ALADIN: All Layer Adaptive Instance Normalization for Fine-grained Style Similarity

Dan Ruta[1] Saeid Motiian[2] Baldo Faieta[2] Zhe Lin[2] Hailin Jin[2] Alex Filipkowski[2]

Andrew Gilbert[1] John Collomosse[1,2] [1]CVSSP, University of Surrey [2]Adobe Research

Figure 1. ALADIN enables style-based visual search, by learning a fine-grained embedding for artistic style similarity. Left: Three queries (blue) and their results *i.e.* nearest neighbours in the ALADIN search embedding. Each result set exhibits fine-grained style coherence; a consistent style of sketching. Right (red box): a search in a prior style search embedding [6] returns results with only coarse-grained style coherence; all are different variants of watercolor style.

Abstract

We present ALADIN (All Layer AdaIN); a novel architecture for searching images based on the similarity of their artistic style. Representation learning is critical to visual search, where distance in the learned search embedding reflects image similarity. Learning an embedding that discriminates fine-grained variations in style is hard, due to the difficulty of defining and labelling style. ALADIN takes a weakly supervised approach to learning a representation for fine-grained style similarity of digital artworks, leveraging BAM-FG, a novel large-scale dataset of user generated content groupings gathered from the web. ALADIN sets a new state of the art accuracy for style-based visual search over both coarse labelled style data (BAM) and BAM-FG; a new 2.62 million image dataset of 310,000 fine-grained style groupings also contributed by this work.

1. Introduction

Digital artwork spans a broad range of content depicted in diverse visual styles. Learning a representation suitable for searching artwork based on visual style is an open challenge, particularly when discriminating between subtle, *fine-grained* [39, 37, 25] variations in style. This is due to the difficulties of both (i) defining a suitable fine-grained ontology to label styles and (ii) the expert annotation task. Research to date has therefore focused upon coarse-grain

discrimination of a limited number of styles [17, 6].

Artistic style is the distinctive appearance of an artwork; *i.e.* how an artist has depicted their subject matter [8]. Style may be characterized, non-exhaustively, by visual attributes such as the texture, strokes, media, shading, or layout of an artwork; the challenge of identifying a complete list of attributes is long-standing and unsolved [7]. Our core contribution is to learn *fine-grained* artistic style similarity (Fig. 1) and do so via a weakly supervised approach that does not rely upon explicit labelling of style or style attributes in images. Our technical contributions are three-fold:

1. ALADIN Fine-grained Style Embedding. We propose ALADIN; a novel architecture to learn a search embedding for image style. ALADIN is an encoder-decoder (E-D) network, that pools Adaptive Instance Normalization (AdaIN) statistics across its style encoder layers to learn a discriminative search embedding capable of both discriminating subtle fine-grained of style (*e.g.* variations in sketching style) as well as coarse-grained styles (*e.g.* sketch, watercolor, etc.). Image stylization networks [14, 10, 15] have previously used AdaIN for style transfer, but these perform poorly for measuring similarlity (c.f. subsec 5.2). For the first time, ALADIN explicitly disentangles content and style within an E-D network to show how AdaIN may be harnessed for fine-grained style search.

2. Behance Artistic Media Fine-Grained (BAM-FG) dataset. We contribute a new 2.62 million image dataset of artwork within 310K fine-grained style groupings, gath-

ered from a creative portfolio website (Behance.net). Digital artists publish to Behance.net in micro-collections ('projects') comprising images of a related visual theme. On the assumption that image co-occurrence within these groups implies a weak cue for style similarity, we sample millions of such co-occurrences. Further, we partition and clean this noisy co-occurrence data via a large-scale crowd annotation task in which distinct style-coherent sub-groups of images within projects are identified with high consensus (yielding 1.62 million images and 135K groups).

3. Weakly supervised learning of fine-grained style. We present the first study into representation learning for *fine-grained* artistic style similarity, taking a weakly supervised approach. Prior style-based visual search learns only *coarse-grain style* discrimination directly from *explicitly labelled* data (*e.g.* via proxy classification tasks [17] or deep metric learning [6]). We train ALADIN using supervised contrastive learning [4, 19] to achieve a state of the art performance at both coarse and fine-grained style search without requiring any explicit coarse of fine-grained categorization of image style.

Note that we distinguish between noisy and weak supervision. The supervision is weak because there is no fine-grained style ontology to label images explicitly; instead, a weak proxy via implicit project groupings is the basis for learning. These groupings may be noisy or be cleaned up via crowd annotation, but supervision remains weak, as the data is not explicitly labelled to fine-grained styles. We show via objective and subjective user trials that raw project groupings are sufficient to train a state of the art model for fine-grained style similarity. Our cleaned data is used both for evaluating and enhancing fine-grained style discrimination of ALADIN.

2. Related Work

Visual style has been researched primarily from the perspectives of synthesis (*e.g.* style transfer) and classification. Early style transfer work learned visual analogies from photo-artwork pairs [13, 20]. More recently, deep representations enabled stylization from unpaired data [9]. Notably, Gatys *et al.* computed the Grammian across layers of a pre-trained model (*e.g.* VGG-19 [31]) to abstract content from style in diverse imagery; a representation that has been exploited for texture description [22], style-coherent in-painting [11], and fast neural style transfer (NST) via style-specific encoder-decoder networks [33, 16]. Extensions to multi-scale [36] and video [28] NST were later presented. Variants of this representation (*e.g.* cosine-of-Grammian) have been explored to represent style [5], and the concepts of image analogy and NST combined for style transfer [21, 35]. Instance normalization was proposed to enhance the quality of style transfer [34], and building upon this, mean-variance statistics (AdaIN) between content and style features [14, 10]. Recently unsupervised style transfer was enabled via MUNIT [15], which disentangled content and style via AdaIN [14]; learning a latent code for style

without labelled data within an encoder-decoder (E-D) architecture. A similar architecture was later used to swap style between images [26]. These approaches can be considered to embed the notion of style within architectural design choices. We explore the complementary problem of learned search representations via E-D models.

By contrast, trained representations for style learn disentangled embeddings for content and style explicitly via deep metric learning, supervised using triplets [6]. Their work leverages an extensive public dataset of coarsely labelled digital artwork (Behance Artistic Media – BAM [38]). Smaller collections of labelled data have been used to supervise the classification of style [41, 17], or product designs [2] and even painters [3] and artistic genres [29]. All these approaches use direct supervision of coarse-grain class labels on style, *e.g.* of fine-art [32]. A related field models image aesthetics using votes on social media [23], and recently the link between style and emotion is explored [1, 24]. Generative adversarial networks (GAN) such as cycle-consistent GAN [40] have been trained to map images from one domain to another, including between styles, and require labelled sets of (unpaired) images. Recently Style-GAN [18] explored the injection of a learned style-code at multiple stages in a convolutional encoder. Our work tackles style representation learning with weak labels without requiring explicit class supervision while focusing (being able to) discriminate fine-grained styles (Fig. 1).

3. Learning Fine-Grained Style Similarity

Our goal is to learn a fine-grained representation of style from a weak proxy (group co-membership) rather than direct supervision under labels from a style ontology.

3.1. ALADIN Architecture

We propose ALADIN; All-Layer ADaptive Instance Normalization designed for content and style disentanglement using adaptive instance normalization (AdaIN eq. 1) [14, 10] in an encoder-decoder (E-D) network (Fig. 2). AdaIN has been applied to neural style transfer (NST), where content is disentangled from style to enable modification of the style code [15, 26] before recombination of the two. However, style codes for these models perform very poorly when used as a search embedding (c.f. subsec. 5.2). ALADIN implements a disentangled E-D network design but extracts the mean and variation (as per AdaIN) values of activations from several style encoder layers in the latent code ('style embedding'). By training both for high reconstruction fidelity and with a contrastive approach to encourage metric properties in this latent space, we show ALADIN can learn a *search embedding* suitable for fine-grained style search.

ALADIN comprises dual encoder branches: (i) content encoder; (ii) style encoder. The content encoder uses 4 convolutional layers to downsample image features into a series of semantically-focused feature maps; instance normalization is applied to each layer. The style encoder branch com-
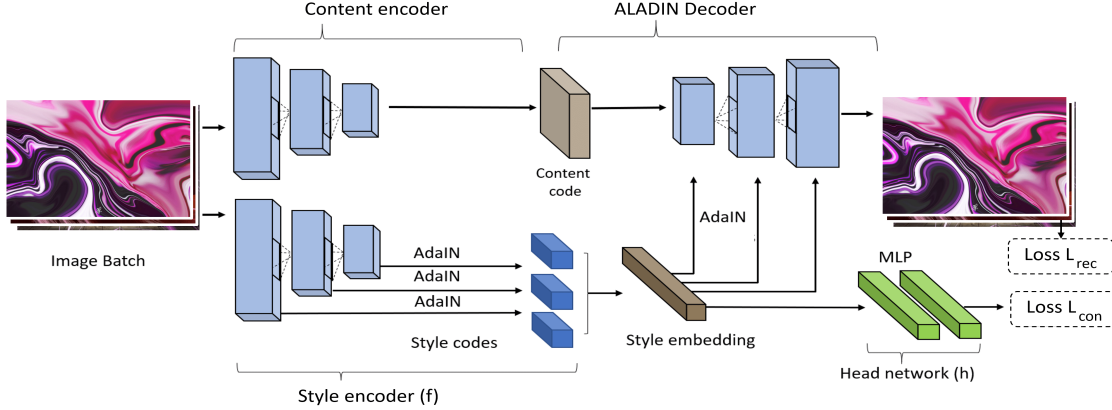
Figure 2. Proposed ALADIN architecture for learning a fine-grained style embedding. ALADIN uses a multiple stage encoder where AdaIN values are aggregated from each encoder layer and passed to the corresponding decoder stages. A concatenation of AdaIN features from encoder layers on the style branch is trained via a dual reconstruction (L_{rec}) and constrastive loss (L_{con}) under weak supervision from project group co-membership. The style encode/decoder backbone may take the form of several convolutional layers (ALADIN-S) or VGG-16 backbone (ALADIN-L).

prises three convolutional layers, comprising of 64, 128, 256 filters, respectively. We extract the style information using the AdaIN mean and variance statistics of the feature maps, rather than a fully connected (FC) layer, as this encoder is used only for style extraction. The style code is composed of twice as many values as filters in the encoder, with a mean and variance of each output feature map. Fig. 2 shows the ALADIN backbone built to include adaptive instance normalization (AdaIN) on the style encoder stages and the aggregation of activations ('style codes') from multiple encoder stages via concatenation to produce an 896-D search embedding.

The decoder mirrors the encoder shape, such that the style code can be split back into the same sized segments and applied to decoder filters. Both this mirroring and the multi-layer encoding of AdaIN differ from stylization E-D networks (*e.g.* MUNIT [15]) and enable a more effective search embedding to be learned from the resulting style codes. In the decoder stylization stage, pairs of [mean $\mu(.)$, variance $\sigma^2(.)$] values from style encoder layer activations (*e.g.* x) are applied to activations from a mirrored layer in the ALADIN decoder (*e.g.* y), following eq. 1.

$$AdaIN(x,y) = \sigma^2(y)\left(\frac{x - \mu(x)}{\sigma^2(x)}\right) + \mu(y) \qquad (1)$$

For both ALADIN-S and ALADIN-L variants we apply a multi-layer perceptron (MLP) with a hidden layer of size 512 and L_2 normalized output vector 128-D to the encoded embedding feature $f(.)$. We refer to this as projection network as $h(f(.))$ and later explore the efficacy of computing the loss on this projection embedding rather than the learned style embedding (c.f. Tbl. 2).

3.2. Training with Implicit Project Groups

We train ALADIN using a variant of supervised contrastive learning adapted with logit accumulation (sec. 3.2.1) to enable larger batch sizes. Training is weakly supervised, eschewing explicit style labels for the similarity in style implied by the co-presence of images in projects (groups) sampled from Behance.net creative portfolios (c.f. subsec 4).

Classic approaches to deep metric learning for search exploit pair-wise comparison (*e.g.* via triplet loss) to encourage correlation between visual similarity and embedding proximity. For a given training set T, a project group $G \subset T$ is selected at random, and an image a picked at random from within G as the 'anchor'. The remaining images within that group form positive examples $G^+ = G \setminus \{a\}$, and an equal number of negative samples are selected from other project groups $G^- \subset T \setminus G^+$. For a given minibatch B triplets (a, p, n) are formed where $p \in G^+$ and $n \in G^-$, and backpropagation applied to minimize $\sum_{(a,p,n)\in B}[\epsilon + |f(a) - f(p)|_2 - |f(n) - f(p)|_2]_+$, where $|.|_2$ denotes the L_2 loss and ϵ a small margin.

Recently, contrastive learning has shown improved performance using larger batch sizes. We form a minibatch B by sampling pairs of images $\{a, p\} \in G_i$ for $i = [1, N]$ groups (we use $N = 1024$ groups). Thus a batch comprises $2N$ images, $B = \{b_1, b_2, ..., b_{2N}\}$, where b_{2i} and b_{2i-1} come from the same group. For a given image b_i we therefore have a positive group $G_i^+ = \{b_p\}$ and a negative group $G_i^- = B \setminus \{b_i \, b_p\}$ for which the loss is:

$$L_{\text{con}}(B) = \sum_{i=1}^{2N} L(i) \;, \text{where}$$

$$L(i) = -\log \sum_{p \in G_i^+} \frac{\exp(f(i) \cdot f(p)/\tau)}{\sum_{n \in G_i^-} \exp(f(i) \cdot f(n)/\tau)} \qquad (2)$$

where $\tau > 0$ is a temperature parameter as in the self-supervised SimCLR [4].

ALADIN is an encoder-decoder network, so we employ a dual loss comprising also a reconstruction loss term L_{rec}:

$$L_{\text{rec}}(B) = \sum_{b \in B} |f(b) - b|. \qquad (3)$$

$$L_{\text{total}}(B) = L_{\text{con}}(B) + \lambda L_{\text{rec}}(B), \qquad (4)$$

where $|.|$ denotes the L_1 loss and we weight the reconstruction loss $\lambda = 10^{-2}$.

3.2.1 Logit Accumulation

Training ALADIN with large batches $|B| = 1024$ is impractical on contemporary GPUs (without resorting to federated or distributed compute) due to the extensive VRAM requirements. To combat this issue, we propose a *logit accumulation* strategy. The large batch size is first split into several smaller chunks (*gradient* batch size) for which the model in inference mode generates logits. Once the *target* batch size of 1024 is reached, these are concatenated and used for computing the contrastive loss. Backpropagation is carried out to compute gradients, stopping and retaining them at the logits level. The original chunks are next passed through the model again one by one, storing gradients in the model. The logit gradients corresponding to samples in a given chunk are backpropagated through the model before finally applying the gradients to the weights. When all chunks have been re-forwarded through the model the batch ends (see sup-mat. for visualizations of this process). With this technique, a single GPU with 12GB VRAM can fit the required batch size of 1024. Multiple GPUs could, in theory, handle larger batches as the bottle-beck becomes storing the MLP head gradients in VRAM, which may be parallelized across GPUs.

4. BAM-FG: Behance Artistic Media Fine-Grained dataset

We propose BAM-FG, a novel dataset of 2.62 million digital artworks in 310K project groups sampled from Behance.net – a creative portfolio website. We assume that image co-occurrence within these groups implies a weak cue for style similarity, and it is this signal we harness to train our model. We split BAM-FG into two distinct partitions:

BAM-FG-Raw. The raw, 'noisy' data where images are grouped as on Behance; we refer to this dataset as BAM-FG-Raw comprising 1M images in 175K projects. We train our model with this data.

BAM-FG-C_N. Data that has been 'cleaned' via a large-scale crowd-sourcing exercise so that image groups are known (rather than assumed) to be style consistent. This dataset initially comprises 1.62M images and 135K groups. We derive image groupings from this data to varying levels of confidence (participant consensus), written as $C_N =$

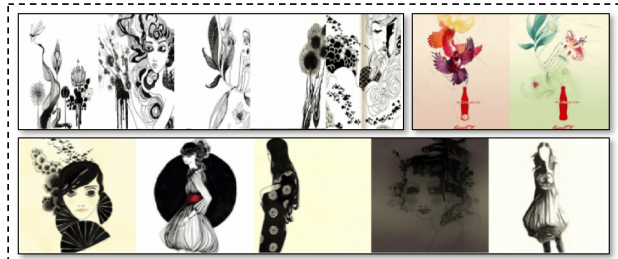| Consensus C_N	Groups	Samples	Avg samples/group
Reference	135,000	1,620,000	12.00
1	114,907	1,328,167	11.56
2	116,453	1,149,159	9.87
3	113,906	854,316	7.50
4	86,330	483,602	5.60
5	37,756	160,508	4.25

Figure 3. Top: BAM-FG-C_N dataset statistics, cleaned at each worker consensus level $C_N = [1, 5]$. Bottom: Three sub-groupings within a single raw project group, identified at consensus level $C_N = 3$

$1, ..., 5$. Fig. 3 (top) describes the number of images (samples) in BAM-FG at each consensus level, where $C_N = 5$ returns the highest data quality but lowest data volume. This data is used to evaluate ALADIN and fine-tune its performance (train/test split details are in subsec. 5.1).

4.1. Cleaning Style Groups (BAM-FG-C_N)

We constructed BAM-FG-C_N by manually curating the 135K project groups into sub-groups of coherent style. Annotation was crowd-sourced via Amazon Mechanical Turk (AMT) using 1073 workers. Workers were presented with images from a project and invited to tag any number of images (including zero) sharing the same visual style (creating a group). If there were multiple styles present, they picked the largest group. We sent each annotation task to 5 workers. The consensus was determined using a graph-based vote pooling method in which edges coded by an affinity matrix $A_{i,j}$ reflecting the number of times both images $\{i, j\}$ were simultaneously selected within an annotated cluster/moodboard. Edge counts are incremented each time an image pair is co-indicated. Thresholding $A_{i,j}$ at a given consensus level $C_N = [1, 5]$ partitions the group into sub-groups This yields groups of images at different strengths of style similarity, measured by the number of workers who grouped them together. Fig. 3 (bottom) shows an example of three sub-groups formed at $C_N = 3$ from a single raw group. The supplementary material contains a task example and a visualization of the graph-vote algorithm.

5. Experiments and Discussion

We evaluate the performance of ALADIN for both coarse and fine-grained retrieval tasks. We use a learning rate of 10^{-4} with decay 0.9 and the ADAM optimizer on a

| | Coarse grained | | | | | | Fine grained | | |
| | BAM | | | BAM-X | | | BAM-FG | | |
Model	Top-1	Top-3	mAP	Top-1	Top-3	mAP	IR T-1	IR T-5	IR T-10
Karayev [17]	29.95	36.44	0.34	21.62	23.06	0.18	3.28	5.43	6.59
NST [9]	36.56	39.43	0.38	31.01	32.03	0.28	6.31	7.82	9.86
NST-CGM [5]	34.74	42.96	0.36	32.85	36.84	0.31	3.54	8.32	10.89
DML-BAM [6]	93.16	99.32	0.61	67.12	88.69	0.50	3.57	6.87	9.95
DML-BAMX [6]	98.16	**99.95**	0.49	79.97	92.68	**0.69**	3.04	6.18	8.70
ResNet50 [12]	73.59	96.01	0.162	51.05	78.01	0.163	1.97	5.18	8.41
MUNIT-Unsup [15]	33.79	45.05	0.44	45.76	74.41	0.22	4.12	6.92	9.45
MUNIT-Triplet [15]	43.29	46.47	0.589	66.28	88.66	0.271	17.68	25.89	31.67
MUNIT-Listwise [15]	42.63	46.45	0.404	70.98	91.85	0.226	18.04	24.91	29.62
MUNIT-Contrastive [15]	53.80	86.36	0.284	27.87	58.49	0.137	6.43	10.02	12.96
ALADIN-Unsup.	93.86	99.79	0.394	83.13	96.66	0.308	11.48	16.38	19.84
ALADIN-Triplet	96.59	99.90	0.638	78.60	95.28	0.336	26.70	35.08	40.47
ALADIN-Listwise	95.59	99.90	0.550	76.16	94.12	0.316	37.90	47.88	53.83
ALADIN-Contrastive	**99.48**	**99.95**	**0.737**	**85.28**	**98.07**	0.479	**56.89**	**66.80**	**71.94**

Table 1. Coarse (BAM/BAM-X) and Fine grained (BAM-FG) style discrimination for the ALADIN model compared to baselines on coarse and fine-grained retrieval (BAM-FG). The larger ALADIN-L model is used.

single NVidia Titan-X 12GB GPU for all experiments. We trained all models to convergence with early stopping.

5.1. Datasets and Partitions

BAM-FG/-Raw/-C_N. We train all our models from scratch using 1M images with noisy grouping (BAM-FG-Raw). We split the 1.62M clean images in BAM-FG-C_N into training (115K groups) and test (20K groups) partitions. BAM-FG-C_N is crowd-annotated to form sub-groups as described in Section 4. The data is thresholded at different consensus levels $C_N = [1, 5]$ (groupings referred to as BAM-FG-$C_{1..5}$). Fine-grained retrieval is evaluated over the 78K images test of $BAM - FG - C_3$; we use C_3 as the majority consensus level with the highest data volume.

BAM/-X. Behance Artistic Media (BAM) is a public dataset of 2M contemporary artworks gathered from Behance.net and annotated using a large-scale active learning pipeline [38]. We sample a 70k subset of this dataset, the annotations for which include seven coarse style labels for non-photographic media (3D renderings, comics, pencil/graphite sketches, pen ink, oil paintings, vector art, watercolor). To increase the coarse-scale category count on BAM, we add a further 70K images gathered from web photo collections focusing on photographic styles (luxury, neon, minimalist photography, metallic, abstract shots, geometric forms, pastel shades), bringing the total to 14 coarse style classes each of 10K images. This Extended BAM (BAM-X) dataset of 140K images is used for coarse-grain evaluation. None of these are included in BAM-FG.

5.2. Baseline Methods and Losses

1. Methods. We compare against several existing style representations. **Karayev** *et al.* sample activations from a late FC layer of CaffeNet [17]. Neural Style Transfer [9] **NST** features using VGG-19 [30] pre-trained on ImageNet [22] are compared with a variant taking cosine of Gram ma-

trices **NST-CGM** across layers [5]. A baseline discriminative network *ResNet50* [12], extracting the style representation (embedding) from the penultimate FC layer (2048-D) of a ResNet50 network. We compare also against the coarse-grain style embedding of Collomosse *et al.* [6] applying Deep Metric Learning (DML) via triplet loss. Their model **DML-BAM** was trained on BAM [38], and we retrain it also on BAM-X **DML-BAMX**. We compare against the style codes available from MUNIT [15] which explicitly trains a disentangled representation for stylization. We evaluate MUNIT style codes for search, trained unsupervised (**MUNIT-Unsup** [15]) and trained via our contrastive scheme, per ALADIN (**MUNIT-Contrastive**). In Sec. 5.6 we further explore style code fusion of ALADIN and ResNet.

2. Alternative losses. We also evaluate alternative training strategies for ALADIN, and the baselines. Both **Triplet** loss and **Listwise** loss [27] are explored as smaller batch alternatives to contrastive training *e.g.* **MUNIT-Triplet**, and **MUNIT-Listwise**. For triplet training we train using random negative mining, sampling G^- from random projects, and also hard-negative mining (HN) sampling images from semantically similar projects. For HN, $n \in G^-$ are samples that satisfy a semantic threshold $|S(n) < S(a) + \sum p \in G^+ S(p)| < T$ set empirically to encourage disentanglement similar to deep metric learning (DML) over BAM [6]. We use a pre-trained auxillary ResNet/ImageNet embedding for $S(.)$. The listwise loss [27] is a differentiable approximation to mean Average Precision (mAP), recently applied to deep metric learning for search embeddings. Using list-wise loss several style groups are chosen at random, and all samples are added as query (anchor) images to the batch $B = \{a_1, ..., a_{32}\} \subset T$. For each a we define G^+ as images co-present in the project, and $G^- = T \setminus G^+$. The listwise loss rewards higher ranking of G^+ versus G^-.

3. Alternative supervisions. We baseline against self-

Loss	Backbone	Aug.	MLP	Sup.	Coarse-grained		Fine-grained		
					BAM mAP	BAMX mAP	P@k=1	P@k=5	P@k=10
Triplet	MUNIT [15]			✓	0.589	0.271	17.68	25.89	31.67
Triplet	MUNIT [15]			HN	0.483	0.255	12.96	19.48	24.58
Triplet	MUNIT [15]	✓		✓	0.482	0.252	7.39	12.70	17.16
Triplet	MUNIT [15]	✓			0.501	0.259	7.79	12.81	16.95
Listwise	MUNIT [15]			✓	0.447	0.244	20.15	27.40	32.27
Listwise	MUNIT [15]	✓		✓	0.404	0.226	18.04	24.91	29.62
Listwise	MUNIT [15]	✓			0.410	0.211	16.03	22.61	27.23
Contrastive	Resnet50 [12]	✓	✓	✓	0.231	0.143	40.14	50.90	57.19
Contrastive	Resnet50 [12]	✓	✓		0.209	0.104	28.51	36.72	41.96
Triplet	ALADIN-S			✓	0.558	0.297	25.60	33.71	38.93
Triplet	ALADIN-L			✓	0.638	0.336	26.70	35.08	40.47
Triplet	ALADIN-L			HN	0.609	0.312	23.75	31.87	37.10
Triplet	ALADIN-L	✓		✓	0.614	0.335	22.12	31.55	37.60
Triplet	ALADIN-L	✓			0.417	0.214	21.07	28.20	32.81
Listwise	ALADIN-L			✓	0.463	0.247	25.14	33.44	38.80
Listwise	ALADIN-S	✓		✓	0.523	0.284	30.03	38.42	43.67
Listwise	ALADIN-L	✓		✓	0.550	0.316	37.90	47.88	53.83
Listwise	ALADIN-L	✓			0.453	0.244	21.22	27.98	32.36
Contrastive	ALADIN-S	✓	✓	✓	0.565	0.308	36.42	44.49	49.32
Contrastive	ALADIN-L	✓	✓	✓	**0.737**	**0.479**	**56.89**	**66.80**	**71.94**
Contrastive	ALADIN-L	✓	✓		0.443	0.222	30.47	38.43	43.39

Table 2. Coarse and fine-grained style discrimination of different architectures and training losses, exploiting data augmentation (Aug.) and projection networks (MLP) under self or weak supervision (Sup.). HN indicates hard negative mining. Trained over 1M images in BAM-FG-Raw.
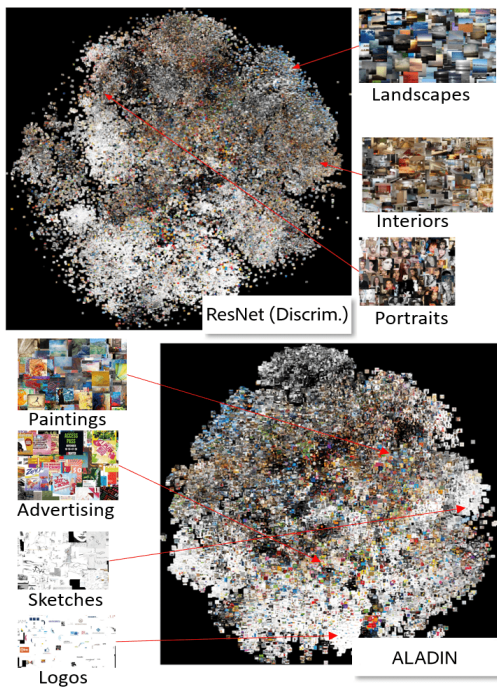

Figure 4. t-SNE visualization of BAM-FG test set within the ALADIN and discriminative embeddings; qualitatively fewer semantic clusters form in the ALADIN vs. Discriminative embedding.

supervised training of ALADIN using no project group supervision *i.e.* $L_{\text{total}} = L_{\text{rec}}$ with minibatches selected from groups randomly. Whilst self-supervised, such loss does not encourage metric learning. Therefore we also baseline our dual loss eq. 4, where L_{con} uses vanilla contrastive learning (*i.e.* self-supervised SimCLR [4]), applying data augmentation to generate G^+ from a using random cropping, hori-
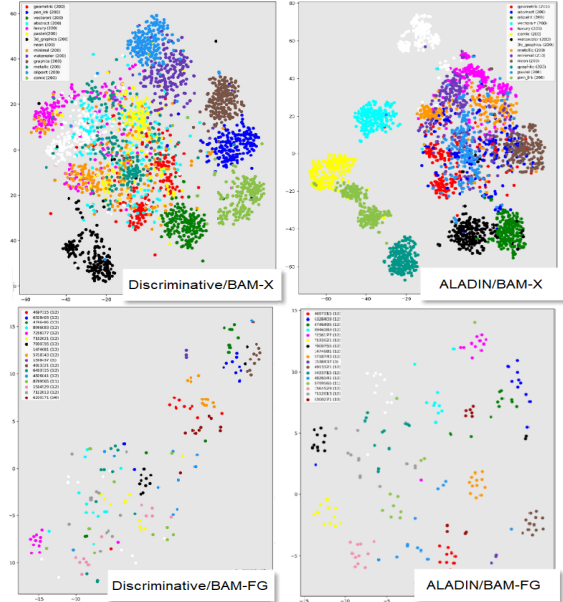

Figure 5. t-SNE visualizations in style embeddings (Discriminative and ALADIN-L). Top: BAM-X coarse grain, color shows 14 test classes. Bottom: BAM-FG, color shows 20 test projects.

zontal flipping, and color jittering, and $G^- = B \setminus \{a, G^+\}$. Finally, we demonstrate the benefit of contrastive learning for weakly supervised learning, by baselining performance against **softmax** loss for an $n-$way classification using project group membership as labelled data. Thus we use our weak supervision signal as explicit labels for strong supervision, a 175K-way classification (one label per group).

5.3. Evaluating Coarse and Fine grained Retrieval

We evaluate retrieval performance of ALADIN, computing precision at k (P@k), an instance retrieval metric that expresses the percentage of queries for which a relevant result (same project) occurs within the top k ranked results, for ranks $k = \{1, 5, 10\}$. Tbl. 1 reports the performance of several baselines at this task; whilst coarse-grain performance is satisfactory on the BAM, and BAM-X datasets, the fine-grained style discrimination (BAM-FG) and generalization to new styles is poor for the baseline approaches, which supports our motivation to develop a new model (ALADIN) that is effective for fine grained style search. The style code learned in **MUNIT-UnSup** [15] model is known to be suitable for image stylization, and we explore it here for style search. We observe that for coarse-grain style discrimination, MUNIT performs similar (BAM P@k=1: 33.79) to the lower-performing prior work that does not explicitly train for style (BAM P@k=1: 34.74) [5]. However **MUNIT-UnSup** performs poorly as a search embedding for fine-grained style, in line with all prior models. Our proposed approach, **ALADIN-Contrastive** (using the ALADIN-L backbone), sets the new state of the art in fine grained retrieval (BAM-FG P@k=1: 56.89).

Tbl 1 also presents the results for coarse and fine grain

Data	Data size	ResNet50 [12]			ALADIN-L		
		P@k=1	P@k=5	P@k=10	P@k=1	P@k=5	P@k=10
BAM-FG-Raw	250K	20.72	28.72	34.30	40.08	49.62	55.27
BAM-FG-Raw	500K	33.80	43.74	49.71	44.26	54.20	59.75
BAM-FG-Raw	750K	38.73	49.11	55.09	47.78	57.89	63.53
BAM-FG-Raw	1M	40.14	50.90	57.19	56.89	66.80	71.94
BAM-FG-C_3	114K	42.33	52.14	57.72	59.03	68.67	73.62
BAM-FG-C_4	86K	43.85	53.86	59.37	59.31	68.80	73.92
BAM-FG-C_5	38K	45.22	54.81	60.52	**59.51**	**69.07**	**74.05**

Table 3. The effect of data volume and consensus level on a fine-tuned model, training with contrastive loss.

Data	Model	P@k=1	P@k=5	P@k=10
BAM	DML [6]	3.57	6.87	9.95
BAM-FG-C_5	Ours (ALADIN-L)	58.98	68.79	73.73
BAM-FG-C_5	Ours (ResNet50 [12])	45.22	54.81	60.52
BAM-FG-Raw	Ours (Fused)	60.76	70.26	75.12
BAM-FG-C_5	Ours (Fused)	**62.18**	**71.50**	**76.33**

Table 4. Performance of fused embeddings Discriminative and ALADIN for noisy (BAM-FG-Raw) and cleaned (BAM-FG).

for our proposed ALADIN network trained unsupervised (via reconstruction loss only; **ALADIN-UnSup**) on BAM-FG_Raw. The fine grained IR performance (BAM-FG P@k=1: 26.70) of ALADIN-L exceeds all prior baseline architectures even without supervision.

5.4. Evaluating weak supervision

Next, we evaluate different ways to weakly supervise ALADIN training on BAM-FG-Raw noisy image groupings, comparing triplet, listwise and contrastive losses (Tbl. 2). The ALADIN model (the ALADIN-L model structure) shows gains at both coarse-grain discrimination (+ 10-12% mAP), and as shown in Tbl 1, fine-grained style discrimination (+ 10-15% IR Top-1). Our proposed dual loss (reconstruction and contrastive learning) outperform others with listwise loss gives superior performance to triplet — reflecting observations that even human annotators find it harder to assess style coherence in paired images versus groups of images. A discriminative Resnet50 backbone was also trained using a softmax loss, with a ~175k-way classifier for every project group in the 1M BAM-FG-Raw dataset. However, it did not perform well for fine-grained discrimination. ALADIN achieves a state of the art performance on fine-grained style discrimination, but interestingly outperforms existing models for coarse grain style. We achieve this without using any of the coarse training labels, demonstrating our model's generalization capabilities.

5.5. Ablation Studies

Tbl 2 reports an ablation study for both coarse and fine grained style discrimination, exploring contrastive learning, data augmentation and use of a projection network (applying the loss to $h(f(.))$ versus $f(.)$), the latter yields performance gain for both models. The pair-wise (triplet) losses consistently underperform the list-wise loss. Performing hard negative mining to encourage disentanglement between semantics and style is ineffective, perhaps due to the

disentanglement in the ALADIN architecture. ALADIN-L (56.89%) outperforms baselines for fine-grained discrimination, including a recent self-supervised baseline (SimCLR [4]). In contrast, a discriminative model [12], ResNet50, similarly trained, achieves 40.14%.

Tbl. 3 explores the impact of data volume on overall performance for reductions of 25-75% of BAM-FG-Raw data, for the best performing models ALADIN-L (P@k=1: 56.89%) and Discriminative (P@k=1: 40.14%). We also explored fine-tuning these two models with the cleaned BAM-FG training data at consensus levels $C_N = [3, 4, 5]$. Despite lower training volumes at higher consensus levels, we show the most significant gains are at these levels (up to 5% at C_5).

5.6. Visualizing Disentanglement

We explore the complementary properties of ALADIN-Contrastive and a similarly trained discriminative network (ResNet50-Contrastive) that does not explicitly disentangle style from content in its architecture. Fig. 6, shows search results for both embeddings, offering a more significant correlation between content and style under the Discriminative model, which does not explicitly disentangle the two within its architecture.

Visualizing a t-SNE distribution of all 78K images in BAM-FG-C_3 (Fig. 4) shows similar groupings. It reveals ALADIN correlates more strongly on lower level features such as color. Smaller-scale visualization of Fig. 5 shows good discrimination between a random subset of 20 fine-grained groups and on coarse-grain styles despite no direct supervision on these classes; consistent with the quantitative performance in Tbl. 1.

Fused. We explored simply concatenating both embeddings given their information appeared complementary. Interestingly, the incorporation of semantic cues gave a small performance boost (around 4%) to ALADIN. Further small gains can be made by further fine-tuning ALADIN on cleaned annotated data from BAM-FG-C_N. Tbl. 4 reports our best performing fine-grained search embeddings are the fused models on noisy data (60.76%) and cleaned data at the highest consensus threshold (C_5 at 62.18%).

5.7. User Evaluation of Style Similarity

We ran crowd-sourced user studies on image retrieval results to compare ALADIN with the fine-grained ResNet-Contrastive model trained on the same data and a coarse-grain baseline (DML [6]). 1000 queries were each presented independently to 5 workers. Each worker was asked to flag images within the 25 results that were style inconsistent with the query. Consensus thresholding (Sec. 4) was applied and Precision at rank k recorded for majority consensus levels in Tab. 5. The fused embeddings consistently achieve the highest accuracy, both where we explore the complete results and when we consider only results where the query style group is held out from the search corpus.

We further explore image retrieval using the models

Figure 6. Representative retrieval results for 4 queries; Coarse-grain style embedding baseline (DML) [6] versus our proposed fine-grained embedding (ALADIN architecture under weak supervision) and for comparison, a similarly trained (discr)iminative network; ResNet.

Consensus C_N	ResNet50 [12]			Ours (ALADIN)			Ours (Fused)			DML-BAMX [6]		
	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10	P@1	P@5	P@10
3	0.959	0.914	0.895	0.965	0.918	0.893	**0.974**	**0.937**	**0.918**	0.780	0.607	0.449
4	0.840	0.681	0.624	0.832	0.653	0.582	**0.867**	**0.727**	**0.666**	0.633	0.397	0.277
5	0.562	0.324	0.256	0.612	0.307	0.221	**0.642**	**0.363**	**0.277**	0.467	0.196	0.120
3 (c)	0.822	0.686	0.519	0.834	0.831	0.826	**0.887**	**0.890**	**0.875**	0.681	0.571	0.428
4 (c)	0.522	0.432	0.322	0.558	0.458	0.450	**0.575**	**0.569**	**0.560**	0.364	0.307	0.229
5 (c)	0.153	0.117	0.096	0.115	0.106	0.101	**0.173**	**0.166**	**0.159**	0.079	0.065	0.049

Table 5. Image retrieval accuracy via Amazon Mechanical Turk (AMT) study; Precision (P@k) at rank k=[1,5,10] over 1000 queries on BAM-FG. (c) indicates "held out" evaluation i.e. removing results in the same style group as the query. These show that when images from the same group are removed, ALADIN (and fused) maintains style similar results for longer. Results at 3 majority consensus levels.

C_N	Ours (ALADIN)			Ours (Fused)		
	P@1	P@5	P@10	P@1	P@5	P@10
3	1.0	0.999	0.999	1.0	0.999	0.999
4	1.0	**0.975**	**0.976**	1.0	0.971	0.972
5	1.0	0.921	0.906	1.0	0.921	**0.909**

Table 6. Results for multi-image queries

Figure 7. Multi-image query results; a query moodboard (top left) and the top 25 ranked result images (bottom). Quantiative results from MTurk user study at majority consensus levels (top right).

when multiple images are used together to form a query. We use our annotated style groups from the test set as such queries, mean-pooling the encoder features of all images to create the query. We repeat our user study and ask groups of 5 workers to label 1000 retrieval results. The retrieval corpus does not contain images from the same project as the queries. Fig. 7 shows multi-image queries, and a representative result with high relevance scores for both ALADIN

and the fused variants, implying the mean well represents the query group and suitable for use as a query.

6. Conclusion

We proposed an encoder-decoder (E-D) architecture; 'ALADIN', to learn a fine-grained representation for measuring fine-grained artistic style similarity. ALADIN is inspired by the E-D models used to drive content stylization [15] by explicitly disentangling content and style across network branches. We present and use a weakly supervised approach to learn this representation using contrastive training made practical using a logit accumulation strategy. We sample 310K user-generated content groupings across 2.62M images which we release as the first fine-grained artistic style dataset, BAM-FG. We exploit these group co-occurrences as an implicit weak proxy for fine-grained style similarity. ALADIN sets the state of the art performance for fine-grained discrimination and improves further when trained with cleaned grouping data and when fused. ALADIN also sets a new state of the art in coarse grain style discrimination, despite not being trained with explicit style labels. Future work could explore the potential of ALADIN as a unified embedding for both stylization and search.

References

[1] P. Achlioptas, M. Ovsjanikov, K. Haydarov, M. Elhoseiny, and L. Guibas. Artemis: Affective language for visual art, 2021. 2

[2] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. In *Proc. ACM SIGGRAPH*, 2015. 2

[3] E. Cetinic and S. Grgic. Automated painter recognition based on image feature extraction. In *Proc. ELMAR*, 2013. 2

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 4, 6, 7

[5] W. Chu and Y. Wu. Image style classification based on learnt deep correlation features. *IEEE Transactions on Multimedia*, PP:1–1, 02 2018. 2, 5, 6

[6] J. Collomosse, T. Bui, M. Wilber, C. Fang, and H. Jin. Sketching with style: Visual search with sketches and aesthetic context. In *Proc. ICCV*, 2017. 1, 2, 5, 7, 8

[7] D. Cooper. *'Ontology of artworks', in A Companion to Aesthetics*. Basil Blackwell, Oxford, 1992. 1

[8] Eric Fernie. *Art History and its Methods: A critical anthology*. Phaidon, London, 1995. 1

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016. 2, 5

[10] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. 2017. 1, 2

[11] A. Gilbert, J. Collomosse, H. Jin, and B. Price. Disentangling structure and aesthetics for content-aware image completion. In *Proc. CVPR*, 2018. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5, 6, 7, 8

[13] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proc. ACM SIGGRAPH*, pages 327–340, 2001. 2

[14] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. ICCV*, 2017. 1, 2

[15] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 1, 2, 3, 5, 6, 8

[16] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, 2016. 2

[17] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoller. Recognizing image style. In *Proc. BMVC*, 2014. 1, 2, 5

[18] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, 2019. 2

[19] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 2

[20] H. Lee, S. Seo, S. Ryoo, and K. Yoon. Directional texture transfer. In *Proc. ACM NPAR*, 2000. 2

[21] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017. 2

[22] T.-Y. Lin and S. Majo. Visualizing and understanding deep texture representations. In *Proc. CVPR*, 2016. 2, 5

[23] L. Marchesotti, F. Perronnin, and F. Meylan. Learning beautiful (and ugly) attributes. In *Proc. BMVC*, 2013. 2

[24] S. M. Mohammad and S. Kiritchenko. Wikiart emotions: An annotated dataset of emotions evoked by art. In *Proc. LREC*, 2018. 2

[25] K. Pang, Y. Yang, T. M. Hospedales, T. Xiang, and Y. Song. Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10344–10352, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 1

[26] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. Efros, and R. Zhang. Swapping autoencoder for deep image manipulation. In *Proc. ECCV*, 2020. 2

[27] J. Revaud, J. Almazán, R. Sampaio de Rezende, and C. R. de Souza. Learning with average precision: Training image retrieval with a listwise loss. *2019 IEEE/CVF ICCV*, pages 5106–5115, 2019. 5

[28] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. In *Proc. GCPR*, 2016. 2

[29] L. Shamir, T. Macura, N. Orlov, and D. Eckley. Impressionism, expressionism, surrealism: automated recognition of painters and schools of art. *IEEE Trans. Applied Perception*, 2010. 2

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 5

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 2

[32] G. Strezoski and M. Worring. Omniart: Multi-task deep learning for artistic data analysis. *arXiv preprint arXiv:1708.00684*, 2017. 2

[33] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proc. ICML*, 2016. 2

[34] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2

[35] P. Upchurch, N. Snavely, and K. Bala. From a to z: supervised transfer of style and content using deep neural network generators. *arXiv preprint arXiv:1603.02003*, 2016. 2

[36] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proc. CVPR*, 2017. 2

[37] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval, 2017. 1

[38] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *Proc. ICCV*, 2017. 2, 5

[39] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR 2011*, pages 1577–1584, 2011. 1

[40] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial net-

works. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. 2

[41] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, and Pappas T. Classifying paintings by artistic genre: an analysis of features and classifiers. In *Proc. IEEE Workshop on Multimedia Sig. Proc. (MMSP)*, 2009. 2