Proceedings of CONF-CDS 2025 Symposium: Data Visualization Methods for Evaluatio
DOI: 10.54254/2755-2721/2025.PO24682

A Survey of Transformer Optimization Techniques: Progress and Challenges from Computational Efficiency to Multimodal Fusion

Chuhao Xiong

Faculty of Innovation Engineering, Macau University of Science and Technology, Macau, China
1220029107@student.must.edu.mo

Abstract. Since its proposal in 2017, the Transformer model has achieved revolutionary breakthroughs in natural language processing and even in computer vision tasks. However, its huge number of parameters and high computational complexity have posed substantial difficulties in training and inference efficiency, model knowledge updating, and multimodal information fusion. This paper reviews recent research progress on Transformer optimization techniques, including: (1) Structural optimization and computational efficiency – model architecture improvements, pruning compression, and efficient attention mechanisms to reduce computational cost; (2) Parameter-efficient fine-tuning and task adaptation – new fine-tuning methods with high parameter efficiency, as well as few-shot and zero-shot learning paradigms, to improve adaptability in low-resource and multi-task scenarios; (3) External knowledge integration – incorporating knowledge graphs, retrieval-based external memory, etc., into Transformers to fill knowledge gaps and enhance commonsense reasoning; (4) Multimodal fusion – designing cross-modal Transformer architectures and alignment mechanisms to effectively fuse information from modalities such as vision and language. Analyze representative methods, underlying principles, and experimental results for each direction, discuss the main challenges, and predict forthcoming research directions, such as unified efficient attention theories, trustworthy knowledge injection mechanisms, green AI training strategies, and next-generation interpretable Transformer architectures.

Keywords: Transformer, Model optimization, Computational efficiency

1. Introduction

Since Vaswani et al. introduced the Transformer architecture in the landmark paper "Attention Is All You Need" [1], Transformer models have fundamentally transformed AI research. By exclusively depending on self-attention, Transformers removed the sequential dependency of RNNs and enabled highly parallelizable training, leading to breakthrough results in machine translation and other sequence modeling tasks. In the past few years, Transformers have rapidly scaled from the original model (~65 million parameters) to extremely large models with hundreds of billions of parameters (e.g., GPT-3 with 175 billion [2]) and even trillion-scale models (e.g., Google's Switch Transformer

DOI: 10.54254/2755-2721/2025.PO24682

with 1.6 trillion parameters using sparse expert mixing [3].)This scaling has driven paradigm shifts in natural language processing (NLP) and even computer vision, as seen in Vision Transformer (ViT), which applies Transformers to image patches and rivaled convolutional networks [4].]At the same time, these massive models exhibit impressive capabilities (e.g., GPT-3 demonstrating strong few-shot learning), but their size and complexity also bring huge computational costs and make deployment on real-time or resource-constrained systems very difficult. In addition, Transformers pre-trained on static corpora have "common sense blind spots" and factual obsolescence – they may lack knowledge not present in training data or struggle to update with new facts. Integrating multiple modalities (text, images, speech, etc.) into one Transformer is also challenging, due to the need to align heterogeneous features and balance learning objectives across modalities. All these issues limit the broader application of Transformers. To tackle these issues, the research community and industry have proposed a variety of innovative Transformer optimization techniques. Several surveys have reviewed the evolution of Transformer models. For instance, Lin et al [5]. systematically summarized improvements of various "X-former" variants in architecture, pre-training, and applications, and proposed a new taxonomy of Transformer models. yTay et al [6]. focused on efficient Transformers, surveying approaches such as Reformer, Linformer, Performer, Longformer, etc., that reduce computational and memory complexity. Xu et al [7]. reviewed Transformer-based multimodal learning techniques and challenges. Compared to existing surveys, this paper centers on a range of optimization strategies for Transformers from improving computational efficiency to knowledge acquisition and multimodal expansion. The author discusses multiple aspects including model structure, training and fine-tuning, external knowledge injection, and multimodal fusion. The author not only reviews the principles of representative methods in each area, but also summarizes their experimental performance on typical tasks, in order to give a complete and in-depth understanding of Transformer optimization techniques. The remainder of this paper is organized into four main sections covering structural optimizations, parameter-efficient tuning, knowledge integration, and multimodal fusion, followed by a conclusion with a future outlook.

2. Structural optimization and computational efficiency

The powerful expressiveness of Transformers comes at the price of high computational cost. self-attention mechanism has quadratic $O(n^2)$ complexity in sequence length, and model size grows substantially with more layers and larger hidden dimensions. processing long texts or high-resolution images can thus be very inefficient, and training large-scale models demands enormous computing power and energy. To mitigate these issues, a great deal of research has focused on optimizing Transformer architectures and improving computational efficiency, by cutting redundant computation and approximating the expensive components. In general, efforts in this area can be grouped into three aspects: model compression, efficient attention mechanisms, and training/inference optimization.

Model Compression and Pruning: Large Transformers often contain substantial parameter redundancy. Pruning and quantization are classic compression techniques to reduce model size. Pruning removes weights or even entire structures that have minimal impact on model performance (e.g., unstructured weight pruning, or structured pruning of attention heads, neurons, or whole layers).ang et al. conducted a comprehensive survey of static and dynamic pruning algorithms and showed that low-bit quantization can significantly shrink model size and speed up inference while maintaining accuracy [8]. For example, BERT models can prune 40–90% of parameters with usually under 5% performance drop [8]. Furthermore, knowledge distillation compresses models by having a small student model learn to mimic a large teacher Transformer. For instance, DistilBERT was

DOI: 10.54254/2755-2721/2025.PO24682

distilled from BERT and achieves ~97% of BERT's performance with only half the number of parameters [9]. Another approach is parameter sharing – ALBERT ties weights across layers to greatly reduce parameters while retaining performance comparable to BERT [10]. By combining pruning, quantization, distillation, and parameter sharing, researchers have managed to create lightweight Transformers that can be deployed on resource-constrained devices (e.g., mobile and edge computing) without drastic loss of capability.

Lightweight architecture improvement: In addition to compressing the existing model, researchers have also proposed an alternative Transformer architecture with higher parameter efficiency or computational efficiency. For example: (1) General-purpose transformers adopt a cyclic mechanism to share parameters across depths, effectively reducing model parameters and achieving adaptive computing time [11]. (2) Transformer-XL introduces segment-level recursion to capture remote dependencies outside the fixed window and reuse hidden states to improve efficiency [12]. (3) For visual tasks, hierarchical transformers such as Swin Transformer use patch merging (down sampling) layers to reduce spatial resolution. Increasing depth significantly reduces the computational load of subsequent layers [13]. These architectural modifications have improved efficiency while retaining (or even enhancing) the model performance in their respective fields.

Effective attention mechanism: The secondary cost of complete self-attention in long sequences has stimulated a large number of methods to improve the efficiency of attention. The research has determined the characteristics of the attention matrix that can be utilized (locality, sparsity, low-rank structure, etc.). Overall, three methods have been explored:

Structured sparse attention: Use a fixed pattern to limit the attention range of each token. Examples include a sparse transformer with predefined stride patterns [14], a Longformer that uses a local sliding window (for example, a window size of 512) along with some global tokens [15], and a Big Bird that combines local, random, and global attention and has been proven to be approximately full attention [16]. These methods reduce the complexity to linear or $O(n \cdot w)$, where w is the window size.

Dynamic sparse attention: Allowing the model to learn sparse patterns or group tokens dynamically. Reformer uses locality-sensitive hashing to bucket similar tokens and only attends within buckets, greatly lowering memory usage [17]. Routing Transformer performs k means clustering of tokens and restricts attention to tokens in the same cluster [18]. Sinkhorn Transformer learns to reorder sequences via differentiable sorting and then apply block-sparse attention [19]. Such methods achieve sparsity adaptively based on content.

Low-rank and approximate attention: Approximating the full attention matrix with lower dimensional structures. Linformer projects sequences into a lower-dimensional representation to achieve linear time attention [20]. Performer uses random feature mappings to approximate the SoftMax kernel, providing linear complexity with theoretical error bounds [21]. These approaches leverage the low-rank nature of attention matrices to speed up computation.

By incorporating these efficient attention mechanisms, Transformers can handle much longer sequences or higher-resolution inputs with reduced time and space complexity. In addition, a range of training **optimization strategies** have been proposed (often in conjunction with the above): for example, gradient checkpointing and memory optimization to enable training larger models on limited hardware, mixed-precision and distributed pipeline parallelism to accelerate training, and at inference time techniques like 8-bit or 4-bit quantization, knowledge distillation to smaller student models, and input length truncation to lower latency. Recent studies show that end-to-end optimization across data, models, and systems can dramatically improve efficiency, bringing us closer to green AI training and deployment.

DOI: 10.54254/2755-2721/2025.PO24682

3. Parameter-efficient fine-tuning and task adaptation

Pre-trained Transformers are typically fine-tuned for downstream tasks by updating all their parameters, which becomes impractical for extremely large models. This has led to research in **parameter-efficient fine-tuning (PEFT)** techniques that adapt a model to new tasks with only a small fraction of parameters updated. Such methods aim to retain the full model's knowledge and capacity while avoiding the cost of full fine-tuning for each task.

One line of work adds small adapter modules to the Transformer and leaves the original weights mostly frozen. For example, Houlsby et al. introduced adapter layers between Transformer sublayers, with only the adapter parameters being trained for each task. Adapter-based methods have proven effective in multi-task and cross-lingual scenarios: by keeping the large pre-trained model shared across tasks and only learning task-specific adapters, they enable efficient fine-tuning and storage of multiple task models. AdapterDrop further improves adapter efficiency by dropping certain adapters during training/ inference to trade off performance and speed [24]. These approaches achieve performance on par with full fine-tuning in many cases while updating far fewer parameters.

Another direction exploits the low-rank properties of parameter updates. LoRA (Low-Rank Adaptation) injects trainable low-rank matrices into each layer's weights and keeps the main weights fixed [22]. Despite updating only, a tiny percentage of parameters (e.g. 0.1%), LoRA has shown that on NLP tasks its fine-tuned models can match the accuracy of full-model fine-tuning, while being much more parameter efficient. Similarly, BitFit proposes to only fine-tune the bias terms of the Transformer [23]. Remarkably, BitFit can achieve nearly the same performance as full fine-tuning on medium-sized datasets, although on some data-rich tasks it is slightly lower – a gap that can often be closed by combining it with other techniques.

Prefix- or prompt-tuning offers a parameter-free alternative to conventional fine-tuning: the model stays frozen and learns only a short, continuous prompt that conditions it on a new task. Li & Liang [25]. showed that a small prompt vector can steer GPT-3 to new tasks with no weight updates Across benchmarks, parameter-efficient methods rival full fine-tuning: LoRA matches GLUE scores while training orders of magnitude fewer weights [22]; adapters support cross-lingual or continual learning by slotting tiny task modules into the backbone; and BitFit, which tweaks merely ~0.1 % of parameters, often suffices, with minor gaps recoverable by hybrid schemes [23]. Such lightweight tuning means a handset can download only a prompt or adapter to customize a large model, and a cloud service can host one shared backbone plus many small heads, drastically cutting storage and update costs while retaining state-of-the-art performance.

4. External knowledge integration

Despite training on massive text corpora, Transformers can still struggle with facts and commonsense knowledge that were never seen or not memorized during training. The model's knowledge is also static once training is done, making it hard to update with new information. To address these limitations, researchers have explored methods to **integrate external knowledge sources** into Transformers, with the goal of equipping models with knowledge beyond what is in their frozen parameters and improving their reasoning accuracy and credibility.

Knowledge-graph injection. K-BERT expands inputs with retrieved KG triples and uses soft positional masks to limit interference, improving medical and financial QA over a plain BERT [26]. K-Adapter freezes the backbone and trains parallel adapter modules—e.g., factual (Wikidata) or linguistic (dependency parse)—that plug into each layer; combining adapters yields further gains on

DOI: 10.54254/2755-2721/2025.PO24682

relation classification, entity typing and QA [27]. Overall, adding knowledge graphs gives the model extra context, helping it track complex connections between entities that regular memory might overlook.

Knowledge-enhanced pre-training. Baidu's **ERNIE** augments BERT objectives with entity/relation prediction to unify text and KG representations [28], while **KnowBERT** inserts an entity-embedding layer and is jointly trained for language modeling and entity linking [29]. Such methods "bake in" a slice of external knowledge, lifting QA and disambiguation versus vanilla BERT. A survey by Hu et al. shows that effective knowledge infusion must match source type and architecture; simply adding extra pre-training tasks rarely suffices [30]. Tailored fusion modules—adapters, graph nets, or retrieval links—are often required.

Retrieval-augmented models. Instead of embedding facts in parameters, **REALM** jointly trains a Transformer and differentiable retriever that fetches helpful passages during masked-LM pre-training, boosting open-domain QA by 4-16 points [31]. **RAG** couples a BART generator with a dense retriever; answers are conditioned on retrieved passages, producing verifiable long-form responses [32]. Retrieval thus gives a live, updatable memory without retraining the model.

External memory. Recent work equips Transformers with writable memories. Agents can store salient observations during reinforcement learning and re-attend to them later; **MemGPT** logs dialogue notes to external memory, effectively extending the context window for long conversations. Such memories let knowledge evolve within a session and mitigate forgetting.

Summary. Injecting knowledge graphs, using retrieval, or adding memory all improve factuality and reasoning depth, overcoming limits of static, data-only pre-training. The open challenge is trustworthy use: models must judge source reliability, resolve conflicts, and expose uncertainty. Current research therefore explores credibility metrics, misinformation filters, and evidence-aware generation—critical steps toward dependable, knowledge-rich Transformers.

5. Multimodal fusion

Transformers, first designed for text, now underpin multimodal systems that must fuse heterogeneous signals while enabling cross-modal reasoning. Early vision–language models adopted a dual-stream design: ViLBERT and LXMERT run separate Transformer encoders over image regions and text, then exchange information through stacked cross-attention, achieving strong Visual-Question-Answering (VQA) and captioning scores by aligning objects with words [33,34].

Later work shifted to single-stream fusion. UNITER concatenates visual and textual tokens into one sequence and applies ordinary self-attention over the joint input, slightly outperforming dual-stream baselines on image–text retrieval and VQA while using a simpler architecture [35].

Model–scale success depends heavily on **large-scale multimodal pre-training.** VinVL enlarged pre-training corpora and improved object detection to learn richer visual vocabularies, setting state-of-the-art results on several benchmarks [36]. OpenAI's CLIP used contrastive learning on 400 M image–text pairs scraped from the web, producing a shared embedding space that enables zero-shot ImageNet classification at ResNet-50 accuracy without task-specific fine-tuning [37]. BLIP mixes discriminative (retrieval) and generative (captioning, VQA) objectives in a single framework, yielding strong generalization across understanding and generation tasks [38].

Research is now pursuing unified multimodal Transformers that process more than two modalities in one backbone. Meta-AI's FLAVA trains on images, text, and paired data; modality-type embeddings let a single encoder act as image-only, text-only, or jointly multimodal, and it delivers competitive results across diverse tasks [39]. Google's Perceiver and Perceiver IO go further, treating images, audio, point clouds, and text as generic sequences projected into a latent

DOI: 10.54254/2755-2721/2025.PO24682

array processed by cross-attention, matching or surpassing specialized models in each domain [40]. Such architectures foreshadow a single foundation model capable of unified reasoning over multi-sensory input.

Challenges remain, Efficiently scaling to long videos or high-resolution 3-D scenes requires hybrid attention or sparse computation while robust cross-modal alignment still demands curated data or auxiliary supervision. Web-scale pre-training, as in CLIP, introduces noise and bias, so ensuring that learned semantics are reliable—not spurious correlations—remains an active area. Nevertheless, multimodal Transformers have already overtaken bespoke systems in captioning, VQA, retrieval, and entailment, grounding fine-grained correspondences between pixels and words. With continuous progress in unified architectures, data curation, and efficient attention, they are set to become for multimodal AI what BERT and GPT are for NLP.

6. Conclusion

Transformer optimization remains a vibrant and crucial area of research as researchers push the limits of model size, efficiency, and applicability. In this survey, the author reviewed advances ranging from architectural tweaks for computational efficiency to new fine-tuning methods, knowledge integration techniques, and multimodal model designs. These innovations have substantially improved the practicality and performance of Transformer models: modern Transformers can run faster, adapt with fewer parameters, access external knowledge, and handle multiple modalities, compared to their initial incarnation.

However, several challenges and open questions persist, pointing to promising future directions:

Unified Efficient Attention Theory: There is ongoing work to develop a unified theoretical framework for efficient attention mechanisms. Most current efficient Transformers are designed case by case (sparse patterns, low-rank approximations, etc.). A deeper understanding of why attention matrices are sparse or low-rank in practice could lead to a general solution that adapts to different data characteristics. The goal is to maintain full attention quality with significantly lower complexity in a theoretically grounded way.

Trustworthy Knowledge Injection: Future research will explore techniques for validating and updating the knowledge that a model uses, handling conflicts between model predictions and external facts, and enabling models to explain which knowledge was used for a given prediction. Developing trustworthy and interpretable knowledge integration mechanisms (potentially with symbolic reasoning components or neurosymbolic hybrids) is an exciting avenue.

Green and Sustainable Training: The trend of ever larger Transformers raises concerns about energy consumption and carbon footprint. There is a growing movement toward “green AI” finding ways to achieve comparable performance with less data and compute. Future optimizations may include smarter pre-training curricula, model recycling and compression techniques, federated or distributed training strategies that minimize redundancy, and hardware software co-design (such as leveraging new accelerators or sparsity support). The aim is to enable training next-generation Transformers in a more resource-efficient and environmentally friendly manner.

Next-Generation Interpretability: Large Transformers often act as black boxes. Building on optimization techniques, future models might incorporate architecture changes that improve interpretability (for example, modules that explicitly model causal relationships or logic, which could be inspected). Research on probing attention and attributions will continue, but there may also be architectural innovations that make Transformers more inherently transparent without sacrificing performance.

DOI: 10.54254/2755-2721/2025.PO24682

In conclusion, the evolution of Transformer optimization is far from over. Each of the areas discussed efficiency, parameter tuning, knowledge, multimodality – continues to be actively researched, and breakthroughs in one area often complement others. By combining efficient architectures, adaptive fine tuning, external knowledge, and multimodal understanding, we are moving toward Transformer-based AI systems that are not only powerful and general, but also efficient, adaptable, and knowledgeable. Such systems hold the promise of widespread deployment across domains, from tiny embedded devices to comprehensive multi-modal assistants. The challenges are significant, but the rapid progress to date gives reason to be optimistic that the Transformer will continue to be the workhorse of AI, increasingly optimized to meet both our practical constraints and ambitious goals.

References

[1] Vaswani A., Shazeer N., Parmar N., et al. (2017). Attention Is All You Need. Proceedings of NeurIPS 2017, 5998-6008.
[2] Brown T., Mann B., Ryder N., et al. (2020). Language Models are Few-Shot Learners. NeurIPS 2020, 1877-1901.
[3] Fedus W., Zoph B., Shazeer N. (2022). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. Journal of Machine Learning Research, 23(120), 1-39.
[4] Dosovitskiy A., Beyer J., Kolesnikov A., et al. (2021). An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. ICLR 2021.
[5] Lin T., Wang Y., Liu X., Qiu X. (2022). A Survey of Transformers. AI Open, 3, 111-132.
[6] Tay Y., Dehghani M., Bahri D., Metzler D. (2020). Efficient Transformers: A Survey. arXiv: 2009.06732.
[7] Xu P., Zhu X., Clifton D. (2023). Multimodal Learning with Transformers: A Survey. IEEE TPAMI (early access).
[8] Liang T., Glossner J., Wang L., Shi S. (2021). Pruning and Quantization for Deep Neural Network Acceleration: A Survey. Neurocomputing, 461, 370-403.
[9] Sanh V., Debut L., Chaumond J., Wolf T. (2019). DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. NeurIPS 2019 Energy-Efficient ML Workshop.
[10] Lan Z., Chen M., Goodman S., et al. (2020). ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. ICLR 2020.
[11] Dehghani M., Gouws S., Vinyals O., Uszkoreit J., Kaiser Ł. (2019). Universal Transformers. ICLR 2019.
[12] Dai Z., Yang Z., Yang Y., et al. (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. ACL 2019, 2978-2988.
[13] Liu Z., Lin Y., Cao Y., et al. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. ICCV 2021, 10012-10022.
[14] Child R., Gray S., Radford A., Sutskever I. (2019). Generating Long Sequences with Sparse Transformers. arXiv: 1904.10509.
[15] Beltagy I., Peters M. E., Cohan A. (2020). Longformer: The Long-Document Transformer. arXiv: 2004.05150.
[16] Zaheer M., Guruganesh G., Dubey K. A., et al. (2020). Big Bird: Transformers for Longer Sequences. NeurIPS 33, 17283-17297.
[17] Kitaev N., Kaiser Ł., Levskaya A. (2020). Reformer: The Efficient Transformer. ICLR 2020.
[18] Roy A., Saffar M., Vaswani A., Grangier D. (2021). Efficient Content-Based Sparse Attention with Routing Transformers. TACL, 9, 53-68.
[19] Tay Y., Bahri D., Yang L., et al. (2020). Sparse Sinkhorn Attention. ICML 2020, 9438-9447.
[20] Wang S., Li B. Z., Khabsa M., Fang H., Ma H. (2020). Linformer: Self-Attention with Linear Complexity. arXiv: 2006.04768.
[21] Choromanski K., Likhosherstov V., Dohan D., et al. (2021). Rethinking Attention with Performers. ICLR 2021.
[22] Hu E. J., Shen Y., Wallis P., et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv: 2106.09685.
[23] Ben Zaken E., Ravfogel S., Goldberg Y. (2022). BitFit: Simple Parameter-Efficient Fine-Tuning for Transformer-Based Masked Language Models. ACL 2022, 1-9.
[24] Rücklé A., Glavaš G., Reimers N., Gurevych I. (2021). AdapterDrop: On the Efficiency of Adapters in Transformers. EMNLP 2021, 7930-7946.
[25] Li X. L., Liang P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. ACL 2021, 4582-4597.

DOI: 10.54254/2755-2721/2025.PO24682

[26] Liu W., Zhou P., Zhao Z., et al. (2020). K-BERT: Enabling Language Representation with Knowledge Graph. AAAI 2020, 2901-2908.
[27] Wang R., Tang D., Duan N., et al. (2021). K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. Findings of ACL-IJCNLP 2021, 1405-1418.
[28] Zhang Z., Han X., Liu Z., et al. (2019). ERNIE: Enhanced Language Representation with Informative Entities. ACL 2019, 1441-1451.
[29] Peters M. E., Neumann M., Logan IV R. L., et al. (2019). Knowledge Enhanced Contextual Word Representations (KnowBERT). EMNLP 2019, 43-54.
[30] Hu L., Liu Z., Zhao Z., et al. (2023). A Survey of Knowledge-Enhanced Pre-Trained Language Models. ACM Computing Surveys (in press), arXiv: 2211.05994.
[31] Guu K., Lee K., Tung Z. D., Pasupat P., Chang M. (2020). REALM: Retrieval-Augmented Language Model Pre-Training. ICML 2020, 3929-3938.
[32] Lewis P., Perez E., Karpukhin A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP. NeurIPS 33, 9459-9474.
[33] Lu J., Batra D., Parikh D., Lee S. (2019). ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations. NeurIPS 32, 13-23.
[34] Tan H., Bansal M. (2019). LXMERT: Learning Cross-Modality Encoder Representations from Transformers. EMNLP 2019, 5100-5111.
[35] Chen Y.-C., Li L., Yu L., et al. (2020). UNITER: UNiversal Image-TExt Representation Learning. ECCV 2020, 104-120.
[36] Zhang P., Li X., Hu X., et al. (2021). VinVL: Revisiting Visual Representations in Vision-Language Models. CVPR 2021, 5579-5588.
[37] Radford A., Kim J. W., Hallacy C., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. ICML 2021, 8748-8763.
[38] Li J., Li D., Xiong C., Hoi S. (2022). BLIP: Bootstrapping Language-Image Pre-Training for Unified Vision-Language Understanding and Generation. ICML 2022, 12888-12900.
[39] Singh A., Goswami V., Srinivasan P., et al. (2022). FLAVA: A Foundational Language and Vision Alignment Model. CVPR 2022, 15638-15650.
[40] Jaegle A., Borgeaud S., Alabdulmohsin I., et al. (2021). Perceiver: General Perception with Iterative Attention. ICML 2021, 4651-4664.