_y c_j)/\tau\right)}{\exp\left(l_c(s_i c_j)^T l_c(s_y c_j)/\tau\right) + \sum \exp\left(l_c(s_i c_j)^T l_c(s_m c_n)/\tau\right)}\right) (6)$$

The $\mathcal{L}_{\text{patch}}$ term defined in Eq 7 is our patch discriminator (D_{patch} loss, guided by Sobel Maps (SM).

$$\mathcal{L}_{\text{patch}} = \mathbb{E}_{I_s \sim S}[-\log(D_{\text{patch}}\,(\text{crop}(I_{sc}, SM_{sc}), \text{crops}(I_s, SM_s)))] \qquad (7)$$

The final objective is shown below, with SM, sc, s, S, D, $\mathcal{C}$, and I representing Sobel Maps, stylized image, style image, style dataset, discriminator, cropping function, and output image. We would like to stress that these loss terms have been explored in previous literature [3, 40]. Please refer to these published works for further ablations. We newly introduce the Sobel-guided patch co-occurrence discriminator loss, which we show in Figure 5 to solve the style halos artefacts.

$$\mathcal{L}_{\text{final}} := \lambda_1 \mathcal{L}_s + \lambda_2 \mathcal{L}_{\text{adv}} + \lambda_3 \mathcal{L}_c + \lambda_4 \mathcal{L}_{\text{identity}} + \lambda_5 \mathcal{L}_{\text{s-contra}} \dots$$

$$+\lambda_6 \mathcal{L}_{\text{c-contra}} + \lambda_7 \mathcal{L}_{\text{patch\_simple}} + \lambda_8 \mathcal{L}_{\text{patch\_complex}} \qquad (8)$$

$$\mathcal{L}_{\text{patch}} = \mathbb{E}_{I_s \sim S}[-\log(D_{\text{patch}}\,(\mathcal{C}(I_{sc}, SM_{sc}), \mathcal{C}(I_s, SM_s)))] \qquad (9)$$

5   Experiments

We can perform interpolation in the style space, by interpolating between the feature maps fed through the decoder. Fig 6 visualizes interpolation between the feature map of a content image, and the feature map of the same content image, stylized by the style image shown. For full interpolation, we additionally interpolate the content image prior due to our pre-processing pipeline. We use an alpha α value to signify the interpolation strength between these.
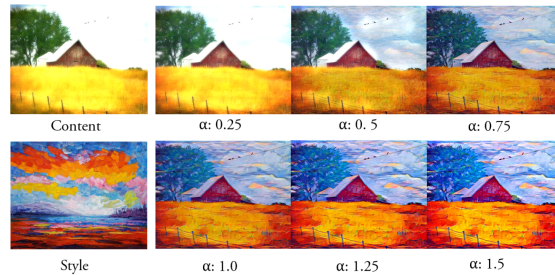

Fig. 6: Interpolation between the original image and its stylized variant, by interpolating the features input to the decoder.

Interestingly, we can *amplify* the stylization strength by increasing α (interpolation strength) beyond 1, thus further boosting the effect of the stylization. A couple such samples, denoted in the figure as $\alpha = 1.25$ and $\alpha = 1.5$ show how the

stylized image's style even more strongly resembles that of the style image. Yet, structural details in the image are still maintained - such as the fence posts and the birds on the wire. Similar to the content prior's blur parameters, changing the alpha value during inference can be used by an artist as a tool to control the stylization process further.
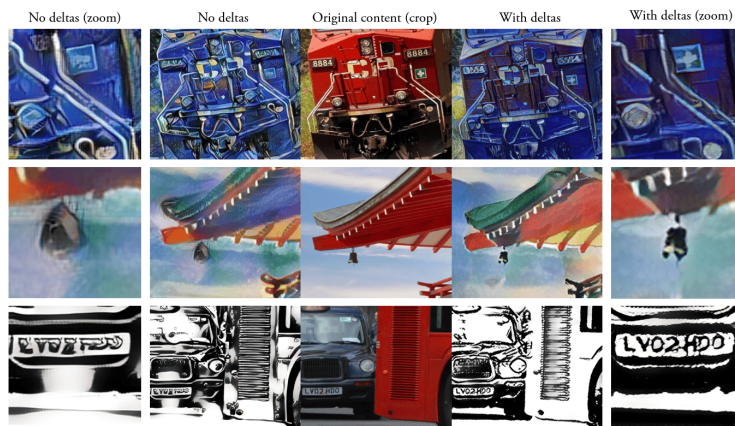

Fig. 7: Ablation showing detail gain from use of prior deltas. Quantified in Table 1.

5.1   Training and evaluation

The strengths of the content prior blurs can be varied, but for training, we settle on a balanced amount of 7px blurring, followed by bilateral filtering with filter size 25, and a sigma value of 100. During early tests, we experimented with unbalanced averaging of the losses from the two sobel guided patch co-occurrence discriminators. Based on the idea that high frequency information requires more work to learn correctly, we provide more of the loss budget to the high frequency patches, weighing them at 0.75 (λ_{pc}), versus 0.25 (λ_{ps}) for the low frequency patches. The training time for NeAT is roughly 3 days on an A100.

We form 750 test images from 15 content images and 50 style images sampled from publicly available NST literature test sets, unseen during training. We evaluate four aspects of our model's generated images compared to literature, in automated metrics: (1) colour consistency in regards to the style image, (2) content structure preservation in regards to the content image, (3) style consistency compared to the style image, (4) inference run-time of the model to run the style transfer, end-to-end.

We measure colour consistency using the Chamfer distance. We use a version of this normalized by the number of pixels in the image, such that the score is independent of the image resolution. We use LPIPS [39] to measure content preservation, and we calculate the average inference run-time, averaged over our

test set. To measure style consistency between the style and stylized images, we use the SIFID metric introduced in SinGAN [32]. This metric measures FID score, but only amongst individual images. SIFID is used in previous works such as Swapping Autoencoders [23] for measuring photorealistic style transfer - we use it for measuring artistic style transfer. Results are shown in Tables 1 and 2.

We additionally measure structure similarity using SSIM, and style similarity using Gram. Our method scores favourably against baselines. We note that $StyTr^2$ is limited to square aspect ratios unlike our method which operates at arbitrary resolution. We observe that the few baselines which score slightly better than our method on content similarity do so only at the cost of significantly worse style similarity. For example, S2WAT narrowly scores better on LPIPS but significantly worse on SIFID/Gram values, implying low quality stylization.

We further include results from our model where the prior image blurring is turned off. This is a control vector that an artist can use for adjusting how many details from the content image should come through into the final, stylized image. We also expose these user controls, to control a trade-off between stronger style or content similarity. Turning the blurring off can further improve the preservation of details, measured in the table via LPIPS.

Finally, we undertake a user study to measure stylization quality and consistency with the target image.

5.2   User study

We conduct a user study on Amazon Mechanical Turk (AMT), to measure real life preference between our model, and baseline models. We show users the content and style image in the middle of the screen, and on either side, a stylized sample using either our model, or a baseline comparison model. The whole sample is shown at the top, and a centre crop shown below, to highlight smaller details in the image - critical when working with high resolution images. The left/right columns for either models is randomly shuffled. We test our model against the baselines, across three different experiments: *Overall*, *Content*, and *Style*. All three use NeAT trained on WikiArt/MSCoco, for a fair comparison against literature.

The *Content* experiment asks users to evaluate the semantic structure and detail preservation between the content and stylized images. The *Style* experiment asks users to assess the similarity of artistic style between the output and target style, irrespective of content. The *Overall* experiment asks users to evaluate both criteria simultaneously to gauge the overall combined quality of the style transfer.

We randomly sub-sample 200 stylized pairs from our test set for each experiment. We ask 5 different workers to pick their preference for each pair, and we use consensus voting to determine the overall decision for each pair. In total, 74 total workers contributed to the user study. We collect the results in Table 3. A value higher than 50% indicates workers preferred our method - we beat all baselines, on each of the 3 separate experiments. We underline the closest baseline to NeAT. The star (*) symbol represents experiments where images were stylized at 512x512 resolution (including for our method, for fairness), either due to

Model	LPIPS ↓	SIFID ↓	Chamfer ↓	SSIM ↑	Gram ↓ (×10³)
ContraAST [3]	0.643	2.519	51.696	0.419	0.807
PAMA [20]	0.656	2.301	271.630	0.408	1.480
SANet [22]	0.684	2.837	58.297	0.341	0.391
CAST [40]	0.579	1.729	214.075	0.491	0.315
AdaAttn [19]	0.598	3.157	196.391	0.522	1.670
MCCNet [6]	0.646	2.689	393.348	0.467	0.299
ArtFlow [2]	0.346	4.646	656.336	0.673	0.393
NNST [16]	0.626	3.784	2617.187	0.393	0.628
AdaIN [13]	0.666	2.235	204.790	0.330	0.194
NeAT (Ours - WikiArt+MSCoco)	0.655	1.171	78.780	0.464	0.149
⟶ *no prior blurring*	0.596	2.343	96.541	0.513	0.149
NeAT (Ours - BBST-4M)	0.687	1.053	41.742	0.507	0.115
⟶ *no prior blurring*	0.635	1.306	54.227	0.491	0.115
⟶ *no prior deltas*	0.697	1.384	21.505	0.409	0.461
$StyTr^2$ [7] (512x512)	*0.636*	*1.07*	*20.423*	*0.453*	*0.075*
S2WAT [36] (512x512)	*0.585*	*2.884*	*27.294*	*0.564*	*1.330*
NeAT (Ours - 512x512 WikiArt+MSCoco)	*0.587*	*1.062*	*16.278*	*0.482*	*0.058*

Table 1: Quantitative evaluation metrics. We include results with our method for 512px images, for fair comparisons to baselines limited to 512x512 resolution ($StyTr^2$, S2WAT). Results computed over 512x512 images are presented in *italics*.

Model	256x256 ↓	512x512 ↓	1920x1080 ↓
ContraAST [3]	0.030	0.043	0.150
PAMA [20]	0.035	0.051	0.189
SANet [22]	0.030	0.043	0.163
CAST [40]	0.050	0.055	0.184
AdaAttn [19]	0.053	0.068	0.221
MCCNet [6]	0.040	0.054	0.178
ArtFlow [2]	0.153	0.247	1.178
NNST [16]	17.647	17.266	65.367
AdaIN [13]	0.037	0.0517	0.185
$StyTr^2$ [7]	0.023	0.115	-
S2WAT [36]	0.146	0.146	-
NeAT (Ours)	0.049	0.068	0.226
⟶ *no prior blurring*	0.047	0.058	0.196

Table 2: Timing (seconds/image) of all methods.

Model	Overall	Content	Style
ContraAST [3]	62.5%	64.0%	66.5%
PAMA [20]	63.5%	71.0%	54.5%
SANet [22]	78.5%	78.0%	64.0%
CAST [40]	60.0%	57.0%	69.0%
AdaAttn [19]	52.5%	54.5%	56.0%
MCCNet [6]	58.5%	55.5%	66.0%
ArtFlow [2]	60.0%	51.5%	72.0%
NNST [16]	84.5%	83.0%	81.5%
AdaIN [13]	72.5%	74.5%	70.0%
$StyTr^2$ [7]*	57.8%	54.4%	57.1%
S2WAT [36]*	62.2%	61.7%	61.7%
Average	64.77%	64.10%	65.3%

Table 3: Preference of our method on AMT, compared to baselines (61% agreement).

aspect ratio limitations, or prohibitively high VRAM requirements. Finally, we qualitatively visualize random stylization results across NeAT and the baselines in Figure 8.

5.3   Generalization

We use BBST-4M, our newly collected dataset, to improve style transfer quality, especially across styles not covered in typical style datasets such as WikiArt. Such datasets are limited in their scope to mainly fine art images. In contrast, the collection process for the style subset of BBST-4M covered a much larger variety of styles, as typically found on *Behance.net*. We train our model using BBST-4M, and we evaluate via two further user studies how the quality of the style transfer changes. As before, we first test the BBST-4M variant with the test set. Second, we separately evaluate how well the model generalizes to non fine art images, by evaluating using an *out of distribution* set of style images, respective to WikiArt. We build this small set of images from a separate data scrape from Behance, by selecting images that are, on average, far away in ALADIN style space relative to the images in the WikiArt dataset. These are mainly digital art, vector art, and other such images not typically seen in such fine art collections.

As before, users on AMT evaluate stylized pairs of the same content and style image for two variants. The user studies indicate a preference for the BBST-4M variant at 56.0% over the WikiArt variant. Similarly, there is a preference of 60.0% when using the *out of distribution* style test set. This more significant gap indicates that the model has stronger generalization than the WikiArt model.

6   Limitations and future work

We use simple adjustments to the mean and covariance for the pre-recoloring of the content image prior. While effective, this process could be better and can lead to occasional inconsistencies and artifacts in the image colors. We visualize this example in Figure 9, where imperfections in the color adjustments can lead to lost details.

Additional future work can involve a stronger focus on architectural changes, which was not a focus in this work. Finally, though not an issue with our method specifically, different resolutions of the content image lead to noticeably different stylized outputs. An artist can remedy this by attempting the process with multiple resolutions. However, this is an additional parameter that needs to be explored during inference. Though it could be considered a positive, as it allows control over the level of style complexity in the edited image.

7   Conclusion

We presented NeAT, a new style transfer model, achieving state-of-the-art performance through novel technical design decisions. We stylize images through
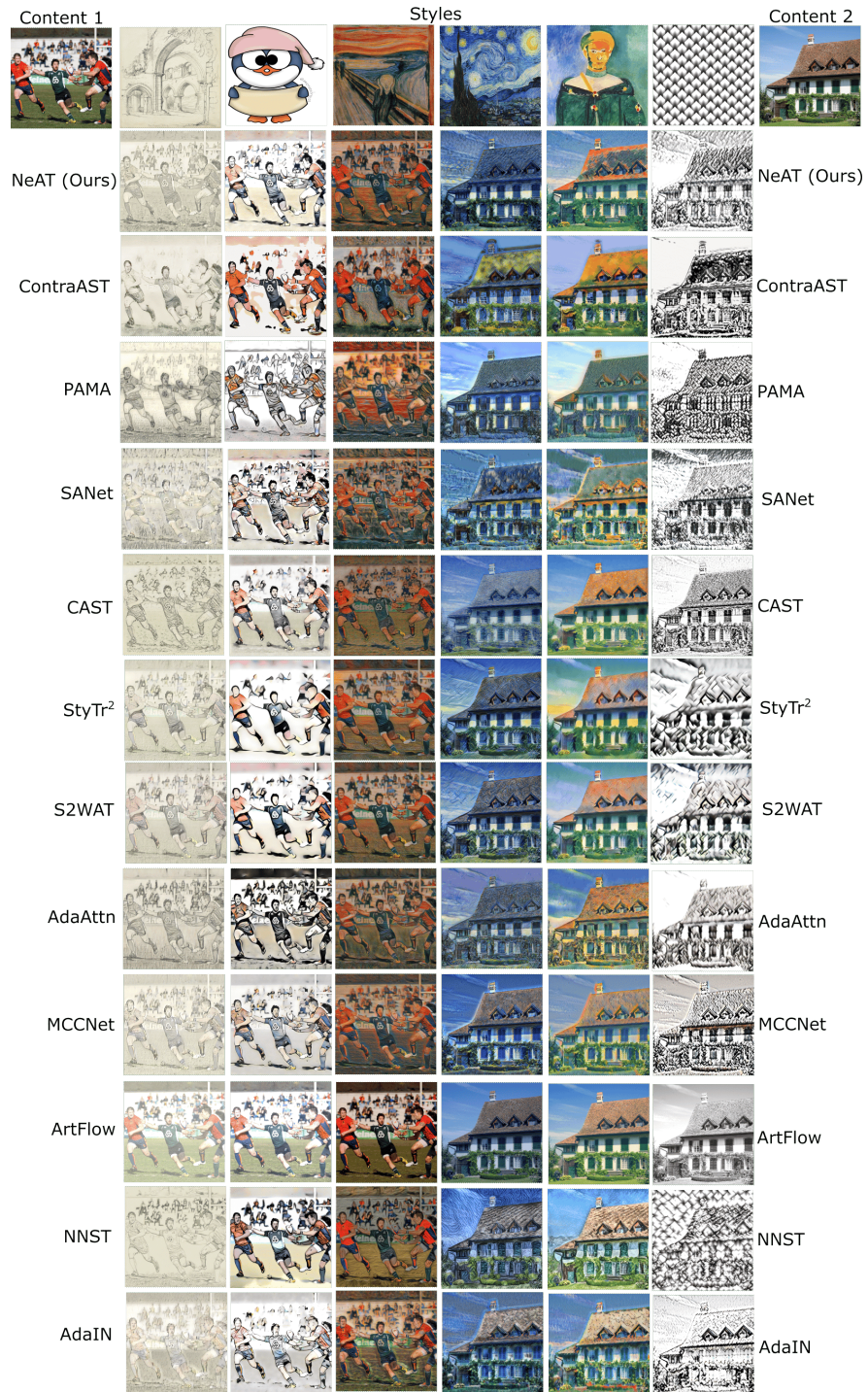
Fig. 8: NST using NeAT, and baselines. Please zoom for more details.

Fig. 9: Example of limitation on color adjustments. The squirrel's tail is overexposed, and details are lost.

editing rather than regeneration, leading to better content preservation and improved artistic style transfer. We provide a solution to style halos which sometimes appear in stylized images, through careful guidance during patch selection for a patch co-occurrence loss. We use Sobel edge maps to sort and separate high frequency and low frequency areas, thereby better matching appropriate areas of the style image to match regions in the content image for the stylization. We also release BBST-4M, the first large scale, highly diverse, high resolution NST dataset, cleaned with a novel stylistic prediction model that we also release.

References

1. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.H.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing (2021)
2. An, J., Huang, S., Song, Y., Dou, D., Liu, W., Luo, J.: Artflow: Unbiased image style transfer via reversible neural flows. CoRR **abs/2103.16877** (2021), `https://arxiv.org/abs/2103.16877`
3. Chen, H., Zhao, L., Wang, Z., Ming, Z.H., Zuo, Z., Li, A., Xing, W., Lu, D.: Artistic style transfer with internal-external learning and contrastive learning. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), `https://openreview.net/forum?id=hm0i-cunzGW`
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020)
5. Collomosse, J., Bui, T., Wilber, M., Fang, C., Jin, H.: Sketching with style: Visual search with sketches and aesthetic context. In: Proc. ICCV (2017)
6. Deng, Y., Tang, F., Dong, W., Huang, H., Ma, C., Xu, C.: Arbitrary video style transfer via multi-channel correlation. CoRR **abs/2009.08003** (2020), `https://arxiv.org/abs/2009.08003`
7. Deng, Y., Tang, F., Dong, W., Ma, C., Pan, X., Wang, L., Xu, C.: Stytr^2: Image style transfer with transformers (2022)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR **abs/2010.11929** (2020), `https://arxiv.org/abs/2010.11929`

9. Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., Taigman, Y.: Make-a-scene: Scene-based text-to-image generation with human priors (2022). `https://doi.org/10.48550/ARXIV.2203.13131`, `https://arxiv.org/abs/2203.13131`

10. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion (2022). `https://doi.org/10.48550/ARXIV.2208.01618`, `https://arxiv.org/abs/2208.01618`

11. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proc. CVPR. pp. 2414–2423 (2016)

12. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control (2022). `https://doi.org/10.48550/ARXIV.2208.01626`, `https://arxiv.org/abs/2208.01626`

13. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. CoRR **abs/1703.06868** (2017), `http://arxiv.org/abs/1703.06868`

14. Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., Winnemoller, H.: Recognizing image style. In: Proc. BMVC (2014)

15. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models (2022). `https://doi.org/10.48550/ARXIV.2210.09276`, `https://arxiv.org/abs/2210.09276`

16. Kolkin, N., Kucera, M., Paris, S., Sykora, D., Shechtman, E., Shakhnarovich, G.: Neural neighbor style transfer (2022). `https://doi.org/10.48550/ARXIV.2203.13215`, `https://arxiv.org/abs/2203.13215`

17. Kolkin, N.I., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. CoRR **abs/1904.12785** (2019), `http://arxiv.org/abs/1904.12785`

18. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.: Universal style transfer via feature transforms. CoRR **abs/1705.08086** (2017), `http://arxiv.org/abs/1705.08086`

19. Liu, S., Lin, T., He, D., Li, F., Wang, M., Li, X., Sun, Z., Li, Q., Ding, E.: Adaattn: Revisit attention mechanism in arbitrary neural style transfer. CoRR **abs/2108.03647** (2021), `https://arxiv.org/abs/2108.03647`

20. Luo, X., Han, Z., Yang, L., Zhang, L.: Consistent style transfer. CoRR **abs/2201.02233** (2022), `https://arxiv.org/abs/2201.02233`

21. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations (2021). `https://doi.org/10.48550/ARXIV.2108.01073`, `https://arxiv.org/abs/2108.01073`

22. Park, D.Y., Lee, K.H.: Arbitrary style transfer with style-attentional networks. CoRR **abs/1812.02342** (2018), `http://arxiv.org/abs/1812.02342`

23. Park, T., Zhu, J.Y., Wang, O., Lu, J., Shechtman, E., Efros, A.A., Zhang, R.: Swapping autoencoder for deep image manipulation. In: Advances in Neural Information Processing Systems (2020)

24. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents (2022). `https://doi.org/10.48550/ARXIV.2204.06125`, `https://arxiv.org/abs/2204.06125`

25. Revaud, J., Almazán, J., de Rezende, R.S., de Souza, C.R.: Learning with average precision: Training image retrieval with a listwise loss. 2019 IEEE/CVF ICCV pp. 5106–5115 (2019)

26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021). `https://doi.org/10.48550/ARXIV.2112.10752`, `https://arxiv.org/abs/2112.10752`

27. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation (2022). `https://doi.org/10.48550/ARXIV.2208.12242`, `https://arxiv.org/abs/2208.12242`

28. Ruta, D., Motiian, S., Faieta, B., Lin, Z., Jin, H., Filipkowski, A., Gilbert, A., Collomosse, J.: Aladin: All layer adaptive instance normalization for fine-grained style similarity. arXiv preprint arXiv:2103.09776 (2021)

29. Ruta, D., Gilbert, A., Aggarwal, P., Marri, N., Kale, A., Briggs, J., Speed, C., Jin, H., Faieta, B., Filipkowski, A., Lin, Z., Collomosse, J.: Stylebabel: Artistic style tagging and captioning (2022). `https://doi.org/10.48550/ARXIV.2203.05321`, `https://arxiv.org/abs/2203.05321`

30. Ruta, D., Gilbert, A., Motiian, S., Faieta, B., Lin, Z., Collomosse, J.: Hypernst: Hyper-networks for neural style transfer (2022). `https://doi.org/10.48550/ARXIV.2208.04807`, `https://arxiv.org/abs/2208.04807`

31. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: Laion-5b: An open large-scale dataset for training next generation image-text models (2022)

32. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. CoRR **abs/1905.01164** (2019), `http://arxiv.org/abs/1905.01164`

33. Sheng, L., Lin, Z., Shao, J., Wang, X.: Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on. pp. 1–9 (2018)

34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)

35. Valevski, D., Kalman, M., Matias, Y., Leviathan, Y.: Unitune: Text-driven image editing by fine tuning an image generation model on a single image (2022). `https://doi.org/10.48550/ARXIV.2210.09477`, `https://arxiv.org/abs/2210.09477`

36. Zhang, C., Yang, J., Wang, L., Dai, Z.: S2wat: Image style transfer via hierarchical vision transformer using strips window attention (2022)

37. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks (2018). `https://doi.org/10.48550/ARXIV.1805.08318`, `https://arxiv.org/abs/1805.08318`

38. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields (2022). `https://doi.org/10.48550/ARXIV.2206.06360`, `https://arxiv.org/abs/2206.06360`

39. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)

40. Zhang, Y., Tang, F., Dong, W., Huang, H., Ma, C., Lee, T.Y., Xu, C.: Domain enhanced arbitrary image style transfer via contrastive learning. In: ACM SIGGRAPH (2022)