Enabling Perspective-Aware Ai with Contextual Scene Graph Generation

information

Article

Enabling Perspective-Aware Ai with Contextual Scene Graph Generation

Daniel Platnick [1,*,†], Marjan Alirezaie [1,*,†] and Hossein Rahnama [2]

[1] Flybits Labs, Creative School, Toronto Metropolitan University, Toronto, ON M5B 2G9, Canada
[2] MIT Media Lab, Cambridge, MA 02139-4307, USA; rahnama@mit.edu
* Correspondence: daniel.platnick@flybits.com (D.P.); marjan.alirezaie@flybits.com (M.A.)
† These authors contributed equally to this work.

Abstract: This paper advances contextual image understanding within perspective-aware Ai (PAi), an emerging paradigm in human–computer interaction that enables users to perceive and interact through each other's perspectives. While PAi relies on multimodal data—such as text, audio, and images—challenges in data collection, alignment, and privacy have led us to focus on enabling the contextual understanding of images. To achieve this, we developed perspective-aware scene graph generation with LLM post-processing (PASGG-LM). This framework extends traditional scene graph generation (SGG) by incorporating large language models (LLMs) to enhance contextual understanding. PASGG-LM integrates classical scene graph outputs with LLM post-processing to infer richer contextual information, such as emotions, activities, and social contexts. To test PASGG-LM, we introduce the context-aware scene graph generation task, where the goal is to generate a context-aware situation graph describing the input image. We evaluated PASGG-LM pipelines using state-of-the-art SGG models, including Motifs, Motifs-TDE, and RelTR, and showed that fine-tuning LLMs, particularly GPT-4o-mini and Llama-3.1-8B, improves performance in terms of R@K, mR@K, and mAP. Our method is capable of generating scene graphs that capture complex contextual aspects, advancing human–machine interaction by enhancing the representation of diverse perspectives. Future directions include refining contextual scene graph models and expanding multi-modal data integration for PAi applications in domains such as healthcare, education, and social robotics.

Keywords: perspective-aware Ai; human–computer interaction; contextual scene graph generation; LLM; digital identities

Citation: Platnick, D.; Alirezaie, M.; Rahnama, H. Enabling Perspective-Aware Ai with Contextual Scene Graph Generation. *Information* **2024**, *15*, 766. https://doi.org/10.3390/info15120766

Academic Editors: Arcangelo Merla, Daniela Cardone, Alessia Amelio and David Perpetuini

Received: 30 October 2024
Revised: 16 November 2024
Accepted: 19 November 2024
Published: 2 December 2024

1. Introduction

Perspective-aware Ai (PAi) is a novel approach to human–AI interaction that extends beyond traditional personalization, allowing users to perceive, understand, and interact with the world from another person's perspective. This capability supports the creation of exchangeable identities or borrowable mental models, which simulate the cognitive and behavioural essence of individuals who are modelled [1]. Unlike classic AI models, which primarily focus on personalization through user-specific analytics, PAi aims to learn and model human personality, thus enabling a wide range of applications in social, professional, and collaborative environments [2,3]. PAi's impact goes well beyond adapting to individual users. By creating digital models that capture diverse perspectives, PAi has the potential to bring transformative changes to fields like healthcare, legal services, education, and decentralized recommendations. For example, professionals could leverage PAi to share expertise through digital avatars that capture their unique knowledge and values, simplifying knowledge transfer and reducing costs. Additionally, by enabling people to view information from diverse perspectives, PAi promotes inclusivity and empathy, which are essential elements for reducing bias and enhancing societal well-being.

To achieve PAi, we utilize a two-phase neuro-symbolic process bridged through a reason-ready structure, which we refer to as a *chronicle*. Chronicles are graph-based

models designed to represent an individual's perspective, constructed from temporal and situational data. In PAi, the process begins with a learning phase that analyzes an individual's digital footprints—including images, text, and social media interactions—to build a formal, reason-ready graph structure. This graph aims to encapsulate the individual's mental model or cognitive character while remaining organized for secure querying by others with appropriate access. By providing structured access, insights from the model can be shared responsibly, supporting more informed and inclusive decision-making processes (see Figure 1).

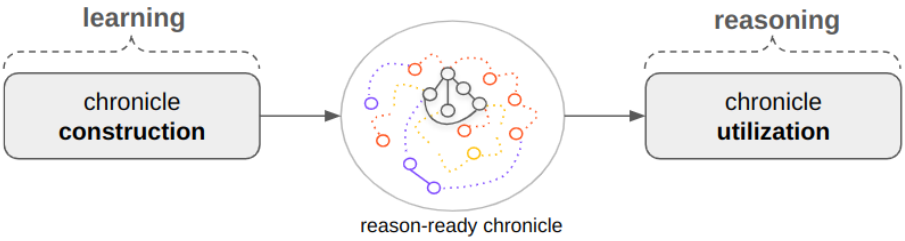

Figure 1. Chronicle pipeline consisting of two phases: (1) construction of the chronicle as a reason-ready structure, and (2) utilization, where the chronicle is communicated and queried by users to share the individual's captured perspective. The colouring indicates new sub-graphs being extended in the chronicle throughout the chronological structural learning process.

Developing PAi presents significant challenges, particularly in data collection and privacy. Building accurate and reliable chronicles requires vast amounts of personal data in various formats, such as images, text, and audio. Since the data originate directly from (source) users, ensuring robust privacy and security measures is essential to address risks effectively. People are often reluctant to share personal information due to concerns about potential misuse, unauthorized access, and the loss of control over their data. In addition to privacy concerns, the sheer volume and variety of data required adds another layer of complexity. To create a comprehensive chronicle, data from various formats must be collected within specific time frames to reflect a person's full experience. This is challenging, however, as some data may be hard to obtain or difficult to align consistently across formats. Even when the data are accessible, training models to process multiple formats can be expensive in terms of time, effort, and resources. These challenges make developing PAi systems difficult and highlight the need for strong data management strategies, user consent, secure handling, and efficient processing of multi-modal data.

To mitigate these complexities, we simplify the process by focusing solely on image data and leveraging state-of-the-art (SOA) methods in SGG. Our contributions are three-fold. First, we evaluate existing state-of-the-art SGG models to assess their suitability for PAi. This evaluation offers insights into current capabilities, identifies limitations in achieving PAi, and examines the feasibility of extending the learning process to other data types. Second, we propose PASGG-LM, which extends state-of-the-art SGG models by integrating LLMs to enhance contextual understanding, thereby improving their applicability to the downstream task of representing individual perspectives within the chronicles used in PAi. PASGG-LM addresses limitations in existing SGG models by capturing nuanced contextual and ambient relational data, providing a more accurate and less biased solution for PAi. Third, to support this endeavour, we developed a PAi-compatible SGG dataset centred around a single individual. This dataset offers a consistent and controlled basis for evaluating SGG model performance in PAi and showcases our approach's potential for broader data modalities in future research. Our work represents a step toward building a robust, privacy-preserving PAi framework that can significantly enhance human–machine interactions.

This paper is structured as follows: in the next section, we review the related works and background of the models utilized in this study. In Section 3, we outline the structure and purpose of the situation graph (SG) for capturing users' chronological experiences, formalize the task of scene graph generation, and introduce the new task of context-

aware scene graph generation. In Section 4, we describe the complexities associated with gathering PAi data and outline the VG benchmark SGG dataset and our PAi benchmark. Section 5 details our approach to evaluating classical SGG models for PAi and describes the proposed PASGG-LM model. In Section 6, we present and analyze the results, followed by a discussion and conclusion in the final sections.

2. Related Work

In this section, we provide an overview of three related areas of work: PAi for the construction and utilization of digital identity models, scene graph generation, and relevant state-of-the-art SGG models, as well as large language models (LLMs) and their capabilities in post-processing. We then discuss extending the PAi architecture with other models capable of interpreting specific types of data, further enhancing its versatility and robustness in multimodal tasks.

2.1. PAi for the Construction and Utilization of Digital Identities

Perspective-aware AI (PAi) is an emerging computational innovation where users of a trusted system can view and interact with each other's points of view without requiring a centralized recommendation system [4]. One primary goal of PAi is to construct *chronicles*: a query-able graph-based digital identity model that reflects a source user's cognitive map, allowing it to act on their behalf, simulating their feelings and responses in a given situation. This can be achieved through a neuro-symbolic structure learning approach that learns to build a *chronicle* of an individual from data contained in their digital footprint [5]. *Chronicle* construction requires extensive personal data collection to accurately model a source user's mental framework across diverse scenarios, allowing for realistic simulations of potential responses. Modelling a source user's cognitive character across a wide range of situations presents a significant level of complexity. Therefore, PAi can best be achieved by incorporating many modalities into the training data, offering the model a comprehensive representation of what the user has seen, heard, thought, or felt, along with the evolutions around these experiences. Furthermore, the dataset gathered for learning a *chronicle* should include a varied distribution of user situations to allow a rich and unbiased depiction of the user's mental map.

PAi has the potential to re-invent entire media landscapes that are fuelled by our growing dependency on privatized and centralized recommendation systems. Some problems caused by bias in current recommendation systems include increased polarization, erosion of public knowledge, reduced autonomy, and privacy leakages [6–8]. As the state of our media landscape progresses toward ubiquitous integration with personalized recommendations, it is crucial to design systems that mitigate bias and the problems associated with it. Leveraging *chronicles* trained on holistic digital footprints that encompass the cognitive essence of a source user, PAi has the potential to contribute to the design of less-biased recommendation systems. This can enhance transparency, inclusivity, and clarity in critical decision-making across various domains, including healthcare, education, and business.

Current recommendation systems, owned and operated by large tech corporations, rely primarily on data gathered from user interactions within their software products. However, accurately modelling a user's identity requires richer data than what can be gathered from these limited interactions. These limitations lead to commercial identity models that are fragmented, failing to offer a holistic picture of the user, resulting in system bias. Furthermore, the centralized and capitalistic nature of the current media landscape leads businesses to focus on maximizing user attention. This metric has contributed to a growing digital economy of recommendation applications that encourage users to endlessly swipe through arbitrary content [9]. The competition inherent in capitalism, along with the goal of maximizing user attention, leads companies to exploit their users through emotional manipulation or fear tactics [10]. PAi has the potential to remedy all of this through holistic

chronicles that promote inclusivity and the consideration of more diverse viewpoints, along with data decentralization and privacy preservation.

In particular, as described in [5], *chronicles* are a multi-modal and knowledge-aware solution for decentralized digital identity modelling. They reduce bias by providing control of the knowledge injected into them. Furthermore, performing inference with a *chronicle* can produce a trace of the reasoning that led to the inference, thereby offering greater transparency. With its vast applicability, PAi can lead to increased transparency and reduced bias of recommendation systems in many areas such as peer-to-peer learning, Digital Immortality, and, in general, decentralized computational social systems.

2.2. Scene Graph Generation from Images

Scene graph generation (SGG) is a vision understanding task aimed at predicting a graphical representation of an image. The goal of SGG is to capture entities and their relationships in the scene as (*subject*, *predicate*, *object*) triplets, such that the resulting scene graph can support downstream graph-reasoning tasks [11]. This formal representation encodes semantic information about the scene and supports a variety of tasks such as image-retrieval [12], visual question-answering (VQA) [13], image captioning [14], and 3D scene synthesis [15]. Moreover, SGG can significantly enhance the capabilities of current reasoning systems by enabling automated reasoning on generated scene graphs and leveraging the rich semantic information embedded within images.

At present, SGG is performed using pipelines that fall into two categories: two-stage and end-to-end approaches [16]. In two-stage approaches, SGG is split into the two following processes: object localization and classification, and relationship detection. In the first stage, objects in the image are detected and classified using a backbone convolutional neural network (CNN) architecture, such as Faster-RCNN [17–19]. The second stage then uses a separate network to predict relationships between the objects localized by the CNN backbone, finally outputting a scene graph in the form of relational triplets. This setup requires separate training procedures for both the backbone and relation network to tune the final model, given a distribution of images labelled with object annotations and ground truth scene graphs. Apart from two-stage approaches, end-to-end pipelines for SGG have also been proposed [16,20]. One-stage approaches often have advantages over two-stage approaches in terms of speed, cost, simplicity, and not exhibiting error propagation between modules [21]. Historically, two-stage approaches have dominated the SGG community; however, due to recent advancements in one-stage approaches, one-stage methods are increasing in popularity [22].

To capture situations the user has been experiencing through image footprints, this paper explores three well-known benchmark SGG models using both one-stage and two-stage approaches.

The first is the Neural Motifs (Motifs) model, a two-stage benchmark SGG model that attempts to predict the relationship occurring most frequently between object pairs as observed in the training data [19]. By incentivizing the model to incorporate global context and focus on structural patterns in the data, the Motifs model achieved state-of-the-art results in SGG on the Visual Genome benchmark dataset [23]. The Neural Motifs model serves as a strong baseline for PAi experiments, given its widespread use within the SGG community.

The second, Relation Transformer (RelTR), introduced by [20], is a one-stage model capable of directly detecting triplet proposals and simultaneously predicting entities and their predicate relations. As RelTR can generate scene graphs using only visual appearance, the model results in fewer parameters used compared to classical two-stage methods. Despite this, RelTR boasts state-of-the-art performance on benchmark datasets, marking a significant improvement in end-to-end scene graph generation.

Finally, in 2020, ref. [24] introduced a benchmark SGG framework, which aims to remove bias in SGG through counterfactual causality by calculating and using the total direct effect (TDE) as a final predicate score. TDE achieves significant improvements over

previous state-of-the-art methods and is used in post-processing after retrieving the output from the trained SGG model to remove the "bad bias", leading to richer, more varied outputs. The SGG benchmark framework is provided in a comprehensive, widely used codebase that supports experimentation with a variety of SGG benchmark models. As the third benchmark model, we tested Neural Motifs with TDE. Our experiments in this study were implemented on top of the codebase proposed by [24].

In the context of PAi, extracting a digital chronicle that captures the mental model of a given user through structure learning on integrated chronological and multimodal data streams is a complex task. As the first step to simplify this endeavour, we remove the temporal constraint and focus solely on image data, allowing us to address a fundamental challenge in constructing chronicles within PAi. Even with this simplification, generating contextual scene graphs from images to support PAi applications remains challenging, as traditional SGG models are designed to capture the physical and spatial relationships of a scene but fail to extract the contextual and perspective-aware information necessary for PAi.

Bridging the gap between SGG and PAi, we introduce *context-aware SGG*, an extended form of traditional SGG. Unlike conventional SGG, which only predicts spatial and physical elements of a scene, context-aware SGG predicts contextual scene information from the perspective of the user. This advancement is a crucial step toward achieving PAi, as it addresses a key challenge.

2.3. Large-Language Models for Post-Processing

The ubiquitous capabilities of LLMs have sparked a surge of research, where LLMs are employed for post-processing between modules or at the end of a pipeline to enhance overall performance [25–27]. Training on vast amounts of textual data, state-of-the-art LLMs implicitly learn to encode rich representations of concepts from the training set, enabling them to perform complex reasoning tasks [28,29]. Graphs can naturally be represented as text, making them highly compatible with LLMs. Consequently, many researchers have explored using LLMs for graph-based tasks, such as inferring relations between entities, finding the shortest path, and inductive reasoning [30–33]. Research suggests that neural-based systems like LLMs can significantly benefit from the structured representations offered by graphs, leading to robust neuro-symbolic pipelines that address the limitations of both neural and symbolic approaches [1].

Wang et al. used LLMs to perform post-processing on speaker diarization systems, which can improve their readability and reduce their error rate [25]. Ref. [27] explored leveraging knowledge stored in LLMs to enhance graph node attributes and operate as standalone predictors. Wang et al. used LLMs to enhance knowledge graph inductive reasoning, where the task is to infer missing facts from KGs that have not been seen before [33]. They achieve this by having the LLM generate a graph structural prompt which improves the output of pre-trained graph neural networks (GNNs), allowing them to outperform the compared baselines in zero-shot, one-shot, and three-shot reasoning tasks.

Building on previous research, our work explores using LLMs to enhance SGG for PAi. Specifically, we investigate LLMs for inferring nodes and edge connections representing various contextual aspects of a scene during post-processing, thereby extending the spectrum of traditional SGG toward more context-aware vision applications such as PAi.

2.4. Enhancing PAi Architecture

In this paper, we acknowledge that while the ultimate goal of PAi is to handle multimodal data, we are currently focusing on image data alone to simplify the approach. In the future, as the PAi architecture expands, it will feature specific sub-systems designed to process and extract semantic information from a diverse range of data types. For example, a dedicated sub-system could handle sound data, leveraging techniques like spectrogram analysis and acoustic modelling to interpret audio signals [34]. This is particularly useful for tasks such as speech recognition, where understanding the nuances of tone and accent

can improve personalized communication, or for emotional state detection, which can offer insights into an individual's well-being. Video data could be processed using temporal convolutional networks or Transformers, which excel at capturing both spatial and temporal features [35]. Video data are critical for tasks like action recognition, where identifying physical gestures or behaviours can help build more dynamic and context-aware models of individuals, such as in security monitoring or personalized fitness coaching. Transactional data, often found in domains like finance and e-commerce, could be processed with graph-based models or recurrent neural networks, which can capture sequential dependencies and relational patterns [36]. Understanding an individual's purchasing behaviour helps prepare recommendations or predict future actions, making the system more adjusted to personal preferences and financial decisions. Additionally, the Kolmogorov–Arnold Network (KAN), which is adept at handling data with complex non-linear relationships, could be employed for specialized tasks like modelling intricate dependencies in sensor data or predicting outcomes in systems with highly interdependent variables [37]. For example, KAN could be useful in predicting individual health outcomes based on various intertwined factors, such as lifestyle, genetic data, and environmental variables, offering a more holistic view of an individual's health trajectory. Given the strength in modelling different types of data with complex dependencies, these sub-models could be integrated as components of PAi's modular architecture to enable a more comprehensive multimodal approach.

3. Preliminaries

In this section, we describe the structure and purpose of the situation graph, which serves as the foundational ontology for capturing chronological, perspective-aware experiences. We will then formalize the task of scene graph generation, and introduce our new task of context-aware scene graph generation.

3.1. Situation Graph

The goal of PAi, as previously mentioned, is to extract a representation of the situations an individual is experiencing from their available digital footprint. Before going through the details of developing PAi, we need to clarify what we mean by situation representation. The situation is represented as a graph, referred to as a situation graph (SG), whose structure is based on the DOLCE Ultralite (DUL) ontology, serving as the foundation ontology model [38].

Situation graphs (SGs) are a unified and fundamental structure in PAi that formally represents a given situation and the mental model of the source user being modelled. To facilitate the formal representation of an individual's digital identity across arbitrary situations based on learning from diverse data streams, we employ the SG structure, as previously described in [5]. The SG structure provides a template for the definition of an arbitrary situation (from the perspective of the source user), adhering to a pre-specified structure. SGs model aspects of an arbitrary situation such as its time, location, ambience, people, activity, emotion, weather, and social context. By encoding the most encompassing aspects of a situation in the form of a graph with entities and relations, we can leverage graph and structure learning to create robust node embeddings which can be used for downstream reasoning tasks. An example of a situation graph is shown in Figure 2.

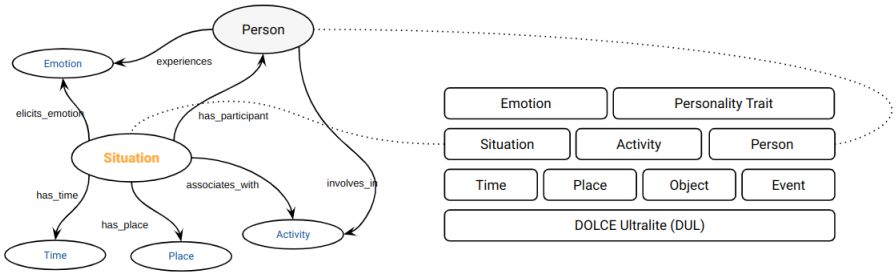

Figure 2. Representation of the situation graph derived from the DOLCE Ultralite (DUL) ontology.

SGs provide a consistent and unified graph structure to represent arbitrary situations in terms of their physical and contextual elements from the perspective of the source user being modelled, whom we refer to as the *main participant*. A situation graph (SG) is defined as $SG(V, E)$, where V corresponds to nodes in the graph which represent entities such as the given situation, activities within the situation, and participants of the situation, and E represents the set of relationships or edges connecting nodes in the graph. Relationships represented by edges between nodes in the SG structure can be spatial, such as ''`performing activity`'', as well as contextual, for instance, ''`has weather`'', ''`has emotion`'', or ''`has ambience`''. By combining entity and relation pairs, SGs are capable of robustly modelling intricate situations comprised of explicit and implicit physical, spatial, and contextual elements.

3.2. Scene Graph Generation

Scene graph generation aims to provide a structured and comprehensive representation of the semantic content within an image by identifying objects and their relationships. Formally, a scene graph G is defined as $G = (B, O, R)$, where we have the following:

- Bounding boxes $B = \{b_i \mid i = 1, \ldots, N\}$ are a set of N bounding boxes, with each $b_i \in \mathbb{R}^4$ representing the spatial location and size of an object within the image.
- Object classes $O = \{o_i \mid i = 1, \ldots, N\}$ are the corresponding set of object class labels, where each $o_i \in \mathcal{C}$ and $\mathcal{C}$ is the predefined set of object categories.
- Relationships $R = \{(i, r_{ij}, j) \mid i, j = 1, \ldots, N; i \neq j\}$ are a set of directed edges representing pairwise relationships between objects, where $r_{ij} \in \mathcal{R}$ and $\mathcal{R}$ is the set of predicate types in the training distribution (e.g., ''`above`'', ''`next to`'').

Given an input image I, the goal of SGG is to predict the most probable scene graph G that accurately reflects the objects and their relationships in I. This involves estimating the joint probability distribution:

$$P(G \mid I) = P(B, O, R \mid I).$$

In traditional SGG, each relationship between objects is represented by a relational triplet (s, p, o), where subject s represents the first object in the relationship (e.g., ''`person`''), predicate p indicates the type of relationship (e.g., ''`sitting on`''), and object o is the second object involved in the relationship (e.g., ''`motorcycle`''). Relational triplets form the edges in the scene graph G, linking object nodes with predicate relationships that define their spatial configurations and physical interactions within the image.

3.3. Context-Aware Scene Graph Generation

While traditional SGG focuses on physical and spatial elements within an image, context-aware scene graph generation extends this framework to include ambient and contextual aspects by leveraging the situation graph (SG) structure described in Section 3.1. Modelling an SG from an image requires predicting not only objects and their spatial relationships, but also incorporating contextual, ambience-based, and perspective-aware information such as participant identities, activities, emotions, weather conditions, and locations.

Formally, a context-aware scene graph SG' is an augmented version of the traditional scene graph, defined as $SG' = (B', O', R')$, where we have the following:

- The set of extended bounding boxes $B' = \{b'_i \mid i = 1, \ldots, N'\}$ includes bounding boxes representing both physical objects and contextual regions within the image. Each $b'_i \in \mathbb{R}^4$ may encompass areas associated with contextual or ambient information, such as emotions, scene locations, or environmental conditions (e.g., a region representing a sunset or rainy weather).
- Extended object classes $O' = \{o'_i \mid i = 1, \ldots, N'\}$ are the set of object and context class labels, where $o'_i \in \mathcal{C}' = \mathcal{C} \cup \mathcal{C}_{\text{context}}$. Here, $\mathcal{C}_{\text{context}}$ includes new classes representing contextual elements such as *situations* (`party`, `meeting`), *emotions* (`happy`, `sad`), *weather conditions* (`sunny`, `rainy`), and *locations* (`beach`, `city`).

- The set of extended relationships $R' = \{(i, r'_{ij}, j) \mid i, j = 1, \ldots, N'; i \neq j\}$ includes relationships capturing both physical interactions and contextual associations, where $r'_{ij} \in \mathcal{R}' = \mathcal{R} \cup \mathcal{R}_{\text{context}}$. The set $\mathcal{R}_{\text{context}}$ contains predicates that express ambient or perspective-based relationships (e.g., `has ambience`, `has location`, `has emotion`).

The goal of context-aware SGG is to predict the most probable situation graph SG' given the image I:

$$P(SG' \mid I) = P(B', O', R' \mid I).$$

In context-aware SGG, relational triplets are extended to capture both physical and contextual information, represented as (s, p, e), where we have the following: subject s represents the first object or context in the relationship, predicate p indicates the relationship type, which can be spatial, interactive, or contextual, and entity e denotes either a physical object or a contextual class (e.g., emotion, location). Thus, context-aware SGG models estimate probabilities over these extended sets, enabling the prediction of context-aware scene graphs that encompass both explicit and implicit information from the image. This extension allows each triplet to capture both object-to-object and object-to-context relationships, such as the following:

- Physical interactions e.g., (`person, sitting on, bench`).
- Ambient associations e.g., (`person, has emotion, happy`).
- Situational contexts e.g., (`situation, has location, party`).

3.4. Key Differences Between SGG and Context-Aware SGG

The key differences between classical SGG and context-aware SGG are as follows:

1. **Incorporation of contextual elements**: Context-aware SGG introduces new object and predicate classes that capture ambient and contextual information, expanding the prediction space beyond physical entities and their spatial relationships.
2. **Bounding boxes for contextual regions**: Unlike classical SGG, where bounding boxes are associated strictly with physical objects, context-aware SGG associates bounding boxes with regions representing contextual aspects of the scene.
3. **Situation graph structure**: Context-aware SGG provides a more holistic understanding of the scene, enabling applications that require deeper semantic reasoning by adhering to the SG structure described in Section 3.1 and incorporating elements such as activities, emotions, weather conditions, and locations.

4. Data

This section outlines and addresses complexities associated with collecting datasets for PAi, the Visual Genome benchmark SGG dataset, and the procedure used to create our PAi SGG dataset (image data labelled with corresponding situation graphs).

4.1. Complexities Associated with PAi Datasets

Constructing a chronicle that accurately models a source user's cognitive map and behaviours across different situations requires gathering extensive personal multimodal data from that individual over time. The chronicle construction pipeline, as outlined by Alirezaie et al. in [5], integrates various streams of chronological multimodal data from the user being modelled, including but not limited to the following:

- Images or videos of the user in various situations participating in different activities.
- Text data in the form of messages or social media posts.
- Songs listened to by the user, or recent phone calls.
- Tabular data such as location information about the user.

Each data stream offers a glimpse through a different window into the behaviours and psyche of the source user. By incorporating diverse data streams, we can create a comprehensive model that reveals how the source user feels in a diverse set of given situations. For instance, texting a friend or posting about a new job reveals information

about the user's employment status and feelings toward it. A photo of the source user performing a certain activity such as surfing or biking can imply their favourite hobbies. Songs a user listens to may be highly correlated with how they are feeling, and events going on in their life. Location data can help ground the data available at a given time step in a real time and place, increasing accuracy and reducing bias in the model. All of these data can be utilized in a comprehensive process of feature extraction and learning to create a rich digital identity model of a given source user.

As described in [5], we can utilize the available data streams at a given time step to generate a situation graph (SG) representing the situation a source user has experienced. For example, the chronicle construction pipeline described in [5] depends on various modes of chronological data such as images, text, and audio, which are used to learn a digital identity model of the source user. In trying to construct such a pipeline, we find that two hurdles naturally arise with the demanding and invasive data collection process surrounding PAi:

1. People are reluctant to share vast amounts of personal data.
2. Annotating and labelling data for PAi is a highly labour-intensive process that requires substantial time and domain expertise.

To make progress despite these challenges, we simplify the problem of chronicle construction by restricting it to a unimodal setting (images) and removing the temporal constraint (no longer considering time steps or chronological ordering in the data). By eliminating the temporal constraint and focusing solely on images, we approach this sub-task in chronicle construction as a form of *context-aware scene graph generation*.

4.2. Visual Genome Benchmark Dataset

In 2017, ref. [23] introduced the Visual Genome (VG) dataset, which has since become the most notable and widely accepted benchmark in the SGG community. It contains over 100 K images with an average of 21 objects and 18 pairwise relationships per image. The VG dataset supports 150 object classes and 50 predicate classes for SGG. Object classes are based on physical entities, and some examples include `airplane`, `woman`, `man`, `person`, `book`, `boat`, `bird`, `face`, and `motorcycle`. Predicate classes are used to represent spatial and physical relations between object entities in the form of relational (*subject*, *predicate*, *object*) triplets. Some examples of predicate classes include `between`, `carrying`, `hanging from`, `in front of`, and `lying on`. The diversity of object and predicate classes in the VG dataset enables SGG pipelines to model relationships across a wide range of everyday concepts effectively. Consequently, SGG models pre-trained on VG learn rich representations that transfer well to fine-tuning or downstream tasks.

The VG dataset is crucial to our study, as many state-of-the-art SGG frameworks are trained and evaluated on it [11,18–20,24]. We believe that, due to the notoriety and traction of VG as a benchmark dataset in the SGG community, there is significant utility in creating new SGG datasets that follow a similar format. By creating new SGG datasets structured similarly to the VG, we can ensure our data remains compatible with the bulk of the state-of-the-art SGG models. Therefore, we format our new PAi SGG dataset to mimic the format of VG, modifying it only by adding new contextual objects and predicate classes to support PAi. We find this strategy leads to simpler integration when testing different SGG models.

4.3. A Scene Graph Generation Dataset for PAi

We created a PAi SGG dataset that contains 112 images and corresponding situation graphs from a single individual (the main participant) across a diverse set of situations. The images were taken from the main participant's iPhone camera roll from 2015 to 2024. The 112 situations vary in terms of time, location, weather, ambience, attendees, activities performed, emotional status of the main participant and others, and social context. The images of situations were collected from a wide variety of experiences involving professional, recreational, casual, and formal settings. Some examples of situations in the PAi

dataset include hiking, holiday and birthday party celebrations, performing activities such as biking or surfing, working in an office, and eating food with friends.

Related to each image, we create and provide corresponding situation graph labels (see Section 3.1) as a list of relational triplets, capturing the mental model of the situation from the perspective of the main participant. The relational triplets contain objects and predicates based on the situation graph structure. To advance PAi, we extend traditional SGG to be more context-aware by adding contextual, ambience-based, and perspective-aware object and predicate classes, in addition to detecting the traditional spatial and physical elements of SGG. In particular, we add new object and predicate classes to the Visual Genome dataset, which facilitates the representation of situations, activities, situation-level and perspective-level emotional states, weather states, locations, etc. These new object and predicate classes expand the prediction space of SGG models beyond purely physical and spatial elements, enabling the incorporation of contextual and implicit ambient information from the scene.

In total, we add 130 new contextual object classes and 12 predicate classes to the ones present in Visual Genome, discussed as follows. The new classes facilitate the representation of contextual entities in the scene, such as the situation itself, different participants, emotions, times of day, social contexts, types of weather, locations, and activities. When combined with our new predicate classes, our proposed context-aware SGG pipeline permits the freedom to express a variety of new relationships with relational triplets. Some examples of new contextual expressions representable by our proposed pipeline include: (`situation, has social context, professional`), (`main participant, activity, laptop`), and (`main participant, has emotion, focused`). Figure 3 shows the distribution of the top 30 most frequent objects appearing in scene graphs of the PAi SGG dataset. A complete list of added PAi object classes can be found in our codebase.

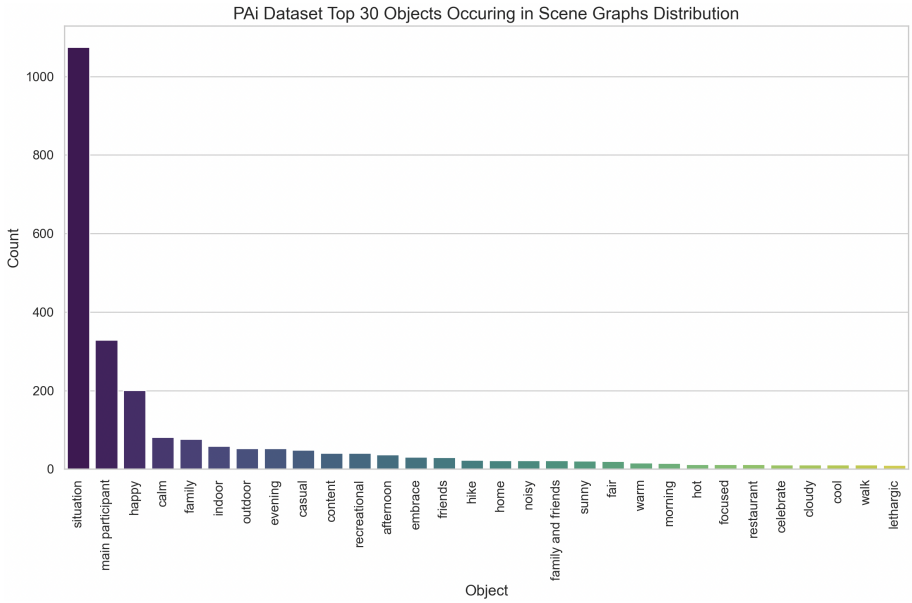

Figure 3. Distribution of the top 30 most frequent objects occurring in scene graphs of the PAi SGG dataset.

The PAi SGG dataset consists of 112 images with a balanced representation of location types, including 59 indoor and 53 outdoor scenes. These images span various times of day, with 15 taken in the morning, 37 in the afternoon, 53 in the evening, and 7 at night, offering temporal diversity. Of these, 9 images are selfies, which provide additional contextual variety for perspective-aware analysis. The dataset encompasses a broad range of locations, with 22 images set in home environments, 12 in restaurants, 7 on boats, 6 on streets, 5 on hills, 4 near lakes, and 4 in arenas or basements. Additionally, three images each capture

scenes in forests, on beaches, in banks, and on driveways, with several other unique locations represented by one or two images each. Weather conditions in the dataset include 21 sunny, 20 fair, 11 cloudy, 3 rainy scenes, and 1 snowy scene, ensuring environmental variability. This diversity in location, time, and weather enriches the PAi dataset's ability to support *context-aware SGG* across a wide range of situational factors.

Our PAi SGG dataset contains 12 new predicate classes that support the representation of diverse contextual relationships between entities, enabling SGG models trained on it to predict not only spatial and physical attributes but also new contextual information. Figure 4 displays the overall distribution of predicates as they appear in the 112 situation graphs contained in the PAi SGG dataset. Our new predicate classes enable the following new types of relations in SGG: `has time`, `has participant`, `has emotion`, `overall emotion`, `has location`, `location type`, `has social context`, `has ambience`, `activity`, `overall activity`, `has weather`, and `temperature`. These types of predicate relations can be combined with objects using relational triplets to express a range of new contextual aspects of a scene.

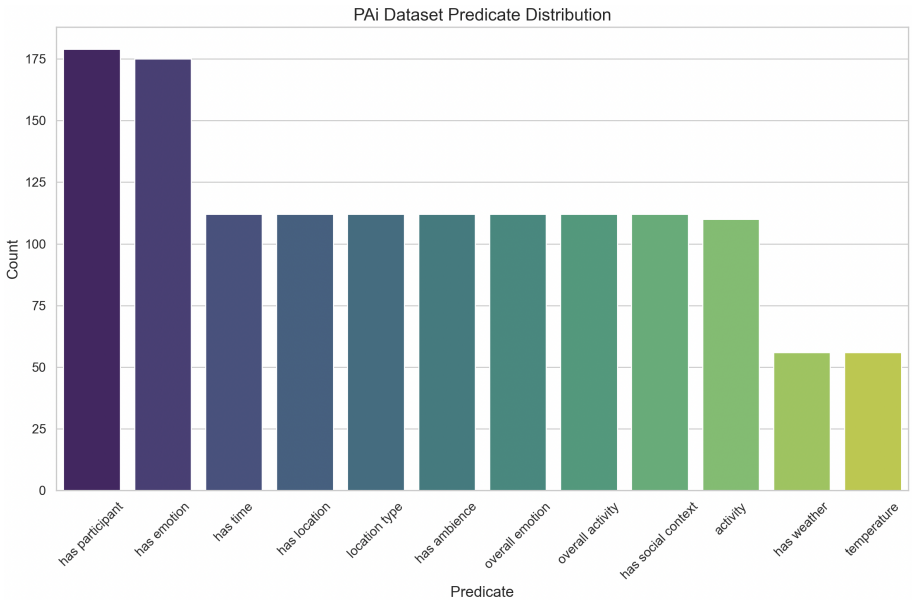

Figure 4. Distribution of the 12 new PAi predicates as they appear in scene graphs of the PAi dataset.

Since the Visual Genome is the most widely accepted benchmark for SGG, we evaluate SGG models for PAi that have previously been trained on the benchmark. Fine-tuning models pre-trained on Visual Genome for our PAi SGG dataset requires carefully designing and manually labelling new types of object boxes that capture emotional and ambient features, in addition to spatial and physical object boxes. Due to the time constraints and labour-intensive nature of this annotation process, we are still working on creating object box annotations and will leave fine-tuning SGG models on the PAi SGG dataset for future work.

4.4. Challenges with Multimodal Data Alignment in PAi

Aligning multimodal data for our proposed PASGG-LM pipeline (described in Section 5.3) presents several challenges, particularly in addressing various biases that can arise during the annotation process. Inductive personal and subjective biases often influence the labelling process, as labellers may have different ideas of what constitutes a happy, anxious, or casual feeling, compared to the ground truth feelings of the participant in the image. Apart from label annotation bias, there can be implicit bias related to the participants of the scene. Sometimes people put on a different face for photos compared to how they truly feel. For example, people may smile but feel bored, sad, or anxious on the inside. These types of inductive biases are inherent to the labelling process and raise

significant challenges for designing unbiased systems capable of robust contextual understanding. We can mitigate this bias by incorporating different multimodal data sources corresponding to the main participant in the scene. For example, supplementing PAi SGG datasets with social media activity data of the participants leading up to the image could add additional context and allow future PAi models to better predict participant-specific emotions in the scene.

On another note, the alignment of PAi images with corresponding textual scene graphs is highly labour-intensive. This alignment requires careful annotation to ensure each image is accurately represented with a contextual situation graph. Leveraging prompt engineering with chain-of-thought reasoning, we can greatly reduce the size of the dataset required to fine-tune the LLMs in PASGG-LM pipelines. This is because prompt engineering and chain-of-thought reasoning are standalone methods to improve LLM performance, thereby reducing the model's dependence on a large fine-tuning dataset.

5. Methods

In this section, we describe our methods and experimental setups used throughout the study.

5.1. Evaluating Classical Scene Graph Models for PAi

We introduce a new metric called PAi similarity score (PSS), which offers a crucial and effective way to empirically measure the performance of benchmark SGG models in context-aware applications, in contrast to traditional metrics like R@K, mR@K, and mAP. As mentioned previously, current SGG methods lack the ability to capture the contextual nuances required for *context-aware SGG*. Our new PAi SGG labels include predicate and object classes that lie outside the distributions on which existing SGG models are trained. Thus, traditional methods used to measure the performance of SGG algorithms will not be as effective in the context of PAi. The SGG models we evaluate (those pre-trained on VG), are not designed to predict the contextual triplet relations found in our PAi SGG dataset described in Section 4.3. As a result, when evaluated on the PAi dataset using traditional metrics, these models score zero, highlighting the inadequacy of conventional metrics to capture their context-aware performance.

To fairly evaluate the performance of existing state-of-the-art SGG models for *context-aware SGG* in PAi, we propose a new metric, called the PAi similarity score (PSS). We use PSS to empirically rank state-of-the-art SGG models on their applicability for PAi. The PAi similarity score (PSS) serves as a metric to empirically assess state-of-the-art scene graph models in the scope of *context-aware SGG*, a key step toward the broader development of PAi. PSS is highly effective for evaluating benchmark SGG models, as it leverages semantic similarity, a widely adopted measure in AI and NLP research, to assess context-related performance accurately. Effectively scoring models on our PAi SGG dataset requires crafting an objective function aimed at inferring contextual, ambient, and participant-specific information, as performed by PASGG-LM. Therefore, traditional metrics remain suitable for evaluating PASGG-LM, given that it is fine-tuned and uses prompt engineering for context-aware SGG on the PAi SGG data distribution.

The PAi similarity score is calculated as follows:

Let Y denote the list of N ground truth relation (i.e., triplets in the PAi SGG dataset), and $\hat{Y}$ denote the set of predicted scene graphs generated by an SGG model, such as Motifs, on a PAi evaluation set. Each element $\hat{y}_i \in \hat{Y}$ and $y_i \in Y$ corresponds to a list of relation triplets for an individual PAi image. For each pair of corresponding predicted and ground truth scene graphs $(\hat{y}_i, y_i)$, as described in [39,40], a tuned LLM, such as GPT-4o [41], is used to convert them into semantically equivalent textual representations, as follows:

$$y_{i,\text{text}} = \text{LLM}(conversion\_prompt, y_i), \quad \hat{y}_{i,\text{text}} = \text{LLM}(conversion\_prompt, \hat{y}_i)$$

Each textual representation, $y_{i,\text{text}}$ and $\hat{y}_{i,\text{text}}$, is then embedded into a vector space using an embedding model:

$$v_{y_i} = \text{Embedding}(y_{i,\text{text}}), \quad v_{\hat{y}_i} = \text{Embedding}(\hat{y}_{i,\text{text}})$$

The semantic similarity for each pair $(\hat{y}_i, y_i)$ is then calculated by computing the cosine similarity (CS) between the corresponding vectors. Finally, the PAi similarity score (PSS) score is the average cosine similarity over total N scene graphs in the evaluation set:

$$\text{PSS}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^{N} \text{CS}(v_{\hat{y}_i}, v_{y_i})$$

This provides an empirical measure of the overall semantic similarity between the predicted scene graphs $\hat{Y}$ and the ground truth scene graphs Y, where values closer to zero correspond to less similarity, and values closer to one indicate greater similarity.

We evaluated Motifs, Motifs-TDE, and RelTR, on the task of context-aware SGG with 112 PAi images, providing insights into their applicability within PAi. The results of evaluating different SGG models for PAi can be found in Section 6.2. The three SGG models were tested using PSS as an empirical metric. In calculating PSS scores, we used GPT-4o to convert scene graphs of relational triplets into an intermediary textual representation. The intermediary representation was then fed into OpenAI's `text-embedding-3-large` embedding model to generate final PSS values. In addition to measuring SGG model scores on the full set of contextual classes present in the PAi SGG dataset, we also remove different variables to see which features in PAi are most and least difficult for the SGG models to handle. In particular, we measured the effects on PSS when simplifying the task by removing different contextual features from the task such as `activity`, `ambience`, `emotion`, `location`, `participant`, `social context`, `temperature`, and `time`. This provides insight into which features should be incorporated into future PAi datasets and which features should be prioritized for improving future context-aware SGG models.

5.2. Perspective-Aware Scene Graph Generation with LLM Post-Processing

To extend the capabilities of state-of-the-art SGG models in capturing more nuanced contextual information for digital identity modelling, we propose a novel SGG algorithm called perspective-aware scene graph generation with LLM post-processing (PASGG-LM) whose steps are captured in Algorithm 1. PASGG-LM helps enrich the output of the current state-of-the-art SGG models by extending their ability to predict contextual and ambient aspects of the scene. Current state-of-the-art models in SGG are trained on VG to capture spatial and physical relationships between elements in a scene. For example, this is part of a scene graph generated by Motifs-TDE, trained on VG evaluated on images from our PAi SGG dataset:

Algorithm 1 Perspective-aware scene graph generation with LLM post-processing.

Inputs: `image`, `rule_based_inference_prompt`, `num_llm_attempts`
Output: Contextual scene graph (modelled after situation graph)

```
scene_graph = generate_scene_graph(SGG_model, image)
text_scene_graph = scene_graph_to_text(scene_graph)
rule_based_prompt = append_scene_graph(rule_based_inference_prompt, text_scene_graph)
for attempt in range(num_llm_attempts) do
    llm_response = perform_inductive_inference(rule_based_prompt)
    if is_valid_response(llm_response) then
        contextual_scene_graph = extract_contextual_scene_graph(llm_response)
        return contextual_scene_graph
    end if
end for
return "No tuples predicted"
```

```
(man, on, surfboard)
(man, wearing, short)
(man, wearing, shirt)
(man, has, hair)
```

In contrast, the ground truth for the same data point in our PAi SGG dataset follows this format:

```
(situation, has location, lake)
(main participant, has emotion, happy)
(main participant, activity, surfboard)
(situation, has weather, sunny)
(situation, temperature, hot)
(situation, has social context, recreational)
```

PASGG-LM leverages prompt engineering and fine-tuning to instruct an LLM to infer aspects of the ground truth PAi SGG data based on the entities output by the benchmark SGG model trained on VG. Our proposed PASGG-LM algorithm effectively bridges the gap between existing SGG methods and the context-aware capabilities required for PAi.

Considering the ground truth PAi SGG data point in the previous example, since there is a man on a surfboard, there is a high probability he is surfing. It follows that if he is surfing and wearing shorts, the location is likely outside and the weather is probably warm or hot. This type of inductive reasoning can be performed by LLMs, bridging the gap between existing SGG methods trained on VG and our desired performance on the PAi SGG dataset. To achieve this, PASGG-LM employs fifteen inductive reasoning rules along with logical constraints through prompt engineering as examples for the model. The set of rules provided in the prompt consists of different rules based on emotion and the scene. In addition to providing hand-crafted inductive rules in the engineered prompt, we instruct the LLM to come up with its own rules based on the input SGG data. We compare the performance of PASGG-LM while including and excluding LLM fine-tuning from the process. For fine-tuning, we train GPT-4o-mini and a quantized Llama 3.1 with 8 billion parameters [42] on PAi SGG labels, further discussed in Section 6.3. The PASGG-LM language model comparison results can be found in Tables 1 and 2.

Table 1. Effects of LLM fine-tuning on overall PASGG-LM pipelines with $K = 11$.

SGG Model	LLM	Recall@11 (%)		Mean Recall@11 (%)	
		No FT	FT	No FT	FT
Motifs	Llama-3.1-8B	5.22	15.67	5.30	16.54
	GPT-4o-mini	11.57	16.42	12.06	18.24
Motifs-TDE	Llama-3.1-8B	4.85	18.66	4.92	20.52
	GPT-4o-mini	8.58	23.51	8.71	25.76
RelTR	Llama-3.1-8B	5.22	19.03	5.30	21.53
	GPT-4o-mini	11.94	25.37	12.12	27.65

Table 2. Results of PASGG-LM performing context-aware SGG on the PAi SGG dataset. PASGG-LM pipelines are compared in terms of Recall@K, Mean Recall@K, and mean average precision.

K	Metric	Motifs		Motifs-TDE		RelTR	
		Llama (8B)	GPT-4o (mini)	Llama (8B)	GPT-4o (mini)	Llama (8B)	GPT-4o (mini)
3	R@K (%)	8.58	8.96	10.07	12.69	9.33	12.31
	mR@K (%)	8.71	9.09	10.23	12.88	9.47	13.45
	mAP (%)	6.48	6.55	6.54	8.66	6.22	8.69
5	R@K	12.69	12.69	14.55	20.15	13.43	19.03
	mR@K	12.88	12.88	14.77	20.45	13.95	20.27
	mAP	8.94	9.02	8.98	13.83	8.07	12.97

Table 2. *Cont.*

K	Metric	Motifs		Motifs-TDE		RelTR	
		Llama (8B)	GPT-4o (mini)	Llama (8B)	GPT-4o (mini)	Llama (8B)	GPT-4o (mini)
7	R@K	13.81	15.67	17.16	22.39	16.04	24.63
	mR@K	14.65	17.49	19.00	24.62	18.50	26.89
	mAP	9.59	11.02	10.45	15.43	9.30	16.47
9	R@K	15.67	16.42	18.66	23.51	18.66	25.37
	mR@K	16.54	18.24	20.52	25.76	21.15	27.65
	mAP	10.31	11.63	11.07	16.18	10.57	17.03
11	R@K	15.67	16.42	18.66	23.51	**19.03**	**25.37**
	mR@K	16.54	18.24	20.52	25.76	**21.53**	**27.65**
	mAP	10.31	11.63	**11.07**	16.18	10.72	**17.03**

5.3. PASGG-LM Algorithm

PASGG-LM starts by receiving an input image, which is passed to a state-of-the-art scene graph model (presumably pre-trained on VG), returning a predicted scene graph composed of relational triplets. The resulting scene graph is then stored in a list and passed to the LLM post-processing module. Next, the scene graph is converted into a textual representation and appended to the end of the *rule-based inference prompt* described in Section 5.4. The resulting prompt is fed into an off-the-shelf or fine-tuned LLM, which performs the post-processing. During post-processing, the LLM conducts inductive inference about contextual aspects of the scene based on the spatial and physical entities presented in the inputted scene graph. The final output is then validated and either accepted or the LLM is re-prompted. This process repeats until the LLM has outputted a valid response. In our PASGG-LM evaluation experiments, we give the LLM 3 chances otherwise it is recorded as the model predicted no tuples. The PASGG-LM algorithm is outlined in Algorithm 1.

5.4. Rule-Based Inference Prompt

Through prompt engineering, we carefully constructed a *rule-based inference prompt*, which guides the LLM to infer various emotional and contextual attributes underlying an inputted scene graph. Our instructions involved hand-crafting 15 inference rules based on inductive reasoning, including nine *scene rules* and six *emotion rules*. An example of a scene rule is as follows: "if the graph contains outdoor entities such as beach, boat, mountain, hill, tree, street, then add tuple (`situation, location type, outdoor`)"; and an example emotion rule is as follows: "if a tuple contains bike, surfboard, ski, skateboard, then add (`situation, has social context, recreational`)". The purpose of these rules is to help the LLM infer additional context and ambience information from the scene.

The nine scene-related rules can be used to infer scene information such as location type (*indoor* vs. *outdoor*) or weather conditions based on objects like *umbrella* or *ski*, and provide additional context about the environment. For instance, the presence of *ski*, *jacket*, *pants*, and *hill*, might prompt the system to infer *cold* temperature and an *outdoor* setting, adding depth to the understanding of the scene. As another example, detecting the objects *desk* and *paper* might cause the model to infer a *professional* social context. The six emotion-related rules help the model capture the likely social atmosphere and emotional undertones within a scene, which are significant for perspective-aware applications. We instruct the model to infer emotions based on the types of activities present in the scene graph, and the social context inferred by the model. For example, if the input graph contains a *skateboard*, the model may infer a *recreational* social context, and if the model infers a *recreational* social context, it may as a result infer that the situation's overall emotion is *happy* (in the absence of sufficient opposing evidence).

By design, PASGG-LM infers implicit context from explicit context during post-processing. For example, scene graphs output by benchmark SGG models contain explicit relations such as (man, on, bike). Using prompt engineering, we guide the LLM to infer

implicit relationships based on given explicit relationships output from benchmark SGG models. Within the engineered prompt, we provide the object classes present in the VG dataset and instruct the LLM to make logical inductions based on these object classes as they appear in the inputted scene graph. For demonstration purposes, we could include a rule similar to the following: "If the inputted scene graph contains objects such as *bike*, *basketball*, or *surfboard*, infer that there is a *recreational* social context". Additionally, there may be a second rule stating the following: "if inferring a *recreational* social context, infer the situation is *happy*". Given the explicit relationship `(man, on, bike)`, PASGG-LM may try to infer implicit contextual information such as `(situation, has social context, recreational)`, `(situation, location type, outside)`, or `(main participant, has emotion, happy)`.

More examples of possible rules include the following: "based on clothing worn and types of objects present in the inputted scene graph, infer the weather, temperature, and location being indoor or outdoor", "if the entity tie is detected in the scene graph, infer `(situation, has social context, formal)`", and "if the entity umbrella is detected, infer `(situation, has weather, rainy)`". To prevent hallucinations, we instruct the LLM to not attempt to infer things that cannot be logically inferred from the inputted scene graph. This is done using additional constraints on top of the 15 rules within the engineered prompt. These additional constraints act as reasoning guardrails.

Within the *rule-based inference prompt*, we also provide some example logical inductions to follow, that, during the reasoning process, likely occur as a consequence of following the 15 inductive reasoning rules. For instance, we add the following constraint: "if a tuple contains tie, and *scene rule 2* resulted in inferring *indoor location*, add (`situation, has social context, formal`)". By providing examples of this type of reasoning in the prompt and instructing the LLM to adhere to similar logical principles, we guide the LLM to output relevant scene graphs enriched with contextual information. We also provide the LLM some examples of how the input-output conversions should look, within the prompt.

These logical inductions and constraints help the model respect interdependencies between the rules. Many inductive rules have interdependencies, where specific emotion rules depend on outputs from scene rules (e.g., "if the location is *indoor* and there are *jacket* and *tie* objects in the scene, infer *formal* `social context`"). This dependency structure reflects real-world scenarios where spatial elements influence emotional and social interpretations. Applying these rules in an order that respects dependencies allows PASGG-LM to generate a cohesive and contextually accurate scene description.

We also instruct the model to avoid predicting attributes in the scene when the input benchmark model's scene graph contains ambiguous or absent information about those attributes. In other words, we reinforce the LLM to only infer tuples that can be logically inferred from the input. One of our constraints included in the engineered prompt is as follows: "if logical, infer additional tuples based on your own inference". Additionally, we provide the constraint: "if multiple social contexts apply (e.g., *recreational* and *professional*), prioritize based on the most likely given the context". Lastly, we include the following constraint: "when inferring new tuples, ensure they are logical, reasonable, and enhance the data's richness". By incorporating logical guardrails, we are largely able to prevent PASGG-LM models from attempting to infer emotional or social context attributes that lack sufficient information.

Since chain-of-thought (CoT) has been shown to improve the performance of LLMs, we incorporate CoT into our prompt [29]. Specifically, we integrate chain-of-thought reasoning with the inductive rules, by providing the LLM examples of the sequential reasoning that adheres to the specified logical principles, rules, and constraints. In this way, PASGG-LM mimics human-like reasoning by sequentially applying rules and making inferences based on prior deductions. This approach increases interpretive accuracy, as the model uses both pre-defined rules and its own logic to enhance understanding. Combining these tactics, we aim to maximize the capabilities of LLMs for the task of post-processing SGG output to support PAi applications.

In terms of flexibility, the inductive reasoning rules used in a PASGG-LM post-processing pipeline are diverse within the training distribution of the benchmark SGG model. In our paper, we used models trained on the VG dataset for experimentation. Therefore, our inductive reasoning rules were designed to post-process SGG outputs found in the VG object and predicate class distribution. Within the context of SGG models trained on VG, PASGG-LM is diversely applicable to different scenarios. While PASGG-LM can be flexibly applied to various dataset distributions, adjustments to the inductive reasoning rules may be necessary to align with the specific object and predicate classes present in new training distributions.

5.5. Metrics for Classical Scene Graph Generation

Recall@K is a widely used metric in SGG, initially introduced by [43]. This metric reframes the SGG problem as a retrieval task, emphasizing not only the correct classification of relationships but also ranking them based on their confidence or likelihood of being correct. Specifically, Recall@K measures the percentage of ground-truth relationships included among the top K predictions, encouraging models to prioritize relevant relationships over unrelated relationship pairs.

Mean Recall@K was developed to address the bias inherent in datasets like Visual Genome, where certain predicates are overrepresented. In typical Recall@K evaluations, models can achieve high scores by correctly predicting only the most common relationships, even if performance on less frequent relationships is poor. To mitigate this issue, Mean Recall@K was proposed by [18], which calculates Recall@K separately for each predicate category and reports the average score across all categories. This approach ensures that each relationship type, regardless of frequency, is given equal weight in the evaluation.

Mean average precision (mAP) is a gold standard metric in information retrieval and evaluation systems [44]. This metric provides a robust way to evaluate models that produce ranked lists of results, so we use it to evaluate our scene graph generation models. mAP is calculated by first determining the average precision (AP) for each category or class within a dataset. The mean of these AP values across all categories provides the mAP score, an overall performance metric that accounts for both precision and recall across the entire dataset. mAP is especially useful in cases where there are imbalances in class frequencies, as it averages performance metrics across all classes, including both frequent and rare classes. We employ all three metrics—R@K, mR@K, and mAP—in our experimental evaluations to ensure consistency and comprehensiveness in comparing our results with prior research.

6. Experiments and Results

This section presents the experimental setups and corresponding results for evaluating the following three settings: state-of-the-art scene graph models for PAi, LLM fine-tuning, and PASGG-LM for *context-aware scene graph generation*. For clarity, each result is discussed directly after its setup.

6.1. Scene Graph Model Evaluation on PAi: Setup

To evaluate state-of-the-art SGG models for PAi, we tested three models pre-trained on VG (VG described in Section 4.2), on our novel PAi SGG dataset: Motifs [19], Motifs-TDE [24], and RelTR [20]. Motifs and Motifs-TDE were both implemented on top of the widely used benchmark framework provided by [24]. RelTR was implemented using the framework provided by [20]. Each framework provided a version of its state-of-the-art model pre-trained on VG, simplifying our implementation process. We will now discuss the setup for evaluating Motifs and Motifs-TDE, followed by the setup for RelTR.

Motifs and Motifs-TDE are both two-stage SGG frameworks that use an object detector backbone followed by a relation detector module. In our case, we use Faster-RCNN as the backbone followed by either Motifs or Motifs-TDE as the relation detector. Faster-RCNN combines region proposal networks with Fast-RCNN to achieve state-of-the-art results in object detection with an overall detection frame rate of 5 fps [17,45]. We use the pre-trained

Faster-RCNN weights and setup provided by the original paper [24]. The Faster-RCNN model uses a ResNeXt-101-FPN [46,47] and input images are scaled to be 1 k pixels in length. The object detector was pre-trained on VG with batch size 8 and an initial learning rate of 8×10^{-3}, which was decayed by a factor of 10 on the 30th and 40th iterations. Using a 0.5 IoU threshold, the detector achieved a final mean-average precision (mAP) of 28.14, as reported in [24].

Motifs and Motifs-TDE were trained with the setup outlined in [24], using stochastic gradient descent (SGD) with a batch size of 12 and an initial learning rate of 0.12. The learning rate was decayed twice by a factor of 10 after performance plateaus on the validation set. Eighty RoIs were sampled per image, using 0.5 IoU for object predictions. We followed [24] and did not assume that non-overlapping subject–object pairs were invalid. On the VG dataset, the pre-trained Motifs and Motifs-TDE models achieved a mean Recall@K (K = 20) of 5.2 and 6.6, respectively [48].

RelTR was trained on VG with a batch size of 2 for 150 epochs using the Adam optimizer with the weight decay equal to 10^{-4} and clipping gradients greater than 0.1 [49,50]. Different from Motifs, RelTR is trained in an end-to-end fashion. The backbone ResNet50 and Transformer module are set with respective initial learning rates of 10^{-4} and 10^{-5}, dropping the learning rate by 0.1 after the first 100 epochs. During training, RelTR was initialized with weights pre-trained on the VG, except for the relation classifiers. RelTR was then fine-tuned on scene graph generation on VG for 100 epochs. The relation classifier learning rate was set to 10^{-4} while a learning rate of 10^{-5} was used for the other modules. Auxiliary losses [51] were used for the triplet decoder, following previous works [52,53]. As reported by [20], when training the RelTR multi-head attention modules, a dropout rate of 0.1 was used for pruning.

6.2. Scene Graph Model Evaluation on PAi: Result

Figure 5 shows the performance of the three pre-trained SGG models—Motifs-TDE, Motifs, and RelTR—on the PAi SGG dataset, evaluated using our PSS metric described in Section 5.1. The first bar for each model represents the PAi similarity score (PSS) compared to the ground truth (PAi SGG data). None of the SGG models achieved a PSS above 0.5. To identify the main bottleneck, we recalculated the PSS by excluding each key PAi feature one at a time to observe any PSDD improvements. All three state-of-the-art models perform similarly, struggling to capture the context-based aspects required for perspective-aware computing. The highest score, though not significantly better, is 0.48, achieved by Motif-TDE when the time feature is excluded. This underscores the limitations of existing scene graph models for PAi, as time is crucial for accurate situation recognition.

Figure 5. Comparison of PSS performance for three SGG models (Motifs-TDE, Motifs, RelTR) on the PAi SGG dataset, illustrating their difficulty in capturing context-based aspects for perspective-aware computing. The performance is also shown with PAi feature exclusion to explore potential improvements.

6.3. PASGG-LM Fine-Tuning: Setup

We compare two benchmark LLMs, exploring their capabilities on context-aware SGG by testing them in our PASGG-LM pipeline. In particular, we compare GPT-4o-mini and Llama-3.1-8B-Instruct with and without fine-tuning on the PAi SGG dataset, measuring their performances in terms of R@K, mR@K, mAP, and PSS. The fine-tuning dataset is comprised of 112 ground truth (Y label) situation graphs represented as text. Each situation graph corresponds to an image from our PAi SGG dataset described in Section 4.3. The X labels of the tuning procedure are generated scene graphs from the three state-of-the-art scene graph models pre-trained on VG: Neural Motifs, Neural Motifs-TDE, and RelTR. In total, we compare 2 LLMs with and without fine-tuning on 3 SGG models, resulting in 12 PASGG-LM models. When fine-tuning both GPT-4o-mini and Llama-3.1-8B-Instruct, we shuffled and split the PAi SGG dataset into 90 samples for training and validation, and 22 for unbiased evaluation of the models.

We created the dataset by combining $X \rightarrow Y$ mappings in the form of message–response pairs. The messages (X labels) consist of generated scene graphs produced by evaluating the three pre-trained state-of-the-art models on our PAi SGG dataset. Scene graphs are represented to the LLM during fine-tuning as a list of relational triplets converted to a text representation. Part of an example X label from the fine-tuning procedure looks like this:

```
(woman, riding, bike)
(man, riding, bike)
(woman, wearing, shirt)
(woman, wearing, shirt)
(man, wearing, shirt)
(man, wearing, short)
```

Similarly, the Y label responses are ground truth situation graphs of the same image from our PAi SGG dataset. The Y labels are formatted as text representations in the same fashion:

```
(situation, has time, morning)
(situation, has location, lake)
(situation, location type, outdoor)
(situation, has ambience, calm)
(situation, has participant, main participant)
(main participant, has emotion, happy)
(main participant, activity, bike)
(situation, overall activity, bike)
(situation, has weather, sunny)
(situation, temperature, warm)
(situation, has social context, casual)
```

We now describe our training setups for fine-tuning GPT-4o-mini on scene graphs from our PAi SGG data followed by the setup for Llama-3.1-8B-Instruct. The `GPT-4o-mini-2024-07-18` model was implemented using OpenAI's API. During fine-tuning, we trained GPT-4o-mini for 3 epochs using a batch size of 1, as recommended by the OpenAI API for a dataset of 90 samples. We used a training–validation split of 0.86–0.24 (68 samples for training, 22 for validation) and set the learning rate multiplier to 1.8.

After fine-tuning on the PAi SGG data scene graphs, GPT-4o-mini reported mean token accuracy scores on the final hold-out set of 22.9%, 26.1%, and 26.9% when using Motifs, Motifs-TDE, and RelTR-predicted SGGs as input, respectively. Additionally, on the same hold-out set, GPT-4o-mini scored a precision of 46.7%, 53.3%, and 52.2% when using Motifs, Motifs-TDE, and RelTR, respectively.

Llama-3.1-8B-Instruct was implemented through the `Transformers` library on a single 4090 RTX GPU using the model hosted on HuggingFace. To reduce memory usage and enable training the 8-billion-parameter model on our GPU, we employed 8-bit quantization

and Low-Rank Adaptation (LoRA) [54,55]. For experimental validity and consistency, the data processing of the Llama training procedure and hyper-parameter selection is identical to the process used in fine-tuning GPT-4o-mini. The batch size for our Llama implementation was 1, and we trained it for 3 epochs. Our learning rate for Llama was 5×10^{-5} and we employed the Adam optimizer with weight decay [49,50]. Upon completing fine-tuning, on the unbiased hold-out set, Llama-3.1-8B-Instruct scored 22.8%, 24.6%, and 23.8% mean token accuracies when using SGG output from Motifs, Motifs-TDE, and RelTR as input, respectively. In the same setup, Llama scored a precision of 39.1%, 22.6%, and 28.7%, when using Motifs, Motifs-TDE, and RelTR, respectively.

6.4. LLM Fine-Tuning: Result

Table 1 describes the effect of fine-tuning on PASGG-LM pipelines. Our results indicate that fine-tuning leads to a significant improvement within *context-aware SGG* and also the development of PAi systems. As shown in Table 1, fine-tuning can boost performance, sometimes doubling or nearly tripling the effectiveness of the same PASGG-LM pipeline compared to when it is not fine-tuned. This improvement is most evident with mR@K at K = 11, where the performance of all PASGG-LM pipelines increased, regardless of the baseline SGG model used in the first module. RelTR combined with GPT-4o-mini increases from 12.12% mR@K (K = 11) to 27.65%. Llama's performance increases from 5.3% to 21.53% R@K with K = 11. The same trend holds when using Motifs and Motifs-TDE as base models, with Motifs (GPT-4o-mini) mR@K increasing from 12.06% to 18.24% and Motifs-TDE increasing from 8.71% to 25.76%.

Across all tested cases, fine-tuning just three epochs provides significant improvements in terms of Recall@K, Mean Recall@K, and mean average precision. Even the smallest improvement was significant. The smallest improvement was with Motifs and GPT-4o-mini, increasing the Recall@K (R@K) with K = 11 from 11.57% to 16.42% on the PAi SGG evaluation set. The largest improvement was seen in the PASGG-LM pipeline using Motifs-TDE and GPT-4o-mini, improving Mean Recall@K (mR@K) with K = 11 from 8.71% to 25.76%. The largest improvement with Llama-3.1-8B was in the PASGG-LM pipeline using RelTR and Llama, increasing mR@K with K = 11 from 5.30% to 21.53%.

Without fine-tuning, the LLMs tend to produce smaller contextual scene graphs, with fewer contextual triplets on average. After fine-tuning, the LLMs generate richer scene graphs with more triplets and often higher precision compared to those produced without fine-tuning. Fine-tuning improved the LLMs' responsiveness to the engineered prompt, resulting in better adherence to the specified output format. Without fine-tuning, Llama struggled to follow the output format specified in the engineered post-processing prompt with inductive rules. This means the base LLM models had to be re-prompted more times on average to retrieve a valid scene graph output compared to their fine-tuned counterparts. Overall, we find fine-tuning to be both highly effective and computationally efficient, requiring only three epochs in our case. Furthermore, incorporating fine-tuning enhances the scalability of the overall system, as fine-tuned outputs are more accurate and valid than untuned ones, reducing the number of failed inferences from the LLM.

Another trend we see is that fine-tuning tends to help Llama-3.1-8B more than GPT-4o-mini. Llama-3.1-8B's baseline performance was very low without fine-tuning. For instance, without fine-tuning, Motifs (Llama-3.1-8B) scores 5.22% R@K (K = 11), which increases significantly to 15.67% R@K with fine-tuning. Similarly, RelTR with Llama-3.1-8B achieved an R@K (K = 11) of 5.22% without fine-tuning, which increased to 19.03% with fine-tuning (a substantial improvement). We find that GPT-4o-mini outperforms Llama-3.1-8B in pipelines without fine-tuning; however, this advantage is largely mitigated when fine-tuning is added to the process.

6.5. PASGG-LM Evaluation on Context-Aware SGG in PAi: Setup

Our novel PASGG-LM algorithm generates context-rich scene graphs from images, capturing both contextual and ambient features within a scene. Figure 6 shows the gener-

ated graphs from Neural Motifs-TDE (left) and our proposed PASGG-LM pipeline using Motifs-TDE with GPT-4o-mini (right), both evaluated on a PAi image.

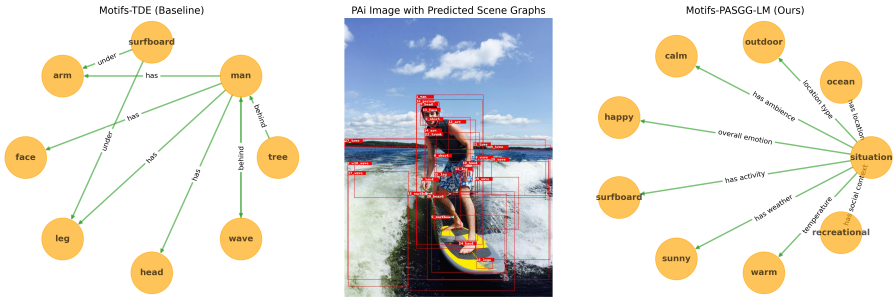

Figure 6. Comparison of generated scene graphs between Neural Motifs-TDE (**left**) and our proposed PASGG-LM pipeline (**right**) evaluated on a PAi image from the unbiased hold-out set. In this example, PASGG-LM uses Motifs-TDE, with the generated scene graphs processed by GPT-4o-mini fine-tuned on situation graphs from our PAi SGG data.

We evaluated PASGG-LM as an end-to-end pipeline for *context-aware SGG* on our PAi SGG dataset, measuring its performance with the classical SGG metrics Recall@K (R@K), Mean Recall@K (mR@K), and mean average precision (mAP) [18,19]. Our PASGG-LM evaluation results are shown in Figure 7 and Table 2. The PASGG-LM pipelines, composed of various SGG and LLM model combinations, were tested on our PAi SGG dataset. In total, 16 models were evaluated, including combinations of Motifs, Motifs-TDE, and RelTR paired with both fine-tuned and base versions of GPT-4o-mini and Llama-3.1-8B. Table 2 shows the results of using PASGG-LM on the PAi SGG data (with fine-tuning). Our results in Table 2 and Figure 7 establish initial baselines for *context-aware SGG* in PAi, significantly advancing the contextual capabilities of current SGG methods.

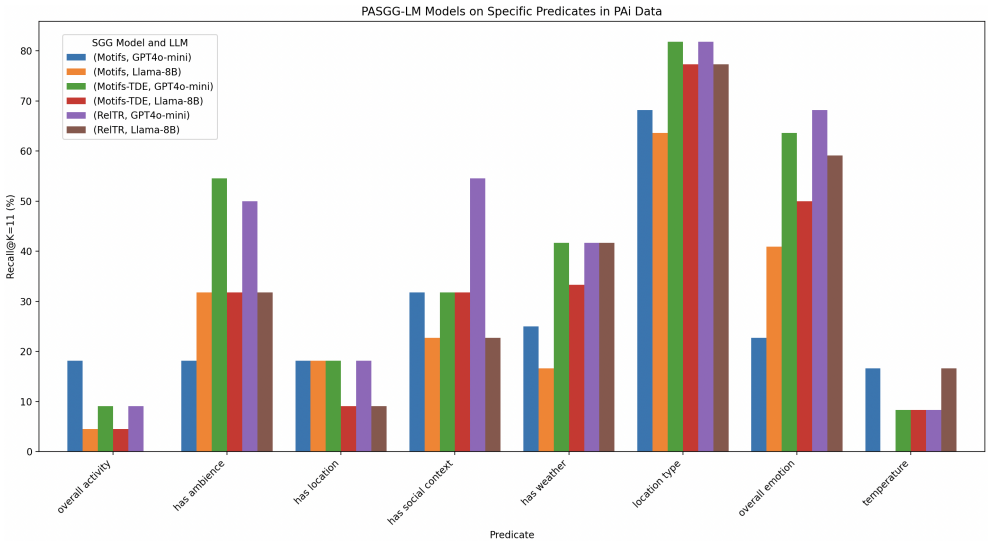

Figure 7. Predicate classes with recall scores of zero, including activity (participant-specific), has emotion (participant-specific), has participant, and has time, were omitted.

To further our understanding of which predicates are challenging in *context-aware SGG*, we also compare predicate-specific recall scores. Figure 7 shows R@K scores of the six fine-tuned PASGG-LM pipelines for predicting contextual predicates correctly on the PAi SGG data. We measured PASGG-LM model performance on the 12 new contextual predicate classes found in our PAi SGG dataset. Of the 12 classes, 4 were never correctly predicted by the model and removed from Figure 7: activity (participant level), has emotion (participant level), has participant, and has time. The results of the PASGG-LM models tested on different contextual predicates are further discussed in Section 6.6.

6.6. PASGG-LM Evaluation on Context-Aware SGG in PAi: Result

Fundamental baseline results of PASGG-LM performing *context-aware SGG* on the PAi SGG data unbiased evaluation set are shown in Table 2. Of the 6 fine-tuned PASGG-LM pipelines, RelTR performed the best in terms of R@K, mR@K, and mAP. With K = 11, RelTR post-processed by GPT-4o-mini scores 25.4, 27.7, and 17.03 on R@K, mR@K, and mAP, respectively. We found the second-best PASGG-LM setup to be Motifs-TDE combined with GPT-4o-mini for post-processing. Setting K = 11, Motifs-TDE, respectively, achieved a R@K, mR@K, and mAP of 23.5, 25.8, and 16.18, which is worse but comparable to the performance of RelTR. In the same setup, Motifs without using TDE performed the worst, resulting in R@K, mR@K, and mAP scores of 16.4, 18.2, and 11.63, respectively. In general, we found RelTR to be the best SGG model for the PASGG-LM pipeline, with Motifs-TDE performing slightly worse.

Motifs-TDE and RelTR performed significantly better as base models compared to base Motifs in our PASGG-LM pipeline. This is illustrated in Table 2, where Motifs-TDE and RelTR, when combined with Llama for post-processing, outperform Motifs with GPT-4o-mini in terms of R@K and mR@K. However, in each case, using GPT-4o-mini fine-tuned on the PAi data resulted in higher mAP compared to Llama-3.1-8B with fine-tuning. Our experiments show that PASGG-LM pipelines built on GPT-4o-mini significantly outperform their similar counterpart when using Llama-3.1-8B instead. This suggests that GPT-4o-mini is better suited than Llama-3.1-8B for contextual inference tasks such as following the *rule-based inference prompt* described in 5.4. While GPT-4o-mini is closed-source, Llama is open-source. Considering trade-offs such as performance and control over the model is essential and is further discussed in Section 7.

To deepen our understanding of which contextual predicates are challenging to model in PAi, we examined the performance of PASGG-LM pipelines in predicting specific predicate classes within the PAi SGG dataset. The results of this analysis are presented in Figure 7. Four classes proved extremely difficult for the explored PASGG-LM pipelines: `activity` (participant-specific), `has emotion` (participant-specific), `has participant`, and `has time`. Of the eight predicate classes the model was capable of predicting, location type was by far the easiest. Intuitively this makes sense, as the location type can usually be inferred from the list of indoor or outdoor objects present in the predicted scene graph. If there are trees and a bird in the graph, the location is likely outside, and in contrast, if there is a desk or bed, it is likely indoors. In general, there was a high degree of variability in the difficulty of predicting specific predicate classes.

The results shown in Figure 7 align with our findings in Table 2, and Motifs-TDE and RelTR paired with GPT-4o-mini achieved the best per predicate R@K scores. PASGG-LM pipelines based on GPT-4o-mini outperformed their counterparts based on Llama-3.1-8B. This trend was consistent across the three SGG models, except in a few cases with PASGG-LM pipelines built on Motifs, where Llama outperformed GPT-4o-mini in predicting contextual classes such as `has ambience` and `overall emotion`. In general, GPT-4o-mini significantly outperformed Llama-3.1-8B in aiding inference in PASGG-LM pipelines.

As indicated in Figure 7, the PASGG-LM models had difficulty with correctly predicting relationships between the situation and its temperature, location, and overall activity. Interestingly, the combination of both Motifs-TDE and RelTR with GPT-4o-mini underperformed comparably to the other models in trying to infer the temperature of the scene. It would require a larger study with more data to draw conclusions on their ability to infer temperature, but this remains an interesting finding. RelTR with GPT-4o-mini was the best at predicting the social context of the scene, achieving a Recall@K with $K = 11$ of over 50%. The models generally performed best on the following predicate classes: `location type`, `overall emotion`, `has ambience`, `has weather`, and `has social context`.

We also measured the effect of fine-tuning on PASGG-LM's PSS and found that it consistently enhances performance. As depicted in Figure 8, the Motif-TDE model with fine-tuned GPT-4o-mini achieved a 70% similarity score, indicating the highest performance.

In contrast, in the non-fine-tuned settings, GPT-4o-mini applied to both the Motifs and RelTR models reached a 58% similarity score, marking the best result in that configuration.

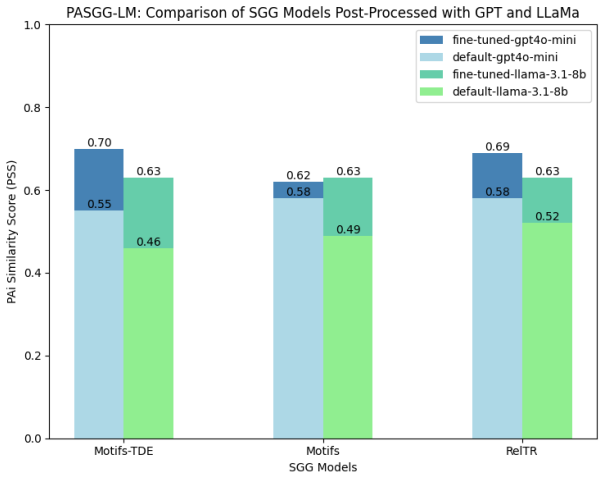

Figure 8. Comparison of PSS performance for PASGG-LM across three SGG models (Motif-TDE, Motifs, RelTR) with and without fine-tuning using two LLM models: GPT-4o and Llama-3.1-8B. The figure highlights the improvements achieved through fine-tuning, with GPT-4o consistently outperforming Llama-3.1-8B in both settings.

6.7. Comparative Computational Costs

The main computational costs associated with our proposed PASGG-LM framework are twofold, depending on the choice between open-source and closed-source large models. First, there is the cost associated with the fine-tuning phase, which, although necessary to adapt the model to the specific task and dataset, is not particularly resource-heavy in our case. Depending on the dataset, this cost could increase, but for our specific application, the fine-tuning process did not incur considerable computational overhead. Second, there are costs associated with query (inference) time, which depend on the model's size, the complexity of the queries, and how efficiently the framework handles them in real-time. The pricing for GPT model queries is outlined in OpenAI's GPT model pricing table (https://openai.com/api/pricing/, accessed on 20 November 2024), which provides an estimate based on the model size and usage. Compared to pure SGG models, which often require retraining on large datasets to refine understanding and improve accuracy, PASGG-LM is generally more efficient in both fine-tuning and inference costs. SGG models typically incur higher costs during the retraining phase, as they require substantial computational resources to process and generate scene graphs.

7. Discussion

Modelling user perspectives is a crucial aspect of developing robust perspective-aware AI (PAi) systems. To accurately model user perspectives based on images, the model should capture contextual aspects from the scene. Existing state-of-the-art SGG models fail to understand contextual aspects of the scene, as their objective function focuses solely on predicting spatial and physical relationships, leaving them unable to capture the broader context of the scene. In contrast, the objective function of PASGG-LM is to infer contextual, ambient, and participant-specific information. The engineered prompt and labels in our PAi SGG dataset define this objective function, which PASGG-LM optimizes through prompt engineering and fine-tuning. PASGG-LM achieves enhanced contextual inference by using LLMs for inductive reasoning, translating standard SGG outputs into a more comprehensive "situation graph" (the PAi ground truth labels). When PASGG-LM includes a fine-tuned LLM, it is specifically trained on PAi SGG labels that incorporate user perspectives, capturing participant-specific details. Details on these labels are provided in Section 4.3.

Even without participant-specific fine-tuning, PASGG-LM's ability to infer general contextual information about the situation significantly enhances the capacity of existing SGG models to capture user perspectives. By enriching scenes with details about the environment, social context, and potential activities, PASGG-LM provides insights that better support PAi applications. For example, inferring not only that a "man" is "on a bike" but also that he is in a "busy" scene taking place in the "city" suggests an active, energetic atmosphere, adding interpretive layers that resonate with how users perceive and interact with their surroundings. This general contextual awareness enables PASGG-LM to deliver outputs that feel more personalized and relevant. By aligning its inferences with common human experiences, the model becomes more attuned to the implicit perspectives users bring to different situations.

It is important to remember that PASGG-LM pipelines are limited to the inferences that the LLMs can draw from the output provided by the benchmark SGG model. If the SGG model output lacks sufficient information to describe certain aspects of the scene, it is unreasonable to expect the LLM to reliably infer those aspects. This can be seen in Figure 7, as the PASGG-LM models do not successfully infer any relations with predicates from the following classes: `activity`, `has emotion`, `has participant`, and `has time`. Our experiments demonstrated that LLMs generally refrain from inferring contextual relations with insufficient evidence, showing a strong capability to avoid false positives in these cases. In most cases, we found that when the input scene graph contained sufficient information, GPT-4o-mini and Llama-3.1-8B with finetuning consistently inferred related contextual features, building a more coherent final scene graph representation. This is qualitatively shown in Figure 6, where the PASGG-LM model based on Motifs-TDE and GPT-4o-mini is capable of inferring robust contextual aspects about the user who is surfing in warm weather recreationally on the water.

The primary obstacle to achieving high PAi similarity scores for complex contexts, such as participant-specific emotions, lies in the ambiguous output of current benchmark SGG models, which often lack sufficient information to infer these contextual aspects. PASGG-LM addresses these obstacles by fine-tuning the LLMs on our ground truth PAi scene graph labels, which contain participant-specific information in the ground truth labels. Through fine-tuning, we optimize the model toward outputting contextual relation triplets which capture participant emotion and activity information. We also instruct the model to not infer attributes that cannot be capably inferred from the inputted scene graph. Therefore, PASGG-LM generally attempts to predict participant-specific information only when it can reasonably be inferred from the inputted benchmark scene graph output. We find this feature tends to override the fine-tuning and reduces hallucinations in the model, improving its precision. As a result, PASGG-LM tends to overly focus on the context of the overall situation, taking into account the limitations in what participant-specific information can be inferred from the scene graph being post-processed.

Although our experimental results on R@K, mR@K, and mAP metrics strongly favour GPT-4o-mini over Llama-3.1-8B for PASGG-LM in context-aware SGG to support PAi, it is important to consider the drawbacks associated with closed-source LLMs. PAi is highly dependent on personal data, and using closed-source LLMs can expose data to the owners of the API. As people generally prefer not to share personal data, limiting the number of third-party stakeholders with access to it is crucial. Furthermore, data decentralization is a core aspect of PAi, and using closed-source LLMs through an API can remove this value. Although slower than GPT-4o-mini through OpenAI's API, Llama is cheap. The open-source nature of Llama also provides much more freedom in terms of customizing the capabilities of the LLM with different hyper-parameters. Considering the rate of progress in AI, it can be assumed that open-source LLMs will continue to improve and become more suitable for PAi, making them more desirable than they are today.

Privacy concerns are an important aspect to consider when working with personal data. Using an open-source model locally can ensure security over the data passing through it. We explored open-source options for all scene graph models and tested one closed-

source LLM (GPT-4o-mini) alongside one open-source LLM (Llama-3.1-8B). Users have the option to select a completely open-source PASGG-LM pipeline, allowing them to maintain full control and security over their data. In addition to using open-source models, PASGG-LM advances toward PAi, a paradigm that leverages data decentralization and federated learning to enhance privacy preservation. The data decentralization and federated learning method are outlined in [4].

Regarding user consent, PASGG-LM, when using models without LLM fine-tuning, aligns with existing SGG models by avoiding participant-specific attributes. Such attributes can only be incorporated through fine-tuning with PAi SGG labels that include participant-specific data. For an optimized PASGG-LM pipeline to effectively capture such personalized contextual attributes, user-specific data are necessary for fine-tuning. Consequently, the model's predictions on personal information are limited to what participants have explicitly shared. Therefore, as long as users do not make extensive personal data publicly available, PASGG-LM pipelines do not face issues regarding user consent.

In terms of future work, it is important to consider how we can design future SGG models that better predict contextual predicate classes such as time and other participant-specific classes. Current SOA scene graph methods struggle to capture participant-specific relations with objects. We posit that constructing a large context-aware SGG or PAi SGG dataset with properly labelled contextual and physical object classes will enable models trained on it to learn contextual relations directly from training data.

Moreover, SOA scene graph methods return predicted object box coordinates along with the predicted class (i.e., person, man, woman, face). Theoretically, this information could be used in post-processing along with the original image to perform further computational inference. We envision a system that leverages auxiliary information in the form of a chronicle identity grounding meta-dataset (CIGM). The CIGM would be provided by a source user (or main participant) before model inference to support subsequent facial, emotional, and identity recognition through post-processing. An example of a CIGM could be an image dataset containing 1-5 images of the participant and each common secondary participant (family and friends) labelled with individual IDs. The CIGM could then be used alongside the localized predicted object coordinates to classify or re-identify participant and 'emotion' information during post-processing, leveraging pre-trained vision models. If successful, this strategy would be a way to inject participant and participant-specific emotion information into SGG and *context-aware SGG* frameworks.

Our approach to PAi differs from traditional multimodal AI models due to its neuro-symbolic foundation, which combines neural learning with symbolic reasoning through a structured, reasoning-ready model known as a chronicle. Chronicles are created through a two-phase process: a learning phase that constructs a graph-based model of an individual's temporal and situational perspectives using digital footprints (such as images, text, and interactions) and a reasoning phase that enables responsible, secure querying of insights to support informed decision-making.

In terms of contextual image understanding, unlike conventional SGG models that focus on detecting explicit attributes and spatial relationships, our PASGG-LM approach extends scene graphs to include ambient, social, and participant-specific attributes, capturing contextual elements and enhancing user perspectives. PASGG-LM accomplishes this by using LLMs in post-processing, which apply inductive reasoning to generate richer contextual representations. Through prompt engineering and fine-tuning on PAi-specific labels, our model captures nuanced contextual details that are often beyond the reach of traditional methods. Altogether, this unique combination supports applications tailored for perspective-aware use cases, prioritizing ethical access and fostering a deeper understanding of individual contexts.

8. Conclusions

In this work, we explored the development of PAi, a novel approach that enhances human–AI interaction by allowing users to perceive and interact with the world from

another person's perspective. By introducing our PASGG-LM pipeline and the novel task of *context-aware SGG*, we integrate SGG with LLMs to more effectively model individual perspectives within chronicles. Our experiments demonstrate the proposed method can detect robust contextual scene graphs from images, addressing a significant limitation of existing SGG models. We found that fine-tuning LLMs, particularly GPT-4o-mini, significantly improved our system's ability to capture contextual aspects in scene graphs, outperforming identical pipelines that used Llama-3.1-8B instead. Challenges remain in improving SGG models' ability to predict certain contextual relationships, such as temporal aspects or participant-specific interactions. To address this, we propose constructing a large *context-aware SGG* dataset, as well as integrating auxiliary tools like the CIGM to enhance identity and emotion recognition. This work has broad implications for human–AI interaction, particularly in areas like social robotics, personalized assistants, and collaborative systems, where understanding diverse perspectives can drive more inclusive and empathetic interactions. Future work will focus on improving the open-source applicability of models and designing better SGG frameworks for capturing nuanced human contexts.

Author Contributions: Conceptualization, D.P., M.A. and H.R.; methodology, D.P. and M.A.; software, D.P. and M.A.; validation, D.P. and M.A.; formal analysis, D.P. and M.A.; investigation, D.P., M.A. and H.R.; resources, D.P., M.A. and H.R.; data curation, D.P.; writing—original draft preparation, D.P. and M.A.; writing—review and editing, D.P., M.A. and H.R.; visualization, D.P. and M.A.; supervision, M.A. and H.R.; project administration, M.A. and H.R.; funding acquisition, H.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Flybits, Toronto Metropolitan University, and The Creative School; Grant Number: 1-51-51973.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this work is divided into two parts: the Visual Genome benchmark and a personal image dataset of an individual. Regarding the first dataset, the original data presented in the study are openly available at https://homes.cs.washington.edu/~ranjay/visualgenome/index.html (accessed on 29 October 2024). The second dataset contains sensitive personal image data from individual volunteers and is not readily available due to privacy considerations.

Acknowledgments: The authors would like to thank the team at Flybits, Toronto Metropolitan University, The Creative School, and MIT Media Lab for their support of our ongoing research in this field.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rahnama, H.; Alirezaie, M.; Pentland, A.S. A Neural-Symbolic Approach for User Mental Modeling: A Step Towards Building Exchangeable Identities. In Proceedings of the AAAI Spring Symposium Combining Machine Learning with Knowledge Engineering, Virtual, 22–24 March 2021. Available online: https://api.semanticscholar.org/CorpusID:232292661 (accessed on 5 November 2024).
2. Kok, C.L.; Ho, C.K.; Tan, F.K.; Koh, Y.Y. Machine Learning-Based Feature Extraction and Classification of EMG Signals for Intuitive Prosthetic Control. *Appl. Sci.* **2024**, *14*, 5784. [CrossRef]
3. Tian, L.; Yang, B.; Yin, X.; Su, Y. A Survey of Personalized Recommendation Based on Machine Learning Algorithms. In Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering (EITCE '20), Xiamen, China, 6–8 November 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 602–610. [CrossRef]
4. Alirezaie, M.; Platnick, D.; Rahnama, H.; Newman, D.J. Alex Paul "Sandy" Pentland. Perspective-Aware AI (PAi) for Augmenting Critical Decision Making. In Proceedings of the IEEE High Performance Extreme Computing Conference (HPEC), Virtual, 23–27 September 2024; pp. 1–9. Available online: https://ieee-hpec.org/wp-content/uploads/2024/09/142.pdf (accessed on 2 November 2024).
5. Alirezaie, M.; Rahnama, H.; Pentland, A. Structural Learning in the design of Perspective-Aware AI Systems using Knowledge Graphs. In Proceedings of the AAAI 2024, Digital Human Workshop, Vancouver, BC, Canada, 20–27 February 2024.

6. Pansanella, V.; Sîrbu, A.; Kertesz, J.; Rossetti, G. Mass media impact on opinion evolution in biased digital environments: A bounded confidence model. *Sci. Rep.* **2023**, *13*, 14600. [CrossRef] [PubMed]
7. Ge, Y.; Liu, S.; Fu, Z.; Tan, J.; Li, Z.; Xu, S.; Li, Y.; Xian, Y.; Zhang, Y. A Survey on Trustworthy Recommender Systems. *ACM Trans. Recomm. Syst.* **2024**. [CrossRef]
8. Yu, X.; Li, W.; Zhou, X.; Tang, L.; Sharma, R. Deep learning personalized recommendation-based construction method of hybrid blockchain model. *Sci. Rep.* **2023**, *13*, 17915. [CrossRef] [PubMed]
9. Dai, X.; Wang, J. Effect of online video infotainment on audience attention. *Humanit. Soc. Sci. Commun.* **2023**, *10*, 421. [CrossRef]
10. Spencer, S.B. The Problem of Online Manipulation. *Univ. Ill. Law Rev.* 2020. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3341653 (accessed on 20 November 2024).
11. Xu, D.; Zhu, Y.; Choy, C.B.; Li, F.-F. Scene Graph Generation by Iterative Message Passing. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3097–3106. Available online: https://api.semanticscholar.org/CorpusID:1780254 (accessed on 20 November 2024).
12. Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Shamma, D.A.; Bernstein, M.S.; Li, F.-F. Image retrieval using scene graphs. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3668–3678.
13. Teney, D.; Liu, L.; van den Hengel, A. Graph-Structured Representations for Visual Question Answering. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3233–3241.
14. Nguyen, K.; Tripathi, S.; Du, B.; Guha, T.; Nguyen, T.Q. In Defense of Scene Graphs for Image Captioning. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 1387–1396.
15. Zhai, G.; Örnek, E.P.; Wu, S.-C.; Di, Y.; Tombari, F.; Navab, N.; Busam, B. CommonScenes: Generating Commonsense 3D Indoor Scenes with Scene Graphs. *arXiv* **2023**, arXiv:2305.16283. [CrossRef]
16. Li, R.; Zhang, S.; He, X. SGTR: End-to-end Scene Graph Generation with Transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 19464–19474. [CrossRef]
17. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef]
18. Tang, K.; Zhang, H.; Wu, B.; Luo, W.; Liu, W. Learning to Compose Dynamic Tree Structures for Visual Contexts. *arXiv* **2018**, arXiv:1812.01880. [CrossRef]
19. Zellers, R.; Yatskar, M.; Thomson, S.; Choi, Y. Neural Motifs: Scene Graph Parsing with Global Context. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5831–5840. Available online: https://api.semanticscholar.org/CorpusID:4379400 (accessed on 20 November 2024).
20. Cong, Y.; Yang, M.Y.; Rosenhahn, B. RelTR: Relation Transformer for Scene Graph Generation. *arXiv* **2022**, arXiv:2201.11460. [CrossRef]
21. Li, H.; Zhu, G.; Zhang, L.; Jiang, Y.; Dang, Y.; Hou, H.; Shen, P.; Zhao, X.; Shah, S.A.A.; Bennamoun, M. Scene Graph Generation: A comprehensive survey. *Neurocomputing* **2024**, *566*, 127052. [CrossRef]
22. Wang, G.; Li, Z.; Chen, Q.; Liu, Y. OED: Towards One-stage End-to-End Dynamic Scene Graph Generation. *arXiv* **2024**, arXiv:2405.16925. [CrossRef]
23. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D.A.; et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *arXiv* **2016**, arXiv:1602.07332. [CrossRef]
24. Tang, K.; Niu, Y.; Huang, J.; Shi, J.; Zhang, H. Unbiased Scene Graph Generation from Biased Training. *arXiv* **2020**, arXiv:2002.11949. [CrossRef]
25. Wang, Q.; Huang, Y.; Zhao, G.; Clark, E.; Xia, W.; Liao, H. DiarizationLM: Speaker Diarization Post-Processing with Large Language Models. In Proceedings of the Interspeech 2024, Kos Island, Greece, 1–5 September 2024; pp. 3754–3758. [CrossRef]
26. Lu, C.; Lu, C.; Lange, R.T.; Foerster, J.; Clune, J.; Ha, D. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv* **2024**, arXiv:2408.06292. [CrossRef]
27. Chen, Z.; Mao, H.; Li, H.; Jin, W.; Wen, H.; Wei, X.; Wang, S.; Yin, D.; Fan, W.; Liu, H.; et al. Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs. *SIGKDD Explor. Newsl.* **2024**, *25*, 42–61. [CrossRef]
28. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020; Curran Associates Inc.: Red Hook, NY, USA, 2020; pp. 1–25.
29. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.H.; Xia, F.; Le, Q.; Zhou, D. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2022**, arXiv:2201.11903. [CrossRef]
30. Jin, B.; Liu, G.; Han, C.; Jiang, M.; Ji, H.; Han, J. Large Language Models on Graphs: A Comprehensive Survey. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 8622–8642. [CrossRef]
31. Nagamochi, H.; Ibaraki, T. *Algorithmic Aspects of Graph Connectivity*, 1st ed.; Cambridge University Press: New York, NY, USA, 2008.

32. Goldberg, A.V.; Harrelson, C. Computing the shortest path: A search meets graph theory. In Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05), Vancouver, BC, Canada, 23–25 January 2005; Society for Industrial and Applied Mathematics: University City, PA, USA, 2005; pp. 156–165.
33. Wang, K.; Xu, Y.; Wu, Z.; Luo, S. LLM as Prompter: Low-resource Inductive Reasoning on Arbitrary Knowledge Graphs. In *Findings of the Association for Computational Linguistics (ACL 2024)*; Ku, L.-W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 3742–3759. [CrossRef]
34. Cramer, A.L.; Wu, H.H.; Salamon, J.; Bello, J.P. Look, listen, and learn more: Design choices for deep audio embeddings. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019 pp. 3852–3856.
35. Yin, X.; Li, J.; Si, H.; Wu, P. Attention marketing in fragmented entertainment: How advertising embedding influences purchase decision in short-form video apps. *J. Retail. Consum. Serv.* **2024**, *76*, 103572. [CrossRef]
36. Cappuzzo, R.; Papotti, P. Thirumuruganathan, S. Creating embeddings of heterogeneous relational datasets for data integration tasks. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 1335–1349.
37. Yanhong, P.; Yuxin, W.; Fangchao, H.; Miao, H.; Zebing, M.; Xia, H.; Jun, D. Predictive modeling of flexible EHD pumps using Kolmogorov—Arnold Networks. In *Biomimetic Intelligence and Robotics*; Elsevier: Amsterdam, The Netherlands, 2024; Volume 4, ISSN 2667-3797. [CrossRef]
38. Borgo, S.; Ferrario, R.; Gangemi, A.; Guarino, N.; Masolo, C.; Porello, D.; Sanfilippo, E.M.; Vieu, L.; Galton, A.; Kutz, O. DOLCE: A descriptive ontology for linguistic and cognitive engineering. *Appl. Ontol.* **2022**, *17*, 45–69. [CrossRef]
39. Kim, K.; Yoon, K.; Jeon, J.; In, Y.; Moon, J.; Kim, D.; Park, C. LLM4SGG: Large Language Models for Weakly Supervised Scene Graph Generation. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 17–21 June 2024; pp. 28306–28316. [CrossRef]
40. Dagdelen, J.; Dunn, A.; Lee, S.; Walker, N.; Rosen, A.S.; Ceder, G.; Persson, K.A.; Jain, A. Structured information extraction from scientific text with large language models. *Nat. Commun.* **2024**, *15*, 1418. [CrossRef]
41. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.08774. [CrossRef]
42. Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. The Llama 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783. [CrossRef]
43. Lu, C.; Krishna, R.; Bernstein, M.S.; Li, F.F. Visual Relationship Detection with Language Priors. *arXiv* **2016**, arXiv:1608.00187. [CrossRef]
44. Beitzel, S.M.; Jensen, E.C.; Frieder, O. MAP. In *Encyclopedia of Database Systems*; Liu, L., ÖZsu, M.T., Eds.; Springer: Boston, MA, USA, 2009._492. [CrossRef]
45. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083. [CrossRef]
46. Lin, T.-Y.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. Available online: https://api.semanticscholar.org/CorpusID:10716717 (accessed on: 2024-11-20).
47. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [CrossRef]
48. Chen, T.; Yu, W.; Chen, R.; Lin, L. Knowledge-Embedded Routing Network for Scene Graph Generation. *arXiv* **2019**, arXiv:1903.03326. [CrossRef]
49. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980. [CrossRef]
50. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2019**, arXiv:1711.05101. [CrossRef]
51. Al-Rfou, R.; Choe, D.; Constant, N.; Guo, M.; Jones, L. Character-Level Language Modeling with Deeper Self-Attention. In Proceedings of the AAAI'19: AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3159–3166. [CrossRef]
52. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229._13. [CrossRef]
53. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2020**, arXiv:2010.04159. [CrossRef]
54. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2021**, arXiv:2106.09685. [CrossRef]
55. Dettmers, T.; Lewis, M.; Belkada, Y.; Zettlemoyer, L. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. *arXiv* **2022**, arXiv:2208.07339. [CrossRef]