Interactive Scene Graph Analysis for Future Intelligent Teleconferencing Systems

Mingyuan Wu[a], Yuhan Lu[a], Shiv Trivedi[a], Bo Chen[a], Qian Zhou[a], Lingdong Wang[b], Simran Singh[c], Michael Zink[d], Ramesh Sitaraman[b], Jacob Chakareski[c], Klara Nahrstedt[a]

[a] Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL

[b] College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA

[c] College of Computing, New Jersey Institute for Technology, Newark, NJ

[d] Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA

Email: {mw34, yuhanlu2, shivvt2, boc2, qianz, klara}@illinois.edu

{ramesh, lingdongwang}@cs.umass.edu, {simran.singh, jacobcha}@njit.edu, {mzink}@cas.umass.edu

***Abstract*—In a real-life meeting environment, individuals often demonstrate a remarkable ability to selectively focus their attention on specific visual information. This ability allows them to naturally concentrate on a specific region of interest while tuning out others. Understanding and exploiting such selective attention remains unexplored in a user-centric teleconferencing system, where there is a potential to customize video streaming and foveated rendering based on the viewer's attention. This paper proposes a novel user-centric scene analysis module that fully leverages the power of selective attention for online meeting scenarios and recognizes the unequal importance of individual pixels in the videos. The module determines the user's selective attention through the meeting contexts. The contextual representation of the meeting is modeled as a combination of two primary components: proactive user interaction within the system and passive real-time analysis of high-level visual semantics from the scenes. As the meeting progresses, the interactive scene analysis module dynamically updates its contextual representation, offering a dual advantage: (a) Videos can be selectively and adaptively streamed within a user's attention, resulting in bandwidth savings of up to 78 percent. (b) The module enhances the overall quality of the user experience by facilitating higher user interactivity, particularly in meeting-related tasks such as screen sharing, privacy-preserving user blocking, background removal, automatic user attention shift detection, etc. Our interactive scene analysis module makes significant progress toward enabling an efficient, immersive, and intelligent teleconferencing system.**

***Index Terms*—Interactive Scene Analysis, Teleconferencing System, User-centric System**

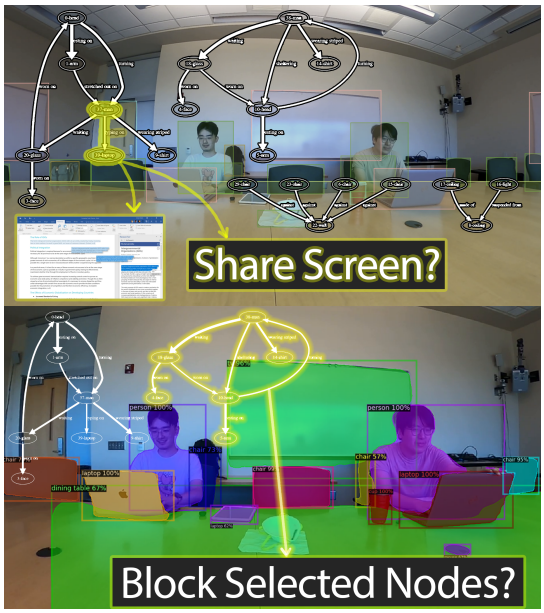

Fig. 1: An interactive example of a user-facing interface, where someone typing on a laptop triggers a prompt to share the screen, or a group of object nodes is selected for foveated rendering or blocking in the conference room. Regions distinguished by color masks indicate associated pixel groups.

I. Introduction

The development of teleconferencing systems has continuously evolved over the past decades due to their critical role in facilitating connections among individuals. Traditional platforms such as Zoom, Microsoft Teams, and Skype offer a cost-effective and convenient way for people to collaborate in real-time, regardless of their physical location. However, these systems have limitations in capturing users' selective attention. They uniformly transmit pixel-level information to users without considering the varying importance of each pixel, based on user interaction and meeting context. This approach results in two primary limitations: (a) limited flexibility for users to manage and direct their visual focus, and (b) constrained interaction pathways that require navigating complex icon lists to perform actions like screen sharing or switching screens.

To address these challenges, our research is motivated by the understanding that **the importance of pixels varies depending on the context**. Certain areas, such as a presenter or a whiteboard during a demo, are more critical to remote participants. The importance of pixels can dynamically change based on user interaction, which reflects interests in specific regions and may fluctuate due to personal interests and privacy concerns. Transmitting all pixels uniformly wastes bandwidth and ignores visual focus. Ideally, a streaming system should detect and utilize discrepancies in pixel importance, consider-

ing both room context and user interactions.

With these motivations, we propose a novel interactive scene analysis module. This module aims to equip video conferencing systems with an understanding of contextual information and natural participant interactions. Contextual information is extracted and summarized in a graph-structured representation, named the scene graph[1]. The scene graph provides users with an interactive interface to identify areas of interest or privacy concerns in the video. It assists the streaming system in determining the importance of each pixel. As illustrated in Figure 1, this scene graph overlays the remote user's view, allowing user interaction to request foveated or blocked functionality, or additional streaming sessions when certain events are detected. In addition to scene graph interactions, the system accommodates direct text and click prompts and produces object masks, supported by the vision foundation model Segment Anything (SAM)[2]. For segmentation efficiency, CondInst[3]-based interactive segmentation is provided as a real-time option.

Our contribution also involves customizing the Scene Graph Generation (SGG) model for teleconferencing. A significant challenge is the absence of a dedicated indoor meeting room dataset with comprehensive ground truth annotations. This data scarcity hinders the possibility of fine-tuning the model. Pre-trained SGG models on large world datasets, such as VG[4], could serve as a naive solution with extra operations of filtering out all the irrelevant category labels of objects and relationships. However, this strategy may not be optimal due to distribution shifts between VG and indoor conference rooms. The distribution shifts are two-fold: First, the VG dataset is a mix of diverse indoor and outdoor scenes, while the desired test environment is completely indoor. Second, the relationship category labels within the dataset are unevenly distributed, creating a long-tail effect. This skews the dataset away from the types of relationship instances pertinent to a meeting room setting, which may be underrepresented in the VG data.

We propose a non-transformer-based SGG model under an adversarial unsupervised domain adaptation framework to tackle this. This model better adapts to indoor conference room scenarios and selected labels without requiring extra ground truth annotation.

II. Related Work

A. Scene Graph Generation Model

Scene Graph Generation (SGG) is a fundamental task that bridges vision and language, garnering attention from both the computer vision and natural language processing communities. Initially presented as visual relation detection in [5], SGG task involves detecting each relationship independently. It was later formulated as a graph representation in [6], a pioneering work incorporating contextual information in images for relationship classification in the scene graph. Some following works, like [7] began to utilize global contextual information for refinement, leveraging the insight that object labels are highly predictive of relation labels. [7] serves as one of the strongest baselines in the pre-transformer era and is adopted in our system. Recently, vision-language models for SGG have become a popular research direction. Works like [8] and [9] propose employing vision-language models for SGG, achieving some level of few-shot ability in other domains. Another line of research, exemplified by [10], addresses the long-tail problem of VG [4] and the ambiguity of the labeling process, arguing that SGG is not well-defined and can lead to uninformative model predictions. Various dataset-balancing methods and modifications to learning objectives have been proposed to tackle this issue.

B. Unsupervised Domain Adaptation

Domain adaptation is relevant when a model is trained on a large-scale source domain with ground truth labels and then applied to an unlabeled target domain. Recent solutions include Adversarial Discriminative training, with theoretical support in [11] that derives the generalization bounds target risks. Approaches like Domain-Adversarial Neural Network [12] and Adversarial Discriminative Domain Adaptation[13] employ adversarial objectives in training to align features from both source and target domains. [14] is among the first works that apply Unsupervised Domain Adaptation to scene graphs and demonstrate promising results in adapting the model to civic domains. In our module, Adversarial Discriminative Domain Adaptation is adopted with a GAN-based loss, leading to a successful model adaptation to meeting room scenarios.

C. Promptable Interactive Segmentation

Recently, SAM [2] from Meta was introduced as a promptable model for segmentation. SAM can encode flexible prompts including points, boxes, text, and masks with a prompt encoder. SAM achieves (amortized) real-time performance in prompt queries, assuming heavyweight image encoding is precomputed. However, this image encoding must be performed on the entire image, making it impractical for mobile devices. The concept of interactive segmentation through clicking predates SAM and the deep learning era, to generate accurate object masks using a minimal number of clicks. In our system, we provide a powerful but non-real-time SAM to prompt user interaction, as well as an efficient real-time interactive segmentation option over Condinst[3]. These segmentation results support functionalities like foveated rendering and blocking as per the user's request. These 2D segmentation algorithms can also be naturally extended to 360-degree video live systems, such as [15].

III. Interactive Scene Analysis

A. User Interactivity

The interactive analysis module supports user interactions through two main categories: passive interactions based on events detected by the scene graph and proactive interactions that can occur at any timestamp during the meeting.

The scene graph representation detects new meeting-related objects and relationships, such as identifying someone typing on a laptop or a person newly seated and starting to write on a whiteboard. For example, if a person is detected to be typing

on his or her laptop, virtual participants will be notified, and prompts will be sent, asking whether they want to view a new streaming session (screen sharing of the computer) as a side session window.

Participants can also interact with the graph, empowering them to select nodes and relationships that fit their interests or they wish to block, at any given time. This feature enables participants to request video content with options like foveated rendering or blocking, benefiting thereby both the system efficiency and their individual preferences. Moreover, participants can provide positive and negative clicks within videos to specify object regions they prefer or wish to avoid, serving as input for interactive segmentation algorithms. Finally, text prompts to specify certain objects are supported, as well.

B. Scene Graph Definition

One of the key components in the interactive scene analysis module is the scene graph. An example visualization of the scene graph is demonstrated in the figure 1. It is a structured representation G of semantic information of important objects and relationships in between. G consists of a set of bounding boxes, a corresponding set of objects, assigning a pre-defined set of class labels to each bounding box, and a set of relationships between those objects. Overall, the SGG model aims at detecting all potential triplets of a subject in a bounding box with a label, relationship label, and object in a bounding box with a label. The result can be easily stored as a JSON file in the system.

C. Model Architecture

Our SGG model architecture is heavily inspired by MotifNet [7] architecture, which was state-of-the-art before the transformer era. Building on the idea that global context in an image aids the SGG task, MotifNet first detects objects in the images and efficiently encodes the global context between local predictors. (i.e. objects and relationships.). The encoding of the global context is achieved by a bidirectional Long Short-term Memory Network (LSTM) [16]. In the final stage, the representations of the global contexts and the local predictors are used to predict objects and relationship labels. Our model follows a similar approach but extends it under an adversarial domain adaptation framework. Here, object and relationship features are forwarded to a neural network-based discriminator instead of being used directly for classification. Further insights into this domain adaptation framework will be discussed in subsequent sections.

Stage I Object Detection predicts bounding boxes, corresponding feature vectors, and probabilities of each category label. Arbitrary object detector architecture can be used in this stage. We use faster-RCNN [17] for its fast inference speed.

Stage II Context objects Encoding passes all proposed object regions with feature vectors and class probabilities into a bidirectional LSTM, which efficiently encodes object context information into a single vector representation.

Stage III Object Decoding uses the object context information to decode labels for each proposed bounding region, conditioned on previous decoded labels with another LSTM. Hidden states of the LSTM can be passed into another learned fully connected layer for object labels.

Stage IV Relations (Edge) Context Encoding employs an additional bidirectional LSTM to encode the context information between relationships and objects.

Stage V Relations (Edge) Context Decoding utilizes an LSTM to decode context information for pair-wise relationships among a quadratic number of object pairs.

Stage VI Domain Discriminator is a feed-forward neural network that classifies whether an object or relationship representation comes from the source or the target domain, addressing data or label distribution shifts.

Stage VII Final Prediction involves an extra fully connected layer to project object and relationship feature representations onto a 1-D vector. A softmax is applied to produce a probability distribution of all object and relationship labels.

D. Unsupervised Domain Adaptation

Inspired by the success of Adversarial Discriminative Domain Adaptation (ADDA) [13] applications in other deep learning-based models, and an attempt in [14] that tries integrating the ADDA with SGG models, we introduce an adversarial training pipeline that adapts the SGG model trained on VG[4] dataset to a domain-specific setting of an indoor conference room.

ADDA training is divided into different stages, similar to standard GAN loss [18]. Initially, the source encoder is pre-trained using source data from the VG dataset [4]. In the second stage, the model performs adversarial adaptation by jointly learning a target encoder and a discriminator that predicts domain labels, with the source encoder frozen. During testing, target images are mapped with the target encoder to a shared feature space to perform tasks like predicate classification. As stated in ADDA literature [13], we use the same general adversarial loss function, the standard GAN loss.

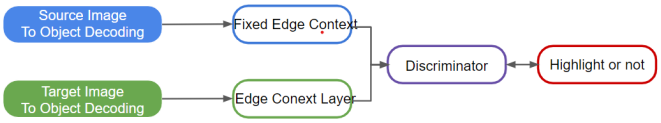

Fig. 2: Stage II of ADDA to deal with label shifts.

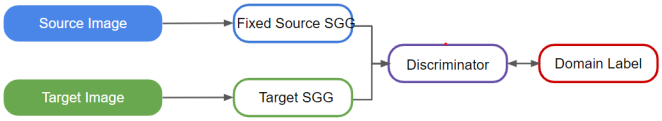

Fig. 3: Stage II of ADDA to deal with data distribution shifts.

We refer readers to more details of the general adversarial loss function in the original ADDA paper [13]. When integrating ADDA with SGG, we have two design choices for the stage II adversarial training when applying the adversarial

training to the SGG model. As mentioned in the previous section, both label shifts and data shifts between train/test data result in a performance drop for the model. To address label shifts, the discriminator works on each edge context (relationship representation) between object pairs. It is learned to distinguish between "highlighted for meeting related relationships" or "non-highlighted" and backpropagates gradient into the edge context layer in motifNet. This procedure is demonstrated in Figure 2. For data shifts, the approach works similarly to other recognition tasks relying on CNN features, shown in Figure 3.

E. Interactive Segmentation

Prompting, recognized as a natural and effective mode of interaction, has attracted significant attention since the evolution of ChatGPT. Within the interactive scene analysis, users can input prompts, either in text or through clicks, to achieve their desired outcomes. As illustrated in Figure 4, users can provide positive or negative clicks to accurately segment areas they wish to foveate or block. Figure 5 further demonstrates the power of text prompts. SAM leverages semantic knowledge from natural language prompts to infer and identify the object with the descriptive term "red thing", instead of directly specifying the identification of a red cup on the table.

The promptable interactive segmentation module offers two models: the computationally intensive SAM[2] from Meta for text and click prompts, and the real-time efficient CondInst[3] for click prompts only. Users can extract important frames and prompt them with text or clicks in SAM. However, this choice cannot be made in real-time due to SAM's heavyweight image encoder (based on the Vision Transformer). For real-time functionality of foveated rendering and blocking, users can switch to interactive CondInst for its efficiency. While we made efforts to enable CondInst to run in a real-time setting, we opted not to include this in the paper as we could not evaluate its quantitative performance on our indoor meeting room dataset lacking ground truth annotations for masks.

Fig. 4: Positive Click in Green and Negative Click in Red.

F. Implementation Details

SGG Model Details. The discriminator used in training consists of three fully connected layers with 2048 hidden units each. Each layer is followed by a batch normalization and ReLU activation. A dropout rate of 0.5 is applied in the training. The model and the discriminator are trained using ADAM with learning rates of 0.001 and 0.01. We use a faster RCNN detector model with a resnet50fpn backbone.

Selection of Categories. We manually select conference event-related categories of object relationships. Among these, the object labels "chair, table, notebook, laptop, board, person" and relationship labels "write on, type on, sit, stand on, beside (the whiteboard)," are representative of conference progress. These labels are essential in initiating, terminating, or creating a new side window session operation during the meeting. In addition to these labels, other meeting room events can be preselected from the original dataset to benefit virtual participants. By leveraging these labels and events, the teleconferencing system can potentially provide a more immersive and interactive experience for all participants.

Integration with Teleconferencing. A server with a powerful GPU plays a crucial role in constantly running machine learning-based scene analysis as a service. The server receives a stream from a room camera and periodically executes the SGG Models, transmitting the scene graph representation to a remote client for interactive visualization. Moreover, when users initiate foveated rendering or blocking interactions sent from the clients, the server must execute an interactive frame-level instance segmentation algorithm in real-time and provide pixel groups of instance masks as requested by the clients.

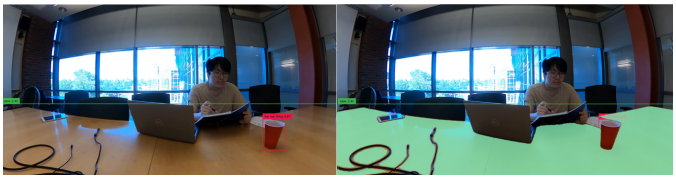

Fig. 5: With the prompt: detect the red thing on the table.

IV. Experiments

We collected a video dataset covering events in a conference room. The dataset consists of 11 full HD video clips at 60 fps with a total length of 997 seconds. We use 8 clips for unsupervised training and 3 for evaluation. We label conference-related relationships and objects at selected frames for evaluation of the model with Label Studio.

Previous studies have evaluated scene graph generation models in PREDCLS, SGCLS, and SGGEN tasks. SGGEN is unique in considering the model's performance in detecting all objects and relationships from scratch. In an SGGEN task, the model has to simultaneously detect the set of objects and predict the right predicate for each pair of objects. **SGGEN** was adopted because our system needs to detect object relationships from scratch rather than relying on two given object ground truths. The conventional metric for evaluating SGG is Recall@K(R@K), which computes the fraction of times the correct relationship is predicted in the top K predictions with the highest probability. Recall @ 20 and Recall @

10 are measured in Table I when testing the model on our dataset with our selected object and relationship labels in the conference room. The results demonstrate that our customized model outperforms the pre-trained model. The performance gain might increase when the training dataset scales.

During inference, the SGG model takes an average of 0.33 seconds for a single 4K frame on an Nvidia RTX 3090, achieving an inference speed of 3 frames per second. It is important to note that methods related to Vision Transformers [19] are prohibitively resource-intensive in the context of the real-time scene analysis setting. They exhibit an inference speed of 0.5 seconds per low-resolution image on TPU for image classification tasks, which is notably slower than the demands of real-time SGG. Regarding segmentation, the SAM model operates at a speed of 0.452 seconds per frame on Nvidia A100. On the other hand, a lightweight segmentation based on Condinst can achieve a speed of 0.03 seconds per frame, making it suitable for real-time applications.

TABLE I: Adversarially/Pre Trained Model Accuracy.

Setting	Pre-trained	w/domain shift	w/label shift
SGGEN Recall20	48.6	**51.2**	50.3
SGGEN Recall10	44.3	45.0	**45.2**

In our video dataset, we manually labeled important meeting events and simulated some user click prompts to imitate foveated or blocking requests. In the most optimal simulation scenario, where a user consistently focuses on a specific person related to an important event, foveated rendering of that person onto a static meeting room background resulted in a remarkable 78% bandwidth savings in the video streams.

V. Conclusion

In this paper, we introduce an innovative interactive scene analysis module designed for teleconferencing. This module enhances the teleconferencing system by enabling functionalities like user-focused foveated streaming, privacy-preserving blocking, and efficient selective streaming. It comprises a scene graph generation model customized for indoor meeting room scenarios, along with real-time interactive segmentation. Through experiments, we demonstrate the module's efficiency, accuracy, and feasibility. This represents a significant stride towards creating an immersive, interactive, and intelligent teleconferencing system.

VI. Acknowledgment

This work was funded by the National Science Foundation under grant contracts NSF 1835834, NSF 1900875, NSF 2106592. This work was also supported in part by the NSF under awards CCF-2031881, ECCS-2032387, CNS-2040088, CNS-2032033, CNS-2106150, CNS-2106463, CNS-1901137 and by the NIH under award R01EY030470, and by the Panasonic Chair of Sustainability at NJIT. Any results and opinions are our own and do not represent the views of the National Science Foundation.

References

[1] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3668–3678.

[2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[3] Z. Tian, B. Zhang, H. Chen, and C. Shen, "Instance and panoptic segmentation using conditional convolutions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 669–680, 2023.

[4] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332

[5] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European Conference on Computer Vision*, 2016.

[6] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," 2017. [Online]. Available: https://arxiv.org/abs/1701.02426

[7] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Conference on Computer Vision and Pattern Recognition*, 2018.

[8] Y. Yao, Q. Chen, A. Zhang, W. Ji, Z. Liu, T.-S. Chua, and M. Sun, "Pevl: Position-enhanced pre-training and prompt tuning for vision-language models," in *Proceedings of EMNLP*, 2022.

[9] Y. Yao, A. Zhang, Z. Zhang, Z. Liu, T.-S. Chua, and M. Sun, "Cpt: Colorful prompt tuning for pre-trained vision-language models," 2021. [Online]. Available: https://arxiv.org/abs/2109.11797

[10] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, "Learning to compose dynamic tree structures for visual contexts," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6619–6628.

[11] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1–2, p. 151–175, may 2010. [Online]. Available: https://doi.org/10.1007/s10994-009-5152-4

[12] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[13] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," 2017. [Online]. Available: https://arxiv.org/abs/1702.05464

[14] S. Kumar, S. Atreja, A. Singh, and M. Jain, "Adversarial adaptation of scene graph models for understanding civic issues," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2943–2949. [Online]. Available: https://doi.org/10.1145/3308558.3313681

[15] Q. Zhou, Z. Yang, H. Guo, B. Tian, and K. Nahrstedt, "360broadview: Viewer management for viewport prediction in 360-degree video live broadcast," in *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, ser. MMAsia '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3551626.3564939

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, oct 2020. [Online]. Available: https://doi.org/10.1145/3422622

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.