ArtContext: Contextualising Artworks with Open-Access Art History Papers and Wikidata Knowledge through a LoRA-Tuned CLIP Model

Student Name: Samuel Waugh
Supervisor Name: Dr Stuart James
Submitted as part of the degree of BSc Computer Science to the
Board of Examiners in the Department of Computer Science, Durham University

Abstract—Many Art History articles discuss artworks and specific parts of the work, be it layout, iconography, or material. However, when viewing an artwork it is not trivial to identify what different articles have said about the piece. Therefore, we propose ArtContext, a pipeline for taking a corpus of Open-Access Art History papers and Wikidata Knowledge and annotating Artworks with this knowledge. We do this using a novel corpus collection pipeline then learn a bespoke CLIP model adapted using Low-Rank Adaptation (LoRA) to make it domain-specific. We show that the new model, PaintingCLIP outperforms CLIP and provides context to a given piece of artwork. The proposed pipeline can easily be applied across numerous different humanities applications.

Index Terms—Computer vision, Knowledge management, Machine learning, Natural language processing

1 Introduction

When a viewer stands before a painting, the scholarly expertise scattered across hundreds of thousands of art-history papers – insights about iconography, technique, and cultural context – remains largely hidden. Bridging that gap is a core challenge at the intersection of computer vision, natural-language processing, and digital humanities. Effectively grounding painting descriptions in authoritative, nuanced labels drawn from the art-history corpus would give the public richer detail about the works they encounter beyond the simple captions available in galleries. Solving this problem will both advance technical research on domain-adapted vision-language models and widen access to art-historical knowledge.

Much of the literature at the intersection of computer vision with art history still treats paintings as a labelling exercise [1]. Convolutional neural networks are tuned to classify paintings according to movement or author [2]. These outputs may help with indexing and aid in the training of generative AI models to produce pastiche renderings of your photo gallery in the style of Pablo Picasso or Hayao Miyazaki, but they fail to explain what the original artworks depict, the significance of the visual content, or how scholars have interpreted them.

The bias begins with the data. Datasets such as ImageNet [3] and COCO [4] contain millions of sharp photographs whose textures, lighting, and perspective are broadly uniform. The uncommon ways scenes are depicted in paintings lie outside this visual vocabulary. The enormous stylistic variation across centuries and styles means the features a network must extract shift dramatically from one movement to the next. Most people struggle to interpret an unfamiliar piece of modern art without accompanying wall text [5]; the image alone rarely suffices to communicate the work's artistic intent or value, and machines inherit the same ignorance.

Even when bounding boxes are correct, today's pipelines remain object-centred: "figure", "leopard skin", "chariot". But *Bacchus and Ariadne* (Titian, 1523) cannot be understood by listing its props. Bacchus' sudden jump towards Ariadne marks the moment he falls in love. A system that merely notes "male figure flying through air" misses the painting's crucial symbolism. To grasp such meaning, object recognition must consider the textual scholarship that describes the significance of each detail of an artwork – a link that existing computer-vision models have yet to master.

CLIP (Contrastive Language–Image Pre-training) [6] pairs a vision transformer with a text transformer and learns from $\approx$ 400 million image–caption pairs scraped from the open web. Because it embeds whole sentences – not just object tags – into the same space as full images, it is an ideal starting point for bridging the semantic gap between a painting and the prose written about it. To tailor such a model to a niche domain we need a lightweight form of fine-tuning. Low-Rank Adaptation (LoRA) [7] meets that need: it keeps the original network fixed and learns only a small set of add-on weights, steering the model towards new data without expensive retraining. LoRA therefore offers a practical way to inject CLIP with art-historical knowledge while preserving its general visual-language skills.

Large, general-purpose LLMs write fluent text but often invent facts [8]. In fields where mistakes carry real cost like medicine or science, hallucinations are unacceptable. The same logic holds for art history: gallery-goers cannot discern the accuracy of a seemingly plausible AI-generated caption without the help of an expert. The solution has been to pre-train domain-specific models on specialist literature: SciBERT [9], built on 1.1 million research papers, tops general BERT at citation-intent labelling and figure extraction, while BioBERT [10], fed billions of PubMed tokens [11], leads

in biomedical entity recognition. However, because the openly licensed art-historical corpus is orders of magnitude smaller and far less tightly aligned with images than the biomedical or scientific text used for BioBERT and SciBERT, our approach must rely on a lightweight adaptation rather than training a standalone art-history language model from scratch.

Wikidata [12] is an open knowledge graph that already stores an extensive set of metadata about art. The structured way it is stored renders it an excellent source for complementing the nuance of academic work with factual information – which often is too obvious for an art historian to state explicitly in the first place.

Thankfully, scholars have produced thousands of monographs, journal articles, and books rich with the exact content we are looking for. One could argue paintings are the best annotated visual artefacts in existence. While most of the corpus is not accessible freely online, enough of a proportion is available to attempt our research into the following question: to what extent can contextualising artworks with open-access art-history papers and Wikidata knowledge, via a LoRA-tuned CLIP model, advance the state of the art in automated painting captioning?
To answer this question, the project aimed to:

1) Create a structured dataset of open-access art-history papers given what was legally downloadable from the web;
2) Build an entire pipeline which given this dataset would produce a fine-tuned CLIP model better capable of captioning paintings than the original CLIP model;
3) Deploy a shareable tool which would allow scholars, art-history students, and enthusiasts to evaluate the quality of our model's captions in a distributed manner.

We achieved the first two aims and plan to develop the third soon.

Before building a pipeline we sought to validate our hypothesis that CLIP would be capable of associating paintings and academic prose in a shared embedding space going beyond its pre-training which associates digital images with simple natural language descriptions. Fig. **??** provides evidence that CLIP can *see* the cheetahs discussed in the extract from the National Gallery's Primary Teachers' Notes [13].

We designed and implemented ArtContext, an end-to-end pipeline that (i) harvests open-access art-history papers with the OpenAlex API, (ii) converts the PDFs to clean text, (iii) splits the text into candidate sentences, (iv) matches those sentences to paintings via Sentence-BERT [14] and Wikidata metadata, and (v) packages the resulting image–sentence pairs as training labels. Using these labels we attached LoRA adapters to the projection heads of CLIP ViT-B/32 and fine-tuned for 20 epochs, monitoring InfoNCE loss [15] to select the best checkpoint. We then evaluated the adapted model, PaintingCLIP, against the baseline with macro-averaged precision–recall curves on a set of canonical works and carried out a qualitative error analysis to classify the most common ways our model fails.

The project delivered:

1) A curated corpus of 27 044 open access art history PDFs grouped by 450 artists and ranked by relevance according to OpenAlex;
2) A labelled dataset of 38 749 paintings paired with scholarly sentences and structured Wikidata fields;
3) The ArtContext code base, covering data collection, sentence mining, label synthesis and model training;
4) The LoRA weight deltas for PaintingCLIP, which measurably improve recall of correct scholarly captions over the original model
5) Evaluation notebooks and plots that reproduce all quantitative results in the paper.

2 Related Work

To contextualise progress in relation to ArtContext, we review advancements in automatic art analysis in section 2.1, focusing on how contextual awareness can be synthesised with visual processing. Approaches to vision language models (VLMs) — and specifically adaptations of the CLIP model — are presented in section 2.2. In section 2.3 we review methods for lightweight domain adaptation and in section 2.4 state-of-the-art NLP and article summarisation techniques are explored with a view to informing an improved textual input for a VLM designed to describe artworks.

2.1 Automatic Art Analysis

Seeking to go beyond style classification, Garcia and Vogiatzis released the SemArt dataset [16]: 21 384 European paintings, each paired with structured metadata and a curatorial comment. Accompanying this, they introduced the Text2Art retrieval task together with simple joint-embedding baselines. Under the benchmark: a model must embed paintings and their curatorial comments metadata in a shared space, then (i) retrieve the correct image given a textual query and (ii) retrieve the correct text given an image. Performance is reported with median rank (MR) and Recall1/5/10, so high scores indicate that the system has captured the semantic link between art-historical prose and visual content rather than relying on shallow cues. The work framed art understanding as matching prose to images, but stopped short of modelling the relationships between artistic attributes or of exploiting external knowledge graphs.

Garcia *et al.* [17] extended SemArt with a pair of ContextNet variants that fuse image and context in two distinct ways. Multitask Learning (MTL) feeds each painting through a ResNet-50 [18] backbone; the shared 2048-D embedding is simultaneously optimised to predict four attributes—Type, School, Time-frame and Author. The network is encouraged to encode visual cues that are useful across tasks, like brushwork that correlates with both author and school. Knowledge-Graph Mode (KGM) instead builds a 33 K-node graph whose edges link paintings to their attribute nodes; `node2vec` [19] produces 128-D embeddings, and an encoder is trained to align the ResNet features with these graph vectors while still solving the same classification tasks. MTL treats context as implicit whereas KGM treats it as explicit. On SemArt, MTL tops the vision-only baseline on School (69.1 %) and Time-frame

(63.2%), whereas KGM leads on Type (81.5%) and Author (61.5 %). The KGM scored best in the Text2Art challenge, showing that lightweight contextual signals sharpen the embedding space. Limitations remain: `node2vec` cannot encode higher-order semantics and at test time the model still relies on the global image crop and its supervised labels rather than fine-grained grounding or zero-shot reasoning.

Castellano *et al.* [20] proposed the ArtGraph knowledge graph which incorporated metadata and visual attributes from a large dataset sourced from WikiArt and DBpedia, creating connections between entities like artists, genres, and historical periods. Node embeddings were learned with a Graph Attention Network (GAT) [21] and concatenated with Vision-Transformer (ViT) features in a multi-task network that jointly classifies style and genre whereby a learned projection lets the model operate at test-time using images only. The proposed model was evaluated against others on the SemArt dataset, including baseline ResNet and ViT models as well as previous KG-integrated models such as MTL. The ViT-GAT combination consistently outperformed all alternatives in both single-task and multi-task classification tasks for artwork style and genre.

Cetinic [22] presented a novel method for generating captions for images of artworks by leveraging iconographic information. The research focused on generating descriptions that incorporate the contextual and symbolic meanings that are important in art historical analysis. This task is particularly challenging due to the complexity and interpretative nature of art. Cetinic fine-tuned on the Iconclass AI Test Set [23], an 86k image corpus whose descriptions are extracted from alphanumeric Iconclass codes — a hierarchical thesaurus used by museums to catalogue the subject matter of artworks. They fine-tuned a vision-language pre-trained model on this dataset and evaluated the quality of the generated captions using both standard image captioning metrics like BLEU [24] and METEOR [25], and more recent reference-free metrics such as CLIPScore [26]. Although many outputs capture the overall iconography, the model still misses fine-grained distinctions. It produces very brief, tag-like phrases rather than full sentences. Because these captions are generated from global image features without explicit region-level grounding, the system cannot show which part of the painting each word refers to, making it hard to disentangle multiple objects or scenes within a single image.

Kadish *et al.* [27] investigated the challenge of detecting objects, specifically people, in art images using deep learning techniques. It addressed the cross-depiction problem, where neural networks trained on photographic data struggle to recognise objects in non-photographic images such as paintings or drawings because such detection mechanisms are usually reliant on the texture of an object to identify it. AdaIN [28] style transfer was used to create a large dataset (StyleCOCO) by applying artistic styles to images from the COCO dataset [4], which is pre-labelled for object detection. The authors fine-tuned a Faster R-CNN (a Region-Based Convolutional Neural Network) with a ResNet-152 backbone using the StyleCOCO dataset. The model was then tested on the People-Art dataset, resulting in improved object detection performance over prior methods, achieving an AP50 score of 0.68. A limitation of the paper is the bias in both the ImageNet backbone and the style transfer dataset, which may cause the model to perform poorly on certain art styles or depictions not well represented in the training data.

Strafforello *et al.* [29] trialled four VLMs - CLIP, LLaVA [30], OpenFlamingo [31], and GPT-4o [32] - on their ability to predict art style, artist and time period in a zero-shot setting, that is, using the models "as is" with only carefully worded prompts and no additional fine-tuning on art images or labels. Using two public datasets and a curated test set named ArTest, the study examined whether these models, trained across multimodal datasets, can match human expertise in art classification. Findings revealed that GPT-4o performed the best overall, yet all models fell short in some areas, with specific misclassifications of challenging or nuanced art styles. The study concluded that while VLMs show potential, they are not yet sufficiently reliable for art history applications without expert supervision. Limitations include the models' frequent misclassifications on complex examples and a limited style vocabulary in some VLMs, indicating a need for further refinement before broader deployment in art historical research.

A body of literature has looked into how VLMs can be modified to boost their descriptive abilities. The next section presents how the CLIP model has been enhanced in a broad variety of ways.

2.2 VLMs and CLIP

Contrastive Language–Image Pre-training (CLIP) [6], developed by OpenAI, leveraged the transformer architecture, introduced in [33], to create a multimodal model capable of understanding both images and text. CLIP utilises dual transformer encoders — one for image input and one for text — to independently embed visual and linguistic information. For the image encoder, based closely on the work of Dosovitskiy *et al.* [34], CLIP uses a Vision Transformer (ViT) that divides each image into fixed-size patches and treats them as tokens, each with a positional embedding to retain spatial structure. These tokens pass through layers of multi-head self-attention and feed-forward networks, capturing long-range dependencies across the image and producing a holistic visual representation. Through contrastive learning, CLIP aligns these image embeddings with those generated from text, associating paired images and descriptions within a shared latent space. By learning from large-scale internet data, CLIP works in a zero-shot fashion, enabling it to interpret and classify unseen images with natural language responses.

Specifically for art, Baldrati *et al.* [35] naively applied CLIP to artwork classification and retrieval tasks on the NoisyArt dataset [36], a collection of web-sourced artwork images. CLIP's performance was tested using a shallow classifier for zero-shot classification and image-text retrieval and it was found that it significantly outperformed traditional ResNet models with higher accuracy and robustness against domain shifts. While the study demonstrated CLIP's effectiveness in handling noisy, weakly-supervised data, the authors admitted limitations and that fine-tuning the architecture specifically for artwork domains and expanding testing to datasets with more curated and annotated examples should enhance the generalisability of future models.

Each adaptation of CLIP presented is benchmarked on different tasks, so no single variant is "best" in every setting. Across all the enhancements in the literature, textual quality is critical. Fan *et al.* [37] introduced LaCLIP, or Language-Augmented CLIP, which built upon the original model by addressing a critical asymmetry in its data augmentation. While the original model applies data augmentations to images during training, its text inputs remain static, leading to potential over-fitting on specific language patterns and limited exposure of the model to diverse linguistic structures. LaCLIP enhanced this by introducing a language augmentation strategy where LLMs like LLaMA generate rewrites of the original text descriptions associated with each image. These rewritten captions vary in sentence structure and vocabulary while retaining essential concepts, thus providing a richer linguistic context for training. By randomly sampling either the original text or its augmented version during training, LaCLIP improved zero-shot transfer capabilities, achieving an 8.2% accuracy boost on ImageNet [38] over CLIP. A drawback of this approach is that the augmented texts can introduce noise if the generated captions misalign with the actual image content, potentially affecting the model's accuracy in certain contexts. The data augmentation can also introduce a large computational overhead although this can be minimised if the dataset for fine tuning is not too large.

Seeking to enhance CLIP in an entirely different direction, Zhong *et al.* [39] extended CLIP's capabilities with RegionCLIP to understand and process details within specific image regions, addressing the limitation of global image-text alignment in the original CLIP model. Whereas CLIP was trained to match entire images to text, RegionCLIP focuses on region-level visual-text alignment, crucial for tasks like object detection. The architecture incorporates a region proposal network (RPN) that generates object-specific regions in an image, which are then aligned with text descriptions created from a pool of object concepts derived from the original caption. This design facilitates the creation of "pseudo" region-text pairs that are used in pre-training, enabling RegionCLIP to learn localised representations without relying solely on full-image labels. RegionCLIP showed marked improvements over CLIP in zero-shot and open-vocabulary object detection tasks, significantly outperforming other baselines on datasets like COCO [4] and LVIS [40]. A limitation of RegionCLIP is its inability to discern the relationship between objects in an image, which means it is unlikely to be suitable for art analysis where a holistic visual comprehension is necessary.

Beyond pairing regions of an image with instances of vocabulary, phrase localisation aims to identify and localise objects in an image described by a text phrase. Li *et al.* [41] proposed a method to adapt the pre-trained CLIP model for this task. The authors achieved this by generating high-resolution spatial feature maps from CLIP's ViT and ResNet models. Per-pixel similarity scores were computed with a text query to produce a heat-map. The object in the text query is localised by finding the bounding box that maximises this score, offering a flexible framework for text-based region identification in images. Compared to RegionCLIP, which focuses on fine-tuning CLIP for object detection, this method operates purely in a zero-shot context without retraining, making it scalable and adaptable to novel phrases and categories. Tested on datasets like Flickr30k [42] and Visual Genome [43], the method achieved state-of-the-art performance for zero-shot phrase localisation, surpassing ZSGNet [44] by an absolute 5% on most zero-shot splits. This indicates strong generalisation, especially on long-tailed object categories (ones which occur only rarely in the training data) which are typically challenging for traditional supervised methods. This is especially relevant to identifying objects in artworks since they can often be obscure and not occur often in training.

Zhang *et al.* [45] expanded CLIP's capabilities with Long-CLIP by enabling it to handle much longer text descriptions, surpassing the original model's 77-token limit. Using knowledge-preserved stretching, Long-CLIP extends the positional embeddings, allowing it to process up to 248 tokens while preserving CLIP's ability to interpret shorter inputs accurately. Additionally, primary component matching allows the model to focus on different levels of detail: for short captions, it captures primary image components, while for longer descriptions, it hones in on finer, more specific image elements. This enhancement made Long-CLIP significantly more effective for tasks like detailed image retrieval and text-to-image matching, with a 25% improvement in long-text image retrieval accuracy over the original model. However, while capable of handling longer inputs, Long-CLIP may still require additional training for highly intricate details, as it benefits most from comprehensive long-form text data.

2.3 Lightweight Domain Adaptation

Houslby *et al.* [46] first showed that large transformers can be adapted to many downstream tasks by adding only a few task-specific parameters instead of fine-tuning everything. They injected tiny "adapter" modules (two-layer bottlenecks with skip connections) after every attention and feed-forward block. During downstream training only the adapter weights and per-layer layer-norm scales were updated (2–4% extra parameters per task). The adapters still add latency at every layer, their size must be hand-tuned, and they were validated mainly on classification.

Prompt-tuning introduced by Lester *et al.* [47] is a parameter-efficient way to adapt a frozen, pre-trained language (or vision–language) model to a new task by learning only the prompt that is fed to the model, rather than updating the model's internal weights. Zhou *et al.* [48] took the idea a step further with CoOp, and showed that prompt-tuning can steer a frozen CLIP model. They replaced the hand-written context "a photo of a" with a handful of learnable embedding vectors optimised on only a few labelled examples per class. Accuracy across eleven datasets was improved by double-digits against zero-shot CLIP. Because no backbone weights change, adaptation is memory-light and inference-time latency is unchanged. However, the learned prompts are task-specific, opaque, and must be re-tuned when the domain shifts.

Zhou *et al.* then extended CoOp with Conditional Prompt Learning (CoCoOp) [49] by turning the prompt from a fixed, class-specific string into a prompt that is dynamically generated from each input image. A small MLP

takes the frozen CLIP image embedding and outputs the context vectors that precede the class name. These image-conditioned prompts are trained on base classes and then reused unchanged at test time. CoCoOp keeps CoOp's gains on seen classes while boosting accuracy on unseen classes and out-of-distribution datasets, reducing the prompt overfitting that static CoOp prompts suffered from.

LoRA (Low-Rank Adaptation) [7] fine-tunes large pre-trained models by freezing all original weights and inserting two rank–k matrices into each linear layer: a "down" projection $\boldsymbol{A} \in \mathbb{R}^{d\times k}$ compresses the hidden dimension, and an "up" projection $\boldsymbol{B} \in \mathbb{R}^{k\times d}$ expands it back, with their product $\boldsymbol{BA}$ added to the base weight $\boldsymbol{W}$. This low-rank update captures the task–specific shift with only ≈ 0.1-1% of the parameters required for full fine-tuning, and the matrices can be merged into $\boldsymbol{W}$ at inference, leaving runtime unchanged. This fraction of parameters is significantly lower than for the adapters introduced by Houslby *et al.* [46] while being free of the per-token prompt generation needed by CoCoOp's conditional prompts. LoRA therefore enables models such as CLIP to absorb new domain knowledge efficiently while preserving their original capabilities. The next section explores how the textual input fed to CLIP could be generated efficiently for the art history domain.

2.4 NLP, LLMs, and Article Summarisation

The technique of text summarisation has been categorised by Sharma *et al.* [50] in a recent review of state-of-the-art methods into *i)* extractive – using original sentences or content to formulate the summary; or *ii)* abstractive – generating a summary which does not necessarily contain a reference back to the original content. Extractive methods, popular for their simplicity, involve selecting key sentences directly from the source text based on statistical features, such as word frequency or sentence position. In contrast, abstractive methods generate summaries by rephrasing the content, often employing deep learning models for a more human-like output. These techniques include sequence-to-sequence neural networks, such as those using encoder-decoder architectures, and transformers like BERT [51] and GPT [52]. Abstractive models utilise advanced NLP tasks like sentence fusion and paraphrasing to produce summaries that are more fluent and cohesive, though they are complex to train and computationally intensive. Recently, hybrid methods have emerged, blending extractive and abstractive approaches to leverage the advantages of both, achieving higher accuracy and readability but still facing challenges in coherence and redundancy management. The paper emphasises that while deep learning-based approaches dominate current research, issues like resource requirements, handling rare words, and summary evaluation persist as key areas for further development.

Abstractive Summarisation: LLMs are advanced deep learning models that excel in natural language processing tasks by leveraging vast amounts of data and computational resources. Unlike earlier techniques, such as Latent Dirichlit allocation [53], which model topics through statistical associations of words, LLMs are built on transformer architectures [33] that capture complex language patterns and dependencies across large contexts. LLMs, like GPT [52] and BERT [51], employ self-attention mechanisms, allowing them to weigh word relationships dynamically within and across sentences. This structure enables LLMs to understand nuanced language, maintain coherence, and preserve context, making them particularly adept at abstractive summarisation; far superior to earlier statistical and topic-modelling approaches.

Nechakhin *et al.* [54] explored the use of LLMs for summarising scientific papers. The potential of large language models (LLMs) such as GPT-3.5, Llama 2, and Mistral to automate the extraction of structured research dimensions within the Open Research Knowledge Graph (ORKG) was investigated. The ORKG traditionally relies on human experts to curate structured properties (e.g., "model family," "methodology") that summarise scientific contributions, but this process is time-intensive and can suffer from inconsistencies. To assess whether LLMs could streamline this curation, the authors tested the models' ability to identify relevant dimensions across various scientific fields. Evaluation methods included semantic alignment with ORKG's human-curated properties, cosine similarity scoring via SciNCL embeddings, and expert surveys for quality assessment. The results indicated that while LLMs capture many core research dimensions, their performance varies in capturing domain-specific nuances. GPT-3.5 exhibited the highest alignment with ORKG's manual annotations, showing promise for LLM integration in summarisation tasks. However, the study highlights a gap in domain-specific accuracy, suggesting that fine-tuning LLMs for specific scientific fields would be necessary to match the depth of expert curation fully.

Extractive Summarisation: Traditional machine learning techniques can be applied to extractive summarisation by classifying sentences based on features like sentence length, word frequency, cue words, and sentence position. These features are used to identify important sentences within a document. Techniques include Naive Bayes [55], which learns the probability distribution of these features to rank sentences, as well as decision trees and support vector machines [56], which rely on feature scoring functions. Additionally, methods like hidden Markov models (HMM) [57] and conditional random fields (CRF) [58] assume dependencies between sentences, unlike other methods that treat sentences as independent.

More recently, deep learning techniques have been employed to improve extractive summarisation by focusing on sentence representation and selection as the two key tasks to be tackled. Models like convolutional neural networks (CNNs) [59] and recurrent neural networks (RNNs) [60], including gated recurrent units (GRUs) [61], are used to generate embeddings that capture sentence semantics and context within the document. For sentence selection, unsupervised models such as Restricted Boltzmann Machines (RBM) [62] have been used to optimise sentence extraction based on statistical and semantic features. These methods show significant performance improvements over traditional techniques like TextRank [63] and LexRank [64], particularly when trained on large datasets.

Sentence-BERT (SBERT) [14] modifies BERT with a Siamese architecture that maps whole sentences to dense, 768-dimensional vectors such that semantically similar texts lie close together. By enabling cosine-similarity comparison in milliseconds, SBERT turns tasks like extractive summarisation, semantic search and clustering from quadratic-time cross-encoding into simple nearest–neighbour look-ups, while retaining much of BERT's representational power. Its efficiency and strong performance on STS benchmarks make it a pragmatic backbone for selecting the most relevant sentences from academic papers.

Qi *et al.* [65] introduced a structure-aware heterogeneous graph model (SAPGraph) specifically for extractive summarisation of scientific papers. It constructs a graph from scientific texts by creating three types of nodes: sections, sentences, and entities (key terms or concepts within the text). The model uses a graph neural network (GNN) [66] to process these interconnected nodes, capturing both local sentence-level details and global document structure to generate more representative summaries. By incorporating explicit structural information, like the Introduction, Method, Result, and Conclusion sections, SAPGraph captures the logical flow of scientific papers and mitigates the "head distribution problem"—where models tend to over-select content from early sections.

Extractive summarisation may be better suited for summarising art history papers compared to abstractive summarisation because it preserves the original terminology, descriptions, and nuanced language critical for accurately conveying artistic styles, techniques, and historical context. Art history texts often contain specific references to periods, movements, and artistic details that need precise phrasing to retain their value. Extractive summarisation directly selects sentences or phrases from the original text, thus maintaining the integrity of these details, which might otherwise be misinterpreted or oversimplified in an abstractive approach using LLMs. Additionally, the descriptive nature of art history, with its focus on visual detail and interpretation, benefits from the reliability of extractive methods, where key descriptive phrases are less likely to be altered, ensuring summaries remain accurate and contextually faithful.

Existing research demonstrates that visual-language models can be made more "art-aware" either by adding contextual metadata (knowledge-graph or multi-task approaches) or by tailoring CLIP with domain-specific augmentations, region proposals, or longer prompts. Parallel work in NLP shows how extractive techniques and large language models can surface precise, expert-written sentences from lengthy papers. Yet, to date these two strands have remained largely disjoint: models that excel at recognising visual style seldom ground their outputs in authoritative scholarship, while text-centric pipelines do not close the loop back to vision. Consequently, no system reliably links an image of a painting to the very sentences art historians have written about it.

3 METHODOLOGY

To contextualise artworks, we propose ArtContext, a pipeline for ingesting and connecting art history article text and knowledge to artworks. Our pipeline consists of four main stages designed to fine-tune the CLIP model to accurately associate paintings with descriptive texts. First, we automatically ingested a structured corpus of academic art history articles. For a given set of artists, articles were gathered using the OpenAlex API, chosen for its comprehensive indexing of open-access research papers with appropriate filters (language, relevance, *etc.*). Second, we converted the downloaded articles from PDF to structured text. PDFs were transformed into Markdown files, then cleaned and split into concise candidate sentences. Each sentence was combined with its immediate neighbours to create richer contextual descriptions before embedding these sentences using Sentence-BERT [14]. Using Wikidata metadata, we dynamically constructed semantic queries for each painting and compared these against the embedded candidate sentences using cosine similarity. This enabled us to effectively choose the sentence most relevant to each painting. Finally, we fine-tuned the CLIP model using the carefully selected and contextually enriched labels paired with their digital images. Each label combined the painting's metadata with the most relevant extracted sentence. To efficiently adapt the model to our specific dataset, we applied Low-Rank Adaptation (LoRA) to CLIP's visual and text projection heads. This allowed us to enhance the model's performance with minimal computational overhead. Together, these stages form a coherent pipeline, culminating in a fine-tuned vision-language model, which we introduce as PaintingCLIP, capa-

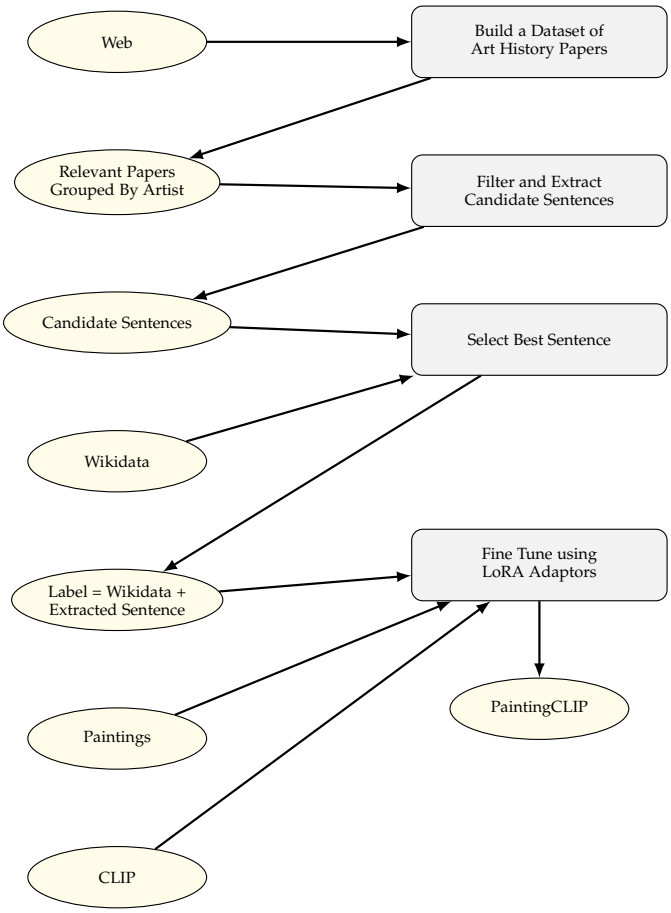

Fig. 1. ArtContext methodology workflow, showing inputs/outputs (ellipses) and processing stages (rounded rectangles).

ble of more accurately describing paintings using academic-level art historical descriptions.

3.1 Relevant Corpus Discovery

The goal of corpus discovery is to collect a set of academic art history articles which can be used to extract expert-written descriptions of paintings. To be useful, articles had to be both *available* and *relevant*. An article was considered available if it could be legally downloaded. We limited our search only to open-access papers, avoiding any sources that shared papers illegally or without permission. We discuss copyright in more detail in Sect. 3.5.

An article was relevant if it discussed art history and described the visual content of paintings. During initial testing, we discovered that many open-access papers were scientific articles. These often mentioned famous artists but did not describe their artworks, adding noise to our dataset. We therefore needed a method to filter these irrelevant papers effectively.

Numerous options for searching for academic articles exist, including Google Scholar, IEEE Xplore, and ACM Digital Library. However, they overly focus on scientific articles and generally do not provide an API. Alternatively, JSTOR provides a humanities-focused portal; however, it requires paid access. Therefore, we opted for OpenAlex [67] to find suitable papers. OpenAlex structures its universe of works $\mathcal{W}$ under a hierarchy of 4 domains, 26 fields, 252 subfields, and 4500 topics $\mathcal{T}_{\text{all}}$. Every work ω indexed by OpenAlex has a set of topic tags

$$\forall\, \omega \in \mathcal{W} :\ \text{tags}(\omega) \subseteq \mathcal{T}_{\text{all}}.$$

So that our dataset included only relevant papers, all of the topics were hand-picked from the subfields of the Art-History field under the domain of Social Sciences. We constructed the set $\mathcal{T}_{\text{art}}$ of 14 topics. A work k was considered for download iff

$$\text{tags}(\omega) \cap \mathcal{T}_{\text{art}} \neq \varnothing \tag{1}$$

A set of artist's names was generated by querying ChatGPT-4o [32] for individuals belonging to diverse movements, cultures, and historical periods to ensure wide and balanced coverage. This step requires manual interaction, although it could be automated using the OpenAI API. We then queried OpenAlex once per artist and downloaded papers to 450 folders, one for each artist. The query required the papers to meet the following criteria: *i)* Written in English *ii)* Open access and downloadable as a PDF and *iii)* (1) is satisfied. Let $\mathcal{A} = \{A_1,\, A_2,\, \ldots,\, A_{450}\}$ be the set of sets of papers for each artist. Then for each artist's name A_i^{Name} the query Q_i returns a set of downloaded works ranked by the relevance score r OpenAlex returns for each work based on the query. We also set a relevance threshold ρ. We found a value of $\rho = 1.0$ struck the right balance between relevance and coverage. A_i is the set of papers downloaded for artist A_i^{Name}.

$$\begin{aligned} A_i &= (A_{i1},\, A_{i2},\, \ldots,\, A_{in}), \\ &\quad \text{with } r_{i1} > r_{i2} > \cdots > r_{in} > \rho. \end{aligned} \tag{2}$$

Using OpenAlex had several advantages:

- It clearly indicated whether papers were open access, ensuring ethical sourcing.
- It provided topic tags and a relevance scoring system, allowing effective filtering.
- The API and its documentation were clear and easy to use, simplifying implementation.

The only technical challenge encountered was occasional request timeouts due to the large number of queries. This was resolved by implementing exponential back-off, ensuring requests respected the API's rate limits. This process produced our first asset: a structured dataset of open-access art history articles. The dataset is not only useful for this project but could also support further research in the digital humanities.

3.2 Corpus Filtering and Visual Sentence Extraction

Each paper A_{ik} was a PDF. $\forall A_i \in \mathcal{A}$ we wanted a set of candidate contexts S_i, not a set of PDFs. To describe how S_i was created, we first define the sub-pipeline Φ.

$$A_{ik} \xrightarrow[\substack{\text{PDF to}\\ \text{Markdown}}]{P_1} M_{ik} \xrightarrow[\substack{\text{NLTK}\\ \text{Sentence}\\ \text{Tokenisation}}]{P_2} C_{ik} \xrightarrow[\substack{\text{SBERT}\\ \text{Embedding}}]{P_3} S_{ik} \qquad (\Phi)$$

3.2.1 P_1: PDF to Markdown

We first converted each PDF A_{ik} into a Markdown file M_{ik} using the Marker software package [68]. Marker was chosen because it reliably converts PDF content into clearly structured text and images and the repository is actively updated. The documentation for Marker was clear, which made implementation straightforward. A major challenge in this step was that Marker takes a long time to process large PDF files. Some of these large files were open-access books with hundreds of pages. Converting these large documents took too long to be practical. Therefore, we decided to convert only PDF files smaller than $10\,\text{MB}$. Although this meant excluding some useful information in large documents, it significantly improved processing speed and efficiency.

3.2.2 P_2: NLTK Sentence Tokenisation

Next, we took each Markdown file M_{ik} and extracted a set of contexts C_{ik}. A context here is defined for each sentence as the concatenation of the previous sentence, the sentence itself, and the subsequent sentence, where possible. Leveraging regular expressions, all images, links, code blocks, and residual Markdown symbols were removed from the file's content. This yielded plain text suitable for extracting sentences. Next, we split the cleaned text into individual sentences using NLTK's Punkt sentence tokeniser [69]. We opted for NLTK because it is fast. We ignored sentences with fewer than four words because they generally did not contain useful descriptions. To assign more semantic meaning to the subsequent embeddings, we concatenated each sentence with the sentence immediately before and after it, if available. These short paragraphs provided more context for the next stage of processing.

3.2.3 P_3: SBERT Embedding

Finally, we embedded these context paragraphs C_{ik} using Sentence-BERT (SBERT) as a semantic vector space using the SBERT (`paraphrase-MiniLM-L6-v2`) model to return the set S_{ik} of embeddings. SBERT was selected because it efficiently produces accurate semantic embeddings. It offered a practical trade-off between accuracy and computational speed.

$$S_{ik} = \{\, S_{ik}^1,\, S_{ik}^2,\, \ldots,\, S_{ik}^m \}$$

3.2.4 The Set of Candidate Contexts

Each candidate context was represented by an SBERT embedding. These embeddings were essential for the next stage, where we matched the best candidate contexts to each painting using its Wikidata metadata.

$$S_i \;=\; \bigcup_{k=1}^{n} \Phi(A_{ik})$$

3.3 Using Wikidata to Extract the Most Appropriate Sentence from Each Painting's Candidate Set

In this stage, the aim was to select the most appropriate descriptive sentence for each painting. We did this by matching candidate contexts from the previous stage with relevant metadata obtained from Wikidata.

We first gathered detailed metadata for paintings by querying the Wikidata knowledge base. Wikidata was selected as it provides extensive and accurate metadata, is straightforward to query, and returns structured data suitable for automated processing. We ranked paintings by their number of Wikidata links, a heuristic indicating a painting's prominence or cultural significance. We used Excel coupled with the Pandas library [70] as this simplified validating intermediate results. There were 37 449 entities with the tag `Q3305213` labelling them as a painting. Let $P = \{\, p_1,\, p_2,\, \ldots,\, p_{37\,449} \}$ be the set of paintings we have Wikidata metadata for. Each $p_l \in P$ has the following metadata fields: *Title*, *Creator*, *Movements*, *Depicts*, *Year*, and *Link Count*.

Next, we used this metadata to dynamically generate a natural-language query for each painting. Namely $\forall p_l \in P$ we define w_l as the SBERT embedding of the sentence: "{*Title*} is a {*Year*} painting by {*Creator*} depicting {*Depicts*}". For example, for the painting *Café Terrace at Night* (Vincent van Gogh, 1888), we embedded "Café Terrace at Night is a 1888 painting by Vincent van Gogh depicting platform, gas burner, La Cité, lamp, sett, Arles, coffeehouse, chair, table, tree, night, sky, star, human". Fig. **??** shows that the information within the *Depicts* field substantially enriches the semantic content of the query. We compared each painting's query embedding against embeddings of the candidate sentences extracted earlier in the pipeline. By matching on first and last names we found the set of candidate sentences S_i such that *Creator* $= A_i^{\text{Name}}$. Letting ψ be the SBERT function which returns the cosine similarity score between two sentence embeddings, then we extracted the best sentence s_l^* as the sentence in the middle of the context s_l where

$$s_l = \arg\max_{s \in S_i} \psi(s, w_l)$$

The sentence with the highest similarity score was selected as the best match for describing the painting. Grouping articles by artist in previous stages proved advantageous here and it was our rationale for doing so in the first place. It significantly reduced the computational load by narrowing each painting p's sentence search to articles associated specifically with its artist A. This approach was effective, as articles mentioning a specific painting almost always reference the painting's creator. Querying OpenAlex using painting names would have been impractical. The selected sentences formed a structured dataset used to construct labels for fine-tuning CLIP in the subsequent stage.

3.4 Domain Adaptation of CLIP to Corpus

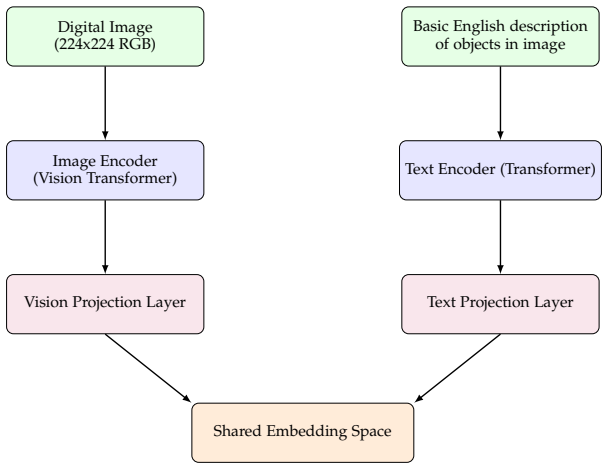

Fig. 2. High-level illustration of CLIP architecture. Images and text are processed by separate encoders and projection layers before being aligned within a shared embedding space.

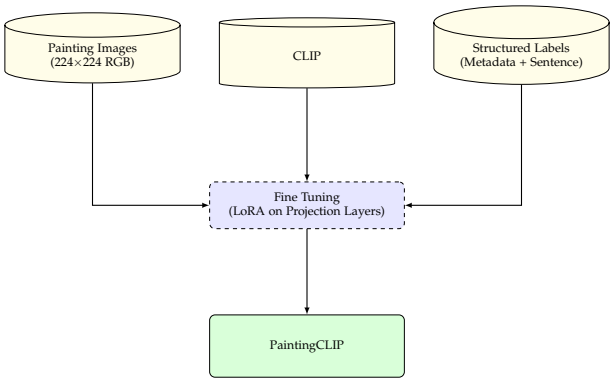

Fig. 3. Fine-tuning stage. Pre-trained CLIP, structured text labels and normalised images are fed to a fine tuning script, producing the PaintingCLIP model.

To link the content of the corpus to the artwork content we fine-tuned the pre-trained CLIP model using the corpus. The goal was to adapt CLIP specifically to the task of matching paintings with accurate textual descriptions. We approached this by constructing detailed labels for each

painting and then applying lightweight fine-tuning using Low-Rank Adaptation (LoRA).

Firstly, we generated structured labels for each painting. These labels combined information from Wikidata with the most relevant extracted sentences identified in the previous stage. This approach ensured that each label included essential descriptive context. Some paintings had limited representation in academic literature. For these cases, incorporating metadata from Wikidata was essential to give the label some factual grounding. Even for popular paintings, the metadata enriched the descriptions, adding valuable context typically omitted in scholarly texts. The labels consisted of the painting's title, artist name, creation year, style, depicted subjects, and a descriptive sentence. The combined text was tokenised and truncated to a maximum of 77 tokens using CLIP's truncation function τ_{77}. The input for fine-tuning is a set of (image, label) pairs $L = \{t_1, t_2, \ldots, t_{29\,697}\}$

$$
\begin{aligned}
t_l &= (p_l^{\text{Image}},\ \tau_{77}(p_l^{\text{Label}})),\\
p_l^{\text{Label}} &= p_l^{\text{Wiki}} \ +\ s_l^*,\\
p_l^{\text{Wiki}} &= b_1 + b_2 + b_3,\\
b_1 &= \text{“}p_l^{\text{Title}}\ \ (p_l^{\text{Year}})\text{ by }p_l^{\text{Creator}}\text{”},\\
b_2 &= \text{“Style: }p_l^{\text{Movement}}\text{”},\\
b_3 &= \text{“Depicts: }p_l^{\text{Depicts}}\text{”}.
\end{aligned}
$$

We initialised the model with the publicly available checkpoint (`openai/clip-vit-base-patch32`) from the Hugging Face Hub [71] and then fine tuned it on the (Image, Label) pairs in L. We selected CLIP because it is highly effective for matching visual and textual information, especially in tasks involving zero-shot predictions. To efficiently fine-tune the model, we integrated LoRA adapters into two critical components of CLIP: the visual projection head and the text projection head. Specifically, we set LoRA parameters with a rank of 16, an α scaling factor of 32, and dropout at 0.05. These settings provided an optimal balance between increased model adaptability and computational efficiency. LoRA was chosen due to its ability to effectively fine-tune large models with very few computational resources. It achieves this by modifying only a small subset of the model's parameters, significantly reducing training time and resource usage. This was essential for our task, given the large scale of our dataset. We split our dataset into training and validation subsets, using a 90/10 ratio. The fine-tuning process ran for 20 epochs, with a batch size of 16, using the AdamW optimiser [72] at a learning rate of 2×10^{-4}. Training was executed on an Apple Silicon GPU using PyTorch's MPS back-end.

We monitored training progress through loss metrics and validation cosine similarity between image and text embeddings. Checkpoints were saved whenever the validation cosine similarity improved. After completing training, we produced graphs of training loss and validation similarity across epochs to verify successful training and identify potential over-fitting. Overall, this fine-tuning approach should efficiently tailor CLIP specifically to the domain of art history, improving its ability to associate paintings with accurate and contextually enriched textual descriptions. We saved just 32 768 LoRA deltas ($\approx 0.04\%$ of full CLIP).

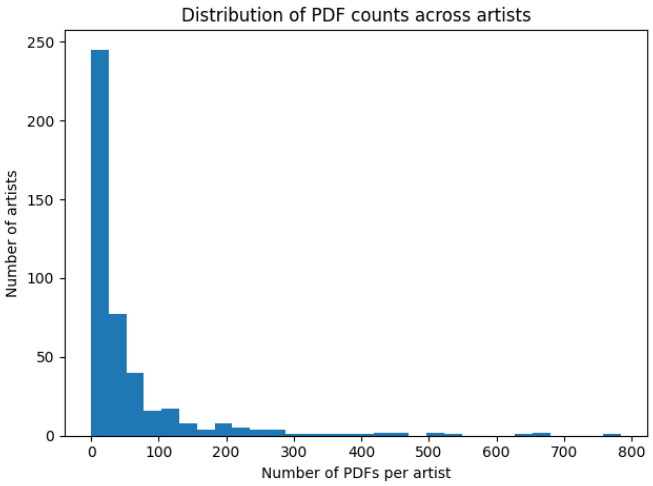

Fig. 4. Histogram of article frequency $|A_i|$ per artist A_i across the entire corpus.

We introduce the CLIP model with these deltas used at inference as PaintingCLIP.

3.5 Corpus discovery Copyright and Ethics

All PDFs were catalogued using the OpenAlex API, restricting queries to records with `is_oa = true` and then reading the associated `best_oa_location.url` field, which, by OpenAlex's own definition, points only to locations where "you can read the full-text of this work without needing to pay money or log in" [73]. Every file therefore carried a lawful open-access licence (typically a Creative Commons variant) or was in the public domain. Because the project is non-commercial research, making local copies for text and data is additionally permitted under the UK copyright exception in s. 29A of the *Copyright, Designs and Patents Act* 1988 [74], which allows computational analysis of works to which the researcher already has lawful access. At no point did we scrape subscription databases, bypass paywalls, or use so-called "shadow libraries" such as Sci-Hub or Library Genesis, sites that courts have repeatedly found to infringe authors' rights [75]. The entire corpus was assembled in full accordance with both legal requirements and accepted scholarly ethics, and materials lacking a clear open-access status were automatically excluded.

4 Results

4.1 Corpus of Open Access Art History Papers

Using the proposed ingestion pipeline (section 3.1) we constructed a structured corpus of open-access art history articles, $\mathcal{A}$, grouped by artist. The corpus consists of freely available art history articles from the web. $\mathcal{A}$ contains 27 044 articles over 450 artists. For each article, the OpenAlex's relevance score is maintained as metadata.

The two distribution plots show that the number of PDFs per artist i varies significantly; $|A_i|$ spans two orders of magnitude. The Fig. 5 frequency plot compares the 20 artists with the most PDFs to the 20 artists with the fewest, showing a stark difference. Prominent artists like Henri

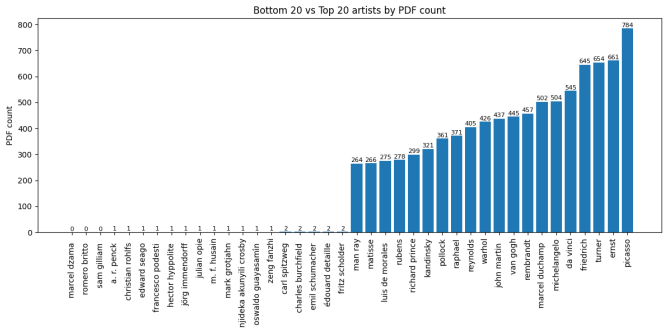

Fig. 5. Frequency plot to illustrate the skew in our dataset

Matisse have hundreds of articles while obscure artists like Julian Opie only have a single article which makes any mention of their work. The Fig. 4 histogram further shows that most artists have fewer than 50 articles, illustrating the uneven availability of art history articles.

A reasonable heuristic for measuring how prominently a painting p features in the academic literature is the number of links pointing to that painting's Wikidata tag, $p^{\text{Link Count}}$. A more accurate measure would be the number of occurrences of the painting name among the papers of $\mathcal{A}$ but the $p^{\text{Link Count}}$ exists $\forall p \in P$ and suffices to present the skew. Fig. 6 seeks to demonstrate that there is a strong bias in what is referenced on the web. It shows data for the 300 most linked to artworks. From the graph we can infer that works which are core to the art history canon are written about much more than obscure works. This skew presents a challenge for fine tuning at scale.

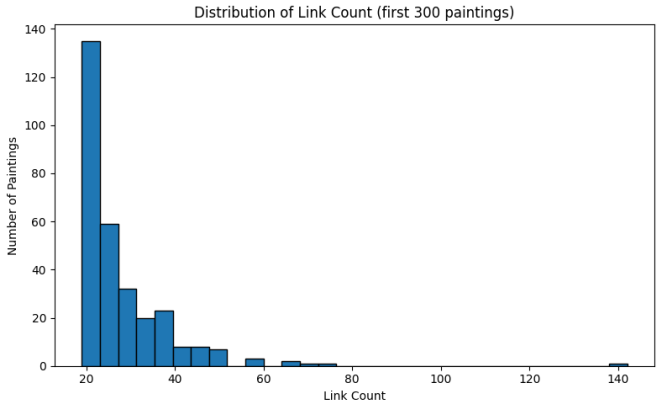

Fig. 6. Histogram of the top-300 $p^{\text{Link Count}}$ values

4.2 PaintingCLIP

4.2.1 *Fine tuning results*

InfoNCE [15] is a symmetric contrastive loss that encourages matched image–text pairs to have high similarity while simultaneously pushing all other (negative) pairs apart. It is particularly well suited as a fine-tuning metric because it exactly matches CLIP's training objective—measuring both positive alignment and negative separation within each batch—and thus faithfully reflects improvements in the joint embedding space. The InfoNCE loss can be defined as:

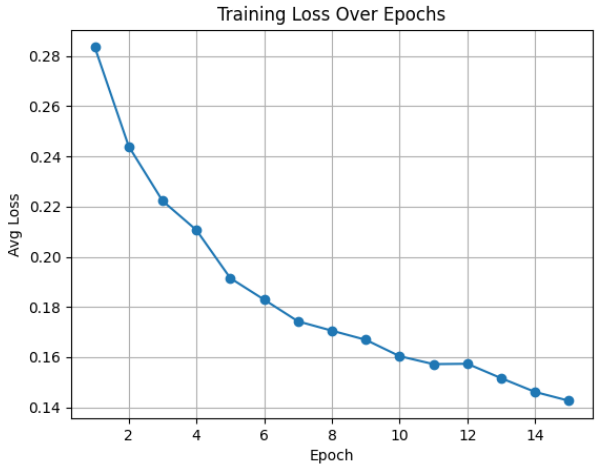

Fig. 7. Average training InfoNCE loss per epoch for the CLIP ViT-B/32 model fine-tuned with high-capacity LoRA adapters on our dataset

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}\left[\log \frac{f(x_{\text{pos}}, \mathbf{c})}{\sum_{x_j \in X} f(x_j, \mathbf{c})}\right] \tag{3}$$

where the scoring function $f(x, \mathbf{c})$ is defined as:

$$f(x, \mathbf{c}) = \exp\left(\frac{\text{sim}(x, \mathbf{c})}{\tau}\right) \tag{4}$$

In practice, we evaluate this loss twice: once with the image embedding as context and the text embedding as query, and once with the roles reversed—and minimise the average of the two terms, exactly mirroring CLIP's original symmetric training objective.

Here, $\text{sim}(x, \mathbf{c})$ is the similarity measure, τ is the temperature hyper-parameter, x_{pos} is the positive sample and X is the set of one positive and $N - 1$ negatives.

In Figure 7 the training-loss plot shows a clear, steady decrease in the average InfoNCE loss over 15 epochs, falling from roughly 0.28 at the start to 0.14 at the end. This downward trend confirms that the LoRA adapters are successfully learning to align each painting with its caption on the training set.

By contrast, the validation InfoNCE loss initially dips slightly—reaching a minimum of about 0.26 around epoch 7—before rising back up to nearly 0.29 by epoch 15. Together, these curves indicate that although the model continues to fit the training data more tightly, its ability to generalise to unseen paintings begins to worsen after mid-training. To mitigate against this overfitting we saved the LoRA values at epoch 8 and use this as our model for PaintingCLIP.

4.2.2 *A mixed methods comparison of the 2 models*

Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_{10}\}$ be the ten paintings with the highest Wikidata–link counts in our corpus. For each painting π_i and for each model $M \in \{\text{CLIP}, \text{PaintingCLIP}\}$ the model returns a ranked list $(x_{ij}, s_{ij})_{j=1}^{10}$, where x_{ij} is the j^{th} candidate sentence and $s_{ij} = M(\pi_i, x_{ij}) \in \mathbb{R}$ is its image–sentence similarity score. Based on our art–historical knowledge and by consulting Wikipedia we assign binary

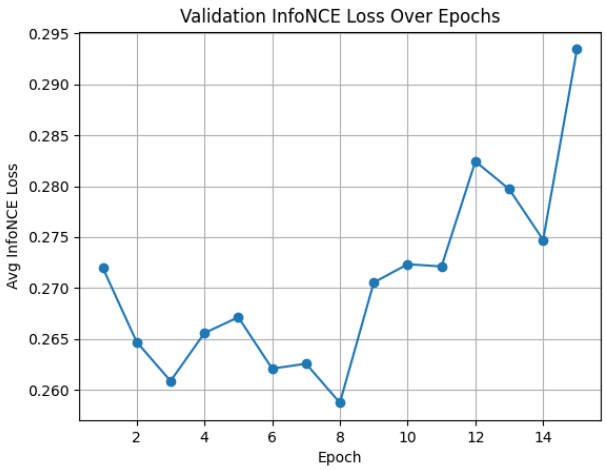

Fig. 8. Average InfoNCE loss per epoch on the validation set of our dataset

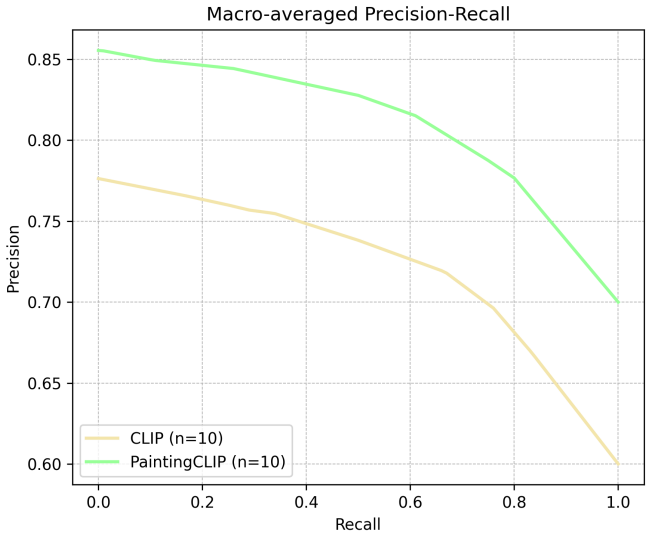

Fig. 9. Macro-averaged $\bar{P}(R)$ curves for CLIP *vs.* PaintingCLIP

relevance labels $y_{ij} \in \{0, 1\}$, where $y_{ij} = 1$ iff x_{ij} is a coherent, factually correct statement about π_i. Hence every painting yields a score vector $\mathbf{s}^{(i)} = (s_{i1}, \ldots, s_{i\,10})$ and a truth vector $\mathbf{y}^{(i)} = (y_{i1}, \ldots, y_{i\,10})$. We pass the pair $(\mathbf{y}^{(i)}, \mathbf{s}^{(i)})$ to `sklearn.metrics.precision_recall_curve` to obtain an empirical precision–recall (PR) curve $C_i = \{(R_i(\tau_k), P_i(\tau_k))\}_k$ [76]. Adhering to the envelope correction of Davis and Goadrich [77], each C_i is (i) linearly interpolated on the common recall grid $\mathcal{R} = \{0, 0.01, \ldots, 1\}$ and (ii) *upper-enveloped* to enforce monotone precision. The macro-averaged curve for a model is then $(r, \bar{P}(r))_{r\in\mathcal{R}}$ with

$$\bar{P}(r) = \frac{1}{10} \sum_{i=1}^{10} P_i(r), \tag{5}$$

thereby enabling a like-for-like comparison of how accurately CLIP and PaintingCLIP retrieve descriptive sentences for the most culturally significant paintings in the dataset.

TABLE 1
Typical issues with PaintingCLIP.

Failure mode	Example candidate sentence
The painting is mentioned but the sentence is about something else	"In this direction, the most noteworthy motif we have found so far involve the elements as well as the partial or complete reproduction of Picasso's Guernica." (Label for *Guernica* (Pablo Picasso, 1937))
Incoherent sentence with some useful information	"], Christ as Salvator Mundi, oil on panel, 45.4 × 65.6 cm (17 × 25)." (Label for *Salvator Mundi* (attrib. Leonardo da Vinci, c. 1500))
Plausibly relevant but requires expert verification	"He paints prostitutes and peasants, struggling for survival, provoking emotional response." (Label for *The Potato Eaters* (Vincent van Gogh, 1885))

4.2.3 *PaintingCLIP's potential and shortfalls*

Fig. **??** presents the 8 sentences with $y_j = 1$ among the top-10 candidates $\{(x_j, s_j)\}_{j=1}^{10}$ that PaintingCLIP retrieved for *The Night Watch* (Rembrandt van Rijn, 1642). The quality of these labels validates our methodology as promising. Table 1 seeks to classify the typical problems that arise.

5 Evaluation

5.1 Evaluation of the evaluation methods adopted

A precision–recall (PR) curve of $P(R)$ *vs.* R captures the essence of our retrieval task from first principles. Given a score vector $\mathbf{s}^{(i)}$ and binary truth vector $\mathbf{y}^{(i)}$ for each painting, precision $P = \frac{\text{TP}}{\text{TP+FP}}$ expresses the reliability of the sentences the model surfaces, while recall $R = \frac{\text{TP}}{\text{TP+FN}}$ expresses their coverage. Varying the decision threshold therefore traces the complete trade-off between selectivity and exhaustiveness in ranking, and macro-averaging the curves assigns equal importance to every painting irrespective of how many positives it contains [78]. Its main drawback, however, is that the ground-truth labels rely on our own judgements. Non-expert annotation is known to introduce noise and bias, and the labour it entails limits the evaluation to a very small sample [79]. Consequently, the curves offer encouraging evidence of PaintingCLIP's promise but cannot yet be regarded as definitive proof of superiority over the baseline.

A more robust method would enlist the help of art-history experts to assess the models' outputs and assign $y_{ij} \in \{0, 1\}$. Non-blind trials whereby academics could explore how the models label paintings they are experts on would yield even better qualitative evaluations and enable us to make stronger claims about where the models excel or falter across different artistic movements and artists. Such an evaluation methodology would require an interactive tool that would make it easy for humanities professors to get a feel for how good the models are.

5.2 Outline of implementation issues

5.2.1 *Difficulty collecting art history articles*

The quality of input data is crucial for any deep learning system; if poor material is fed in, the model's outputs will inevitably suffer [80]. This fact underpins our decision to

leverage art history papers: paintings have received more expert annotation than almost any other image class, making them arguably the best-documented visual artefacts in existence. Yet these annotations are far more complex than the straightforward labels used for tasks such as autonomous-vehicle training, since understanding a painting's visual content, historical context and interpretative nuances demands domain expertise. Nonetheless, it remains vital that the papers we collect support both breadth and scale in our dataset. As Fig 4 shows, a major challenge arises from the skewed focus of the art-history canon: celebrated artists and famous works attract extensive scholarship, while obscure painters and lesser-known pieces are sparsely documented. OpenAlex is limited and does not index every article ever written. Furthermore, we discovered that only 21.4% of the papers OpenAlex indexes are open access and only 8.5% link to documents which are downloadable automatically. This limits the volume of usable training data. In addition to using OpenAlex, we did contact over a dozen notable art history journals requesting access to their content for research purposes. Understandably, journals recognise the value of their data and were not willing to share it. A system which is able to cite annotations to give journals and article authors credit for the labels the system generates could go some way to persuading journals to let us use their data for training in the future.

5.2.2 Issues with Marker and Natural Language Processing

We found that converting PDFs to Markdown with Marker was highly computationally intensive, so we limited processing to files under $10\,\mathrm{MB}$ to keep runtimes and memory use manageable. By including works greater than $10\,\mathrm{MB}$ (a step simply requiring more computational power) our data coverage would grow. Splitting the resulting Markdown into sentences proved non-trivial: articles contain irregular content and punctuation alone cannot reliably mark sentence boundaries, since abbreviations, ellipses and embedded lists often mimic end-of-sentence cues. NLTK's Punkt tokeniser does an adequate job on well-formed prose, but it still splits at novel abbreviations, unconventional punctuation and leftover Markdown artefacts. Ultimately, true sentence segmentation at scale would require some semantic understanding of the text beyond statistical boundary detection. One way to improve this stage of the pipeline would be to categorise works. A bespoke tool tailored to each category of work (book, article, monograph, *etc.*) could be used to convert the PDF to plaintext. The name matching function which matches a painting's creator p^{Creator} to the name of the directory where the PDFs for that artist are stored $A \in \mathcal{A}$ is not perfect. Special characters mean several artists aren't matched properly. An easy fix would be to store a dictionary of all exceptions, mapping from strings containing unusual Unicode characters to their respective partners.

5.2.3 Fine-tuning issues

We performed only a very light-weight fine-tuning, applying low-rank LoRA adapters on ViT-B/32 for twenty epochs with modest batch sizes. We did not have access to a supercomputer — everything ran locally on Apple Silicon MPS. As a result, our effective batch size and learning-rate schedule were constrained by the limited MPS memory which slowed convergence and forced smaller micro-batches. Each full training run took many hours, so exploring more than a handful of hyper-parameter combinations was infeasible. Instead we just picked sensible values to keep wall-time below 10 hours. In particular, testing different learning rates, batch sizes or adapter ranks would have multiplied our wall-time by tens or hundreds, making systematic tuning impractical.

We also confined our LoRA adapters to the final projection heads; we did not experiment with placing adapters on the query/key/value or feed-forward layers within each transformer block, nor at the patch-embedding or token-embedding stages. Alternative adapter architectures—such as standard bottleneck Multi-Layer-Perceptron (MLP) adapters after each attention or MLP block remain unexplored. Beyond LoRA, we might have tried prefix-tuning (learning small prompt vectors for text or image inputs), BitFit [81] (tuning only bias parameters), or even full-model fine-tuning, each offering different trade-offs in parameter efficiency, memory use and flexibility.

Under these constraints over-fitting emerged quickly and we could not test whether deeper or broader adaptation strategies would produce better results. In a larger-scale or production context, access to stronger GPUs or a compute cluster, systematic check-pointing, detailed logging, and automated hyper-parameter sweeps, could all yield insights into how to conduct fine tuning best in this instance.

5.3 Discussion of strengths and weaknesses of the system

We built a coherent end-to-end pipeline, but several limitations stem from our underlying data and model choices. First, our corpus of 27 049 art-history PDFs skews heavily towards the artists most prominent in the canon; many painters have few or no open-access papers. We made no effort to augment this imbalance; no style-transfer or synthetic data to bolster under-represented artists. This uneven coverage risks over-optimising on well-studied painters and under-serving those with sparser literature. 9 044 files out of the 37 749 we had metadata for could not be matched to an image file name out of the 448 352 PNG files we had stored. A further 8 files were malformed meaning our fine-tuning labels derive from only 29 697 paintings. While this was enough to demonstrate ArtContext has potential, a future model would identify why the mismatches are occurring and address them thereby increasing the size of our fine tuning set.

Second, our sentence-selection step uses SBERT because of its speed and ease of batching. While effective for broad semantic matches, SBERT can overlook subtler contextual cues that more sophisticated transformers (e.g., fine-tuned cross-encoders or domain-adapted language models) might capture. A deeper, slower sentence-scoring approach could provide richer, more accurate grounding labels, especially for nuanced art-history descriptions.

Third, our fine-tuning methodology itself was constrained by hardware: lightweight LoRA on the projection heads only, no hyper-parameter sweeps, and a single Apple-

Silicon GPU. We did not explore alternative adapter placements (e.g., within attention or MLP blocks), prefix-tuning, BitFit, or even full-model tuning. Given increased computational resources, future work should carefully consider strategies to leverage them effectively, enabling a model to capture the unique visual and textual characteristics of paintings more comprehensively.

A more fundamental weakness lies in CLIP's own design: it accepts 224 × 224 images and up to 77 tokens of text. Paintings often contain fine brushwork and complex compositions that may vanish at low resolution, and describing them fully will exceed 77 tokens. Future work should evaluate vision–language models with longer text contexts and larger image contexts, as these may better capture the subtleties of style, iconography and brush technique.

Despite these shortcomings, our pipeline's modular architecture is a clear strength. Each stage—from OpenAlex querying through Marker conversion, NLTK tokenisation, SBERT embedding, Wikidata retrieval and CLIP fine-tuning—can be swapped out or extended independently. This design invites experimentation with newer models, alternative summarisation methods, or improved layout parsers without rewriting the pipeline.

Finally, our choice of extractive labelling rests on sound literature precedent: in domains requiring expert precision such as medicine, extractive summarisation often outperforms hallucination-prone abstractive methods.

5.4 Self Evaluation

If I were to do the project again, the most general improvement I would recommend to myself would be to construct small experiments which are easy to conduct quickly to validate or invalidate any intuitions I had. Motivated by the necessity of using time efficiently, this approach would also force me to question my underlying assumptions about the field more rigorously. In service of this aim, I should have taken the time to properly map out visually the entire hypothetical pipeline before I began coding. If I had already formulated the modular design I would later use, namely specifying the inputs and outputs for each module, I would have been able to construct tests to discover how best to solve each sub problem independently of one another. Instead, I worked sequentially through the pipeline. This blocked my progress as dataset collection proved trickier than I had originally anticipated, and I was left with a tight timeline to implement the remainder of the project. Furthermore, I should have consulted the academic literature more than I did. Once my methodology was formulated, I should have zoomed in and consulted the literature on each individual problem I was trying to solve optimally. Instead, I often opted for writing up lightweight solutions which were easy to implement. A quintessential example of this is my use of the Marker module for PDF to text conversion. Once I found the GitHub repository and knew it did the job, I didn't bother to look around for alternatives. This approach meant I did deliver a complete pipeline, but a better marriage between practicality and using state of the art methods was possible in hindsight.

This has been the largest software project I have ever undertaken. It has made clear to me how valuable readable, well commented code is and I'm glad I wrote my code this way from start to finish because it made deciphering how individual scripts worked relatively painless. A software development change I would make is clearly sectioning off unused code so that I don't come back to it later and mistake it for code which is part of the pipeline. A properly written `README.md` file accompanying each directory of Python files would also aid in helping (at a higher level of abstraction than reading code) how different scripts work in concert to achieve a module's aims.

A management technique which helped me clarify my progress each week was writing structured meeting reports based on discussions with my supervisor Dr Stuart James. After each meeting I classified notes into three categories: answers to questions; project next steps; and general wisdom. I neglected this structure in the second half of Epiphany Term and the project's management suffered. Instead of having clear achievable aims agreed upon each week with Dr James, I took a more improvisational approach to just "trying to get something to work". I also undervalued the time needed to write a report of high quality. Taking the time to formulate my methodology with mathematical notation should have been the starting point of the write up instead of a retrospective edit. All told, I'm happy that the project has delivered concrete assets and has arguably made an academic contribution. The prospect of becoming a researcher has become less intimidating and I look forward to extending this work soon.

6 Conclusion

6.1 Project Summary

We set out to determine whether a general-purpose vision–language model could be steered, with modest computational effort, towards the specialised domain of art history. We downloaded 27 044 open-access PDFs, parsed them into 38 749 image–sentence pairs anchored by Wikidata metadata, and applied Low-Rank Adaptation to CLIP ViT-B/32. The resulting model, PaintingCLIP, preserves the zero-shot versatility of the base model while exhibiting measurably stronger alignment between paintings and expert prose. On the ten canonical works with the densest literature, PaintingCLIP exhibited better precision at every recall value. Qualitative inspection confirmed that the top-ranked sentences consistently contain nuanced, authoritative, and esoteric information that an abstractive summarisation model like an LLM could never generate.

Despite the limited material, a lightweight LoRA configuration (rank 16, $\alpha = 32$) attached only to CLIP's projection heads was sufficient to reduce training InfoNCE loss from 0.28 to 0.14 and to lower validation loss until epoch 8. The improvement demonstrates that parameter-efficient tuning is a practical option for domains where compute budgets are tight and full back-propagation through a multi-million-parameter model is infeasible.

6.2 Contributions

The project contributes three assets. First, the cleaned corpus of 27 044 open-access PDFs grouped by 450 artists and ranked by the OpenAlex relevance score based on that

artist's name. Second, ArtContext is a modular end-to-end pipeline that chains OpenAlex harvesting, Marker PDF conversion, NLTK segmentation, SBERT retrieval, and LoRA fine-tuning. Third, we publish the LoRA weight deltas for PaintingCLIP so that anyone can reproduce our results by loading the original OpenAI checkpoint and merging the adapters.

These outputs matter for two reasons. From a computer-science perspective, they show that bridging from dense language (journal prose) to dense vision (paintings rich in symbolic content) is possible without pre-training from scratch. From a cultural perspective, the work moves beyond object identification towards evidence-based description, explanation, and interpretation. Because our model utilises extractive methods instead of abstractive ones, with a bit of extension, it will be able to attribute these descriptions to their authors. This credit is very valuable to the original authors, and they are much more likely to opt-in to allow their work to be used by our model than a generative AI model which steals their scholarship without any recognition.

6.3 Limitations

During corpus construction we verified a data constraint: only 8.5% of the art-history records indexed by OpenAlex are both open-access and machine-downloadable, and many of those are short exhibition notes rather than full journal articles. The highest-quality scholarship – books, biographies, monographs, catalogue raisonnés, conservation reports – remains behind paywalls or is arduous to download automatically.

The dataset also exhibits obvious bias: Western masters are over-represented, while large swathes of non-Western and contemporary art are scarcely documented. Any vision–language system trained solely on public text will therefore inherit the canon's bias.

Fundamentally the model is limited by resolution: the 224×224 crop used by CLIP discards brush-stroke detail and marginal scenes critical to the most insightful analysis. Region-aware adapters or higher-resolution backbones such as SigLIP [82] could preserve this information. Similarly, the 77-token limit truncates many sentences. Integrating Long-CLIP or a dedicated text encoder would enable richer queries and therefore improve fine tuning and recall. With greater computational power, we could experiment with adding LoRA adapters to other parts of the model, such as the encoder, the MLP layers, and the attention blocks. A proper study into finding a better architecture in conjunction with conducting hyper-parameter tuning would yield an even better model which could capture stylistic nuance that our projection-head updates might be missing.

Our evaluation of the model is also biased. Because open access text concentrates on famous Western works, our test set does too, and the reported gains may not generalise to under-discussed paintings. Future studies could stratify evaluation by link-count deciles or artistic movement to report performance where data is scarce. After all, this is the very use-case in which automated extractive assistance is most valuable.

6.4 Future Work

The most obvious extension is data centric. Incorporating museum data, catalogue raisonnés, and digitised books would expand coverage, though it will require formal agreements with rights-holders and more robust layout parsing. A better construction of p^{Label} could replace p^{Wiki} with the SemArt dataset's natural language descriptions for the 21 384 paintings it is available for. Modules like Marker are always seeking to improve the quality of their parsing methods by incorporating the latest state of the art techniques. Improving the document filtering and sentence cleaning part of the pipeline is essential if we want the majority of labels to be readable. Every extracted sentence should be accompanied by a structured citation triple $\langle \text{author}, \text{work}, \text{year} \rangle$.. This would give credit to authors and journals while making it possible to score sources by authority or recency at inference time.

It should also be investigated to what extent phrase localisation [41] can be applied to PaintingCLIP. In multi-figure compositions—*The Last Supper* (Leonardo da Vinci, c. 1495–1498) is a prime example—locating captions near each person could help viewers identify each of Jesus's apostles and illuminate the individual nuances of how each are depicted.

An interactive web interface could showcase PaintingCLIP's top-k sentences for a given painting, permitting experts to evaluate the model. The resulting labels would refine the model and more importantly yield a benchmark absent from the current literature for evaluating a model like ours. A mobile app could recognise a painting via image matching or wall-label text and display the top, coherent sentences, each with links to the respective sources. Gallery goers would have relevant academic knowledge a few taps away, journals would receive traffic, creating an incentive for further open access, and scholars would receive the credit they deserve.

In summary, our work shows that a modest amount of domain-specific text, coupled with low-rank adaptation, can move a general vision–language model towards greater art-historical competence. By publishing a structured corpus ($\mathcal{A}$), a reproducible pipeline (ArtContext), and adapter weight deltas which improve upon CLIP (PaintingCLIP), we provide the basis for others to push beyond our work. With broader data, explicit citations, refined sentence filtering, expert evaluation feedback, and a mobile UI capable of identifying a painting in a gallery, an upgraded version could make reliable, source-backed interpretation available to the public. Lying dormant, the highest-quality analysis and interpretation of paintings is tucked away in obscure journals and academic works. We have demonstrated that in the not-too-distant future, the valuable work of art historians could be read by orders of magnitudes more people, across galleries worldwide.

References

[1] N. Van Noord, E. Hendriks, and E. Postma, "Toward discovery of the artist's style: Learning to recognize artists by their artworks," *IEEE Signal Processing Magazine*, vol. 32, no. 4, pp. 46–54, 2015.

[2] B. Saleh and A. Elgammal, "A unified framework for painting classification," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 1254–1261.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 2014, pp. 740–755.

[5] B. Serrell, *Exhibit labels: An interpretive approach*. Rowman & Littlefield, 2015.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[8] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 2021, pp. 610–623.

[9] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[10] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[11] National Library of Medicine (US), "Pubmed," https://pubmed.ncbi.nlm.nih.gov/, 1996, bethesda (MD): National Center for Biotechnology Information, U.S. National Library of Medicine. Accessed 5 May 2025.

[12] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[13] National Gallery, London, "Primary Teachers' Notes: *"Bacchus and Ariadne"* by titian," Education resource (PDF), National Gallery, London, UK, 2000, accessed: 2025-05-05. [Online]. Available: https://www.nationalgallery.org.uk/media/13681/notes_titian-bacchus-ariadne.pdf

[14] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[15] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International conference on machine learning*. PMLR, 2020, pp. 4182–4192.

[16] N. Garcia and G. Vogiatzis, "How to read paintings: semantic art understanding with multi-modal retrieval," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[17] N. Garcia, B. Renoust, and Y. Nakashima, "Context-aware embeddings for automatic art analysis," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 2019, pp. 25–33.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[20] G. Castellano, V. Digeno, G. Sansaro, and G. Vessio, "Leveraging knowledge graphs and deep learning for automatic art analysis," *Knowledge-Based Systems*, vol. 248, p. 108859, 2022.

[21] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[22] E. Cetinic, "Towards generating and evaluating iconographic image captions of artworks," *Journal of imaging*, vol. 7, no. 8, p. 123, 2021.

[23] ——, "Iconographic image captioning for artworks," in *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part III*. Springer, 2021, pp. 502–516.

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[25] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[26] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, "Clipscore: A reference-free evaluation metric for image captioning," *arXiv preprint arXiv:2104.08718*, 2021.

[27] D. Kadish, S. Risi, and A. S. Løvlie, "Improving object detection in art images using only style transfer," in *2021 international joint conference on neural networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[28] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[29] O. Strafforello, D. Soydaner, M. Willems, A.-S. Maerten, and S. De Winter, "Have large vision-language models mastered art history?" *arXiv preprint arXiv:2409.03521*, 2024.

[30] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.

[31] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa *et al.*, "Openflamingo: An open-source framework for training large autoregressive vision-language models," *arXiv preprint arXiv:2308.01390*, 2023.

[32] OpenAI. (2024, Aug.) GPT-4o system card. OpenAI. Online; accessed 7-May-2025. [Online]. Available: https://cdn.openai.com/gpt-4o-system-card.pdf

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[35] A. Baldrati, M. Bertini, T. Uricchio, and A. Del Bimbo, "Exploiting clip-based multi-modal approach for artwork classification and retrieval," in *International Conference Florence Heri-Tech: The Future of Heritage Science and Technologies*. Springer, 2022, pp. 140–149.

[36] R. Del Chiaro, A. D. Bagdanov, and A. Del Bimbo, "Noisyart: A dataset for webly-supervised artwork recognition." in *VISIGRAPP (4: VISAPP)*, 2019, pp. 467–475.

[37] L. Fan, D. Krishnan, P. Isola, D. Katabi, and Y. Tian, "Improving clip training with language rewrites," *Advances in Neural Information Processing Systems*, vol. 36, pp. 35 544–35 575, 2023.

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[39] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li *et al.*, "Regionclip: Region-based language-image pretraining," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 793–16 803.

[40] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5356–5364.

[41] J. Li, G. Shakhnarovich, and R. A. Yeh, "Adapting clip for phrase localization without further training," *arXiv preprint arXiv:2204.03647*, 2022.

[42] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2641–2649.

[43] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, pp. 32–73, 2017.

[44] A. Sadhu, K. Chen, and R. Nevatia, "Zero-shot grounding of objects from natural language queries," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4694–4703.

[45] B. Zhang, P. Zhang, X. Dong, Y. Zang, and J. Wang, "Long-clip: Unlocking the long-text capability of clip," in *European Conference on Computer Vision*. Springer, 2024, pp. 310–325.

[46] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[47] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.

[48] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[49] ——, "Conditional prompt learning for vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 816–16 825.

[50] G. Sharma and D. Sharma, "Automatic text summarization methods: A comprehensive review," *SN Computer Science*, vol. 4, no. 1, p. 33, 2022.

[51] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[52] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, Tech. Rep., 2018, technical Report. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[53] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey," *Multimedia tools and applications*, vol. 78, pp. 15 169–15 211, 2019.

[54] V. Nechakhin, J. D'Souza, and S. Eger, "Evaluating large language models for structured science summarization in the open research knowledge graph," *Information*, vol. 15, no. 6, p. 328, 2024.

[55] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 1995, pp. 68–73.

[56] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[57] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[58] J. Lafferty, A. McCallum, F. Pereira *et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Icml*, vol. 1, no. 2. Williamstown, MA, 2001, p. 3.

[59] W. Yin and Y. Pei, "Optimizing sentence modeling and selection for document summarization." in *IJCAI*, vol. 15, 2015, pp. 1383–1389.

[60] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," *arXiv preprint arXiv:1603.07252*, 2016.

[61] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[62] G. Sharma, S. Gupta, and D. Sharma, "Extractive text summarization using feature-based unsupervised rbm method," in *Cyber Security, Privacy and Networking: Proceedings of ICSPN 2021*. Springer, 2022, pp. 105–115.

[63] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[64] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[65] S. Qi, L. Li, Y. Li, J. Jiang, D. Hu, Y. Li, Y. Zhu, Y. Zhou, M. Litvak, and N. Vanetik, "Sapgraph: Structure-aware extractive summarization for scientific papers with heterogeneous graph," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2022, pp. 575–586.

[66] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[67] J. Priem, H. Piwowar, and R. Orr, "Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts," Sep. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6936227

[68] V. Paruchuri, "Marker: Convert pdf to markdown + json quickly with high accuracy," GitHub repository, Mar. 2025, version 1.6.2. [Online]. Available: https://github.com/VikParuchuri/marker

[69] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[70] T. pandas development team, "pandas-dev/pandas: Pandas," python library. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

[71] OpenAI, "CLIP ViT-B/32 pretrained weights," Hugging Face Hub, https://huggingface.co/openai/clip-vit-base-patch32, commit `<hash>`, accessed 7 May 2025.

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[73] OurResearch. (2024) Openalex documentation — location object (`best_oa_location` and `is_oa` fields). Accessed: 07 May 2025. [Online]. Available: https://docs.openalex.org/api-entities/works/work-object/location-object

[74] United Kingdom, "Copyright, designs and patents act 1988, section 29a: Copies for text and data analysis for non-commercial research," https://www.legislation.gov.uk/ukpga/1988/48/section/29A, 2014, inserted by the Copyright and Rights in Performances (Research, Education, Libraries and Archives) Regulations 2014; accessed 07 May 2025.

[75] United States District Court, Southern District of New York, "Elsevier inc. *et al.* v. sci-hub *et al.*, no. 1:15–cv–04282 (S.D.N.Y.) : Final default judgment and permanent injunction," https://cip2.gmu.edu/wp-content/uploads/sites/31/2017/06/Elsevier-v-Sci-Hub-Judgment.pdf, 2017, filed 21 June 2017; accessed 07 May 2025.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[77] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[78] D. M. Christopher, R. Prabhakar, and S. Hinrich, "Introduction to information retrieval," 2008.

[79] R. Snow, B. O'connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks," in *Proceedings of the 2008 conference on empirical methods in natural language processing*, 2008, pp. 254–263.

[80] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," *arXiv preprint arXiv:2103.14749*, 2021.

[81] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021.

[82] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.