

# Modern Computer Vision from CNNs to Foundation Models: A Unified Framework for Representation Learning

Abdelmalik Moujahid

`abdelmalik.moujahid@unir.net`

Universidad Internacional De La Rioja

Fadi Dornaika

University of the Basque Country

Salah-Eddine Mechkouri

Mohammed V University

Jesús Cigales Canga

Universidad Internacional De La Rioja

---

## Research Article

**Keywords:** Computer Vision, Vision Transformers, Foundation Models, Self-Supervised Learning, Multimodal Learning, Diffusion Models, Rectified Flow

**Posted Date:** August 4th, 2026

**DOI:** <https://doi.org/10.21203/rs.3.rs-10436723/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

**Additional Declarations:** No competing interests reported.

---

# Modern Computer Vision from CNNs to Foundation Models: A Unified Framework for Representation Learning

Abdelmalik Moujahid<sup>1,\*</sup>, Fadi Dornaika<sup>3,4</sup>, Salah-Eddine Mechkouri<sup>2</sup> and Jesús Cigales Canga<sup>1</sup>

<sup>1</sup>Universidad Internacional de La Rioja (UNIR), Logroño, Spain

<sup>2</sup>Université Mohammed V, Rabat, Morocco

<sup>3</sup>Universidad del País Vasco (EHU)

<sup>4</sup>Ikerbasque, Basque Foundation for Science

July 23, 2026

## Abstract

Modern computer vision is increasingly driven by the search for representations that generalize across tasks, modalities, and learning settings, rather than excelling at a single benchmark. This has produced a broad collection of architectures and paradigms, including convolutional neural networks (CNNs), Vision Transformers (ViTs), foundation models, and diffusion-based generative models. Despite differing in design and training objectives, these approaches increasingly draw on a common set of principles: representation learning, self-attention, self-supervision, multimodal alignment, and generative modeling. As a result, the boundaries between discriminative, multimodal, and generative paradigms are becoming progressively less distinct.

This survey examines that convergence through the lens of representation learning. We first introduce a common mathematical framework for describing vision architectures and learning paradigms across settings, then trace the evolution from CNNs and Vision Transformers, through hybrid CNN-Transformer architectures, foundation models, and multimodal vision-language systems, to recent generative approaches, including diffusion models, Diffusion Transformers, and Rectified Flows. By analyzing these paradigms within a single framework, we show how discriminative, multimodal, and generative learning are converging around shared representational principles, argue for a unified representation-learning perspective on modern computer vision, and outline open challenges and future directions toward general-purpose visual intelligence, multimodal foundation models, and trustworthy deployment.

**Keywords:** Computer Vision, Vision Transformers, Foundation Models, Self-Supervised Learning, Multimodal Learning, Diffusion Models, Rectified Flow.

## 1 Introduction

The development of visual representations that generalize across tasks, domains, and modalities has become a central objective of contemporary computer vision. This objective has driven rapid progress along both discriminative and generative directions, resulting in advances ranging from convolutional and transformer-based architectures to multimodal foundation models and diffusion-based generative systems. Although these developments are often presented within separate research communities, they

---

\*Corresponding author: abdelmalik.moujahid@unir.net

collectively reflect the broader evolution of computer vision toward increasingly versatile representation-learning frameworks. Understanding how these paradigms relate to one another has therefore become an important challenge for organizing the field and identifying its future directions.

35 Convolutional neural networks established the foundations of modern visual recognition through local connectivity, weight sharing, and hierarchical feature extraction. Over time, CNNs evolved from early architectures to influential models such as ResNet [1], DenseNet [2], and ConvNeXt [3], demonstrating that convolution remains highly effective when combined with improved optimization, larger receptive fields, and modern training strategies [4, 5]. Despite their success, convolutional architectures  
40 rely primarily on localized processing, making the modeling of long-range relationships comparatively difficult.

Vision Transformers addressed this limitation by leveraging self-attention mechanisms to model global interactions directly between image regions. Since their introduction by Dosovitskiy et al. [8], Transformers have achieved state-of-the-art performance across a wide range of computer vision tasks,  
45 including image classification, object detection, segmentation, and multimodal learning [9, 10]. At the same time, their substantial data and computational requirements have motivated extensive research into more efficient Transformer variants and hybrid CNN–Transformer architectures that integrate local convolutional inductive biases with global contextual reasoning [6, 20].

A further development has been the emergence of self-supervised and foundation-model paradigms.  
50 Methods such as DINO, DINOv2, and DINOv3 demonstrate that large-scale pretraining can produce highly transferable visual representations without task-specific supervision [56, 58, 59]. Similarly, vision-language models such as CLIP, ImageBind, and LLaVA extend visual learning beyond images by establishing shared representation spaces across multiple modalities and enabling increasingly sophisticated forms of multimodal reasoning and interaction [61, 62, 63].

55 In parallel, generative modeling has become a central component of modern computer vision. Diffusion models have established a powerful probabilistic framework for image synthesis, restoration, editing, generation, and conditional content creation [13, 16, 17]. More recent developments, including Diffusion Transformers and Rectified Flow models, further blur the distinction between discriminative and generative learning by combining large-scale representation learning with high-fidelity generation  
60 [25, 69].

Taken together, these developments reflect a broader transition from task-specific architectures toward increasingly general representation-learning systems. CNNs, Vision Transformers, foundation models, multimodal architectures, and generative frameworks should no longer be viewed as isolated paradigms. Instead, they form an evolving ecosystem of closely related approaches that share common  
65 mathematical principles while addressing complementary aspects of visual perception, reasoning, and generation.

Despite this convergence, the literature remains fragmented. While existing surveys provide comprehensive treatments of individual architectures, learning paradigms, or application domains, a unified account of how convolutional networks, Vision Transformers, self-supervised foundation models,  
70 multimodal systems, and generative frameworks relate to one another within a common mathematical framework remains largely missing.

As summarized in Table 1, prior surveys typically focus on specific model families, architectural innovations, or application areas, and therefore provide only limited discussion of the broader connections among representation learning, self-attention, self-supervision, multimodal alignment, and generative  
75 modeling. Consequently, many of the shared principles underlying modern computer vision remain obscured by architecture-specific or task-specific perspectives, despite the increasing integration of these paradigms in contemporary vision systems.

This survey addresses that gap by providing a unified mathematical perspective on modern computer vision. Rather than considering individual architectures in isolation, we establish a common formal  
80 framework that connects convolutional processing, self-attention mechanisms, self-supervised represen-

tation learning, multimodal alignment, and generative modeling. Within this framework, we review the evolution of CNNs, Vision Transformers, advanced Transformer architectures, hybrid CNN–Transformer models, multimodal foundation models, diffusion models, Diffusion Transformers, and Rectified Flow frameworks, highlighting their shared principles, complementary strengths, and emerging points of convergence.

The contributions of this survey are fourfold:

1. **Architectural Evolution:** We review the progression of modern computer vision architectures, from CNNs and Vision Transformers to advanced Transformer variants and hybrid CNN–Transformer models, highlighting the convergence of local and global representation-learning mechanisms.
2. **Foundation and Multimodal Models:** We analyze recent developments in self-supervised and multimodal foundation models, including DINO, CLIP, ImageBind, and LLaVA, and discuss their role in the emergence of increasingly general-purpose visual intelligence systems.
3. **Generative Frameworks and Future Directions:** We review diffusion models, Diffusion Transformers, and Rectified Flow methods, and discuss open challenges involving efficiency, interpretability, multimodal reasoning, and trustworthy deployment.
4. **Unified Formal Perspective:** Building upon these developments, we synthesize convolutional networks, Vision Transformers, self-supervised learning, multimodal foundation models, and generative architectures within a common representation-learning framework, highlighting the shared principles that underlie modern computer vision.

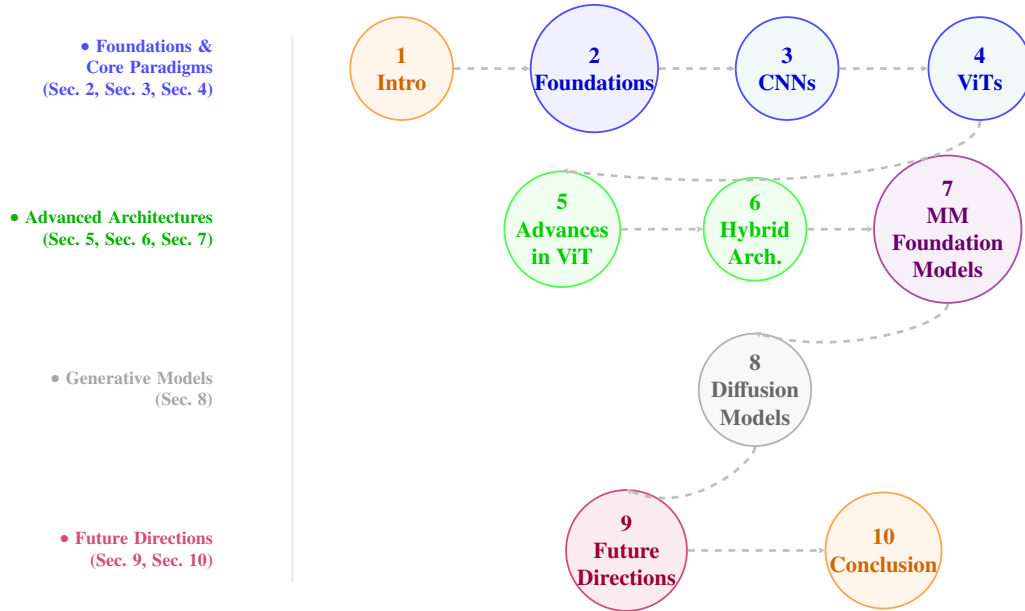


Figure 1: Organization of the survey. The mathematical foundations introduced in Section 2 provide a common framework for describing modern computer vision architectures. The discussion progresses from convolutional and attention-based models to multimodal foundation models, generative architectures, and future directions toward general-purpose visual intelligence.

**Organization of the Survey.** The remainder of this survey follows the progression illustrated in Figure 1. Section 2 establishes the mathematical foundations that provide a common framework for modern

computer vision architectures. Sections 3 and 4 review convolutional neural networks and Vision Transformers, respectively, while Sections 5 and 6 examine advanced Transformer architectures and hybrid CNN–Transformer models. Section 7 discusses multimodal foundation models, including CLIP, Image-Bind, and LLaVA. Section 8 reviews diffusion-based generative frameworks, Diffusion Transformers, and Rectified Flow methods. Finally, Section 9 discusses future directions and open challenges toward general-purpose visual intelligence, and Section 10 concludes the survey.

Table 1: Representative surveys on CNNs, Vision Transformers (ViTs), hybrid architectures, and diffusion models. Existing surveys typically focus on individual model families or application domains. In contrast, this survey provides a unified mathematical perspective that connects convolutional, attention-based, multimodal, foundation-model, and generative paradigms within a common representation-learning framework.

| Year             | Ref. | Focus                       | Main Scope                                                                                                                                                                                                                                     | Unified Mathematical Perspective | Foundation / Multimodal Models |
|------------------|------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|--------------------------------|
| 2025             | [6]  | Hybrid CNN–Transformers     | Taxonomy, comparative analysis, applications, and challenges of hybrid CNN–ViT architectures for vision tasks.                                                                                                                                 | Limited                          | No                             |
| 2025             | [20] | Vision Transformers         | Survey of Transformer architectures, design innovations, applications, and emerging research directions in computer vision.                                                                                                                    | Limited                          | Limited                        |
| 2025             | [30] | ViTs and Hybrid ViTs        | Review of ViT and hybrid CNN–Transformer models for medical image segmentation.                                                                                                                                                                | Limited                          | No                             |
| 2024             | [4]  | CNNs                        | Review of CNN fundamentals, architectures, applications, limitations, and future directions in computer vision.                                                                                                                                | No                               | No                             |
| 2024             | [5]  | CNNs                        | Comprehensive survey of convolution variants, CNN architectures, applications, challenges, and emerging trends.                                                                                                                                | No                               | No                             |
| 2024             | [17] | Diffusion Models            | Survey of diffusion-model theory, efficient sampling, likelihood estimation, and broad multidisciplinary applications.                                                                                                                         | Limited                          | No                             |
| 2023             | [16] | Diffusion Models            | Review of diffusion frameworks, vision applications, relationships to other generative models, and future directions.                                                                                                                          | Limited                          | No                             |
| 2023             | [21] | Medical Diffusion Models    | Comprehensive survey of diffusion-based methods and applications in medical imaging.                                                                                                                                                           | No                               | No                             |
| 2022             | [10] | Vision Transformers         | Comprehensive review of Transformer models for recognition, multimodal learning, video understanding, and generative vision tasks.                                                                                                             | Limited                          | Limited                        |
| 2021             | [9]  | Vision Transformers         | Taxonomy and analysis of Transformer architectures across major computer vision tasks.                                                                                                                                                         | No                               | No                             |
| <b>This Work</b> | –    | Unified Mathematical Survey | Unified mathematical treatment of CNNs, Vision Transformers, hybrid architectures, multimodal foundation models (e.g., CLIP, ImageBind, LLaVA), and diffusion-based generative frameworks within a common representation-learning perspective. | <b>Yes</b>                       | <b>Yes</b>                     |

## 1.1 Research Trends and Core Challenges

The preceding discussion highlights three interrelated research trends that currently shape modern computer vision. The first concerns the evolution of visual representation learning, characterized by a transition from convolution-dominated architectures toward Transformer-based and hybrid CNN–Transformer

systems [8, 3, 11, 12]. The second concerns the emergence of foundation-model paradigms, including self-supervised and multimodal learning approaches, which aim to learn transferable representations across tasks and modalities [56, 61, 62, 63]. The third concerns the rapid development of diffusion-based generative frameworks, which have become the dominant paradigm for image synthesis, restoration, and conditional generation [13, 14, 18, 16].

Together, these trends have significantly expanded the capabilities of vision systems, enabling advances in recognition, reasoning, multimodal interaction, and content generation. At the same time, they have introduced new challenges related to data efficiency, computational scalability, architectural complexity, interpretability, and deployment cost. Understanding these trade-offs is essential for explaining both ongoing architectural innovation and the growing tendency toward integrated vision systems.

**Table 2:** Representative works illustrating core challenges related to data efficiency, computational cost, and generative modeling limitations across CNNs, Vision Transformers, and diffusion-based architectures.

| Year | Ref. | Title                                                                             | Main focus                                                                                                   | Key observations                                                                                                                                       |
|------|------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2022 | [3]  | <i>ConvNeXt: A ConvNet for the 2020s</i>                                          | Revisits and modernizes CNN design to match Transformer performance.                                         | Shows CNNs can outperform ViTs under similar compute budgets, raising questions about true efficiency–accuracy advantages.                             |
| 2022 | [18] | <i>High-Resolution Image Synthesis with Latent Diffusion Models (LDM)</i>         | Introduces latent diffusion operating in compressed latent spaces.                                           | Reduces computation but still requires multi-step denoising, limiting real-time deployment.                                                            |
| 2021 | [22] | <i>Training Data-Efficient Image Transformers (DeiT)</i>                          | Efficient ViT training on mid-scale datasets through distillation tokens and strong data augmentation.       | Improves data efficiency but does not fully solve ViTs’ data hunger or heavy hyperparameter sensitivity.                                               |
| 2021 | [23] | <i>Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows</i>    | Introduces hierarchical shifted-window attention to control complexity while modeling local–global patterns. | Reduces quadratic attention cost but increases architectural complexity and introduces locality/globality trade-offs.                                  |
| 2020 | [8]  | <i>An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale</i> | Introduces Vision Transformers (ViT) as pure transformer architectures applied to image patches.             | Shows that ViTs require very large-scale pretraining data due to weak inductive bias, highlighting high sample complexity and limited data efficiency. |
| 2020 | [13] | <i>Denoising Diffusion Probabilistic Models (DDPM)</i>                            | Establishes denoising diffusion as a powerful generative framework.                                          | Requires hundreds–thousands of sampling steps, making inference extremely slow and computationally expensive.                                          |
| 2020 | [14] | <i>Denoising Diffusion Implicit Models (DDIM)</i>                                 | Proposes deterministic non-Markovian sampling compatible with DDPM training.                                 | Accelerates sampling but still much slower than GANs or autoregressive models; iterative nature remains a bottleneck.                                  |

Table 2 summarizes representative studies that highlight the principal trade-offs associated with contemporary vision architectures. Vision Transformers provide powerful mechanisms for modeling long-range dependencies and global contextual interactions, but often require large-scale training data and considerable computational resources because of relatively weak inductive biases and the quadratic complexity of self-attention [8, 9]. Although approaches such as DeiT [22] and Swin Transformer [23] have substantially improved data efficiency and scalability, these issues remain active areas of research.

Conversely, CNN-based architectures benefit from strong locality priors, efficient optimization, and favorable scaling characteristics. Recent studies such as ConvNeXt [3] demonstrate that modernized convolutional architectures remain highly competitive with Transformer-based approaches across a range of visual recognition tasks. Nevertheless, convolutional models generally possess a more limited capacity for modeling long-range dependencies and global contextual relationships compared with self-attention-based architectures [8, 9, 4].

Diffusion models introduce a different set of challenges associated with generative learning. While DDPMs [13] established a highly effective probabilistic framework for image synthesis, subsequent methods including DDIMs [14] and Latent Diffusion Models [18] have focused on reducing sampling costs and improving scalability. Despite these advances, the iterative denoising process remains computationally demanding and continues to limit efficient large-scale and real-time deployment [16, 17].

Taken together, these observations suggest that no single paradigm simultaneously optimizes representation quality, data efficiency, computational scalability, and generative capability. Consequently, recent research has increasingly emphasized the integration of complementary architectural strengths rather than the development of entirely independent model families.

This trend is particularly evident in the growing interaction between representation-learning architectures and generative frameworks. Contemporary diffusion models increasingly incorporate Transformer-based components and attention mechanisms to improve scalability, conditioning, and global consistency [24, 25, 26]. Likewise, hybrid CNN–Transformer architectures are increasingly employed within diffusion-based systems for image restoration, segmentation, and medical imaging applications [27, 28, 29]. These developments reflect a broader movement toward integrated vision systems that jointly exploit local feature extraction, global contextual reasoning, and generative modeling.

Despite this growing integration, the literature remains largely fragmented. Table 1 summarizes representative surveys covering CNNs, Vision Transformers, hybrid CNN–Transformer architectures, diffusion models, foundation models, and multimodal vision systems. Although these studies provide comprehensive analyses of individual paradigms, comparatively less attention has been devoted to their interactions and shared representational foundations. In particular, few surveys examine how hybrid CNN–Transformer backbones, self-supervised foundation models, multimodal architectures, and diffusion-based generative frameworks collectively contribute to the emergence of unified vision systems.

These developments motivate the central premise of this survey: although modern computer vision is often organized into separate research areas, many of its most influential advances can be understood through a common representation-learning perspective. Identifying the shared principles that connect architectural design, self-supervised learning, multimodal alignment, and generative modeling is therefore an important step toward a more unified understanding of contemporary vision systems.

## 2 Foundations of Modern Vision Models

The architectures and learning paradigms reviewed in this survey vary considerably in their computational mechanisms, training objectives, and application domains. Nevertheless, meaningful comparison requires a common mathematical language capable of describing their core components, learning processes, and representational structures in a consistent manner.

This section introduces the notation and formal framework used throughout the survey. Rather than focusing on architecture-specific details, we establish a generic formulation that is sufficiently broad to encompass discriminative, multimodal, and generative vision models. The resulting framework provides a basis for analyzing how different architectural choices give rise to distinct inductive biases, optimization behaviors, and representational properties while sharing common underlying learning principles.

### 2.1 Formal Framework

To establish a common notation throughout this survey, let  $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C}$  denote the input image space, where  $H$ ,  $W$ , and  $C$  represent the image height, width, and number of channels, respectively. The output space  $\mathcal{Y}$  depends on the task under consideration. For example,  $\mathcal{Y} = \{1, \dots, K\}$  for image classification,  $\mathcal{Y} \subseteq \{0, 1\}^{H \times W \times K}$  for semantic segmentation, and  $\mathcal{Y} \subseteq \mathbb{R}^{H \times W \times C}$  for image generation.

A vision model is represented as a parametric mapping

$$f_{\theta} : \mathcal{X} \rightarrow \mathcal{Y}, \quad (1)$$

where  $\theta \in \Theta \subseteq \mathbb{R}^P$  denotes the set of learnable parameters.

Table 3: Main notation used throughout the survey.

| Notation                                                                                                 | Description                                                                                                            | Section    |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|------------|
| <i>Unified Representation Learning Framework</i>                                                         |                                                                                                                        |            |
| $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}, f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ | Input sample, target output, and parametric vision model with trainable parameters $\theta$ .                          | 2.1        |
| $\mathcal{L}(\theta), \mathcal{R}(\theta), \lambda$                                                      | Task-dependent loss, regularization functional, and regularization strength.                                           | 2.1        |
| $H, W, C, d$                                                                                             | Image height, width, channels, and latent embedding dimension.                                                         | 2.1, 4.1   |
| <i>Convolutional Neural Networks</i>                                                                     |                                                                                                                        |            |
| $\mathbf{h}^{(\ell)}, \phi^{(\ell)}, \rho$                                                               | Feature map, convolutional block, and task-dependent readout operator.                                                 | 3.1        |
| $\Phi, \tilde{\mathbf{W}}^{(\ell)}, \mathbf{b}^{(\ell)}$                                                 | Patch extraction operator, reshaped convolution kernels, and bias parameters.                                          | 3.1        |
| $k_\ell, p_\ell, s_\ell, T_s$                                                                            | Kernel size, padding, stride, and spatial translation operator.                                                        | 3.1, 3.2   |
| $\tilde{\mathbf{W}}_{\text{pw}}, \tilde{\mathbf{W}}_{\text{dw}}$                                         | Pointwise ( $1 \times 1$ ) and depthwise convolution operators.                                                        | 4          |
| <i>Vision Transformers</i>                                                                               |                                                                                                                        |            |
| $P, N, \mathbf{x}_n, \mathbf{E}, \mathbf{p}_n$                                                           | Patch size, number of patches, flattened patch, patch embedding matrix, and positional encoding.                       | 4.1        |
| $\mathbf{Z}, \mathbf{Z}', \mathbf{Z}''$                                                                  | Token matrices before and after Transformer processing stages.                                                         | 4.1, 4.3   |
| $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V, d_k$                                                    | Query, key, value projections and per-head attention dimension.                                                        | 4.2        |
| $\mathbf{A}_h, \text{MHSA}(\cdot), \text{FFN}(\cdot), \text{LN}(\cdot)$                                  | Attention matrix, multi-head self-attention, feed-forward network, and layer normalization.                            | 4.2, 4.3   |
| $r_{nm}, r_\Delta, \Delta$                                                                               | Relative positional bias, displacement-dependent positional bias, and relative spatial displacement.                   | 5.1        |
| $\mathbf{z}_{\text{distill}}, \mathbf{y}_{\text{teacher}}$                                               | Distillation token and teacher prediction in DeiT.                                                                     | 5.4        |
| $\mathbf{G}, \mathbf{B}$                                                                                 | Global-context tokens and relative positional bias matrix in GC ViT.                                                   | 5.5        |
| $\mathbf{X}_{\text{sp}}$                                                                                 | Spatial feature representation obtained by reshaping tokens into a two-dimensional structure.                          | 5.6        |
| <i>Hybrid and Integrated Architectures</i>                                                               |                                                                                                                        |            |
| $\mathcal{G}, g_i^{(\ell)}, K^{(\ell)}$                                                                  | Set of local/global operators, $i$ -th operator within a hybrid block, and number of operators composing the block.    | 6.1        |
| $\text{Block}_M(\cdot), \text{Grid}_G(\cdot)$                                                            | Partition operators used in MaxViT, where $M$ denotes the local window size and $G$ the sparse-grid size.              | 6.3        |
| $\text{MHSA}_{\text{block}}, \text{MHSA}_{\text{grid}}$                                                  | Block-attention and grid-attention mechanisms in MaxViT.                                                               | 6.3        |
| $\tilde{\mathbf{h}}^{(\ell)}, \hat{\mathbf{h}}^{(\ell)}$                                                 | Intermediate feature representations after MBConv and Block Attention in MaxViT.                                       | 6.3        |
| $\Phi_{\text{embed}}$                                                                                    | Convolutional token-embedding operator in CvT.                                                                         | 6.4        |
| <i>Foundation and Multimodal Models</i>                                                                  |                                                                                                                        |            |
| $g_\psi, \mathbf{y}_s, \mathbf{y}_t, \mathbf{p}_s, \mathbf{p}_t, \tau_s, \tau_t$                         | Projection head, student/teacher outputs, probability distributions, and temperatures in DINO-style self-distillation. | 7.2        |
| $\mathbf{z}_I, \mathbf{z}_T, \mathbf{S}, p_{i \rightarrow j}$                                            | Image/text embeddings, similarity matrix, and contrastive matching probability in CLIP-style learning.                 | 7.3        |
| $I, M, \mathbf{q}, \mathbf{k}, \mathbf{Q}, \mathbf{K}$                                                   | Image and auxiliary modality, normalized embeddings, and embedding matrices used in ImageBind.                         | 7.4        |
| $\mathbf{X}_v, \mathbf{F}_v, \mathbf{W}, \mathbf{H}_v$                                                   | Input image, visual feature tokens, projection matrix, and projected visual tokens in LLaVA.                           | 7.5        |
| $\mathbf{E}_{\text{instr}}, \mathbf{C}, f_\phi$                                                          | Instruction embeddings, multimodal context, and autoregressive language model in LLaVA.                                | 7.5        |
| <i>Diffusion and Generative Models</i>                                                                   |                                                                                                                        |            |
| $\{\mathbf{x}_t\}_{t=0}^T, \beta_t, \tilde{\alpha}_t$                                                    | Diffusion trajectory, variance schedule, and cumulative noise retention factor.                                        | 8.1        |
| $q(\mathbf{x}_t \mathbf{x}_{t-1}), p_\theta(\mathbf{x}_{t-1} \mathbf{x}_t)$                              | Forward and reverse diffusion processes.                                                                               | 8.1, 8.2   |
| $\epsilon_\theta(\mathbf{x}_t, t), \mathbf{c} \in \mathcal{C}$                                           | Noise predictor and conditioning variable (class label, text, mask, etc.).                                             | 8.3, 8.4.3 |
| $\mathbf{z} = \mathcal{E}(\mathbf{x}), \hat{\mathbf{x}} = \mathcal{D}(\mathbf{z})$                       | Latent encoding and decoding operators in latent diffusion models.                                                     | 8.4.2      |

Given a dataset  $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^n$ , where  $\mathbf{x}^{(i)} \in \mathcal{X}$  is the  $i$ -th input sample and  $\mathbf{y}^{(i)} \in \mathcal{Y}$  its corre-

sponding target, training consists of minimizing the empirical risk

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f_{\theta}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}) + \lambda \mathcal{R}(\theta), \quad (2)$$

where  $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$  is a task-dependent loss function,  $\mathcal{R}(\theta)$  is a regularization term, and  $\lambda \geq 0$  controls the strength of regularization.

Equations (1) and (2) provide a generic learning framework encompassing the diverse architectures and training paradigms considered throughout this survey. Although convolutional neural networks, Vision Transformers, foundation models, multimodal systems, and generative frameworks differ substantially in their architectural design and training objectives, they can all be interpreted as alternative parameterizations of the mapping  $f_{\theta}$ , distinguished by their inductive biases, optimization strategies, and representational capacities.

Table 3 summarizes the principal mathematical notation used throughout the survey. The notation has been chosen to emphasize commonalities across paradigms while preserving the conventions adopted within each research community. It includes the core operators of convolutional networks, such as the patch extraction operator  $\Phi$  and the convolutional kernel matrix  $\widetilde{\mathbf{W}}^{(\ell)}$ ; key Transformer quantities, including the token representation matrix  $\mathbf{Z}$  and attention matrix  $\mathbf{A}_h$ ; as well as the probabilistic operators used in diffusion models, such as the reverse process  $p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)$  and the denoising network  $\epsilon_{\theta}$ . The table also indicates the corresponding sections in which these quantities are formally defined and discussed.

The notation introduced in this section serves as a common language for describing the models reviewed in subsequent sections. As discussed in Section 9.1, these seemingly distinct paradigms can ultimately be viewed from a unified representation-learning perspective, revealing common principles that underlie modern computer vision systems.

### 3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) established the foundations of modern deep learning for computer vision and remained the dominant architectural paradigm for visual recognition for more than a decade. Their success stems from the incorporation of domain-specific inductive biases, including local connectivity, weight sharing, and translation equivariance, which enable efficient learning from high-dimensional visual data [7, 31]. These properties allow CNNs to construct hierarchical feature representations that progressively transform low-level structures into increasingly abstract semantic concepts.

Over time, CNNs evolved from early convolutional architectures to deeper and more expressive models such as ResNet, DenseNet, and ConvNeXt, which improved optimization, representation capacity, computational efficiency, and scalability [1, 2, 3]. These developments established CNNs as the dominant framework for visual representation learning and inspired many of the architectural principles that continue to influence contemporary computer vision systems [4, 5].

Although Vision Transformers have recently emerged as a powerful alternative for modeling long-range dependencies and global contextual relationships, CNNs remain highly competitive across numerous vision tasks [3, 4]. Moreover, many modern architectures, including hybrid CNN–Transformer models, continue to incorporate convolutional operations because of their favorable inductive biases, computational efficiency, and data efficiency [6, 11, 12]. Consequently, understanding CNNs remains essential for understanding both the historical evolution of computer vision and the architectural developments that followed.

This section introduces a mathematical formulation of convolutional operations and their associated inductive biases. Building on this foundation, we then examine the principal families of CNN architectures, analyze their strengths and limitations, and discuss their role within the broader evolution of modern representation-learning systems.

### 3.1 Layer-wise Matrix Formulation of Convolution operations

A CNN parameterizes  $f_\theta$  as a composition of  $L$  convolutional layers interleaved with nonlinear activations. Let  $\mathbf{h}^{(0)} = \mathbf{x} \in \mathbb{R}^{H \times W \times C}$  denote the input feature map. At each layer  $\ell \in \{1, \dots, L\}$ , the feature map  $\mathbf{h}^{(\ell-1)} \in \mathbb{R}^{H_{\ell-1} \times W_{\ell-1} \times C_{\ell-1}}$  is transformed into  $\mathbf{h}^{(\ell)} \in \mathbb{R}^{H_\ell \times W_\ell \times C_\ell}$  via a *patch extraction operator*  $\Phi$ , which constructs a matrix whose row indexed by spatial location  $(i, j)$  contains the vectorized receptive field patch of  $\mathbf{h}^{(\ell-1)}$  centered at  $(i, j)$ , with zero-padding applied outside the valid domain. Defining the *reshaped filter matrix*  $\widetilde{\mathbf{W}}^{(\ell)} \in \mathbb{R}^{(k_\ell^2 C_{\ell-1}) \times C_\ell}$ , whose columns stack the  $C_\ell$  convolutional filters, and viewing  $\mathbf{h}^{(\ell-1)}$  as a matrix of spatial rows, the layer admits the compact form:

$$\mathbf{h}^{(\ell)} = \sigma\left(\Phi(\mathbf{h}^{(\ell-1)}) \widetilde{\mathbf{W}}^{(\ell)} + \mathbf{1}\mathbf{b}^{(\ell)\top}\right) \in \mathbb{R}^{(H_\ell W_\ell) \times C_\ell} \quad (3)$$

where  $\mathbf{b}^{(\ell)} \in \mathbb{R}^{C_\ell}$  is the bias vector,  $\mathbf{1} \in \mathbb{R}^{H_\ell W_\ell}$  is the all-ones vector, and  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  is a nonlinear activation function applied elementwise to the pre-activation matrix. Typical choices include the Rectified Linear Unit (ReLU), the Gaussian Error Linear Unit (GELU), and the sigmoid function.

This matrix representation recasts the convolution operation as a General Matrix Multiply (GEMM), enabling highly efficient execution on modern GPU architectures [31]. The output spatial dimensions follow:

$$H_\ell = \left\lfloor \frac{H_{\ell-1} + 2p_\ell - k_\ell}{s_\ell} \right\rfloor + 1, \quad W_\ell = \left\lfloor \frac{W_{\ell-1} + 2p_\ell - k_\ell}{s_\ell} \right\rfloor + 1,$$

where  $p_\ell \geq 0$  and  $s_\ell \geq 1$  are the padding and stride.

The total learnable parameters at layer  $\ell$  amount to  $|\theta^{(\ell)}| = C_\ell(k_\ell^2 C_{\ell-1} + 1)$ , independent of the spatial resolution  $H_{\ell-1} \times W_{\ell-1}$ , a direct consequence of *weight sharing*: the same  $\widetilde{\mathbf{W}}^{(\ell)}$  is applied at every spatial location  $(i, j)$  [31].

Denoting the  $\ell$ -th convolutional block as:

$$\phi^{(\ell)}(\mathbf{h}^{(\ell-1)}) = \sigma\left(\Phi(\mathbf{h}^{(\ell-1)}) \widetilde{\mathbf{W}}^{(\ell)} + \mathbf{1}\mathbf{b}^{(\ell)\top}\right), \quad (4)$$

The full network is expressed as the sequential composition of  $L$  blocks,

$$\mathbf{h}^{(\ell)} = \phi^{(\ell)}(\mathbf{h}^{(\ell-1)}), \quad \ell = 1, \dots, L, \quad (5)$$

followed by a task-specific readout  $\rho$  applied to the final feature map:

$$f_\theta(\mathbf{x}) = \rho(\mathbf{h}^{(L)}), \quad (6)$$

where  $\rho$  is typically global average pooling followed by a linear classifier for image classification, or a decoder network for dense prediction tasks.

### 3.2 Translation Equivariance

A central inductive bias of CNNs is *translation equivariance*, which formalizes the prior that the presence of a visual feature is independent of its absolute spatial location. To state this precisely, let  $T_{\mathbf{s}}$  denote the operator that shifts a feature map by  $\mathbf{s} = (s_1, s_2) \in \mathbb{Z}^2$  pixels, defined as:

$$(T_{\mathbf{s}} \mathbf{h})[i, j, c] = \mathbf{h}[i - s_1, j - s_2, c], \quad (7)$$

so that  $(T_{\mathbf{s}} \mathbf{h})[i, j, c]$  returns the value of  $\mathbf{h}$  at the location displaced by  $\mathbf{s}$  from  $(i, j)$ , with out-of-bounds entries set to zero.

This property arises directly from weight sharing: because  $\widetilde{\mathbf{W}}^{(\ell)}$  is applied at every spatial location  $(i, j)$  via  $\Phi$ , the network response to a local pattern depends only on the pattern itself and not on where it appears.

In matrix notation, and noting that the spatially uniform bias  $\mathbf{1b}^{(\ell)\top}$  is invariant under  $T_s$ :

$$\Phi(T_s \mathbf{h}^{(\ell-1)}) \widetilde{\mathbf{W}}^{(\ell)} = T_s \left( \Phi(\mathbf{h}^{(\ell-1)}) \widetilde{\mathbf{W}}^{(\ell)} \right), \quad \forall \mathbf{s} \in \mathbb{Z}^2, \quad (8)$$

which in terms of the output feature map reads:

$$\phi^{(\ell)}(T_s \mathbf{h}^{(\ell-1)}) = T_s \left( \phi^{(\ell)}(\mathbf{h}^{(\ell-1)}) \right), \quad \forall \mathbf{s} \in \mathbb{Z}^2, \quad (9)$$

confirming that translating the input by  $\mathbf{s}$  shifts the output feature map by the same amount. Equivariance is preserved under composition: since each block  $\phi^{(\ell)}$  satisfies  $\phi^{(\ell)} \circ T_s = T_s \circ \phi^{(\ell)}$ , it follows that:

$$\phi^{(L)} \circ \dots \circ \phi^{(1)} \circ T_s = T_s \circ \phi^{(L)} \circ \dots \circ \phi^{(1)}, \quad (10)$$

so equivariance propagates through the entire network depth. When followed by a global pooling operation  $\rho : \mathbb{R}^{(H_L W_L) \times C_L} \rightarrow \mathbb{R}^{C_L}$  that aggregates over all spatial locations, equivariance promotes *translation invariance* of the final representation:

$$\rho(\phi^{(L)} \circ \dots \circ \phi^{(1)}(T_s \mathbf{x})) = \rho(\phi^{(L)} \circ \dots \circ \phi^{(1)}(\mathbf{x})), \quad \forall \mathbf{s} \in \mathbb{Z}^2, \quad (11)$$

since permuting the spatial rows of the output does not alter a spatially aggregating pooling function. This distinction is important: equivariance retains the *where* information (the feature map shifts with the pattern), whereas invariance discards it (the global descriptor is unchanged), making the two properties suited to different tasks, dense prediction and classification, respectively.

More generally, CNNs can be understood within the framework of group-equivariant networks [32], where a symmetry group  $\mathcal{G}$  acts on both the input and output spaces and the network commutes with this action; translation equivariance is the special case  $\mathcal{G} = (\mathbb{Z}^2, +)$ . The receptive field of a neuron at layer  $\ell$  grows as  $\mathcal{O}(\ell \cdot k_\ell)$ , establishing that depth induces a hierarchical progression from local features (edges, textures) to globally meaningful representations, a limitation that motivates the architectures discussed in Section 4.

### 3.3 Taxonomy of Convolutional Architectures

Building on the layer-wise formulation of Section 3.1, contemporary CNN architectures can be organized into four families according to how they extend or augment the basic formulation. These families differ primarily in how they address the fundamental limitations of standard convolutional networks, including the restricted growth of the receptive field, the absence of explicit mechanisms for modeling long-range dependencies, and the deterministic and predominantly discriminative nature of the learned representations. The four families, summarized in Table 4, occupy different positions within the broader CNN design space and are described as follows:

1. **Classical hierarchical CNNs.** These models implement the standard convolutional formulation with progressive downsampling, expressing the full network as a composition of layer-wise transformations:

$$f_\theta(\mathbf{x}) = \phi^{(L)} \circ \phi^{(L-1)} \circ \dots \circ \phi^{(1)}(\mathbf{x}), \quad (12)$$

where each  $\phi^{(\ell)}$  implements the convolutional block defined in Section 3.1. In practice, each block is typically augmented with batch normalization [33] and a ReLU nonlinearity [34], instantiating the general  $\sigma$  of Section 3.1 as:

$$\phi^{(\ell)}(\mathbf{h}^{(\ell-1)}) = \text{ReLU} \left( \text{BN} \left( \Phi(\mathbf{h}^{(\ell-1)}) \widetilde{\mathbf{W}}^{(\ell)} + \mathbf{1b}^{(\ell)\top} \right) \right), \quad (13)$$

where BN normalizes each channel across the spatial and batch dimensions [33] and  $\text{ReLU}(z) = \max(0, z)$  [34] is applied elementwise. Optional pooling between blocks progressively reduces the

spatial resolution  $H_\ell \times W_\ell$ , building a hierarchy of representations at increasing semantic levels. Their strong inductive bias promotes data efficiency and training stability, but the  $O(\ell \cdot k_\ell)$  receptive field growth limits their ability to model global context without significant depth.

A key advancement is the introduction of residual connections [35]:

$$\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)} + \phi^{(\ell)}(\mathbf{h}^{(\ell-1)}), \quad (14)$$

which allow gradient propagation across many layers and enable training of very deep networks. The residual block that established this design is shown in Figure 2.

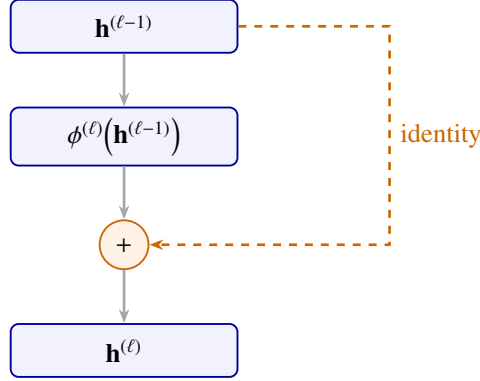


Figure 2: The identity shortcut adds the input  $\mathbf{h}^{(\ell-1)}$  directly to the output of the convolutional block  $\phi^{(\ell)}$ , so the block learns a residual correction rather than a full transformation. This facilitates gradient propagation across many layers and enables training of very deep networks.

2. **Modernized CNNs with Transformer-inspired design.** Architectures such as ConvNeXt [3] reinterpret the standard convolutional block  $\phi^{(\ell)}$  by incorporating three design principles from Vision Transformers: depthwise separable convolutions, layer normalization, and inverted bottleneck structures.

*Depthwise separation* factorizes the convolutional operator into a spatial aggregation operator  $\Phi_{\text{dw}}^{(\ell)}$  and channel-mixing projections. Instead of a single filter matrix  $\widetilde{\mathbf{W}}^{(\ell)}$ , the block applies a depthwise convolution independently to each channel and then mixes channels through pointwise linear maps. The parameter count is reduced from  $k_\ell^2 C_{\ell-1} C_\ell$  to approximately  $k_\ell^2 C_{\ell-1} + C_{\ell-1} C_\ell$  while preserving expressive power.

*Layer normalization* [36] replaces batch normalization by normalizing across the channel dimension, making the block behavior independent of batch size.

*Inverted bottleneck* structures expand the hidden channel dimension to  $r C_{\ell-1}$  before projecting back to  $C_\ell$ , increasing representation capacity while keeping spatial computation unchanged.

Together these modifications yield the ConvNeXt block:

$$\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)} + \mathcal{F}^{(\ell)}(\mathbf{h}^{(\ell-1)}), \quad (15)$$

with

$$\mathcal{F}^{(\ell)}(\mathbf{h}) = \widetilde{\mathbf{W}}_{\text{proj}}^{(\ell)} \sigma(\widetilde{\mathbf{W}}_{\text{exp}}^{(\ell)} \Phi_{\text{dw}}^{(\ell)}(\text{LN}(\mathbf{h}))), \quad (16)$$

where  $\widetilde{\mathbf{W}}_{\text{exp}}^{(\ell)} \in \mathbb{R}^{C_{\ell-1} \times r C_{\ell-1}}$  and  $\widetilde{\mathbf{W}}_{\text{proj}}^{(\ell)} \in \mathbb{R}^{r C_{\ell-1} \times C_\ell}$  denote the expansion and projection maps of the inverted bottleneck, respectively. The operator  $\Phi_{\text{dw}}^{(\ell)}$  preserves translation equivariance because the same depthwise filter is applied at every spatial location  $(i, j)$ . These choices closely mirror the Transformer FFN and normalization design while retaining the inductive biases of convolution [3].

3. **CNNs augmented with global context modules.** These designs supplement the local aggregation operator  $\Phi^{(\ell)}$  with non-local interactions that compute pairwise dependencies across all spatial positions. A non-local block [37] produces

$$\mathbf{h}_{(i,j),:}^{(\ell)} = \frac{1}{C^{(\ell)}} \sum_{(i',j')} \mathbf{A}_{(i,j),(i',j')}^{(\ell)} \mathbf{h}_{(i',j'),:}^{(\ell-1)} \mathbf{W}_g^{(\ell)}, \quad (17)$$

where  $\mathbf{A}_{(i,j),(i',j')}^{(\ell)}$  contains pairwise affinity scores analogous to the attention matrix  $\mathbf{A}_h$  of MHSA (Section 4.2),  $\mathbf{W}_g^{(\ell)}$  is a learnable channel projection, and  $C^{(\ell)}$  is a normalization factor. This raises the per-layer complexity from  $O(H_\ell W_\ell k_\ell^2 C_{\ell-1} C_\ell)$  to  $O((H_\ell W_\ell)^2 C_\ell)$ , matching the quadratic scaling  $O(N^2 d)$  of ViT self-attention (Section 4.2).

4. **Hybrid convolutional-attention architectures.** These pipelines decompose processing into convolutional early stages and attention-based late stages:

$$f_\theta(\mathbf{x}) = f_{\text{attn}}(\mathbf{E} \text{vec}(\mathbf{h}^{(L_{\text{cnn}})}) + \mathbf{P}), \quad (18)$$

where  $\mathbf{h}^{(L_{\text{cnn}})}$  is the output of the convolutional front-end,  $\mathbf{E}$  is a linear projection producing token embeddings analogous to the patch-embedding matrix of Section 4.1,  $\mathbf{P}$  encodes positional information, and  $f_{\text{attn}}$  denotes a Transformer operating on the resulting token sequence. This design combines the translation equivariance and data efficiency of CNNs with the global context modeling of ViTs, leveraging the complementary strengths of the two paradigms described in Section 4.

**Table 4:** Representative mathematical formulations and key practical characteristics of major CNN architectural families.

| Architecture                                  | Representative Equation                                                                                                                                                         | Practical Characteristic  |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Classical CNN<br>(LeNet, VGG, ResNet)         | $\mathbf{h}^{(\ell)} = \sigma(\Phi(\mathbf{h}^{(\ell-1)})\widetilde{\mathbf{W}}^{(\ell)})$                                                                                      | ✓ High data efficiency    |
| Modernized CNN<br>(ConvNeXt, RepLNet)         | $\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)} + \widetilde{\mathbf{W}}_{\text{pw}} \sigma(\Phi_{\text{dw}}(\text{LN}(\mathbf{h}^{(\ell-1)}))\widetilde{\mathbf{W}}_{\text{dw}})$ | ✓ Efficient scaling       |
| Global-Context CNN<br>(GCNet, Non-local Net)  | $\mathbf{h}_{(i,j)}^{(\ell)} = \Phi(\mathbf{h}^{(\ell-1)})\widetilde{\mathbf{W}}^{(\ell)} + \sum_{(i',j')} \mathbf{A}_{(i,j),(i',j')}^{(\ell)} \mathbf{h}_{(i',j')}^{(\ell-1)}$ | ✓ Long-range dependencies |
| Hybrid CNN–Transformer<br>(CvT, CoAtNet, CMT) | $f_\theta(\mathbf{x}) = f_{\text{attn}}(\mathbf{E} \text{vec}(\mathbf{h}^{(L_{\text{cnn}})}) + \mathbf{P})$                                                                     | ✓ Multimodal integration  |

The four architectural families outlined above illustrate several fundamental trade-offs that characterize the evolution of convolutional vision models.

First, they lie along a locality–globality spectrum. Classical and modernized CNNs rely primarily on the local aggregation operator  $\Phi$ , whose effective receptive field grows only as  $O(\ell \cdot k_\ell)$ . In contrast, non-local modules introduce explicit pairwise interactions through affinity matrices  $\mathbf{A}_{(i,j),(i',j')}^{(\ell)}$ . Hybrid CNN–Transformer architectures occupy an intermediate position by integrating convolutional operators  $\Phi$  and self-attention matrices  $\mathbf{A}_h$ . Rather than replacing local processing, these models seek to combine the translation equivariance, parameter efficiency, and strong inductive biases of convolution with the unrestricted receptive field and global dependency modeling capabilities of self-attention.

Second, the taxonomy reflects a continuum between strong inductive bias and representational flexibility. Classical CNNs exploit translation equivariance, weight sharing, and locality constraints to achieve high data efficiency and stable optimization. Modernized CNNs preserve these properties while adopting architectural elements inspired by Vision Transformers. Conversely, architectures incorporating

non-local operations or attention mechanisms gradually relax these constraints in exchange for improved capacity to capture long-range spatial relationships.

Finally, all four families remain fundamentally deterministic discriminative models. Given an input  $\mathbf{x}$ , the network implements a deterministic mapping  $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$  optimized through a supervised objective  $\mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y})$ . This property distinguishes them from the generative and stochastic architectures introduced later in this chapter. For example, diffusion models (Section 8) learn a probabilistic reverse process  $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ , while modern hybrid systems increasingly combine deterministic convolutional encoders with stochastic generative mechanisms.

These trade-offs provide a useful conceptual framework for understanding the subsequent evolution of vision architectures. Vision Transformers (Section 4) primarily address the locality limitations of convolution, diffusion models (Section 8) extend visual representation learning into the generative setting, and hybrid CNN–Transformer architectures (Section 6) seek to balance locality, globality, inductive bias, and representational capacity within a unified framework.

## 4 Vision Transformers

The introduction of Vision Transformers (ViTs) marked a major shift in the development of visual representation learning. Inspired by the Transformer architecture originally proposed for natural language processing [38], ViTs demonstrated that self-attention can serve as a viable alternative to convolution for modeling visual data [8]. Their success challenged the long-standing dominance of convolutional neural networks and initiated a new research direction centered on global context modeling, large-scale pretraining, and increasingly general-purpose visual representations [9, 10, 20].

From the perspective of the unified framework introduced in Section 2, Vision Transformers provide an alternative parameterization of the mapping  $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$  based on self-attention rather than convolution. Instead of processing an image as a spatial grid through localized operators, ViTs represent the input image  $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$  as a sequence of visual tokens and allow every token to interact directly with every other token through attention mechanisms. This design enables global dependency modeling from the earliest layers of the network and has become the foundation of many contemporary architectures, including self-supervised foundation models, multimodal vision-language systems, and diffusion-based Transformer backbones [56, 61, 24].

Compared with CNNs, Vision Transformers introduce a substantially different inductive bias. Convolutional networks exploit translation-equivariant local operators and construct increasingly global representations through hierarchical composition, with receptive fields that expand gradually across depth. In contrast, ViTs model pairwise interactions explicitly through self-attention, allowing global information exchange at every layer. This capability comes at the cost of increased computational complexity and weaker built-in spatial priors, motivating many of the architectural innovations discussed later in this section.

The standard Vision Transformer architecture can be decomposed into three fundamental components: (i) patch tokenization, which converts images into sequences of visual tokens; (ii) multi-head self-attention, which models pairwise interactions among tokens; and (iii) a position-wise feed-forward network that refines token representations within each Transformer block. Together, these components define the basic computational framework upon which most contemporary Vision Transformer architectures are built. The following subsections formalize each component and establish the notation used throughout the remainder of this section.

## 4.1 Patch Tokenization

The input image  $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$  is partitioned into  $N = HW/P^2$  non-overlapping patches of size  $P \times P$  pixels.  
 385 Each patch is flattened into a vector  $\mathbf{x}_n \in \mathbb{R}^{P^2 C}$  and linearly projected to a  $d$ -dimensional embedding:

$$\mathbf{z}_n = \mathbf{E} \mathbf{x}_n + \mathbf{p}_n \in \mathbb{R}^d, \quad (19)$$

where  $\mathbf{E} \in \mathbb{R}^{d \times (P^2 C)}$  is the shared patch embedding matrix and  $\mathbf{p}_n \in \mathbb{R}^d$  is a learnable positional encoding associated with patch  $n$ . The positional encoding is necessary because the subsequent attention mechanism is permutation-invariant with respect to the token sequence: without  $\mathbf{p}_n$ , the model has no means of distinguishing patch  $n$  at spatial location  $(i, j)$  from any other patch at a different location. A learnable  
 390 classification token  $\mathbf{z}_0 \in \mathbb{R}^d$  is prepended to aggregate global information during the forward pass [8], yielding the token matrix:

$$\mathbf{Z} = [\mathbf{z}_0^\top; \mathbf{z}_1^\top; \dots; \mathbf{z}_N^\top] \in \mathbb{R}^{(N+1) \times d}, \quad (20)$$

where  $n = 0$  denotes the classification token and  $n \geq 1$  the patch tokens. Note that whereas the CNN feature map  $\mathbf{h}^{(\ell)} \in \mathbb{R}^{H_\ell \times W_\ell \times C_\ell}$  preserves an explicit spatial grid structure through  $\Phi$ , the token matrix  $\mathbf{Z}$  treats the image as an unordered set of patches, with spatial relationships encoded solely through  $\mathbf{p}_n$ .

## 4.2 Multi-Head Self-Attention

Given the token matrix  $\mathbf{Z} \in \mathbb{R}^{(N+1) \times d}$ , each Transformer block applies multi-head self-attention (MHSA) to model pairwise interactions among all tokens. For each attention head  $h \in \{1, \dots, H_{\text{heads}}\}$ , the query, key, and value representations are obtained through learned linear projections  $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d \times d_k}$ , where  $d_k = d/H_{\text{heads}}$ :

$$\mathbf{Q}_h = \mathbf{Z} \mathbf{W}_h^Q, \quad \mathbf{K}_h = \mathbf{Z} \mathbf{W}_h^K, \quad \mathbf{V}_h = \mathbf{Z} \mathbf{W}_h^V \in \mathbb{R}^{(N+1) \times d_k}, \quad (21)$$

400 The attention weight matrix and head output are computed as:

$$\mathbf{A}_h = \text{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_k}}\right) \in \mathbb{R}^{(N+1) \times (N+1)}, \quad \text{Attn}_h(\mathbf{Z}) = \mathbf{A}_h \mathbf{V}_h \in \mathbb{R}^{(N+1) \times d_k}, \quad (22)$$

where softmax is applied row-wise and  $\sqrt{d_k}$  prevents excessively large inner products in high-dimensional feature spaces. The entry  $\mathbf{A}_h[n, m]$  quantifies the influence of token  $m$  on token  $n$ , for  $n, m \in \{0, \dots, N\}$ , so that unlike a convolutional feature  $\mathbf{h}_{(i,j),c'}^{(\ell)}$  whose receptive field is bounded by the kernel support of  $\Phi$ , each token representation aggregates information from all  $N + 1$  tokens globally. The outputs of all  
 405 heads are concatenated and projected via  $\mathbf{W}^O \in \mathbb{R}^{d \times d}$ :

$$\text{MHSA}(\mathbf{Z}) = [\text{Attn}_1(\mathbf{Z}) \parallel \dots \parallel \text{Attn}_{H_{\text{heads}}}(\mathbf{Z})] \mathbf{W}^O \in \mathbb{R}^{(N+1) \times d}. \quad (23)$$

## 4.3 Position-Wise Feed-Forward Network and Transformer Block

While self-attention models interactions between tokens through  $\mathbf{A}_h$ , the feed-forward network (FFN) applies a nonlinear transformation independently to each token. The mapping  $\text{FFN} : \mathbb{R}^{(N+1) \times d} \rightarrow \mathbb{R}^{(N+1) \times d}$  is defined as:

$$\text{FFN}(\mathbf{Z}) = \sigma(\mathbf{Z} \mathbf{W}_1^\top + \mathbf{1} \mathbf{b}_1^\top) \mathbf{W}_2^\top + \mathbf{1} \mathbf{b}_2^\top, \quad (24)$$

410 where  $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{ff}} \times d}$ ,  $\mathbf{W}_2 \in \mathbb{R}^{d \times d_{\text{ff}}}$ ,  $\mathbf{b}_1 \in \mathbb{R}^{d_{\text{ff}}}$ ,  $\mathbf{b}_2 \in \mathbb{R}^d$ ,  $\mathbf{1} \in \mathbb{R}^{N+1}$  is the all-ones vector producing broadcasted bias addition consistent with Section 3, and  $\sigma$  is a pointwise nonlinear activation, typically GELU [39].

The MHSA and FFN sublayers are combined within each Transformer block through residual connections and layer normalization. The original architecture of Vaswani et al. [38], applies layer normalization *after* the residual addition (Post-LN):

$$\mathbf{Z}' = \text{LN}(\mathbf{Z} + \text{MHSA}(\mathbf{Z})), \quad \mathbf{Z}'' = \text{LN}(\mathbf{Z}' + \text{FFN}(\mathbf{Z}')). \quad (25)$$

Subsequent work [40] demonstrated that applying normalization *before* the sublayer (Pre-LN) improves optimization stability, yielding the formulation adopted in most modern Vision Transformers:

$$\mathbf{Z}' = \mathbf{Z} + \text{MHSA}(\text{LN}(\mathbf{Z})), \quad \mathbf{Z}'' = \mathbf{Z}' + \text{FFN}(\text{LN}(\mathbf{Z}')), \quad (26)$$

where  $\mathbf{Z}''$  serves as the input to the next Transformer block. In both variants, the residual connections facilitate gradient propagation analogously to the skip connections of Section 3.3 [35].

## 5 Advances in Vision Transformer Architectures

The foundational ViT architecture introduced in Sections 4.1–4.3 demonstrated that global self-attention over a sequence of visual tokens can serve as an effective alternative to the convolutional operator  $\phi^{(\ell)}$  discussed in Section 3. Despite its conceptual simplicity and strong empirical performance, the original formulation exhibited several limitations. In particular, it lacked the strong spatial inductive biases naturally provided by convolution, incurred the quadratic computational cost associated with the attention matrix  $\mathbf{A}_h \in \mathbb{R}^{(N+1) \times (N+1)}$ , and required large-scale training datasets to compensate for the absence of convolutional priors. Subsequent developments in the Vision Transformer literature can largely be interpreted as attempts to address these challenges.

### 5.1 Positional Awareness

A fundamental challenge in Vision Transformers is encoding spatial relationships within an architecture whose attention mechanism is intrinsically permutation-invariant. The original ViT addressed this issue through learned absolute positional embeddings  $\mathbf{p}_n \in \mathbb{R}^d$  added to the input token sequence. However, absolute encodings are tied to a specific input resolution and do not explicitly model relative spatial relationships.

Conditional Positional Encodings (CPE) [41] generate positional information adaptively from local token neighborhoods, allowing the model to generalize more naturally across varying image resolutions. An alternative strategy introduces relative positional information directly into the attention computation through learnable biases:

$$\mathbf{A}_h[n, m] = \text{softmax}\left(\frac{\mathbf{Q}_h[n, :] \mathbf{K}_h[m, :]^\top + r_{nm}}{\sqrt{d_k}}\right), \quad (27)$$

where  $r_{nm}$  depends on the relative displacement between tokens. Unlike absolute embeddings, this formulation allows the model to reason about spatial relationships independently of absolute coordinates.

The importance of relative positional information extends beyond spatial awareness. Cordonnier et al. [42] demonstrated that multi-head self-attention can, in principle, reproduce the behavior of a convolutional layer when relative positional offsets are incorporated into the attention mechanism. Specifically, a MHSA layer with  $H_{\text{heads}} \geq k_\ell^2$  heads can emulate a convolution with kernel size  $k_\ell \times k_\ell$  by assigning individual heads to specific relative offsets  $\Delta = \mathbf{s}_n - \mathbf{s}_m \in \mathbb{Z}^2$ . Through suitable relative positional biases  $r_\Delta$ , the resulting attention distribution becomes concentrated on predefined spatial neighborhoods, reproducing the weighted aggregation performed by a convolutional kernel. This result establishes a theoretical connection between self-attention and convolution, suggesting that Transformer architectures can recover convolution-like inductive biases directly from data rather than imposing them explicitly.

Other approaches, such as deformable attention, further relax the rigid structure of global self-attention by allowing each query token to attend selectively to content-dependent spatial locations. In this manner, deformable attention introduces a compromise between the locality of convolutional receptive fields and the unrestricted receptive field of full self-attention.

## 5.2 Hierarchical Transformers

A second line of research seeks to recover the hierarchical processing structure that characterizes convolutional networks. In the original ViT, the number of tokens remains fixed throughout the network, so every layer operates on representations  $\mathbf{Z}^{(\ell)} \in \mathbb{R}^{(N+1) \times d}$  of identical spatial extent. This design contrasts with CNNs, where spatial resolution is progressively reduced while semantic abstraction increases.

The Swin Transformer [23] reintroduces hierarchical processing by partitioning the token sequence into non-overlapping windows and restricting self-attention to tokens within each window. To enable cross-window communication, consecutive layers alternate between standard window attention and shifted-window attention:

$$\mathbf{Z}^{(\ell+1)} = \mathbf{Z}^{(\ell)} + \text{MHSA}_w(\text{LN}(\mathbf{Z}^{(\ell)})), \quad \mathbf{Z}^{(\ell+2)} = \mathbf{Z}^{(\ell+1)} + \text{MHSA}_{sw}(\text{LN}(\mathbf{Z}^{(\ell+1)})), \quad (28)$$

where  $\text{MHSA}_w$  and  $\text{MHSA}_{sw}$  denote window and shifted-window attention, respectively. Combined with patch-merging operations, this strategy reduces the complexity from  $O(N^2d)$  to  $O(NM^2d)$ , which scales linearly with the number of tokens for a fixed window size  $M$ , while simultaneously recovering a multi-scale feature hierarchy analogous to that of CNNs.

## 5.3 Computational Efficiency

Although self-attention provides a global receptive field, the computation of  $\mathbf{A}_h$  requires pairwise interactions between all tokens, resulting in a computational complexity of  $O(N^2d)$ . This quadratic scaling rapidly becomes prohibitive for high-resolution inputs and has motivated numerous approximation strategies.

Linear attention methods approximate the softmax kernel through a feature mapping that converts the attention operation into a computation that scales linearly with the number of tokens. Other approaches exploit the two-dimensional structure of images through factorized attention, which decomposes dense token interactions into separate row-wise and column-wise operations. Dynamic token sparsification further reduces computational requirements by retaining only the most informative subset of tokens at each layer, reducing the effective sequence length processed by self-attention.

Collectively, these approaches seek to preserve the expressive power of global attention while improving scalability to larger image resolutions and longer token sequences.

## 5.4 Data-Efficient Training

Although architectural modifications can improve spatial modeling and computational efficiency, Vision Transformers also exhibit substantial data requirements due to the absence of the strong inductive biases naturally provided by convolution. DeiT [22] addresses this limitation through a knowledge-distillation framework in which a convolutional teacher network guides the training of the Transformer.

Specifically, DeiT introduces a distillation token  $\mathbf{z}_{\text{distill}} \in \mathbb{R}^d$ , which is processed jointly with the visual tokens throughout the Transformer encoder. Let  $\mathbf{z}_{\text{distill}}''$  denote its final-layer representation. During training,  $\mathbf{z}_{\text{distill}}''$  receives supervisory information from a convolutional teacher model, leading to the objective

$$\mathcal{L} = \frac{1}{2} \mathcal{L}_{\text{CE}}(\psi(\mathbf{z}_0''), \mathbf{y}) + \frac{1}{2} \mathcal{L}_{\text{CE}}(\psi(\mathbf{z}_{\text{distill}}''), \mathbf{y}_{\text{teacher}}), \quad (29)$$

where  $\psi$  denotes a linear classification head and  $\mathbf{y}_{\text{teacher}}$  corresponds to the prediction produced by the teacher network. This strategy effectively transfers convolutional inductive biases into the Transformer during training and substantially improves sample efficiency without modifying the architecture itself.

## 5.5 Locality-Aware Attention Mechanisms

Although self-attention provides global interactions from the earliest layers, it is not specifically optimized for extracting fine-grained local visual structures. Several architectures therefore seek to augment Transformer representations with locality-aware mechanisms that preserve short-range spatial dependencies while retaining the flexibility of attention-based processing.

**Vision Outlooker (VOLO).** VOLO (Vision Outlooker) [43] introduces Outlook Attention, a locality-aware attention mechanism designed to enrich token representations before global reasoning is performed. Instead of computing pairwise interactions across all tokens, Outlook Attention aggregates information within a local neighborhood and dynamically adapts the aggregation weights according to the content of the center token.

Given a token  $\mathbf{Z}_{(i,j),:} \in \mathbb{R}^d$  and a local neighborhood  $\mathcal{N}(i, j)$ , the neighboring values are first projected as

$$\mathbf{V}_{(i,j)} = \{\mathbf{Z}_{(i',j'),:} \mathbf{W}_V : (i', j') \in \mathcal{N}(i, j)\} \in \mathbb{R}^{k^2 \times d}, \quad (30)$$

while the center token generates a position-dependent attention map

$$\widehat{\mathbf{A}}_{(i,j)} = \text{softmax}(\text{reshape}(\mathbf{Z}_{(i,j),:} \mathbf{W}_A)) \in \mathbb{R}^{k^2 \times k^2}, \quad (31)$$

which is used to aggregate neighborhood information:

$$\mathbf{Y}_{(i,j)} = \widehat{\mathbf{A}}_{(i,j)} \mathbf{V}_{(i,j)}. \quad (32)$$

Unlike standard self-attention, Outlook Attention performs adaptive local aggregation, occupying an intermediate position between convolution and global self-attention by combining locality-aware processing with content-dependent weighting.

**Global Context Vision Transformer (GC ViT).** A complementary strategy is adopted by the Global Context Vision Transformer (GC ViT) [44], which explicitly combines local and global context modeling within a hierarchical Transformer architecture. Rather than relying on shifted windows to propagate information across regions, GC ViT augments local attention with globally informed query tokens extracted from the entire feature hierarchy.

Let  $\mathbf{Z} \in \mathbb{R}^{N \times d}$  denote the token representation within a local window. Local self-attention captures short-range spatial interactions, while a global query generator produces a compact set of context tokens  $\mathbf{G} \in \mathbb{R}^{M \times d}$ ,  $M \ll N$ , that encode information aggregated from multiple image regions. These global tokens are subsequently used as queries while local tokens provide the corresponding keys and values. The resulting attention operation is

$$\text{Attention}(\mathbf{G}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{G}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{B}\right) \mathbf{V}, \quad (33)$$

where  $\mathbf{B}$  denotes a learnable relative positional bias term [44]. By allowing globally extracted queries to interact with local key-value representations, GC ViT enlarges the effective receptive field while preserving the computational efficiency of window-based attention.

Furthermore, GC ViT incorporates modified Fused-MBCConv blocks within its hierarchical backbone, introducing convolutional inductive biases and improving inter-channel feature modeling. Consequently,

the architecture combines local feature extraction, efficient global context aggregation, and hierarchical representation learning within a unified framework [44].

Together, VOLO and GC ViT illustrate two complementary approaches to locality-aware Transformer design. VOLO emphasizes adaptive local aggregation through Outlook Attention, whereas GC ViT combines local window processing with globally informed context tokens. These developments foreshadow the more explicit integration of convolution and attention mechanisms discussed in Chapter 6.

## 5.6 Alternatives to Self-Attention

The remarkable success of Vision Transformers has motivated a broader investigation into the fundamental components responsible for their performance. In particular, several studies have questioned whether self-attention itself is the critical factor behind the effectiveness of Transformer-based vision models, or whether other architectural characteristics, such as tokenized representations, deep residual architectures, large-scale pre-training, and global information mixing, play a more decisive role. This line of research has led to the development of architectures that preserve the token-based processing paradigm of Vision Transformers while replacing self-attention with alternative mechanisms for information exchange.

One of the earliest and most influential examples is MLP-Mixer [45]. Similar to ViT, an input image is first divided into non-overlapping patches and projected into a sequence of visual tokens. However, instead of employing multi-head self-attention, MLP-Mixer relies entirely on multilayer perceptrons (MLPs) that are repeatedly applied across either spatial locations or feature dimensions. This design separates spatial communication from feature processing through two complementary operations: token mixing and channel mixing. For an input representation,  $\mathbf{X} \in \mathbb{R}^{N \times d}$ , where  $N$  denotes the number of image patches and  $d$  the embedding dimension, MLP-Mixer relies entirely on multilayer perceptrons (MLPs) that are repeatedly applied across either spatial locations or feature dimensions. Specifically, the token-mixing MLP exchanges information among different image patches while keeping the embedding dimension fixed:

$$\mathbf{U}_{*,i} = \mathbf{X}_{*,i} + \mathbf{W}_{\text{token},2} \sigma(\mathbf{W}_{\text{token},1} \text{LN}(\mathbf{X})_{*,i}), \quad i = 1, \dots, d, \quad (34)$$

where  $\mathbf{X}_{*,i} \in \mathbb{R}^N$  denotes the  $i$ -th embedding dimension across all tokens. Subsequently, a channel-mixing MLP is applied independently to each token, aggregating information across the embedding dimension:

$$\mathbf{Y}_{j,*} = \mathbf{U}_{j,*} + \mathbf{W}_{\text{ch},2} \sigma(\mathbf{W}_{\text{ch},1} \text{LN}(\mathbf{U})_{j,*}), \quad j = 1, \dots, N, \quad (35)$$

where  $\mathbf{U}_{j,*} \in \mathbb{R}^d$  represents the embedding of the  $j$ -th token. The token-mixing MLP therefore enables communication among spatial locations, whereas the channel-mixing MLP combines semantic information within each token representation. By repeatedly alternating these two operations, MLP-Mixer captures both spatial and feature-level dependencies without explicitly constructing an attention matrix.

A related perspective is provided by the Vision Permutator (ViP) [46], which seeks to overcome the limitations of purely token-based mixing by explicitly exploiting the two-dimensional structure of images. Unlike MLP-Mixer, which performs a linear transformation across all tokens simultaneously, ViP encodes spatial information by separately modeling interactions along the height and width dimensions of the image.

Let  $\mathbf{X}_{\text{sp}} \in \mathbb{R}^{H \times W \times d}$  denote the spatial feature representation obtained by reshaping the token sequence into its underlying two-dimensional arrangement. A Permutator block consists of two components: a Permute-MLP responsible for spatial information encoding and a Channel-MLP responsible for feature mixing. The overall block can be expressed as

$$\mathbf{Y} = \text{Permute-MLP}(\text{LN}(\mathbf{X}_{\text{sp}})) + \mathbf{X}_{\text{sp}}, \quad (36)$$

$$\mathbf{Z} = \text{Channel-MLP}(\text{LN}(\mathbf{Y})) + \mathbf{Y}, \quad (37)$$

where, LN refers to LayerNorm, Permute-MLP( $\cdot$ ) models spatial interactions by separately transforming features along the horizontal and vertical directions before aggregating them, while Channel-MLP( $\cdot$ ) performs feature mixing in a manner analogous to the feed-forward network of a Transformer block.

Consequently, ViP achieves long-range information exchange through structured permutations and linear transformations rather than through content-dependent attention weights. This design preserves the ability to propagate information globally while avoiding the explicit construction of an attention matrix [46].

Taken together, these developments reveal a common trend in the evolution of Vision Transformers. Rather than abandoning the original ViT paradigm, subsequent architectures progressively introduce mechanisms that improve spatial awareness, hierarchical feature extraction, computational efficiency, and data efficiency while retaining the core self-attention framework. At the same time, many of these advances reintroduce properties traditionally associated with convolutional networks, including locality, multi-scale processing, and data-efficient learning.

This convergence between convolutional and attention-based design principles naturally motivates a subsequent class of architectures that explicitly combine both mechanisms. These hybrid models are examined in Section 6.

**Table 5:** Representative Vision Transformer architectures and their principal mathematical innovations.

| Model                 | Year | Main Innovation                                            | Representative Formulation                                                                                                             | Primary Benefit                                           |
|-----------------------|------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| ViT [8]               | 2020 | Global self-attention on image patches                     | $\mathbf{A}_h = \text{softmax}(\mathbf{Q}_h \mathbf{K}_h^\top / \sqrt{d_k})$                                                           | Long-range dependency modeling                            |
| DeiT [22]             | 2021 | Knowledge-distillation token                               | $\mathcal{L} = \frac{1}{2} \mathcal{L}_{\text{CE}} + \frac{1}{2} \mathcal{L}_{\text{distill}}$                                         | Improved data efficiency                                  |
| Swin Transformer [23] | 2021 | Shifted-window attention                                   | $\mathbf{Z}^{(\ell+2)} = \mathbf{Z}^{(\ell+1)} + \text{MHSA}_{\text{SW}}(\mathbf{Z}^{(\ell+1)})$                                       | Hierarchical features with near-linear complexity         |
| CvT [47]              | 2021 | Convolutional projections for Q, K, and V                  | $\mathbf{Q} = \Phi(\mathbf{Z})\widetilde{\mathbf{W}}^Q, \quad \mathbf{K} = \Phi(\mathbf{Z})\widetilde{\mathbf{W}}^K$                   | Injects convolutional inductive biases into attention     |
| CoAtNet [11]          | 2021 | Sequential convolution-attention integration               | $C \rightarrow C \rightarrow A \rightarrow A$                                                                                          | Balances generalization and scalability                   |
| VOLO [43]             | 2021 | Outlook Attention                                          | $\mathbf{Y}_{(i,j)} = \widehat{\mathbf{A}}_{(i,j)} \mathbf{V}_{(i,j)}$                                                                 | Adaptive local feature aggregation                        |
| GC ViT [44]           | 2023 | Global context tokens for efficient local-global attention | $\text{Attention}(\mathbf{G}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{G}\mathbf{K}^\top / \sqrt{d_k} + \mathbf{B})\mathbf{V}$ | Efficient long-range interactions without shifted windows |
| MLP-Mixer [45]        | 2021 | Token and channel mixing using MLPs                        | $\mathbf{U}_{*,i} = \mathbf{X}_{*,i} + \text{MLP}_{\text{token}}(\mathbf{X}_{*,i})$                                                    | No self-attention required                                |
| ViP [46]              | 2021 | Dimension-wise feature permutation                         | $\mathbf{Y} = \text{PermuteMLP}(\text{LN}(\mathbf{X}_{\text{sp}})) + \mathbf{X}_{\text{sp}}$                                           | Efficient spatial interaction                             |
| MaxViT [48]           | 2022 | Multi-axis attention                                       | $\text{MBCConv} \rightarrow \text{BlockAttn} \rightarrow \text{GridAttn}$                                                              | Global and local interactions with linear complexity      |

Table 5 summarizes the principal Vision Transformer architectures discussed in this section. Although all of these models originate from the same patch-token representation framework, they introduce different mathematical modifications to the standard self-attention formulation. These modifications aim to improve spatial inductive bias (CvT, CoAtNet, VOLO), computational scalability (Swin Transformer), training efficiency (DeiT), dense prediction capability (DPT), or even replace self-attention entirely (MLP-Mixer and ViP). Together, these architectures illustrate the diverse directions that have emerged from the original ViT design.

## 6 Hybrid and Integrated Architectures

As discussed in Section 5, many recent Vision Transformer architectures reintroduce convolution-inspired mechanisms to improve locality modeling, data efficiency, and computational scalability. This trend reflects a broader convergence between convolutional and attention-based paradigms, motivating architectures that explicitly combine both forms of computation within a unified framework.

Several of the architectures introduced in Section 5, including CvT, CoAtNet, and MaxViT, already exhibit varying degrees of convolution–attention integration. In the present section, these models are revisited from a different perspective, namely as representative instances of broader hybrid design strategies.

Hybrid architectures seek to integrate the strong local inductive biases of CNNs with the global context modeling capabilities of Transformers. In the notation introduced throughout this survey, these models combine the translation-equivariant convolutional operator  $\Phi$  with the global interaction mechanism defined by the attention matrix  $\mathbf{A}_h$ . Although implementations vary considerably, most designs can be organized into three architectural strategies, illustrated in Figure 3.

### 6.1 Taxonomy of Hybrid Architectures

Hybrid architectures seek to integrate the strong local inductive biases of CNNs with the global context modeling capabilities of Transformers. In the notation introduced throughout this survey, these models combine the translation-equivariant convolutional operator  $\Phi$  with the global interaction mechanism defined by the attention matrix  $\mathbf{A}_h$ . Although implementations vary considerably, most designs can be categorized into three architectural strategies, illustrated in Figure 3.

**Sequential Integration.** The most direct strategy performs convolutional processing in the early stages of the network and attention-based reasoning at later stages. The output of a convolutional backbone  $\mathbf{h}^{(L_{\text{cnn}})} \in \mathbb{R}^{H_L \times W_L \times C_L}$  is transformed into a token sequence

$$\mathbf{Z} = \mathbf{E} \text{vec}(\mathbf{h}^{(L_{\text{cnn}})}) + \mathbf{P}, \quad (38)$$

which is subsequently processed through Transformer layers

$$\mathbf{Z}^{(\ell+1)} = \mathbf{Z}^{(\ell)} + \text{MHSA}(\text{LN}(\mathbf{Z}^{(\ell)})). \quad (39)$$

This strategy preserves convolutional inductive biases at early stages while enabling global contextual reasoning at deeper semantic levels.

**Interleaved Processing.** A second strategy alternates convolutional and attention operations throughout the network depth, applying each mechanism to the same evolving representation rather than maintaining separate convolutional and attention pathways. In its most general form, the transformation applied at layer  $\ell + 1$  is expressed as

$$\mathbf{Z}^{(\ell+1)} = \mathbf{Z}^{(\ell)} + f^{(\ell+1)}(\mathbf{Z}^{(\ell)}), \quad f^{(\ell+1)} = g_{K^{(\ell+1)}}^{(\ell+1)} \circ \dots \circ g_1^{(\ell+1)}, \quad g_i^{(\ell+1)} \in \mathcal{G}, \quad (40)$$

where  $\mathcal{G}$  denotes the set of available local and global sub-operators, including convolutional blocks, windowed attention modules, and global self-attention mechanisms. The operator  $g_i^{(\ell+1)}(\cdot)$  implicitly includes any normalization, projection, reshaping, or residual processing required by the corresponding architectural component.

Two practically important special cases arise from Equation (40):

- **Depth-alternating architectures** ( $K^{(\ell+1)} = 1$  for all  $\ell$ ). Each layer applies a single processing mechanism, with the selected operator varying throughout the network depth. In this case,  $f^{(\ell+1)} \in \{\phi^{(\ell+1)}, \text{MHSA}(\text{LN}(\cdot))\}$ , where successive layers may alternate between convolutional and attention-based processing. The specific scheduling strategy, including strict alternation, grouped stages, or fixed convolution-to-attention ratios, depends on the architecture.
- **Block-composed architectures** ( $K^{(\ell+1)} = K$  fixed across all layers). A predetermined sequence of local and global operators is applied within every computational block, so that convolution and attention interact at every depth rather than alternating across depths. MaxViT (Section 6.3) provides a canonical example, with  $(g_1, g_2, g_3) = (\text{MBConv}, \text{MHSA}_{\text{block}}, \text{MHSA}_{\text{grid}})$ , applied identically within each block.

Rather than separating local and global processing into distinct network stages, both mechanisms operate on the same evolving representation. Consequently, each feature embedding accumulates local and global contextual information throughout the network depth, yielding a more tightly coupled integration of convolutional and attention-based computation than the sequential strategy described above.

**Convolution-Augmented Attention.** A third strategy incorporates convolutional structure directly into the attention computation. Instead of computing queries, keys, and values through purely linear projections, convolutional aggregation is applied first:

$$\mathbf{Q}_h = \Phi(\mathbf{Z})\widetilde{\mathbf{W}}_h^Q, \quad \mathbf{K}_h = \Phi(\mathbf{Z})\widetilde{\mathbf{W}}_h^K, \quad \mathbf{V}_h = \Phi(\mathbf{Z})\widetilde{\mathbf{W}}_h^V. \quad (41)$$

The resulting attention mechanism operates on representations that already encode local spatial structure, thereby injecting translation-aware inductive biases into self-attention.

These three strategies occupy different positions along the locality–globality spectrum. Sequential integration maintains a clear separation between local and global processing, interleaved architectures combine both mechanisms throughout the hierarchy, and convolution-augmented attention embeds convolutional priors directly within the Transformer computation itself. The following subsections examine representative instances of these strategies through CoAtNet, MaxViT, and CvT, respectively.

Contemporary vision foundation models increasingly combine elements of all three approaches, reflecting the complementary strengths of CNN and Transformer representations.

A major motivation for hybrid designs is reducing the  $O(N^2d)$  cost of  $\mathbf{A}_h$  while retaining global context modeling. This has led to a family of efficient hybrid architectures that balance performance and computational efficiency. MobileViT [49], for instance, combines lightweight depthwise convolutional blocks  $\phi_{\text{dw}}^{(\ell)}$  with compact Transformer blocks whose token sequence is restricted to a local window, achieving favorable accuracy on mobile and edge devices. LeViT [50] takes a different approach, using a convolutional stem to produce  $\mathbf{h}^{(L_{\text{cnn}})} \in \mathbb{R}^{H_L \times W_L \times C_L}$  at reduced spatial resolution, followed by Transformer stages with shrinking token sequences via attention pooling, combined with knowledge distillation to improve convergence. EfficientFormer [51] further pushes efficiency by replacing  $\mathbf{A}_h \in \mathbb{R}^{N \times N}$  in early stages with  $\Phi$ -based local aggregation and applying global attention only in the final stage, maintaining hardware-friendly tensor layouts throughout.

Several influential architectures illustrate how these integration strategies can be realized in practice, often incorporating efficiency considerations into their core design. We now examine representative examples, following the order established in the taxonomy above.

## 6.2 CoAtNet (Convolution and Attention Network)

CoAtNet [11] is a representative example of the sequential integration strategy introduced in Section 6.1. The architecture adopts a progressive transition from convolutional processing to self-attention, following the hierarchy

$$C \rightarrow C \rightarrow A \rightarrow A.$$

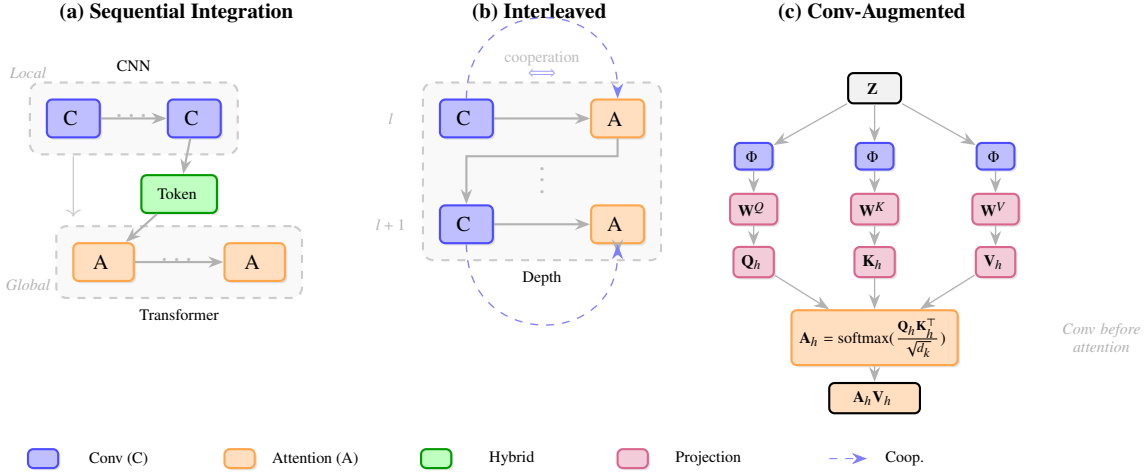


Figure 3: Hybrid CNN-Transformer architectures. (a) **Sequential integration**: the convolutional front-end extracts local visual primitives and reduces resolution; the resulting feature maps are tokenized and processed by Transformer blocks for global reasoning. Tokenization acts as a bridge between 2D and 1D representations. (b) **Interleaved processing**: convolutional and attention layers alternate across the network depth. Each MHSA block performs tokenization internally, allowing both mechanisms to cooperate continuously at all levels of abstraction. (c) **Convolution-augmented attention**: a convolutional operator  $\Phi$  is applied to the input  $\mathbf{Z}$  prior to the linear projections that produce  $\mathbf{Q}_h$ ,  $\mathbf{K}_h$ , and  $\mathbf{V}_h$ . This incorporates the convolutional inductive bias directly into the attention mechanism while preserving its global receptive field.

This design is motivated by the observation that convolution and self-attention possess complementary properties. Convolution provides strong inductive biases, such as locality and translation equivariance, which promote data-efficient learning and improved generalization. In contrast, self-attention offers input-adaptive weighting and a global receptive field, enabling the modeling of long-range dependencies and increasing the representational capacity of the network [11].

The convolutional stages employ Mobile Inverted Bottleneck Convolution (MBConv) blocks, which constitute the primary mechanism for local feature extraction in CoAtNet. Following the pre-normalization formulation adopted by the architecture, each MBConv block first normalizes the input representation, applies a depthwise convolution to perform spatial mixing, and subsequently projects the resulting features through pointwise (channel-wise) transformations. The output is then combined with the input through a residual connection, yielding

$$\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)} + \widetilde{\mathbf{W}}_{\text{pw}}^{(\ell)} \sigma \left( \Phi_{\text{dw}} \left( \text{LN}(\mathbf{h}^{(\ell-1)}) \right) \right) \widetilde{\mathbf{W}}_{\text{dw}}^{(\ell)}, \quad (42)$$

where  $\text{LN}(\cdot)$  denotes layer normalization,  $\Phi_{\text{dw}}(\cdot)$  represents the depthwise convolution operator,  $\widetilde{\mathbf{W}}_{\text{dw}}^{(\ell)}$  and  $\widetilde{\mathbf{W}}_{\text{pw}}^{(\ell)}$  denote depthwise and pointwise learnable projections, respectively, and  $\sigma(\cdot)$  is a nonlinear activation function. This formulation captures the characteristic MBConv structure, in which computationally efficient depthwise convolutions extract local spatial patterns while pointwise projections perform channel mixing within an inverted bottleneck design [52, 11].

While MBConv blocks efficiently capture local spatial interactions, their receptive field remains inherently constrained by the size of the convolutional kernel. To model long-range contextual dependencies, CoAtNet progressively replaces convolutional processing with attention-based operations at deeper levels of the hierarchy. As the network depth increases and the spatial resolution decreases, the architecture transitions from convolutional stages to attention stages. This design is motivated by the quadratic complexity of global self-attention, which becomes computationally feasible only after sufficient spatial

downsampling has been performed by the preceding convolutional layers.

The attention stages employ relative self-attention, whose attention weights are defined as

$$\mathbf{A}_h[n, m] = \text{softmax}\left(\frac{\mathbf{Q}_h[n, :] \mathbf{K}_h[m, :]^\top + r_\Delta}{\sqrt{d_k}}\right), \quad (43)$$

where  $r_\Delta$  is a learnable relative positional bias and  $\Delta = \mathbf{s}_n - \mathbf{s}_m$  denotes the relative displacement between spatial locations. Unlike absolute positional embeddings, the relative bias depends solely on the displacement between positions rather than their absolute coordinates, allowing the model to encode relative spatial relationships while simultaneously benefiting from the adaptive interactions characteristic of self-attention [11].

A key insight underlying CoAtNet is that relative attention provides a unifying formulation that bridges convolution and self-attention. The attention logits combine a content-based similarity term,  $\mathbf{Q}_h[n, :] \mathbf{K}_h[m, :]^\top$ , with a position-dependent bias,  $r_\Delta$ , thereby integrating semantic relationships and spatial structure within a single mechanism. Consequently, the model retains important inductive biases traditionally associated with convolution while exploiting the global context modeling capabilities of self-attention [11].

The sequential arrangement of  $C$  and  $A$  stages is further supported by the empirical analysis of Dai et al. [11]. Their results indicate that architectures in which convolutional stages precede attention stages exhibit superior generalization on moderate-scale datasets, whereas architectures incorporating a larger proportion of attention layers achieve greater representational capacity when trained on large-scale datasets. Consequently, the progressive transition

$$C \rightarrow C \rightarrow A \rightarrow A$$

provides an effective balance between inductive bias, model capacity, and computational efficiency.

From a representation-learning perspective, this hierarchy can be interpreted as a gradual expansion of the effective receptive field. Convolutional stages initially aggregate information from localized neighborhoods to construct increasingly abstract visual representations. Subsequently, attention stages integrate information across distant spatial locations, enabling semantically related regions to interact regardless of their geometric separation. This staged progression reflects the empirical observation that local inductive biases are most beneficial during early visual processing, whereas global interactions become increasingly important as semantic abstraction increases. The resulting

$$C \rightarrow C \rightarrow A \rightarrow A$$

hierarchy therefore constitutes the central architectural principle of CoAtNet and illustrates how sequential integration can effectively combine the complementary strengths of convolution and self-attention [11].

### 6.3 MaxViT (Multi-Axis Vision Transformer)

MaxViT [48] exemplifies the block-composed instance of the interleaved processing strategy defined in Section 6.1. Rather than separating convolution and attention into distinct stages, or alternating between them across depth, MaxViT applies a fixed composition of three sub-operators, MBConv, Block Attention, and Grid Attention, within every block at every depth.

The overall architecture follows a conventional hierarchical design consisting of a convolutional stem and multiple stages of decreasing spatial resolution. However, each stage is constructed from repeated MaxViT blocks, which combine convolutional and attentional processing within a single computational unit. Specifically, as illustrated in Fig. 4, a MaxViT block consists of three sequential sub-operators: (1) a Mobile Inverted Bottleneck Convolution (MBConv) layer for local feature extraction, (2) a Block

Attention module for local window-based self-attention, and (3) a Grid Attention module for global grid-based self-attention. Each sub-operator is followed by a residual connection that facilitates gradient propagation and feature reuse. This design enables the network to capture both local and global dependencies throughout its depth, including early high-resolution stages where global interactions are typically omitted due to computational constraints.

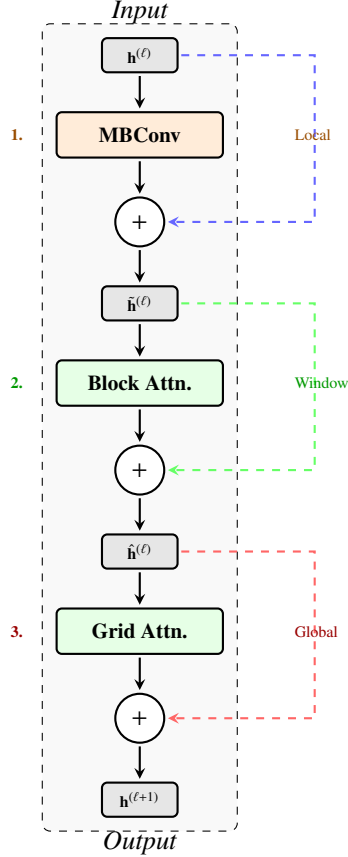


Figure 4: Compact structure of the MaxViT block, with MBConv, Block Attention, and Grid Attention connected sequentially through residual connections.

The convolutional component of the block is implemented through MBConv layers, which provide efficient local feature extraction while preserving the beneficial inductive biases of convolution. Furthermore, the depthwise convolution used within MBConv can be interpreted as a form of conditional positional encoding, allowing the architecture to incorporate spatial information without relying on explicit positional embeddings [48].

A key contribution of MaxViT is the introduction of *multi-axis self-attention*, which forms the attention component of the MaxViT block. Consider an intermediate feature representation  $\mathbf{h}^{(\ell)} \in \mathbb{R}^{H \times W \times C}$ , containing  $N = HW$  spatial locations. Rather than computing attention over all pairs of tokens, MaxViT decomposes the interaction process into two complementary sparse attention operations that are applied sequentially after a convolutional sub-step.

The first stage consists of a Mobile Inverted Bottleneck Convolution (MBConv) block, which performs efficient local feature extraction while preserving convolutional inductive biases. The output of this stage is

$$\tilde{\mathbf{h}}^{(\ell)} = \mathbf{h}^{(\ell)} + \text{MBConv}(\mathbf{h}^{(\ell)}), \quad (44)$$

where the residual connection facilitates gradient propagation and feature reuse.

Following MBConv, MaxViT applies *Block Attention*, which partitions the feature map into non-overlapping windows of size  $M \times M$ , performs self-attention independently within each window, and then restores the attended representations to their original spatial positions:

$$\widehat{\mathbf{h}}^{(\ell)} = \widetilde{\mathbf{h}}^{(\ell)} + \text{Unblock}_M(\text{MHSA}_{\text{block}}(\text{LN}(\text{Block}_M(\widetilde{\mathbf{h}}^{(\ell)})))), \quad (45)$$

where  $\text{Block}_M(\cdot)$  partitions the feature map into non-overlapping windows of size  $M \times M$ , and  $\text{Unblock}_M(\cdot)$  restores the attended window representations to their original spatial positions once self-attention has been computed within each window.

Although Block Attention efficiently models regional context, its receptive field remains restricted to the extent of each local window. To enable global communication, MaxViT subsequently applies *Grid Attention*. Instead of grouping neighboring tokens, the feature map is reorganized into sparse spatial grids whose elements are drawn from distant image locations, self-attention is performed within each grid, and the result is restored to its original spatial arrangement:

$$\mathbf{h}^{(\ell+1)} = \widehat{\mathbf{h}}^{(\ell)} + \text{Ungrid}_G(\text{MHSA}_{\text{grid}}(\text{LN}(\text{Grid}_G(\widehat{\mathbf{h}}^{(\ell)})))), \quad (46)$$

where  $\text{Grid}_G(\cdot)$  partitions the feature map into sparse grids of size  $G \times G$  whose elements originate from distant spatial locations, and  $\text{Ungrid}_G(\cdot)$  restores the attended grid representations to their original spatial arrangement once self-attention has been computed within each grid.

Equation (45) captures local interactions among neighboring tokens while maintaining computational efficiency. Equation (46) complements this by enabling global information exchange: because each grid contains elements originating from distant spatial locations, self-attention within a grid propagates information across the entire image. Consequently, Block Attention and Grid Attention provide complementary local and global interactions that together approximate the representational benefits of dense self-attention while remaining computationally efficient [48].

Instead of forming a dense attention matrix  $\mathbf{A}_h \in \mathbb{R}^{N \times N}$ , MaxViT factorizes global interactions into these two sparse attention mechanisms. Both Block Attention and Grid Attention operate on windows of fixed size —  $M \times M$  and  $G \times G$  respectively — independent of the total number of tokens  $N$ , so the cost of each scales linearly with  $N$  rather than quadratically. The resulting computational complexity can be expressed as

$$\mathcal{O}(M^2N + G^2N) = \mathcal{O}((M^2 + G^2)N), \quad (47)$$

which scales linearly with the number of spatial tokens  $N$ , in contrast to the quadratic complexity  $\mathcal{O}(N^2)$  of conventional self-attention [48]. This efficiency allows global receptive fields to be maintained throughout the entire network, even in high-resolution stages where full attention would be prohibitively expensive.

Equations (44)–(46) together instantiate the block-composed form of the interleaved-processing strategy defined by Equation (40) in Section 6.1. By combining MBConv, Block Attention, and Grid Attention within each computational block, MaxViT integrates convolutional locality and global contextual reasoning throughout the feature hierarchy rather than assigning them to separate network stages.

## 6.4 CvT (Convolutional Vision Transformer)

CvT (Convolutional Vision Transformer) [47] represents the convolution-augmented attention paradigm introduced in Section 6.1. As shown in Equation (41), CvT replaces the purely linear query, key, and value projections of a standard Vision Transformer with convolutional projections: a shared convolutional operator  $\Phi$  first aggregates local spatial context from  $\mathbf{Z}$ , after which per-head linear maps  $\widetilde{\mathbf{W}}_h^Q$ ,  $\widetilde{\mathbf{W}}_h^K$ ,  $\widetilde{\mathbf{W}}_h^V$  produce the query, key, and value representations for each attention head. In practice,  $\Phi$  is instantiated as a depthwise-separable convolution, and the key/value branches are additionally applied with stride greater than one, subsampling the token sequence used for  $\mathbf{K}_h$  and  $\mathbf{V}_h$  relative to  $\mathbf{Q}_h$ ; this

is what allows CvT to reduce the computational cost of self-attention relative to a standard ViT block, since the number of key/value tokens is reduced along with their content. The convolutional operator  $\Phi$  aggregates information from local spatial neighborhoods before attention is computed, enabling each  
790 attention head to operate on representations that already encode local contextual structure.

From an architectural perspective, the convolutional projections serve two complementary purposes. First, they introduce translation-aware inductive biases that improve optimization and generalization, particularly in moderate-data regimes where purely attention-based models often require large-scale pre-training. Second, they perform local feature aggregation prior to the computation of pairwise attention  
795 similarities, enabling the self-attention mechanism to reason over richer spatial representations [47].

Beyond the attention mechanism itself, CvT also modifies the tokenization process. Rather than relying on the fixed patchification operation used by the original Vision Transformer, CvT employs convolutional token embedding layers that jointly perform feature extraction and spatial downsampling. Let  $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$  denote the input image. The initial token representation can be expressed as

$$\mathbf{Z}^{(0)} = \Phi_{\text{embed}}(\mathbf{x}), \quad (48)$$

800 where  $\Phi_{\text{embed}}$  is a convolutional embedding operator producing a sequence of visual tokens. Unlike the patch embedding matrix  $\mathbf{E}$  introduced in Section 4.1, the convolutional embedding preserves local spatial continuity while simultaneously reducing the spatial resolution.

Similar to modern convolutional backbones, CvT adopts a hierarchical multi-stage architecture in which the spatial resolution is progressively reduced while the feature dimensionality increases. Consequently, deeper stages operate on increasingly compact and semantically rich token representations. This hierarchical design improves computational efficiency and more closely resembles the multi-scale  
805 feature hierarchy characteristic of convolutional neural networks.

An additional advantage of convolutional token embedding is that the resulting feature representations implicitly encode positional information. Unlike ViT, which requires explicit positional embeddings to recover the spatial arrangement of image patches, CvT captures spatial relationships through the  
810 convolutional operator itself. Because the receptive field of each convolution depends on the relative arrangement of neighboring pixels, the generated tokens naturally retain spatial awareness. Consequently, CvT eliminates the need for explicit positional embeddings without degrading performance [47].

The complete CvT block can therefore be interpreted as the composition

$$\widehat{\mathbf{Z}}^{(\ell+1)} = \mathbf{Z}^{(\ell)} + \text{MHSA}(\mathbf{Q}_h^{(\ell)}, \mathbf{K}_h^{(\ell)}, \mathbf{V}_h^{(\ell)}), \quad (49)$$

815 with  $\mathbf{Q}_h^{(\ell)}, \mathbf{K}_h^{(\ell)}, \mathbf{V}_h^{(\ell)}$  obtained from  $\text{LN}(\mathbf{Z}^{(\ell)})$  via Equation (41), followed by a standard Transformer feed-forward sub-block with its own residual connection,

$$\mathbf{Z}^{(\ell+1)} = \widehat{\mathbf{Z}}^{(\ell+1)} + \text{FFN}(\text{LN}(\widehat{\mathbf{Z}}^{(\ell+1)})). \quad (50)$$

In contrast to conventional Vision Transformers, where locality is introduced through positional encodings or attention constraints, locality in CvT is embedded directly into the feature representations participating in the attention computation.

## 820 6.5 The Locality–Globality Spectrum

Hybrid CNN–Transformer architectures can be understood as operating along the locality–globality spectrum identified in Section 4: the three integration strategies formalized above differ primarily in *where* in the network the transition from local  $\Phi$ -based processing to global  $\mathbf{A}_h$ -based processing occurs. Sequential integration places the boundary between the convolutional front-end  $\mathbf{h}^{(L_{\text{cnn}})}$  and the Trans-  
825 former back-end  $\mathbf{Z}$ ; interleaved processing composes local sub-operators  $\phi^{(\ell)}$  and global sub-operators MHSA within the same computational stream at every depth, either by alternating between them across

layers or, as in MaxViT (Section 6.3), by applying a fixed composition of both within every block; and convolution-augmented attention embeds  $\Phi$  within the query, key, and value projections of each attention head. The convergence of CNNs and ViTs evidenced by architectures such as ConvNeXt (Section 3.3), CvT, CoAtNet, and MaxViT suggests that future vision systems will increasingly rely on adaptive mix-  
830 tures of local and global processing, with the architectural boundary between the two paradigms becoming progressively less distinct. While the preceding examples focus primarily on CNN–Transformer hybrids, the broader landscape of cross-paradigm architectures is considerably richer.

### 6.5.1 Generative-Discriminative Integration

The clearest mathematical expression of cross-paradigm integration is the Diffusion Transformer (Sec-  
835 tion 8.4.4), where the denoising network  $\epsilon_\theta(\mathbf{Z}_t, t, \mathbf{c})$  is parameterized by a ViT operating on latent patch tokens  $\mathbf{Z}_t$ . The latent space itself is produced by a convolutional encoder  $\mathcal{E}$  applied once to the clean image,  $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$ , after which the noised latents  $\mathbf{z}_t$  for  $t > 0$  are obtained by the forward diffusion process applied directly in latent space, rather than by re-encoding pixel-space observations at each step.  
840 The full generative pipeline is therefore:

$$\mathbf{x}_0 \xleftarrow{\mathcal{D}} \mathbf{z}_0 \xleftarrow{p_\theta(\mathbf{z}_0|\mathbf{Z}_1, \mathbf{c})} \mathbf{z}_1 \xleftarrow{p_\theta(\mathbf{z}_1|\mathbf{Z}_2, \mathbf{c})} \dots \xleftarrow{p_\theta(\mathbf{z}_{T-1}|\mathbf{Z}_T, \mathbf{c})} \mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where  $\mathcal{D}$  is a convolutional decoder applied only at the final step to recover  $\mathbf{x}_0$ , and each reverse step is computed by the Transformer  $\epsilon_\theta(\mathbf{Z}_t, t, \mathbf{c})$  with  $\mathbf{Z}_t = \mathbf{E} \text{vec}(\mathbf{z}_t) + \mathbf{P}$ . Similarly, multimodal systems such as CLIP align a ViT image encoder producing an image embedding  $\mathbf{v} \in \mathbb{R}^d$  with a text Transformer producing a text embedding  $\mathbf{u} \in \mathbb{R}^d$ , via a contrastive objective over paired embeddings in  $\mathcal{Y}$ , enabling  
845 zero-shot transfer without task-specific supervision.

## 7 Foundation Models and Multimodal Learning

Foundation models are built upon the premise that large-scale pretraining on diverse and heterogeneous data sources can produce general-purpose representations that transfer across tasks, domains, and modal-  
850 ities. Rather than optimizing a model for a single downstream objective, the goal is to learn reusable representations that can be adapted to a broad range of perceptual, reasoning, and generative capabilities. Figure 5 illustrates the foundation-model paradigm, in which large-scale pretraining on diverse visual and multimodal data produces transferable representations that can be adapted to a broad spectrum of downstream capabilities, ranging from traditional computer vision tasks to multimodal reasoning and generative modeling.

The central idea underlying foundation models is that scale, diversity of training data, and self-supervised learning can jointly produce representations that generalize beyond the tasks observed during training. Unlike conventional models trained to optimize a single task, foundation models learn trans-  
855 ferable representations that can be reused across multiple downstream settings with little or no task-specific adaptation. In computer vision, this paradigm encompasses both self-supervised visual representation learning and multimodal learning. Methods such as DINO, DINOv2, and DINOv3 demonstrate  
860 that large-scale self-supervised pretraining can produce highly transferable visual representations, while models such as CLIP, ImageBind, and LLaVA extend these principles to vision–language and multimodal settings. Together, these developments illustrate how the general learning framework introduced in Section 2.1 can be extended beyond task-specific optimization toward increasingly general-purpose  
865 visual intelligence.

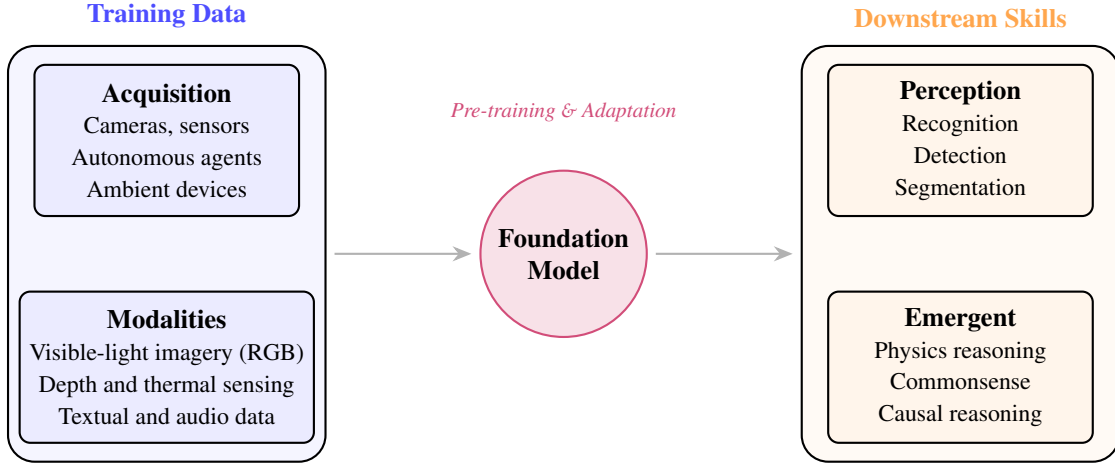


Figure 5: Conceptual illustration of the foundation-model paradigm. Large-scale multimodal data originating from diverse acquisition sources and modalities are used during pre-training to learn a general-purpose foundation model. The resulting representations can subsequently be adapted to a broad spectrum of downstream capabilities, ranging from traditional perception tasks to more advanced reasoning and emergent capabilities.

## 7.1 Promptable Segmentation: Segment Anything Model

The Segment Anything Model (SAM) [54] represents a major step toward general-purpose vision systems by extending the foundation model paradigm to image segmentation. Unlike conventional segmentation models that are trained for specific datasets, object categories, or segmentation tasks, SAM is designed to segment arbitrary objects in previously unseen images through prompt-based interaction. This ability is enabled by large-scale pretraining on the SA-1B dataset and by a promptable architecture that decouples image understanding from task-specific supervision.

From a mathematical perspective, SAM extends the conventional vision mapping  $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$  to a promptable mapping

$$f_\theta : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{Y},$$

where  $\mathcal{P}$  denotes a space of sparse prompts (points and bounding boxes) or dense prompts (masks), and  $\mathcal{Y} \subseteq \{0, 1\}^{H \times W \times K}$  is the segmentation output space introduced in Section 2.1. The architecture is composed of three principal modules: an image encoder  $\mathcal{E}_{\text{img}}$ , a prompt encoder  $\mathcal{E}_{\text{prompt}}$ , and a mask decoder  $\mathcal{D}_\theta$ . Together, these modules define the mapping

$$f_\theta(\mathbf{x}, \mathbf{p}) = \mathcal{D}_\theta(\mathbf{z}_{\text{img}}, \mathbf{z}_{\text{prompt}}), \quad (51)$$

where the latent representations are produced by

$$\mathbf{z}_{\text{img}} = \mathcal{E}_{\text{img}}(\mathbf{x}), \quad \mathbf{z}_{\text{prompt}} = \mathcal{E}_{\text{prompt}}(\mathbf{p}).$$

More specifically:

- The image encoder produces

$$\mathbf{z}_{\text{img}} = \mathcal{E}_{\text{img}}(\mathbf{x}) \in \mathbb{R}^{(H'W') \times d},$$

where  $\mathcal{E}_{\text{img}}$  is a large Vision Transformer backbone (ViT-H/16) pretrained using masked autoencoding [55]. The resulting token matrix follows the same representation structure as the Transformer tokens  $\mathbf{Z}$  introduced in Section 4. Since the image embedding depends only on  $\mathbf{x}$ , it can be computed once and reused across multiple prompts.

- The prompt encoder produces

$$\mathbf{z}_{\text{prompt}} = \mathcal{E}_{\text{prompt}}(\mathbf{p}) \in \mathbb{R}^{M \times d},$$

where  $M$  denotes the number of prompt elements. Sparse prompts are mapped to positional embeddings analogous to  $\mathbf{p}_n$ , while dense prompts are processed through convolutional operators  $\Phi$  to generate spatial feature maps.

- The mask decoder  $\mathcal{D}_\theta$  is a lightweight Transformer that performs cross-attention between  $\mathbf{z}_{\text{prompt}}$  and  $\mathbf{z}_{\text{img}}$  and produces segmentation masks in the output space  $\mathcal{Y}$ .

The model is trained using a combination of focal loss and Dice loss:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{focal}}(f_\theta(\mathbf{x}, \mathbf{p}), \mathbf{y}) + \lambda \mathcal{L}_{\text{Dice}}(f_\theta(\mathbf{x}, \mathbf{p}), \mathbf{y}), \quad (52)$$

where  $\mathbf{y} \in \mathcal{Y}$  denotes the ground-truth mask and  $\lambda$  controls the relative contribution of the Dice term.

A key design insight of SAM is the separation between the computationally expensive image encoder  $\mathcal{E}_{\text{img}}$  and the lightweight prompt-conditioned decoder  $\mathcal{D}_\theta$ . The image encoding stage incurs a cost of approximately  $O(N^2d)$  and is executed only once per image, while prompt decoding scales approximately as  $O(Md)$  and can be evaluated repeatedly for different user queries. This decomposition enables interactive real-time segmentation while preserving the global receptive field provided by the attention matrix  $\mathbf{A}_h$ .

## 7.2 Self-Supervised Visual Representations: From DINO to DINOv3

The emergence of Vision Transformers enabled a new family of self-supervised learning methods capable of learning semantic visual representations directly from unlabeled images. Among the most influential approaches is DINO (Distillation with No Labels), which reformulates self-supervised learning as a teacher–student self-distillation problem. Unlike contrastive methods that rely on negative samples, DINO learns visual representations by enforcing consistency between different views of the same image through a momentum-updated teacher network.

Let  $\mathbf{x} \in \mathcal{X}$  denote an input image and let  $\mathcal{V} = \{v_1, \dots, v_M\}$  be a collection of augmented views generated from  $\mathbf{x}$ . Two Vision Transformers are considered: a student encoder  $f_{\theta_s}$  and a teacher encoder  $f_{\theta_t}$ .

For each view  $v \in \mathcal{V}$ , the student and teacher Vision Transformers produce final token matrices

$$\mathbf{Z}_s^{(v)}, \mathbf{Z}_t^{(v)} \in \mathbb{R}^{(N+1) \times d},$$

where

$$\mathbf{Z}^{(v)} = [\mathbf{z}_0^{(v)}, \mathbf{z}_1^{(v)}, \dots, \mathbf{z}_N^{(v)}].$$

The first token  $\mathbf{z}_0^{(v)} \in \mathbb{R}^d$  corresponds to the classification token and serves as a global image representation, whereas the remaining tokens  $\mathbf{z}_1^{(v)}, \dots, \mathbf{z}_N^{(v)}$  encode local patch-level information.

The self-distillation objective is applied exclusively to the classification tokens  $\mathbf{z}_{0,s}^{(v)}$  and  $\mathbf{z}_{0,t}^{(v)}$ . These are passed through a projection head

$$g_\psi : \mathbb{R}^d \rightarrow \mathbb{R}^K,$$

parameterized by  $\psi$ , yielding

$$\mathbf{y}_s^{(v)} = g_\psi(\mathbf{z}_{0,s}^{(v)}), \quad \mathbf{y}_t^{(v)} = g_\psi(\mathbf{z}_{0,t}^{(v)}), \quad (53)$$

where  $\mathbf{y}_s^{(v)}, \mathbf{y}_t^{(v)} \in \mathbb{R}^K$  are the student and teacher outputs, respectively, and  $K$  denotes the dimensionality of the prototype space used by the self-distillation objective.

The outputs are transformed into probability distributions using a temperature-scaled softmax,

$$\mathbf{p}_s^{(v)} = \text{softmax}\left(\frac{\mathbf{y}_s^{(v)}}{\tau_s}\right), \quad \mathbf{p}_t^{(v)} = \text{softmax}\left(\frac{\mathbf{y}_t^{(v)}}{\tau_t}\right), \quad (54)$$

920 where  $\tau_s$  and  $\tau_t$  denote the student and teacher temperatures, respectively.

The student is trained to match the teacher predictions across different views of the same image through the cross-entropy objective

$$\mathcal{L}_{\text{DINO}} = - \sum_{\substack{v_i, v_j \in \mathcal{V} \\ v_i \neq v_j}} \mathbf{p}_t^{(v_i)\top} \log \mathbf{p}_s^{(v_j)}. \quad (55)$$

The teacher parameters are not optimized directly through gradient descent. Instead, they are updated using an exponential moving average of the student parameters,

$$\theta_t \leftarrow \mu \theta_t + (1 - \mu) \theta_s, \quad \mu \approx 1, \quad (56)$$

925 which provides stable targets throughout training [56].

A remarkable property of DINO is that semantic organization emerges despite the fact that the loss is applied only to the classification token. The patch tokens  $\mathbf{z}_1'', \dots, \mathbf{z}_N''^{(v)}$  do not appear explicitly in the optimization objective. Nevertheless, the self-attention mechanism induces structured semantic information within these spatial embeddings, causing object regions and semantic parts to emerge without any  
930 segmentation labels or pixel-level supervision [56].

**iBOT and Patch-Level Self-Supervision.** A complementary development is represented by iBOT (Image BERT Pre-Training with Online Tokenizer) [57], which extends the teacher–student paradigm to patch-level representations. Rather than constraining only the classification token, iBOT additionally applies self-supervision to the patch tokens  $\mathbf{z}_1'', \dots, \mathbf{z}_N''$ , encouraging the Vision Transformer to learn semantically meaningful dense representations. This extension substantially improves the quality of local  
935 features and forms one of the building blocks of later DINOv2 models.

**DINOv2: Visual Foundation Models.** Building upon these observations, DINOv2 was proposed with the explicit goal of constructing a visual foundation model capable of producing general-purpose visual features across image distributions and downstream tasks [58]. Unlike the original DINO, whose training  
940 was primarily limited to ImageNet-scale datasets, DINOv2 scales both model capacity and training data. A central contribution is the creation of a curated corpus containing approximately 142 million images obtained through an automatic data filtering, clustering, and deduplication pipeline designed to maximize data quality and diversity [58].

DINOv2 retains the teacher–student formulation of DINO while combining it with patch-level self-supervision inspired by iBOT [57]. Given a set of global views  $\mathcal{V}_{\text{global}}$  and local views  $\mathcal{V}_{\text{local}}$ , the student learns representations that remain invariant across different image transformations while preserving fine-grained spatial structure. Denoting the teacher and student outputs for view  $v$  by  $\mathbf{p}_t^{(v)}$  and  $\mathbf{p}_s^{(v)}$ , respectively, the image-level objective can be expressed as

$$\mathcal{L}_{\text{DINOv2}} = - \sum_{v \in \mathcal{V}_{\text{global}} \cup \mathcal{V}_{\text{local}}} \mathbf{p}_t^{(v)\top} \log \mathbf{p}_s^{(v)}. \quad (57)$$

The resulting representations exhibit strong transferability at both the image level and the pixel level.  
950 In particular, the patch-token matrix

$$\mathbf{Z}'' = [\mathbf{z}_0'', \mathbf{z}_1'', \dots, \mathbf{z}_N'']$$

encodes rich semantic information that frequently separates objects, parts, and background regions without explicit supervision. Principal component analysis applied to the patch embeddings often reveals coherent semantic structures and object correspondences across images, demonstrating the emergence of dense visual understanding from purely self-supervised learning [58].

**DINOv3: Toward Universal Dense Visual Representations.** While DINOv2 established self-supervised Vision Transformers as powerful visual foundation models, DINOv3 further improves the quality and stability of patch-level representations, with the explicit objective of learning dense visual features that remain semantically consistent across images, viewpoints, scales, and appearance variations [59].

Let

$$\mathbf{Z}''^{(v)} = [\mathbf{z}_0''^{(v)}, \mathbf{z}_1''^{(v)}, \dots, \mathbf{z}_N''^{(v)}] \in \mathbb{R}^{(N+1) \times d}$$

denote the token matrix produced by the Vision Transformer for an augmented view  $v$  of an image. As in previous sections,  $\mathbf{z}_0''^{(v)}$  represents the classification token, whereas  $\mathbf{z}_n''^{(v)}$ ,  $n = 1, \dots, N$ , correspond to the patch tokens. While DINO and DINOv2 primarily enforce consistency at the image level through the classification token, DINOv3 places additional emphasis on preserving semantic correspondences within the patch-token representations themselves.

Given two views  $v_i$  and  $v_j$  of the same scene, the objective is to encourage corresponding patch embeddings to occupy nearby locations in the representation space  $\mathbb{R}^d$ . Let  $C(v_i, v_j)$  denote the set of corresponding patch indices between both views. The dense consistency objective can be expressed as

$$\mathcal{L}_{\text{dense}} = \sum_{(m,n) \in C(v_i, v_j)} \left\| \mathbf{z}_m''^{(v_i)} - \mathbf{z}_n''^{(v_j)} \right\|_2^2, \quad (58)$$

which encourages semantically equivalent regions to remain aligned under view transformations. As a consequence, the learned patch embeddings become more stable and discriminative than those produced by previous self-supervised approaches.

This emphasis on dense feature consistency yields representations that are simultaneously useful at both global and local scales. The classification token  $\mathbf{z}_0''$  continues to provide a compact image-level descriptor suitable for recognition and retrieval tasks, whereas the patch-token matrix

$$\mathbf{Z}_p'' = [\mathbf{z}_1'', \dots, \mathbf{z}_N''] \in \mathbb{R}^{N \times d}$$

encodes fine-grained spatial information that can be exploited for correspondence estimation, dense retrieval, object discovery, segmentation, image matching, and other pixel-level vision tasks without task-specific supervision [59].

Consequently, DINOv3 further strengthens the foundation-model paradigm introduced by DINOv2. Whereas DINOv2 demonstrated that large-scale self-supervised learning can produce general-purpose visual representations, DINOv3 extends this capability to dense visual understanding by improving the semantic consistency, stability, and transferability of local patch descriptors. Together, DINOv2 and DINOv3 represent a significant step toward universal visual foundation models whose representations remain effective across both image-level and pixel-level downstream tasks with minimal supervised adaptation [58, 59].

Table 6 highlights the progression of the DINO family. DINO established the teacher–student self-distillation framework, DINOv2 transformed this paradigm into a large-scale visual foundation model through data and model scaling, and DINOv3 further emphasizes dense semantic consistency at the patch level. Together, these developments illustrate the evolution from image-level visual representations toward universal visual descriptors capable of supporting both global recognition and fine-grained spatial reasoning.

A particularly compelling demonstration of the representational quality of DINOv3 is provided by INSID3 [60]. Unlike conventional segmentation pipelines, which require dedicated segmentation heads

Table 6: Evolution of the DINO family of self-supervised visual representation models.

| Model       | Training Signal                                          | Representation Level                     | Main Contribution                                                                                                                   |
|-------------|----------------------------------------------------------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| DINO [56]   | Cross-view self-distillation $\mathcal{L}_{\text{DINO}}$ | CLS token $\mathbf{z}''_0$               | Introduces teacher-student self-distillation without labels and reveals emergent semantic structure in Vision Transformers.         |
| DINOv2 [58] | DINO + patch supervision (iBOT-inspired)                 | Image and patch tokens $\mathbf{Z}''$    | Scales self-supervision using curated data, larger ViTs, and improved training, yielding a general-purpose visual foundation model. |
| DINOv3 [59] | Dense semantic consistency $\mathcal{L}_{\text{dense}}$  | Dense patch descriptors $\mathbf{z}''_n$ | Improves patch-level stability and semantic correspondence for retrieval, matching, segmentation, and dense visual reasoning.       |

or auxiliary foundation models, INSID3 operates directly in the dense DINOv3 feature space. The method establishes semantic correspondences between reference and target images using patch-token similarities and performs segmentation through clustering and similarity propagation. The success of this entirely training-free framework provides strong evidence that DINOv3 patch embeddings encode sufficiently rich semantic information to support object-, part-, and instance-level segmentation without task-specific adaptation.

### 7.3 Multimodal Vision-Language Representations: CLIP

While Vision Transformers were originally developed for visual recognition, their scalability and representation capacity have enabled their adoption within multimodal foundation models. A landmark example is CLIP (Contrastive Language-Image Pretraining) [61], which extends the Vision Transformer paradigm to joint vision-language representation learning by constructing a shared embedding space for images and text.

Contrastive pre-training has emerged as one of the dominant paradigms for aligning visual and textual representations in a shared embedding space. Given a batch of image-text pairs, separate image and text encoders are used to project both modalities into a common latent space. Training is then performed using a contrastive objective that increases the similarity between corresponding image-text pairs while decreasing the similarity between mismatched pairs.

Given a mini-batch of aligned image-text pairs

$$\{(\mathbf{x}^{(i)}, \mathbf{T}^{(i)})\}_{i=1}^B,$$

two independent encoders are used to extract modality-specific feature representations. A Vision Transformer image encoder produces image features,  $\mathbf{f}_{\text{img}}^{(i)} \in \mathbb{R}^{d_i}$ , while a text Transformer generates text features,  $\mathbf{f}_{\text{text}}^{(i)} \in \mathbb{R}^{d_t}$ .

Following the notation introduced in Section 4, the visual feature is obtained from the final classification token,

$$\mathbf{f}_{\text{img}} = \mathbf{z}''_{0,\text{img}} = \mathbf{Z}''_{\text{img}}[0, :].$$

The modality-specific representations are subsequently projected into a shared embedding space through learnable projection matrices  $\mathbf{W}_{\text{img}}$  and  $\mathbf{W}_{\text{text}}$ ,

$$\mathbf{z}_{\text{img}}^{(i)} = \frac{\mathbf{f}_{\text{img}}^{(i)} \mathbf{W}_{\text{img}}}{\|\mathbf{f}_{\text{img}}^{(i)} \mathbf{W}_{\text{img}}\|_2}, \quad \mathbf{z}_{\text{text}}^{(i)} = \frac{\mathbf{f}_{\text{text}}^{(i)} \mathbf{W}_{\text{text}}}{\|\mathbf{f}_{\text{text}}^{(i)} \mathbf{W}_{\text{text}}\|_2},$$

where  $\mathbf{z}_{\text{img}}^{(i)}, \mathbf{z}_{\text{text}}^{(i)} \in \mathbb{R}^{d_e}$  are  $\ell_2$ -normalized embeddings. The use of normalization allows the similarity between image and text representations to be measured through cosine similarity.

Pairwise similarities are computed between all image and text embeddings in the mini-batch, yielding a similarity matrix

$$S_{ij} = \exp(\tau) \mathbf{z}_{\text{img}}^{(i)\top} \mathbf{z}_{\text{text}}^{(j)}, \quad (59)$$

where  $\tau$  is a learnable temperature parameter. The diagonal entries  $S_{ii}$  correspond to matching image–text pairs, whereas off-diagonal entries represent mismatched pairs. Consequently, the training objective encourages the diagonal similarities to be larger than the remaining entries of the matrix.

The resulting contrastive objective can be interpreted as two symmetric classification tasks: image-to-text retrieval and text-to-image retrieval. For the image-to-text direction, the probability assigned to the matching text of image  $i$  is

$$p_{i \rightarrow j} = \frac{\exp(S_{ij})}{\sum_{k=1}^B \exp(S_{ik})}, \quad (60)$$

while an analogous distribution is defined for the text-to-image direction. Let

$$\mathcal{L}_{\text{I2T}} = -\frac{1}{B} \sum_{i=1}^B \log p_{i \rightarrow i}, \quad \mathcal{L}_{\text{T2I}} = -\frac{1}{B} \sum_{i=1}^B \log p_{i \leftarrow i}, \quad (61)$$

denote the image-to-text and text-to-image cross-entropy losses, respectively. The final CLIP objective is obtained by averaging both directions,

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2} (\mathcal{L}_{\text{I2T}} + \mathcal{L}_{\text{T2I}}). \quad (62)$$

where each term treats the aligned image–text pair as the correct classification target within the mini-batch [61].

The learned shared embedding space enables zero-shot transfer to unseen tasks. Given a set of  $K$  candidate classes represented by textual prompts with embeddings  $\mathbf{z}_{\text{text},k}$ , classification is performed through

$$\hat{y} = \arg \max_{k \in \{1, \dots, K\}} \mathbf{z}_{\text{img}}^\top \mathbf{z}_{\text{text},k}, \quad (63)$$

where  $\mathbf{z}_{\text{img}}$  denotes the embedding of the query image. Consequently, CLIP reformulates visual recognition as a similarity-matching problem in a shared vision–language embedding space rather than learning a task-specific classifier [61].

## 7.4 ImageBind: A Unified Multimodal Embedding Space

While CLIP aligns visual and textual representations through contrastive learning, IMAGEBIND extends this paradigm by learning a single embedding space shared across multiple modalities, including images, text, audio, depth, thermal imagery, and inertial measurement unit (IMU) signals [62]. The central observation is that images naturally co-occur with many other sensory modalities and can therefore serve as a universal anchor for multimodal representation learning. Remarkably, IMAGEBIND demonstrates that direct supervision between all pairs of modalities is unnecessary: aligning every modality with images is sufficient to induce emergent alignments among modalities that are never observed together during training.

Let  $I \in \mathcal{X}_I$  denote an image observation and  $M \in \mathcal{X}_M$  an observation from another modality, such as text, audio, depth, thermal imagery, or IMU data. IMAGEBIND considers naturally aligned pairs  $(I, M)$ , obtained from datasets where images co-occur with a second modality. Examples include image–text pairs collected from web data, video–audio pairs, image–depth pairs, image–thermal pairs, and video–IMU pairs [62].

For a modality pair  $(I, M)$ , modality-specific encoders  $f : \mathcal{X}_I \rightarrow \mathbb{R}^d$ ,  $g : \mathcal{X}_M \rightarrow \mathbb{R}^d$  produce normalized latent representations

$$\mathbf{q} = f(I), \quad \mathbf{k} = g(M), \quad (64)$$

with

$$\mathbf{q}, \mathbf{k} \in \mathbb{R}^d, \quad \|\mathbf{q}\|_2 = \|\mathbf{k}\|_2 = 1.$$

Given a collection of  $B$  paired samples  $(I_i, M_i)$ , let

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^\top \\ \vdots \\ \mathbf{q}_B^\top \end{bmatrix} \in \mathbb{R}^{B \times d}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{k}_1^\top \\ \vdots \\ \mathbf{k}_B^\top \end{bmatrix} \in \mathbb{R}^{B \times d},$$

where each row corresponds to the embedding of one image or modality sample. The pairwise similarity matrix is defined as

$$\mathbf{S} = \frac{1}{\tau} \mathbf{Q} \mathbf{K}^\top \in \mathbb{R}^{B \times B}, \quad (65)$$

where  $\tau$  is a temperature parameter. Its entries

$$S_{ij} = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\tau}$$

measure the similarity between image  $I_i$  and modality sample  $M_j$ .

IMAGEBIND aligns both modalities through a contrastive InfoNCE objective,

$$\mathcal{L}_{I,M} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(S_{ii})}{\sum_{j=1}^B \exp(S_{ij})}, \quad (66)$$

where the diagonal entries  $S_{ii}$  correspond to matching image–modality pairs and the off-diagonal entries correspond to negative pairs. Following the original formulation, a symmetric objective is applied in both directions,

$$\mathcal{L} = \sum_{M \in \mathcal{M} \setminus \{I\}} (\mathcal{L}_{I,M} + \mathcal{L}_{M,I}), \quad (67)$$

with  $\mathcal{L}_{M,I}$  obtained by exchanging the roles of  $\mathbf{Q}$  and  $\mathbf{K}$ .

From a mathematical perspective, IMAGEBIND can be interpreted as a generalization of CLIP in which the image modality acts as a common anchor. Whereas CLIP aligns only images and text, IMAGEBIND optimizes multiple image-centered contrastive objectives, inducing a single shared representation space across all modalities.

A distinctive property of IMAGEBIND is that it never requires direct supervision between non-image modalities. For example, the model is trained using image–text and image–audio pairs, but never using audio–text pairs. Nevertheless, because every modality is aligned to the same image representation space, an emergent alignment naturally appears among all modalities. Consequently, semantically related observations from text, audio, depth, thermal imagery, or IMU signals tend to occupy neighboring regions of the common latent space despite the absence of explicit supervision between them [62].

This emergent property enables a variety of previously unseen capabilities, including cross-modal retrieval, open-vocabulary classification, multimodal embedding arithmetic, audio-conditioned image generation, and cross-modal transfer learning. More broadly, IMAGEBIND demonstrates that a sufficiently expressive anchor modality—in this case images—can serve as the foundation for learning a general multimodal representation space without requiring exhaustive pairwise supervision among modalities. As such, IMAGEBIND represents an important step toward universal multimodal foundation models capable of integrating heterogeneous sensory information within a single latent representation space [62].

## 7.5 LLaVA: Visual Instruction Tuning and Multimodal Reasoning

Building upon the multimodal alignment paradigms introduced by CLIP and generalized by ImageBind, LLaVA (Large Language and Vision Assistant) integrates visual representations into a large language model to enable multimodal reasoning and instruction following. Rather than learning representations primarily for retrieval or classification, LLaVA integrates visual features into a large language model, enabling image-grounded dialogue, instruction following, and complex visual reasoning tasks. The architecture combines a pretrained CLIP vision encoder with an instruction-tuned language model connected through a lightweight trainable projection layer [63].

From a broader perspective, LLaVA can be viewed as a natural evolution of CLIP-style multimodal learning. Whereas CLIP aligns images and text within a joint embedding space and ImageBind extends this alignment to additional sensory modalities, LLaVA injects visual representations directly into a large language model, transforming multimodal alignment into multimodal reasoning and instruction following. Consequently, the focus shifts from learning shared embeddings to generating coherent responses grounded in both visual and linguistic context.

Let  $\mathbf{X}_v$  denote an input image. The image is first processed by a pretrained CLIP vision encoder  $g(\cdot)$ , which extracts a sequence of visual feature vectors

$$\mathbf{F}_v = g(\mathbf{X}_v) \in \mathbb{R}^{N_v \times d_v}, \quad (68)$$

where  $N_v$  denotes the number of visual tokens produced by the vision encoder and  $d_v$  is the dimensionality of each visual feature vector. These features capture high-level semantic information learned during large-scale vision–language pretraining.

Since the visual feature space generally differs from the embedding space of the language model, LLaVA introduces a trainable projection layer that maps visual representations into language-compatible tokens. Let  $\mathbf{W} \in \mathbb{R}^{d_v \times d_\ell}$  be a learnable projection matrix, where  $d_\ell$  denotes the embedding dimension of the large language model. The projected visual tokens are obtained as

$$\mathbf{H}_v = \mathbf{F}_v \mathbf{W}^\top \in \mathbb{R}^{N_v \times d_\ell}, \quad (69)$$

where  $\mathbf{H}_v$  represents the visual token sequence expressed in the language-model embedding space. Consequently, visual and textual tokens share a common representation space and can be processed jointly by the language model [63].

Given a textual instruction  $X_{\text{instr}} = \{x_1, \dots, x_{N_t}\}$ , consisting of  $N_t$  discrete text tokens, the language-model embedding layer transforms the instruction into

$$\mathbf{E}_{\text{instr}} = \text{Embed}(X_{\text{instr}}) \in \mathbb{R}^{N_t \times d_\ell}, \quad (70)$$

where each row corresponds to the embedding of one instruction token. The projected visual tokens and textual embeddings are then concatenated along the sequence dimension to construct a unified multimodal context,

$$\mathbf{S} = [\mathbf{H}_v; \mathbf{E}_{\text{instr}}] \in \mathbb{R}^{(N_v + N_t) \times d_\ell}, \quad (71)$$

which is provided as input to the language model.

The multimodal sequence is subsequently processed by the autoregressive language model  $f_\phi(\cdot)$ , which generates a response conditioned jointly on the visual content and the user instruction,

$$X_a = f_\phi(\mathbf{S}), \quad (72)$$

where  $X_a$  denotes the generated response. Because both modalities are represented within the same embedding space, the self-attention mechanism of the language model can attend simultaneously to visual and textual tokens, thereby integrating image perception and language reasoning within a single generative framework.

Unlike conventional vision–language pipelines that separate perception from reasoning, this formulation embeds visual information directly into the autoregressive generation process, enabling the language model to reason over image content while producing natural-language responses [63].

The most significant contribution of LLaVA, however, lies not in the projection layer itself but in the training paradigm known as *visual instruction tuning*. Inspired by the success of instruction-tuned systems such as InstructGPT and ChatGPT, the authors extend the same principle to multimodal learning. A key challenge is the scarcity of large-scale image–instruction datasets. To address this limitation, LLaVA leverages GPT-4 to transform existing image–caption pairs into instruction-following examples. Using captions and object annotations as symbolic descriptions of an image, GPT-4 generates conversational questions, detailed image descriptions, and complex reasoning tasks, thereby creating multimodal instruction datasets without extensive manual annotation [63].

LLaVA is trained using the standard autoregressive objective employed by large language models. Given the multimodal context

$$\mathbf{S} = [\mathbf{H}_v; \mathbf{E}_{\text{instr}}],$$

and a target response

$$X_a = \{x_1, \dots, x_L\},$$

the model learns to predict each output token conditioned on the visual content, the instruction, and all previously generated tokens. The corresponding conditional probability is

$$p(X_a | \mathbf{S}) = \prod_{i=1}^L p_{\theta}(x_i | \mathbf{S}, x_1, \dots, x_{i-1}), \quad (73)$$

where  $x_i$  denotes the  $i$ -th response token and  $\theta = \{\mathbf{W}, \phi\}$  represents the trainable parameters of the projection layer and language model. Training therefore maximizes the likelihood of the target response given the multimodal context. In contrast to traditional vision–language models optimized primarily for retrieval or classification, this autoregressive formulation enables the model to learn visual understanding, instruction following, conversational interaction, and multimodal reasoning within a unified generative framework [63].

To facilitate stable multimodal learning, LLaVA adopts a two-stage training procedure. During the first stage, referred to as *feature alignment*, both the CLIP encoder and the language model remain frozen while only the projection matrix  $\mathbf{W}$  is optimized. This stage effectively learns a visual tokenizer that maps image features into the language embedding space. In the second stage, the projection layer and language model are jointly fine-tuned on multimodal instruction data, while the visual encoder remains frozen. This strategy preserves the visual knowledge encoded by CLIP and the linguistic reasoning capabilities of Vicuna while enabling effective multimodal integration [63].

Conceptually, LLaVA represents a shift from representation alignment to multimodal reasoning. Whereas CLIP aligns images and text within a shared embedding space, LLaVA integrates visual perception directly into the reasoning process of an instruction-following language model. As a result, the system can perform image description, visual question answering, commonsense reasoning, conversational interaction, and instruction-guided problem solving within a single autoregressive framework. Experimental evaluations demonstrate that visual instruction tuning substantially improves multimodal reasoning and conversational abilities, enabling LLaVA to exhibit behaviors comparable to proprietary multimodal systems on previously unseen images and instructions [63].

More broadly, recent multimodal architectures are evolving toward general-purpose assistants that combine perception, language understanding, reasoning, retrieval, planning, and generation within a shared representation space. This trend increasingly blurs the distinction between visual recognition systems and conversational AI, suggesting a future in which foundation models function as unified multimodal agents capable of interpreting and reasoning about the physical world through natural interaction [63].

## 8 Diffusion-Based Generative Models

Whereas CNNs and ViTs parameterize deterministic mappings  $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ , diffusion-based generative models define a probabilistic framework over the input space  $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C}$ , learning to synthesize images by progressively denoising a Gaussian random variable.

Denoising Diffusion Probabilistic Models (DDPMs) [13] achieve this by defining a latent variable model over a sequence of representations  $\{\mathbf{x}_t\}_{t=0}^T$ , where  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$  denotes a sample from the true data distribution and  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  the fully corrupted endpoint. The model operates through three coupled mechanisms formalized in the following subsections.

### 8.1 Forward Process

The forward process is defined as a fixed Markov chain that gradually corrupts a data sample  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$  by injecting Gaussian noise according to a prescribed variance schedule  $\{\beta_t\}_{t=1}^T$ , where  $\beta_t \in (0, 1)$ . At each diffusion step, the transition distribution is given by

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad t = 1, \dots, T. \quad (74)$$

This formulation progressively reduces the signal-to-noise ratio of the representation while preserving a tractable probabilistic structure. Defining

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s, \quad (75)$$

the distribution of  $\mathbf{x}_t$  conditioned directly on the original sample  $\mathbf{x}_0$  admits the closed-form expression

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}). \quad (76)$$

A key consequence of Equation (76) is that samples at arbitrary noise levels can be generated without explicitly simulating the entire diffusion trajectory. Specifically,

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (77)$$

which provides a reparameterized sampling procedure used extensively during training. The first term retains a progressively attenuated version of the original signal, whereas the second term injects Gaussian noise whose magnitude increases with  $t$ .

As the diffusion process approaches its terminal state,  $\bar{\alpha}_T \rightarrow 0$ , the influence of  $\mathbf{x}_0$  vanishes and the marginal distribution converges to

$$q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (78)$$

independently of the underlying data sample. Consequently, the forward process gradually transforms structured visual data into pure Gaussian noise, establishing the stochastic trajectory that the reverse process must subsequently learn to invert.

### 8.2 Reverse Process

The reverse process defines a learned Markov chain whose objective is to invert the progressive corruption introduced by the forward process. Starting from a Gaussian latent variable

$$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

the model gradually reconstructs a clean sample by traversing the diffusion trajectory in the reverse temporal direction. The resulting generative model factorizes as

$$p_{\theta}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t), \quad (79)$$

where

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (80)$$

serves as the tractable prior distribution from which generation begins.

Each reverse transition is modeled as a Gaussian distribution whose parameters are learned from data:

$$p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\theta}(\mathbf{x}_t, t), \Sigma_{\theta}(\mathbf{x}_t, t)), \quad (81)$$

where  $\theta \in \Theta$  denotes the learnable parameters of the denoising network,  $\boldsymbol{\mu}_{\theta}(\mathbf{x}_t, t)$  is the conditional mean, and  $\Sigma_{\theta}(\mathbf{x}_t, t)$  is the conditional covariance of the reverse transition distribution.

In practice, the model receives the noisy sample  $\mathbf{x}_t$  together with a timestep embedding encoding the current diffusion step  $t$ , allowing a single network to be shared across the entire trajectory.

Unlike the forward process, whose transitions are fixed by the variance schedule  $\{\beta_t\}_{t=1}^T$ , the reverse transitions must be learned. Consequently, the reverse process can be interpreted as estimating the conditional distribution of a less corrupted sample  $\mathbf{x}_{t-1}$  given the current noisy observation  $\mathbf{x}_t$ . By repeatedly applying the transition distribution in Equation (81), the model progressively transforms an unstructured Gaussian sample into a structured image belonging to the data distribution.

The reverse diffusion trajectory therefore performs the opposite operation of Equation (74). Whereas the forward process gradually destroys information by injecting noise, the reverse process incrementally restores information by estimating the statistics of the underlying clean signal at every timestep. The quality of the generated samples consequently depends on the accuracy with which the learned model approximates the true reverse conditionals

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t), \quad (82)$$

which are intractable in general and must therefore be learned from data.

### 8.3 Training Objective

The parameters  $\theta$  of the reverse process are learned by maximizing the likelihood of the training data. Direct optimization of the marginal log-likelihood  $\log p_{\theta}(\mathbf{x}_0)$  is generally intractable; therefore, DDPMs optimize a variational lower bound (ELBO) on the data likelihood [13]. A key result of Ho et al. [13] is that this objective can be simplified to a denoising score-matching loss, yielding a considerably more stable and efficient training procedure.

Specifically, a timestep  $t \sim \mathcal{U}(\{1, \dots, T\})$  is sampled uniformly, together with a clean image  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$  and Gaussian noise  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ .

Using the reparameterized forward process introduced in Equation (77), a noisy sample  $\mathbf{x}_t$  is generated according to

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}. \quad (83)$$

The denoising network  $\boldsymbol{\epsilon}_{\theta}$  is then trained to predict the noise realization  $\boldsymbol{\epsilon}$  from the noisy observation  $\mathbf{x}_t$  and the timestep  $t$ . The corresponding objective is

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} [\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\mathbf{x}_t, t)\|_2^2], \quad (84)$$

or, equivalently, by substituting Equation (83),

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[ \left\| \epsilon - \epsilon_\theta \left( \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t \right) \right\|_2^2 \right]. \quad (85)$$

The denoising network therefore learns to estimate the corruption process applied at each diffusion step. Since the added noise uniquely determines the corresponding clean sample through Equation (83), accurate noise prediction implicitly enables recovery of the underlying data distribution.

In practice, the noise predictor

$$\epsilon_\theta : \mathbb{R}^{H \times W \times C} \times \{1, \dots, T\} \rightarrow \mathbb{R}^{H \times W \times C}$$

is typically implemented using a U-Net backbone [64]. Modern diffusion architectures frequently augment the convolutional encoder–decoder structure with self-attention layers of the form introduced in Section 4.2, thereby combining local spatial processing through convolutional blocks  $\phi^{(\ell)}$  with global context modeling through the attention matrices  $\mathbf{A}_h$ . As a result, contemporary diffusion models naturally integrate the convolutional and Transformer paradigms discussed in the previous sections.

From the perspective of score-based generative modeling, the denoising objective in Equation (85) can be interpreted as learning the score function of the noisy data distribution at each diffusion step. Let

$$q(\mathbf{x}_t) = \int q(\mathbf{x}_t | \mathbf{x}_0) q(\mathbf{x}_0) d\mathbf{x}_0 \quad (86)$$

denote the marginal distribution of noisy samples at timestep  $t$  induced by the forward process. The associated score function is defined as

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t), \quad (87)$$

which specifies the direction of maximum increase in the log-density of the distribution  $q(\mathbf{x}_t)$  [15]. Intuitively, the score indicates how a noisy sample should be adjusted in order to move toward regions of higher probability under the data distribution.

For the Gaussian corruption process of Equation (76), it can be shown that the score is directly related to the injected noise  $\epsilon$ . Consequently, training the network  $\epsilon_\theta$  to predict the corruption noise as in Equation (85) is equivalent to learning the score function of the noisy data distribution [13, 15]. The denoising network therefore provides an estimate of the local geometry of the probability density at every noise level.

This connection establishes a bridge between diffusion models and score-based generative modeling, where the reverse process can be viewed as following the learned score field from noise toward the data manifold. Under this interpretation, diffusion models correspond to discrete approximations of continuous stochastic differential equations, a perspective that will be revisited in Section 8.4.1.

At inference time, image synthesis proceeds by sampling  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and repeatedly applying the learned reverse transitions defined in Equation (81). After traversing the entire trajectory from  $t = T$  to  $t = 0$ , the resulting sample  $\mathbf{x}_0$  approximates a draw from the underlying data distribution  $q(\mathbf{x}_0)$ . Consequently, generation may be interpreted as the iterative refinement of an initially random Gaussian sample into a structured image through successive denoising steps.

## 8.4 Extensions of the DDPM Framework

The foundational DDPM framework of Section 8.1 established the forward diffusion process  $q(\mathbf{x}_t | \mathbf{x}_{t-1})$  and the denoising objective of Equation (85) as a stable alternative to adversarial training. Despite its success, the original formulation exhibits several practical limitations, including slow sampling due to large diffusion horizons, high computational cost arising from pixel-space processing, and limited controllability during generation. The following developments progressively address these challenges.

### 8.4.1 Continuous-Time Diffusion

The discrete-time diffusion process introduced in Section 8.1 can be generalized to a continuous-time formulation through stochastic differential equations (SDEs) [15]. This perspective establishes a direct connection between diffusion models and score-based generative modeling, viewing the diffusion trajectory as a continuous stochastic process rather than a finite sequence of noise-injection steps.

Let  $\mathbf{x}(t) \in \mathbb{R}^{H \times W \times C}$  denote the data representation at continuous time  $t \in [0, T]$ . The forward diffusion process is governed by the Itô SDE

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t) dt + g(t) d\mathbf{w}, \quad (88)$$

where  $\mathbf{f} : \mathbb{R}^D \times \mathbb{R} \rightarrow \mathbb{R}^D$  is the drift function,  $g : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$  is the diffusion coefficient,  $D = HWC$ , and  $\mathbf{w}$  denotes a standard Wiener process. The drift term controls the deterministic evolution of the trajectory, whereas the diffusion term injects stochastic perturbations.

The variance-preserving diffusion process used by DDPMs is recovered as a particular instance of Equation (88) by choosing

$$\mathbf{f}(\mathbf{x}, t) = -\frac{1}{2}\beta(t)\mathbf{x}, \quad g(t) = \sqrt{\beta(t)}, \quad (89)$$

where  $\beta(t)$  is the continuous counterpart of the discrete variance schedule  $\{\beta_t\}_{t=1}^T$ . Under this parameterization, the continuous-time process gradually transforms samples from the data distribution into Gaussian noise, exactly mirroring the behavior of the discrete forward diffusion chain.

A key result established by Song et al. [15] is that the corresponding reverse-time process is itself an SDE:

$$d\mathbf{x} = \left[ \mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt + g(t) d\bar{\mathbf{w}}, \quad (90)$$

where  $p_t(\mathbf{x})$  denotes the marginal distribution of the forward process at time  $t$ ,  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  is the score function introduced in Equation (87), and  $d\bar{\mathbf{w}}$  represents a reverse-time Wiener process. The reverse SDE therefore depends on the score of the noisy data distribution at every diffusion level.

Since the score function is generally unknown, it is approximated by a neural network

$$s_{\theta}(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x}), \quad (91)$$

which is learned through the denoising objective of Equation (85). Consequently, the noise predictor  $\epsilon_{\theta}(\mathbf{x}_t, t)$  employed in DDPMs can be interpreted as an implicit estimator of the score function, thereby linking denoising diffusion models with continuous score matching frameworks.

An additional consequence of the SDE formulation is the existence of an equivalent deterministic process known as the probability-flow ordinary differential equation (ODE):

$$d\mathbf{x} = \left[ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt. \quad (92)$$

Unlike Equation (90), the probability-flow ODE contains no stochastic term and therefore defines a deterministic trajectory from

$$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

to a sample

$$\mathbf{x}_0 \sim q(\mathbf{x}_0).$$

This observation provides a unified mathematical framework encompassing both stochastic diffusion sampling and deterministic generation procedures, while offering valuable theoretical insight into the dynamics of modern diffusion models.

### 8.4.2 Latent Diffusion Models

While continuous-time formulations improve the efficiency and theoretical understanding of diffusion models, the denoising network  $\epsilon_\theta$  still operates directly in the high-dimensional image space  $\mathbb{R}^{H \times W \times C}$ . As image resolution increases, both memory consumption and computational cost scale accordingly. Latent Diffusion Models (LDMs) [18] address this limitation by decoupling perceptual compression from generative modeling through a pretrained autoencoder.

The encoder

$$\mathbf{z} = \mathcal{E}(\mathbf{x}_0) \in \mathbb{R}^{h \times w \times c}, \quad (93)$$

maps images into a compact latent representation, while the decoder

$$\hat{\mathbf{x}}_0 = \mathcal{D}(\mathbf{z}) \quad (94)$$

reconstructs the image from the latent code. Here,  $h = H/f$ ,  $w = W/f$ , for a downsampling factor  $f > 1$ , and typically  $hwc \ll HWC$ .

Rather than applying diffusion in pixel space, the forward and reverse processes are defined directly on the latent representation. The forward transition becomes

$$q(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1 - \beta_t} \mathbf{z}_{t-1}, \beta_t \mathbf{I}), \quad (95)$$

while the denoising objective is

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \mathbf{z}_0, \epsilon} [\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t)\|_2^2], \quad (96)$$

where  $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0)$ . Since diffusion operates in a substantially lower-dimensional space, training and sampling become significantly more efficient while perceptual quality is preserved through the decoder  $\mathcal{D}$ .

### 8.4.3 Conditional Generation and Classifier-Free Guidance

Although latent diffusion significantly improves computational efficiency, the generative process remains unconditional. Consequently, the model provides no direct mechanism for controlling the content of the generated sample. Conditional diffusion models address this limitation by introducing an auxiliary conditioning variable  $\mathbf{c} \in C$ , which may encode class labels, textual descriptions, segmentation masks, or other forms of side information.

The reverse process is modified as

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{c}), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathbf{c})), \quad (97)$$

and the training objective becomes

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \epsilon, \mathbf{c}} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})\|_2^2]. \quad (98)$$

A particularly effective conditioning strategy is *classifier-free guidance* [65], which avoids the need for an external classifier by jointly training conditional and unconditional denoisers. During sampling, the two predictions are combined through

$$\tilde{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}) = \epsilon_\theta(\mathbf{x}_t, t, \emptyset) + w(\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) - \epsilon_\theta(\mathbf{x}_t, t, \emptyset)), \quad (99)$$

where  $w \geq 1$  is the guidance scale and  $\emptyset$  denotes null conditioning.

Increasing  $w$  strengthens adherence to the conditioning signal at the expense of sample diversity, thereby providing an explicit mechanism for trading off fidelity and variability. This technique forms the basis of modern text-to-image systems and enables applications including image editing, inpainting, super-resolution, and segmentation-guided generation [66, 26].

#### 8.4.4 Transformer-Based Diffusion Architectures

The convolutional U-Net backbone  $\phi_{\text{dec}}^{(\ell)}$  of the original DDPM inherits the  $\mathcal{O}(\ell \cdot k_\ell)$  receptive-field growth characteristic of CNNs. Diffusion Transformers (DiT) [24] replace this backbone with a Transformer architecture, directly connecting diffusion modeling with the Vision Transformer formulation introduced in Sections 4.1–4.3. Rather than operating in pixel space, DiT follows the Latent Diffusion Model (LDM) framework, where diffusion is performed on the latent representation  $\mathbf{z} = E(\mathbf{x})$  produced by a pretrained variational autoencoder (VAE) [24]. This substantially reduces computational cost while preserving image quality.

The latent representation  $\mathbf{z}_t \in \mathbb{R}^{h \times w \times c}$  at diffusion timestep  $t$  is partitioned into non-overlapping patches of size  $P \times P$ , yielding  $N = hw/P^2$  latent tokens. Following the ViT pipeline, each patch is linearly embedded and augmented with positional embeddings to form the token matrix

$$\mathbf{Z}_t = [\mathbf{z}_{t,0}^\top; \mathbf{z}_{t,1}^\top; \dots; \mathbf{z}_{t,N}^\top] \in \mathbb{R}^{(N+1) \times d}, \quad (100)$$

which is processed by a stack of Transformer blocks. As in ViTs, the patch size  $P$  directly controls the sequence length and therefore the computational cost of self-attention: reducing  $P$  increases the number of latent tokens and consequently the model GFLOPs [24].

Among several conditioning strategies explored by DiT, including cross-attention and in-context conditioning, adaptive Layer Normalization (adaLN) was found to provide the best trade-off between simplicity and performance [24]. Instead of appending conditioning tokens or introducing dedicated attention layers, the timestep embedding and optional conditioning signal are used to modulate the Transformer activations through learned scale and shift parameters:

$$\text{adaLN}(\mathbf{Z}, t, \mathbf{c}) = \boldsymbol{\gamma}(t, \mathbf{c}) \odot \text{LN}(\mathbf{Z}) + \boldsymbol{\beta}(t, \mathbf{c}), \quad (101)$$

where  $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$  are predicted from the conditioning embeddings and applied uniformly across all tokens. Unlike cross-attention, conditioning is injected directly into the normalization layers, preserving the standard Transformer structure while introducing only a small computational overhead [24].

The final DiT architecture adopts the adaLN-Zero variant, in which the conditioning embedding generated from the diffusion timestep  $t$  and optional condition  $\mathbf{c}$  predicts six modulation parameters

$$(\boldsymbol{\beta}_1, \boldsymbol{\gamma}_1, \alpha_1, \boldsymbol{\beta}_2, \boldsymbol{\gamma}_2, \alpha_2) = g(t, \mathbf{c}), \quad (102)$$

where  $\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i \in \mathbb{R}^d$  are shift and scale vectors applied by adaLN, and  $\alpha_i \in \mathbb{R}$  are residual gating coefficients. The pairs  $(\boldsymbol{\beta}_1, \boldsymbol{\gamma}_1)$  and  $(\boldsymbol{\beta}_2, \boldsymbol{\gamma}_2)$  modulate the MHSA and FFN branches, respectively, while the residual gates control their contribution to the block output. The pairs  $(\boldsymbol{\gamma}_1, \boldsymbol{\beta}_1)$  and  $(\boldsymbol{\gamma}_2, \boldsymbol{\beta}_2)$  instantiate Equation (101) for the MHSA and FFN branches, respectively.

A key finding of [24] is that Transformer-based diffusion models exhibit remarkably predictable scaling behaviour: increasing model depth, hidden dimension, or token count consistently improves sample quality when measured by the Fréchet Inception Distance (FID). Unlike earlier diffusion architectures, where scaling laws were less evident, DiT demonstrates a strong correlation between computational complexity (measured in GFLOPs) and generative performance. The largest model, DiT-XL/2, achieved a state-of-the-art ImageNet 256×256 FID of 2.27 at the time of publication, establishing Transformer backbones as a viable alternative to convolutional U-Nets for diffusion modeling [24]. This paradigm has since become the foundation of numerous large-scale image and video generation systems, including CogVideoX [67] and DiMBa [68].

#### 8.4.5 Acceleration and Sampling Efficiency

A major limitation of DDPMs is that sampling requires simulating the full reverse Markov chain, typically involving hundreds or thousands of sequential evaluations of the denoising network  $\epsilon_\theta$ . DDIM

(Denoising Diffusion Implicit Models) [14] addresses this bottleneck by generalizing the DDPM formulation from a Markovian forward process to a family of non-Markovian processes that preserve the same training objective. Consequently, a model trained with the standard DDPM loss can be sampled using alternative reverse trajectories without retraining [14].

Given the predicted clean sample

$$\widehat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}, \quad (103)$$

DDIM defines the reverse update

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \widehat{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (104)$$

where  $\sigma_t \geq 0$  controls the amount of stochasticity in the reverse process [14]. The DDPM sampler is recovered for a particular choice of  $\sigma_t$ , whereas the limiting case  $\sigma_t = 0$  yields a fully deterministic trajectory known as a Denoising Diffusion Implicit Model (DDIM). In this regime, each latent state  $\mathbf{x}_{t-1}$  becomes a deterministic function of  $\mathbf{x}_t$ , enabling significantly faster sampling while retaining the same denoising network and training objective [14].

An additional advantage is that DDIM permits sampling along a reduced trajectory  $\tau = \{\tau_1, \dots, \tau_S\}$  with  $S \ll T$ , effectively skipping intermediate diffusion steps. Since the training objective depends only on the marginals  $q(\mathbf{x}_t | \mathbf{x}_0)$  rather than on the specific forward trajectory, generation can be performed using only a subset of timesteps, often reducing the number of denoising evaluations from hundreds or thousands to a few tens without retraining the model [14]. The original paper reports speedups of approximately one to two orders of magnitude while maintaining competitive sample quality [14].

Furthermore, DDIM admits an interpretation as the discretization of an ordinary differential equation (ODE), establishing a connection between diffusion sampling and numerical integration methods [14]. This perspective motivated a large family of subsequent samplers based on ODE and SDE solvers, including higher-order integration schemes that further reduce the number of function evaluations required during inference.

The advances reviewed above address the principal limitations of Section 8.1 systematically: continuous-time formulations generalize the discrete diffusion process; latent diffusion models reduce the operating space from  $\mathbb{R}^{H \times W \times C}$  to  $\mathbb{R}^{h \times w \times c}$ ; conditional generation extends  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$  to include conditioning variables  $\mathbf{c}$ ; Diffusion Transformers replace the convolutional backbone  $\phi_{\text{dec}}^{(\ell)}$  with a Transformer-based denoising model  $\boldsymbol{\epsilon}_\theta(\mathbf{Z}_t, t, \mathbf{c})$ ; and DDIM enables efficient inference through deterministic and subsampled reverse trajectories. Together, these developments establish diffusion models as a versatile framework for image generation, video synthesis, multimodal learning, and scientific imaging.

## 8.5 Rectified Flow Models

Recent research has questioned whether high-quality generative modeling necessarily requires the stochastic forward and reverse processes that characterize diffusion models. This observation has led to the development of *Rectified Flow* models [69], which reformulate generation as a deterministic transport problem between probability distributions. Rather than learning to reverse a stochastic corruption process, Rectified Flow directly learns an ordinary differential equation (ODE) that continuously transports samples from a simple source distribution to the target data distribution.

Let

$$\mathbf{x}_0 \sim \pi_0, \quad \mathbf{x}_1 \sim \pi_1, \quad (105)$$

where  $\pi_0$  denotes a simple source distribution, typically a standard Gaussian, and  $\pi_1 = q(\mathbf{x})$  denotes the target data distribution. Rectified Flow begins by defining the linear interpolation

$$\mathbf{x}_t = t\mathbf{x}_1 + (1 - t)\mathbf{x}_0, \quad t \in [0, 1], \quad (106)$$

which follows the shortest Euclidean path connecting the two endpoints. Differentiating with respect to time yields

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{x}_1 - \mathbf{x}_0, \quad (107)$$

showing that each interpolation trajectory moves along a straight line with constant velocity.

Although Equation (107) defines a valid trajectory for a given pair  $(\mathbf{x}_0, \mathbf{x}_1)$ , it does not provide a usable generative dynamical system. The velocity  $\mathbf{x}_1 - \mathbf{x}_0$  depends explicitly on the terminal point  $\mathbf{x}_1$ , which is unknown during generation. Consequently, the dynamics cannot be evaluated from the current state  $\mathbf{x}_t$  and time  $t$  alone, making the interpolation process non-causal. In other words, simulating the trajectory requires access to future information.

Rectified Flow addresses this limitation by learning a velocity field

$$\mathbf{v}_\theta : \mathbb{R}^{H \times W \times C} \times [0, 1] \rightarrow \mathbb{R}^{H \times W \times C}, \quad (108)$$

that approximates the transport direction  $\mathbf{x}_1 - \mathbf{x}_0$  using only the current state and time. The velocity field is learned through the least-squares objective

$$\mathcal{L}(\theta) = \int_0^1 \mathbb{E}[\|\mathbf{x}_1 - \mathbf{x}_0 - \mathbf{v}_\theta(\mathbf{x}_t, t)\|_2^2] dt, \quad (109)$$

where  $\mathbf{x}_t$  follows the interpolation process of Equation (106). Unlike DDPMs, which train a network to predict Gaussian noise, Rectified Flow directly learns the transport direction that moves samples from the source distribution toward the target distribution. Consequently, the resulting dynamics define a causal ordinary differential equation that can be simulated progressively without requiring knowledge of the terminal point  $\mathbf{x}_1$ .

The optimal velocity field associated with Equation (109) is

$$\mathbf{v}^*(\mathbf{x}, t) = \mathbb{E}[\mathbf{x}_1 - \mathbf{x}_0 \mid \mathbf{x}_t = \mathbf{x}], \quad (110)$$

which corresponds to the conditional expectation of the transport direction among all interpolation trajectories passing through the point  $\mathbf{x}$  at time  $t$ . The resulting generative model is then defined by the ordinary differential equation

$$\frac{d\mathbf{z}_t}{dt} = \mathbf{v}_\theta(\mathbf{z}_t, t), \quad \mathbf{z}_0 \sim \pi_0. \quad (111)$$

A fundamental property of the learned flow follows from the uniqueness of ODE solutions. If Equation (111) admits a unique solution, two trajectories cannot intersect while moving in different directions; otherwise multiple solutions would pass through the same point, violating uniqueness. As a consequence, the learned velocity field organizes transport into a collection of deterministic, non-crossing trajectories.

This observation highlights an important difference between the original linear interpolation process and the learned flow. Different pairs  $(\mathbf{x}_0, \mathbf{x}_1)$  may generate interpolation trajectories that pass through the same intermediate location while exhibiting different transport directions. Such ambiguities are incompatible with a well-defined ODE. Rectified Flow resolves this issue by learning a consistent velocity field that effectively reorganizes the transport process into non-crossing trajectories while preserving the desired distributional evolution [69].

A further consequence of this rectification process is the progressive straightening of transport paths. As trajectories become smoother and closer to straight lines, numerical integration can be performed with coarser discretizations and fewer solver steps. In the limiting case of perfectly straight trajectories, the entire transport path can be approximated with a single Euler update. Rectified Flow therefore

departs from classical diffusion modeling by replacing stochastic differential equations with a deterministic transport ODE, establishing a connection between generative modeling, neural ODEs, and optimal transport while enabling substantially faster sampling [69].

## 9 Future Directions and Open Challenges

The developments reviewed throughout this survey reveal an increasing convergence among architectures that were originally proposed for distinct objectives, including convolutional neural networks, Vision Transformers, self-supervised foundation models, multimodal systems, and generative frameworks. Rather than designing separate models for perception, retrieval, reasoning, interaction, and generation, current research increasingly seeks unified systems capable of learning transferable representations across tasks, domains, and modalities. This trend is motivating a broader shift toward general-purpose visual intelligence, where representation learning, multimodal alignment, and generative modeling become integrated within common computational frameworks. The following subsections examine this evolving perspective, its implications for future vision systems, and the challenges that must be addressed to enable their efficient, interpretable, and trustworthy deployment.

### 9.1 A Unified View of Modern Computer Vision

At the highest level, the diverse paradigms reviewed throughout this survey can be interpreted as mechanisms for constructing, transforming, aligning, or generating latent representations.

Given an input  $\mathbf{x} \in \mathcal{X}$ , a model learns a latent representation

$$\mathbf{z} = h_\theta(\mathbf{x}) \in \mathcal{Z}, \quad (112)$$

where  $\mathcal{Z}$  denotes a latent representation space. Depending on the task, this representation is subsequently mapped to an output

$$\mathbf{y} = g_\theta(\mathbf{z}) \in \mathcal{Y}, \quad (113)$$

such that

$$f_\theta = g_\theta \circ h_\theta. \quad (114)$$

Consequently, many contemporary computer vision methods can be viewed as learning either the mapping  $h_\theta : \mathcal{X} \rightarrow \mathcal{Z}$ , the mapping  $g_\theta : \mathcal{Z} \rightarrow \mathcal{Y}$ , or the structure of the latent space itself. Within this perspective, the principal learning paradigms differ primarily in the mechanisms used to construct and manipulate latent representations. Convolutional neural networks generate hierarchical feature maps through local operators,

$$\mathbf{h}^{(\ell)} = \phi^{(\ell)}(\mathbf{h}^{(\ell-1)}), \quad (115)$$

where locality, weight sharing, and translation equivariance act as inductive biases that guide representation learning.

Vision Transformers adopt a different strategy by representing an image as a sequence of tokens  $\mathbf{Z} \in \mathbb{R}^{(N+1) \times d}$ , and refining these representations through self-attention and feed-forward transformations,

$$\mathbf{Z}^{(\ell)} = \text{Block}^{(\ell)}(\mathbf{Z}^{(\ell-1)}), \quad (116)$$

where latent representations emerge from global interactions between tokens rather than local convolutional processing.

Self-supervised foundation models further emphasize the latent space itself as the primary learning objective. Instead of optimizing for task-specific outputs, methods such as DINO learn representations that remain stable across multiple views of the same visual scene,

$$\mathbf{z}_s \approx \mathbf{z}_t, \quad (117)$$

where  $\mathbf{z}_s$  and  $\mathbf{z}_t$  denote student and teacher representations associated with different image transformations.

Multimodal foundation models extend this principle beyond the visual domain by aligning representations originating from different modalities. For example, CLIP, ImageBind, and related approaches learn embeddings  $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$ , such that semantically related observations occupy nearby regions of a shared latent space,

$$\mathbf{z}_{\text{image}} \approx \mathbf{z}_{\text{text}}, \quad \mathbf{z}_{\text{image}} \approx \mathbf{z}_{\text{audio}}, \quad (118)$$

thereby enabling cross-modal retrieval, reasoning, and transfer learning.

Generative models can likewise be interpreted within the same representation-learning framework. Rather than learning a direct mapping from inputs to outputs, diffusion models learn transformations between probability distributions over images or latent representations,

$$p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \quad (119)$$

or, in latent diffusion frameworks,

$$p_{\theta}(\mathbf{z}_{t-1} \mid \mathbf{z}_t). \quad (120)$$

From this perspective, diffusion models can be viewed as learning dynamics within a representation space rather than merely performing image generation.

These observations suggest that modern computer vision is increasingly less defined by individual architectures and more by the representations they learn. CNNs, Vision Transformers, self-supervised foundation models, multimodal systems, and generative frameworks differ in their inductive biases, training objectives, and computational mechanisms, yet all operate on the common principle of constructing, transforming, aligning, or generating latent representations.

This convergence provides the conceptual foundation for the emergence of increasingly general-purpose visual intelligence systems, discussed in the following subsection.

## 9.2 Toward General-Purpose Visual Intelligence

One manifestation of this trend toward increasingly general-purpose visual intelligence is the development of architectures that minimize modality-specific assumptions while retaining the ability to process high-dimensional sensory information efficiently. The Perceiver architecture [70] was introduced as an attempt to move beyond the strong modality-specific assumptions that characterize many contemporary deep learning models.

The Perceiver adopts a Transformer-based formulation that makes minimal assumptions about the organization of the input, enabling a single architecture to process images, audio, video, point clouds, and multimodal combinations within a unified framework.

The central idea is to replace the standard Transformer’s all-to-all attention with a compact latent bottleneck. Let

$$\mathbf{X} \in \mathbb{R}^{N_{\text{data}} \times d_{\text{in}}}$$

denote the array-based representation of an input sample  $\mathbf{x} \in \mathcal{X}$ , where  $N_{\text{data}}$  is the number of input elements and  $d_{\text{in}}$  is the input feature dimension. The model learns a compact latent array

$$\mathbf{Z} \in \mathbb{R}^{N_{\text{latent}} \times d_{\text{lat}}}, \quad N_{\text{latent}} \ll N_{\text{data}},$$

where  $N_{\text{latent}}$  denotes the number of latent vectors and  $d_{\text{lat}}$  their embedding dimension.

Rather than applying self-attention directly to the entire input, the latent array queries the input through an asymmetric cross-attention mechanism,

$$\mathbf{Z}^{(\ell+1)} = \mathbf{Z}^{(\ell)} + \text{CrossAttn}(\mathbf{Q} = \mathbf{Z}^{(\ell)} \mathbf{W}^Q, \mathbf{K} = \mathbf{X} \mathbf{W}^K, \mathbf{V} = \mathbf{X} \mathbf{W}^V), \quad (121)$$

where the latent array provides the queries while the input supplies the keys and values. Since the number of latent vectors is substantially smaller than the number of input elements, the complexity of the cross-attention operation scales as

$$O(N_{\text{data}}N_{\text{latent}}),$$

rather than the quadratic cost

$$O(N_{\text{data}}^2)$$

associated with conventional self-attention. The latent array therefore acts as an information bottleneck through which all input data must pass.

A major consequence of this bottleneck is that the computational cost of the subsequent Transformer layers becomes largely independent of the input size. Once information has been distilled into the latent representation, all deeper processing is performed through self-attention operating exclusively on the latent array. If  $L$  denotes the number of latent Transformer blocks, the overall complexity becomes

$$O(N_{\text{data}}N_{\text{latent}} + LN_{\text{latent}}^2), \quad (122)$$

which effectively decouples network depth from input dimensionality. This design allows the construction of very deep Transformer stacks while maintaining tractable computational costs even for inputs containing tens or hundreds of thousands of elements.

The architecture derives its name from the notion of *iterative attention*. Rather than attending to the input only once, the latent representation repeatedly interacts with the original input by alternating cross-attention modules and latent self-attention blocks. This iterative refinement enables information extracted during earlier stages to influence subsequent queries to the input, thereby allowing the model to focus its limited latent capacity on the most relevant information. When parameters are shared across iterations, the architecture can be interpreted as a recurrent neural network whose recurrent state is represented by the latent array and whose recurrence unfolds in depth rather than in time.

Because attention mechanisms are inherently permutation invariant, the model must explicitly encode structural information. The Perceiver does so through Fourier-feature positional encodings that are concatenated to the input representation. For an input coordinate  $x_d$  along dimension  $d$ , the encoding takes the form

$$\text{pos}(x_d) = [\sin(f_k \pi x_d), \cos(f_k \pi x_d)], \quad (123)$$

where  $\{f_k\}$  denotes a bank of frequencies spanning multiple spatial or temporal scales. By representing positional information through features rather than architectural constraints, the same Perceiver model can process one-dimensional audio signals, two-dimensional images, three-dimensional video streams, and multimodal inputs without modification.

By combining a latent attention bottleneck, iterative cross-attention, and modality-agnostic positional encodings, the Perceiver demonstrates that a single Transformer architecture can achieve competitive performance across a diverse range of perceptual domains while remaining largely independent of modality-specific design choices.

### 9.3 Efficiency, Interpretability, and Trustworthy Deployment

As computer vision systems continue to grow in scale and capability, their practical deployment increasingly depends on achieving an effective balance between predictive performance, computational efficiency, interpretability, robustness, and reliability. While modern foundation models have demonstrated unprecedented generalization capabilities, their growing complexity introduces significant challenges that extend beyond the traditional pursuit of accuracy.

A primary concern is computational efficiency. Vision Transformers replace the localized processing of convolutional networks with self-attention mechanisms whose computational complexity scales

quadratically with the number of input tokens. Similarly, diffusion models often require repeated evaluations of a denoising network during generation, leading to substantial inference costs. Although architectures such as Swin Transformer, Perceiver, latent diffusion models, Diffusion Transformers, and Rectified Flow frameworks have significantly improved computational efficiency, deploying large-scale models on resource-constrained devices remains a major challenge. Consequently, future research will likely continue to explore sparse and linear attention mechanisms, dynamic token selection, model compression, quantization, knowledge distillation, and hardware-aware neural architecture design.

Beyond computational considerations, interpretability remains a fundamental open problem. Modern visual foundation models often contain billions of parameters and learn highly distributed internal representations whose decision-making processes remain difficult to understand. Although attention maps are frequently visualized as explanations of Transformer predictions, it remains unclear to what extent attention weights correspond to causal mechanisms rather than correlations within the learned representation space. Likewise, self-supervised foundation models such as DINOv3 exhibit emergent semantic organization within patch embeddings, yet the precise mechanisms underlying these representations remain only partially understood.

The challenge becomes even more pronounced in generative models. Diffusion-based architectures learn complex distributions over high-dimensional visual spaces, making it difficult to establish direct relationships between model parameters, latent representations, and generated outputs. Understanding how conditioning signals influence the generation trajectory and how specific visual concepts emerge during sampling remains an active area of research. Progress in mechanistic interpretability, causally grounded explanations, and representation analysis may therefore become increasingly important as generative systems continue to scale.

Another critical concern is robustness and reliability. Foundation models are typically trained on massive and heterogeneous datasets whose distribution may differ significantly from real-world deployment conditions. As a result, performance degradation can occur under domain shifts, adversarial perturbations, rare events, or previously unseen visual concepts. While large-scale pretraining generally improves generalization, ensuring reliable behavior in safety-critical environments remains an open problem.

These challenges are particularly important in domains such as medical imaging, autonomous systems, industrial inspection, remote sensing, and scientific analysis. In such settings, inaccurate predictions or poorly understood failures may lead to substantial practical consequences. Future vision systems must therefore provide not only accurate predictions but also calibrated uncertainty estimates, robustness guarantees, and transparent explanations of their decisions. Methods for uncertainty quantification, confidence calibration, out-of-distribution detection, and explainable artificial intelligence are likely to play a central role in the trustworthy deployment of future models.

In parallel, the emergence of multimodal foundation models introduces new challenges related to alignment and controllability. Systems that combine perception, language understanding, reasoning, retrieval, and generation must ensure consistent behavior across modalities while remaining resistant to hallucinations, misleading inputs, and instruction manipulation. Developing principled mechanisms for multimodal alignment, safety monitoring, and human oversight therefore constitutes another important direction for future research.

Ultimately, the long-term success of general-purpose visual intelligence will depend not only on improvements in predictive performance but also on advances in efficiency, interpretability, robustness, and trustworthy operation. The next generation of computer vision systems will likely integrate scalable representation learning, multimodal reasoning, and generative capabilities within architectures that remain transparent, reliable, and deployable in real-world environments.

## 10 Conclusion

This survey has presented a unified perspective on the principal architectural and learning paradigms that define modern computer vision, including convolutional neural networks, Vision Transformers, self-supervised foundation models, multimodal architectures, and generative frameworks. Although these paradigms emerged from distinct research directions, they can be interpreted within a common representation-learning framework in which an input  $\mathbf{x} \in \mathcal{X}$  is transformed into a latent representation  $\mathbf{z} = h_\theta(\mathbf{x}) \in \mathcal{Z}$ , which is subsequently mapped to a task-dependent output  $\mathbf{y} = g_\theta(\mathbf{z}) \in \mathcal{Y}$ .

From this perspective, modern computer vision can be viewed as the construction, transformation, alignment, and generation of latent representations. This unified viewpoint reveals a remarkable convergence of concepts that were historically developed in isolation, including convolutional processing, self-attention, self-supervision, multimodal alignment, and generative modeling.

Convolutional neural networks established the foundations of modern visual representation learning through local receptive fields, hierarchical feature extraction, and translation-equivariant weight sharing. Vision Transformers subsequently demonstrated that global relationships can be modeled directly through self-attention, enabling scalable representation learning with increasingly general visual representations.

More recently, self-supervised approaches such as DINO, DINOv2, and DINOv3 have demonstrated that large-scale pre-training can produce highly transferable visual representations without task-specific supervision. In parallel, multimodal foundation models have extended these ideas beyond the visual domain. CLIP introduced shared image–text embedding spaces through contrastive alignment, ImageBind generalized this paradigm to multiple sensory modalities within a unified latent space, and LLaVA demonstrated how visual representations can be combined with large language models to support instruction following and multimodal reasoning.

A central theme emerging throughout this survey is the progressive unification of representation learning paradigms. Hybrid architectures such as CvT, CoAtNet, and MaxViT demonstrate that convolution and self-attention should be viewed as complementary mechanisms operating at different points of the locality–globality spectrum. Similarly, foundation models increasingly combine ideas originating from discriminative learning, self-supervision, multimodal alignment, and generative modeling. Diffusion models, Diffusion Transformers, and Rectified Flow frameworks further extend this trend by integrating representation learning with high-fidelity generation within shared latent spaces.

This convergence is particularly evident in the emergence of foundation models and multimodal systems. Modern visual encoders are no longer optimized solely for individual tasks but instead aim to learn transferable representations that support recognition, generation, retrieval, segmentation, reasoning, and interaction within a common latent space. These developments suggest a broader transition from task-specific models toward increasingly general-purpose visual and multimodal systems.

Despite this progress, several fundamental challenges remain unresolved. First, the construction of scalable, diverse, and high-quality training data continues to be a critical factor governing the success of foundation models. Second, computational efficiency remains a major obstacle, particularly for attention-based architectures and iterative generative models. Although hierarchical attention mechanisms, latent representations, and accelerated generation techniques have substantially reduced computational costs, efficient deployment on resource-constrained hardware remains an open problem. Third, interpretability, robustness, uncertainty quantification, and trustworthy deployment continue to limit the practical adoption of large-scale foundation and generative models. Finally, the integration of perception, reasoning, memory, and interaction into a unified multimodal framework remains an active area of research.

Ultimately, the evolution documented in this survey reflects a broader shift in modern computer vision: from specialized architectures optimized for individual tasks toward increasingly general representation-learning systems capable of integrating perception, generation, and reasoning across multiple modalities. Rather than converging toward a single dominant architecture, the field appears to be moving toward scal-

able foundation models that combine local and global processing, self-supervised learning, multimodal alignment, and generative capabilities within a unified framework for visual intelligence.

## **Credit authorship contribution statement**

A.M. and F.D. conceived the survey, developed its methodological and theoretical framework, conducted the literature review and synthesis, and prepared the original manuscript draft.

S.M. and J.C. contributed to the methodological development, validation of the technical content, and critical revision of the manuscript.

All authors reviewed, edited, and approved the final version of the manuscript and agree to be accountable for all aspects of the work.

## **Conflict of Interest Statement**

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## **Declaration of generative AI use**

During the preparation of this work the authors used Microsoft Copilot in order to assist with minor language polishing and reference formatting. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

## **Ethics Statement**

The authors declare that all research was conducted in accordance with ethical guidelines.

## **Funding**

This work was supported by the project 2025-I-IDI-00030, funded by the Agencia de Desarrollo Económico de La Rioja. The grant was co-funded at 40% by the European Regional Development Fund (ERDF) for the co-financed project NeuronAI-Connect. This project contributes to Specific Objective RS01.1, aimed at strengthening research and innovation capacities and the uptake of advanced technologies within the ERDF Programme of La Rioja 2021–2027.

## **References**

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.
- [3] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022.

- [4] Xia Zhao, Limin Wang, Yufei Zhang, Xuming Han, Muhammet Deveci, and M. Parmar. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57, 2024.
- 1680 [5] Abolfazl Younesi, Mohsen Ansari, Mohammadamin Fazli, A. Ejlali, Muhammad Shafique, and Joerg Henkel. A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends. *IEEE Access*, 12:41180–41218, 2024.
- 1685 [6] Yunusa Haruna, Shiyin Qin, Abdulrahman Hamman Adama Chukkol, Abdulganiyu Abdu Yusuf, Isah Bello, and Adamu Lawan. Exploring the synergies of hybrid convolutional neural network and vision transformer architectures for computer vision: A survey. *Engineering Applications of Artificial Intelligence*, 144:110057, 2025.
- [7] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- 1690 [8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. ICLR 2021, Oral.
- 1695 [9] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*, 35:7478–7498, 2021.
- [10] Salman Hameed Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, F. Khan, and M. Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 54:1 – 41, 2021.
- 1700 [11] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. Coatnet: marrying convolution and attention for all data sizes. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS ’21, Red Hook, NY, USA, 2021. Curran Associates Inc.
- [12] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12165–12175, 2022.
- 1705 [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.
- [14] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.
- 1710 [15] Yang Song, Jascha Sohl-Dickstein, Diederik Kingma, and et al. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021.
- [16] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion Models in Vision: A Survey . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(09):10850–10869, September 2023.
- 1715 [17] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4), November 2023.

- [18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, Los Alamitos, CA, USA, June 2022. IEEE Computer Society.
- [19] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Lit, Jay Whang, Emily Denton, Seyed Kam-  
yar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Sali-  
mans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion  
models with deep language understanding. In *Proceedings of the 36th International Conference on  
Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA, 2022. Curran Associates  
Inc.
- [20] Vikas Hassija, Balamurugan Palanisamy, Arpita Chatterjee, Arpita Mandal, Debanshi Chakraborty,  
Amit Pandey, G. Chalapathi, and Dhruv Kumar. Transformers for vision: A survey on innovative  
methods for computer vision. *IEEE Access*, 13:95496–95523, 2025.
- [21] Amirhossein Kazerooni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz,  
Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive  
survey. *Medical Image Analysis*, 88:102846, 2023.
- [22] Hugo Touvron, Matthieu Cord, Matthijs Douze, and et al. Training data-efficient image transform-  
ers & distillation through attention. In *ICML*, 2021.
- [23] Ze Liu, Yutong Lin, Yue Cao, and et al. Swin transformer: Hierarchical vision transformer using  
shifted windows. In *ICCV*, 2021.
- [24] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF  
International Conference on Computer Vision (ICCV)*, pages 4172–4182. IEEE, October 2023.
- [25] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF  
International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023.
- [26] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image  
diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages  
3813–3824, 2023.
- [27] Junde Wu, Wei Ji, Huazhu Fu, Min Xu, Yueming Jin, and Yanwu Xu. Medsegdiff-v2: diffusion-  
based medical image segmentation with transformer. In *Proceedings of the Thirty-Eighth AAAI  
Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Ar-  
tificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence,  
AAAI'24/IAAI'24/EAAI'24*. AAAI Press, 2024.
- [28] G Jignesh Chowdary and Zhaozheng Yin. Diffusion transformer u-net for medical image segmenta-  
tion. In *International conference on medical image computing and computer-assisted intervention*,  
pages 622–631. Springer, 2023.
- [29] Hantao Zhang, Yuhe Liu, Jiancheng Yang, Weidong Guo, Xinyuan Wang, and Pascal Fua. DiffAt-  
las: GenAI-fying Atlas Segmentation via Image-Mask Diffusion . In *proceedings of Medical Image  
Computing and Computer Assisted Intervention – MICCAI 2025*, volume LNCS 15975. Springer  
Nature Switzerland, September 2025.
- [30] Asifullah Khan, Zunaira Rauf, Abdul Rehman Khan, Saima Rathore, Saddam Hussain Khan, Naja-  
mus Saher Shah, Umair Farooq, Hifsa Asif, Aqsa Asif, Umme Zahoora, Rafi Ullah Khalil, Suleman

- 1760 Qamar, Umme Hani Tayyab, Faiza Babar Khan, Abdul Majid, and Jeonghwan Gwak. A recent survey of vision transformers for medical image segmentation. *IEEE Access*, 13:191824–191849, 2025.
- [31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- 1765 [32] Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR.
- 1770 [33] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.
- [34] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814. Omnipress, 2010.
- 1775 [35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016.
- [36] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- 1780 [37] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local means neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803. IEEE, 2018.
- 1785 [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [39] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.
- 1790 [40] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR, 13–18 Jul 2020.
- 1795 [41] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *The Eleventh International Conference on Learning Representations*, 2023.
- [42] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020.

- [43] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6575–6586, 2023.
- [44] Ali Hatamizadeh, Hongxu Yin, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org, 2023.
- [45] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *Advances in neural information processing systems*, volume 34, pages 24261–24272, 2021.
- [46] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1328–1334, 2023.
- [47] Haiping Wu, Bin Xiao, Noel Codella, and et al. Cvt: Introducing convolutions to vision transformers. In *ICCV*, 2021.
- [48] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 459–479, Cham, 2022. Springer Nature Switzerland.
- [49] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv*, 2021.
- [50] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet’s clothing for faster inference. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12239–12249, 2021.
- [51] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022.
- [52] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.
- [53] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J., Demszky, D., ... Liang, P. (2021). *On the Opportunities and Risks of Foundation Models*. Stanford Center for Research on Foundation Models (CRFM). arXiv:2108.07258.
- [54] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [55] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Peng Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

- [56] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021.
- 1845 [57] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *International Conference on Learning Representations (ICLR)*, 2022.
- [58] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2024.
- 1850 [59] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.
- [60] Claudia Cuttano, Gabriele Trivigno, Christoph Reich, Daniel Cremers, Carlo Masone, and Stefan Roth. Insid3: Training-free in-context segmentation with dinov3. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.
- 1855 [61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 2021. arXiv:2103.00020.
- 1860 [62] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2023.
- 1865 [63] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023.
- [64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.
- 1870 [65] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [66] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings, SIGGRAPH ’22*, New York, NY, USA, 2022. Association for Computing Machinery.
- 1875 [67] Yang, Zhuoyi and Teng, Jiayan and Zheng, Wendi and Ding, Ming and Huang, Shiyu and Xu, Jiazheng and Yang, Yuanming and Hong, Wenyi and Zhang, Xiaohan and Feng, Guanyu and Yin, Da and Zhang, Yuxuan and Wang, Weihang and Cheng, Yean and Bin, Xu and Gu, Xiaotao and Dong, Yuxiao and Tang, Jie. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, pp. 83048–83077, 2025.
- 1880

- [68] Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, Youqiang Zhang, and Junshi Huang. Dimba: Transformer-mamba diffusion models. *arXiv preprint arXiv:2406.01159*, 2024. arXiv:2406.01159.
- 1885 [69] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *International Conference on Learning Representations (ICLR)*, 2023.
- 1890 [70] Andrew Jaegle, Felix Gimeno, Andrew Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.