GigaMoE: Sparsity-Guided Mixture of Experts for Efficient Gigapixel Object Detection

Xiang Li[1,2*], **Wenxi Li**[3,6*], **Yuetong Wang**[4], **Chenyang Lyu**[7], **Haozhe Lin**[1], **Guiguang Ding**[1,5†], **Yuchen Guo**[1†]

[1]Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China
[2]Department of Automation, Tsinghua University, Beijing, China
[3]KLATASDS-MOE, School of Statistics, East China Normal University, Shanghai, China
[4]Xiuzhong College,Tsinghua University, Beijing, China
[5]School of Software, Tsinghua University, Beijing, China
[6]Zhuoxi Lab, Hangzhou, China
[7]AI Business, Alibaba International Digital Commerce, Hangzhou, China
{xiang-li22, wangyuet23}@mails.tsinghua.edu.cn, {linhz, dinggg}@tsinghua.edu.cn,
wxli@sfs.ecnu.edu.cn, lyuchenyang.dcu@gmail.com, yuchen.w.guo@gmail.com

Abstract

Object detection in High-Resolution Wide (HRW) shots, or gigapixel images, presents unique challenges due to extreme object sparsity and vast scale variations. State-of-the-art methods like SparseFormer have pioneered sparse processing by selectively focusing on important regions, yet they apply a uniform computational model to all selected regions, overlooking their intrinsic complexity differences. This leads to a suboptimal trade-off between performance and efficiency. In this paper, we introduce GigaMoE, a novel backbone architecture that pioneers adaptive computation for this domain by replacing the standard Feed-Forward Networks (FFNs) with a Mixture-of-Experts (MoE) module. Our architecture first employs a shared expert to provide a robust feature baseline for all selected regions. Upon this foundation, our core innovation—a novel Sparsity-Guided Routing mechanism—insightfully repurposes importance scores from the sparse backbone to provide a "computational bonus," dynamically engaging a variable number of specialized experts based on content complexity. The entire system is trained efficiently via a loss-free load-balancing technique, eliminating the need for cumbersome auxiliary losses. Extensive experiments show that GigaMoE sets a new state-of-the-art on the PANDA benchmark, improving detection accuracy by 1.1% over SparseFormer while simultaneously reducing the computational cost (FLOPs) by a remarkable 32.3%.

Introduction

Object detection, a fundamental task in computer vision, aims to identify and localize objects of interest within an image. Over the past decade, the field has seen great progress, with methods developed on benchmarks like MS COCO (Lin et al. 2014) achieving remarkable performance, with iconic architectures evolving from R-CNN variants (Girshick et al. 2014; Girshick 2015; Ren et al. 2015) to the widely-adopted YOLO series (Redmon et al. 2016; Redmon and Farhadi 2018; Ge et al. 2021; Wang et al. 2024a; Tian, Ye, and Doermann 2025), and Transformer-based models (Carion et al. 2020; Zhang et al. 2023). These successes, however, are predominantly rooted in "close-up" scenarios, where images typically capture a limited field of view with objects occupying a significant portion of the frame.

The rapid advancement of imaging systems has propelled a new frontier: object detection in High-Resolution Wide (HRW) shots (Li et al. 2024b,d; Fan et al. 2022; Chen et al. 2022; Lin et al. 2024b; Ma et al. 2024; Liu et al. 2024; Li et al. 2024c). Often referred to as gigapixel imaging, this paradigm analyzes images of immense scale, where a sin-

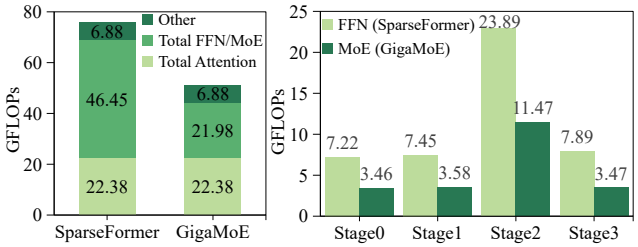

Figure 1: FLOPs analysis of GigaMoE versus SparseFormer. (a) Backbone FLOPs breakdown. Our analysis identifies the Feed-Forward Network (FFN) as the primary computational bottleneck in SparseFormer, accounting for a substantial **61.4%** of the backbone's FLOPs. Our adaptive MoE module targets this specific component, reducing its computational cost by a remarkable **52.7%**. (b) Stage-by-stage comparison illustrating that our MoE design is consistently more efficient than the standard FFN across all stages of the backbone.

*These authors contributed equally.
†These authors are co-corresponding authors.

gle frame can encapsulate vast scenes into billions of pixels, introducing unique challenges. These advances are complemented by progress in related vision tasks such as cross-modal retrieval (Wang et al. 2017) and video understanding (Gao et al. 2017), which also benefit from efficient large-scale visual analysis.

Detecting objects in HRW shots by simply applying methods designed for close-up images proves ineffective. The unique characteristics of HRW data, as found in benchmarks like PANDA (Wang et al. 2020), present a distinct set of obstacles. The most significant of these is **extreme information sparsity**. With objects of interest often occupying less than 5% of the total image area, detectors must sift through a veritable "sea of background noise". This leads to a high rate of false positives in empty regions and, conversely, false negatives where small or low-contrast objects are missed.

To address the inefficiency born from sparsity, the paradigm has shifted towards sparse processing. For HRW shots, methods like SparseFormer (Li et al. 2024d,a) have pioneered the concept of spatial selection. They divide the full image into windows and employ a lightweight scoring network to select only the most important regions for subsequent feature extraction, effectively reducing computation in the spatial dimension.

However, our analysis, illustrated in Figure 1, reveals that even for these sparsely selected regions, a new computational bottleneck emerges. The Feed-Forward Network (FFN) accounts for the vast majority of the computational load (**61.4%**) within the backbone. This inspired us to question whether the principle of sparsity could be extended from the spatial dimension to the computational dimension itself. We posit that the uniform application of a dense FFN to all selected windows—regardless of their intrinsic complexity—is suboptimal. A window with simple content does not require the same computational power as one with dense, complex objects.

To solve this, we propose to dynamically allocate the computational resources of the FFN. We introduce GigaMoE, a novel architecture designed to dynamically allocate computational resources precisely where they are needed most. Our core conceptual leap is to replace the standard, monolithic Feed-Forward Network (FFN) layers within the vision transformer backbone with a Mixture-of-Experts (MoE) module. This architectural shift enables a heterogeneous, on-demand application of computational power. Instead of forcing every selected region through the same fixed-size network, GigaMoE provides a collection of specialized expert networks and allows a dynamic router to decide how many experts should be engaged for each specific region based on its content.

The key to enabling this adaptive computation is our Sparsity-Guided Routing mechanism, which orchestrates the collaboration between shared and specialized experts. Inspired by recent advances in MoE architectures, our model first routes all selected windows through a single, **shared expert**. The designated role of this expert is to capture foundational knowledge and extract common feature patterns that are universally beneficial across all regions. By compressing this shared knowledge into a dedicated component, we allow our other experts to become more specialized.

The core of our innovation is how we engage these specialized experts. We insightfully repurpose the importance scores from the `ScoreNet`—a lightweight network used by methods like SparseFormer (Li et al. 2024d) to guide region selection—to orchestrate a *computational bonus*. Based on this score, our novel router dynamically allocates an additional number of these **specialized experts**, from zero for the simplest regions up to three for the most complex, an idea that resonates with recent findings that more complex inputs can benefit from more experts (Huang et al. 2024). This design allows the model to first build a baseline understanding of all important regions via the shared expert, and then dedicate more powerful, specialized resources only to the areas that require deeper, context-specific analysis.

The efficacy of our approach is validated through extensive experiments on the PANDA benchmark, where GigaMoE significantly improves detection accuracy by 1.1% while reducing computational costs (FLOPs) by 32.3% over the state-of-the-art.

Our contributions are threefold:

- We introduce GigaMoE, a new backbone for gigapixel object detection that, to our knowledge, is the first to leverage Mixture-of-Experts for adaptive, window-wise computational allocation in this domain.
- We propose a novel and efficient Sparsity-Guided Routing mechanism that creates a synergistic link between sparse region selection and adaptive expert allocation, repurposing existing signals to guide computation.
- We demonstrate through extensive experiments that GigaMoE sets a new state-of-the-art on the challenging PANDA benchmark, proving the efficacy and efficiency of our adaptive approach over existing methods.

Related Work

Object Detection for Close-up Shots

Object detection has matured significantly on benchmarks of close-up shots, such as PASCAL VOC (Everingham et al. 2010) and MS COCO (Lin et al. 2014). Methodologies have continuously evolved from classic two-stage and one-stage paradigms (Ren et al. 2015; Cai and Vasconcelos 2018; Redmon et al. 2016; Liu et al. 2016; Lin et al. 2017) to modern Transformer-based architectures (Carion et al. 2020; Zhang et al. 2023), with various architectural innovations improving feature representation (Ding et al. 2019), achieving remarkable performance. These methods, however, excel in standard "close-up" scenarios where objects occupy a significant portion of the frame. Their core designs are fundamentally challenged when scaling to the vast and sparse nature of gigapixel images, necessitating a different approach.

High-Resolution Wide (HRW) Shot Detection

The emergence of HRW detection, benchmarked by datasets like PANDA (Wang et al. 2020), introduced challenges of extreme scale and computational cost. The quadratic complexity of standard Vision Transformers (Dosovitskiy et al. 2021) makes them impractical for such large images. To

address this, hierarchical backbones with windowed attention were proposed to achieve linear complexity (Liu et al. 2021). Building on this, the dominant strategy shifted towards sparse processing, a concept broadly explored in vision transformers to improve efficiency through dynamic token selection (Rao et al. 2021; Meng et al. 2022) and efficient representation learning (Shen et al. 2015, 2018), where, in the context of HRW detection, only a small subset of salient regions are selected for the expensive backbone computation (Li et al. 2024b; Yang, Huang, and Wang 2022; Li et al. 2024d). Although these methods effectively reduce spatial redundancy, they still apply a uniform computational workload to all selected regions. This uniform processing overlooks the varying complexity of different regions, creating a efficiency bottleneck that our work aims to resolve.

Mixture-of-Experts for Conditional Computation

The Mixture-of-Experts (MoE) paradigm enables conditional computation by activating only a subset of expert subnetworks, scaling model capacity more efficiently than dense models. The router design is an active research area, with strategies ranging from top-k gating to more dynamic expert selection schemes (Huang et al. 2024; Yang et al. 2024; Nie et al. 2021; Zheng et al. 2024). While MoE has been successfully applied in NLP and vision for classification (Riquelme et al. 2021; He et al. 2021) and multimodal learning (Lin et al. 2024a), and in object detection (Oksuz et al. 2023), its application to gigapixel object detection remains unexplored.

Method

Our work introduces GigaMoE, a novel backbone architecture designed for efficient and adaptive object detection in gigapixel images. An ideal vision model should not only focus on salient regions but also allocate computational resources adaptively based on content complexity. To this end, GigaMoE enhances the sparse processing paradigm by introducing a dynamic, on-demand computational mechanism. The overall architecture is illustrated in Figure 2.

Overall Architecture

Inspired by Swin Transformer (Liu et al. 2021), GigaMoE is built upon a hierarchical structure consisting of four stages, which produce feature maps at different scales. Each stage begins with a Patch Merging layer that downsamples the feature map by a factor of 2, followed by a series of sequential blocks. This hierarchical design allows the model to capture features at various resolutions, which is crucial for detecting objects of vastly different sizes in HRW images.

Each stage is centered around two distinct types of attention blocks designed to capture features at different scales: a global attention block and a local attention block. The Global Attention block, following the design of SparseFormer (Li et al. 2024d), operates on aggregated features from all windows to efficiently capture coarse-grained, long-range dependencies. Following the global block, our novel GigaMoE Local Block processes the features to extract fine-grained details. Unlike SparseFormer, which applies a uniform FFN to all selected regions, our primary contribution is the redesign of this local processing step. We replace the standard FFN with a Mixture-of-Experts (MoE) module, enabling adaptive, complexity-aware computation. This key architectural shift allows GigaMoE to move beyond mere selective processing to true adaptive processing, allocating more computational power to complex regions while conserving resources on simpler ones.

Sparse Processing Foundation

Our approach is built upon the sparse processing framework pioneered by SparseFormer (Li et al. 2024d), which first enriches local features with global context and then identifies information-rich regions for further processing.

Global Attention on Aggregated Features. To complement the fine-grained local processing, the architecture first incorporates a global attention mechanism to efficiently model long-range dependencies. This is achieved in three steps. First, *Feature Aggregation* generates a coarse representation $\bar{z}$ of the input feature map z by averaging the tokens within each window of size $M \times M$.

$$\bar{z}_{x',y'} = \frac{1}{M^2} \sum_{\Delta x=0}^{M-1} \sum_{\Delta y=0}^{M-1} z_{x+\Delta x, y+\Delta y} \tag{1}$$

where (x', y') are the coordinates in the aggregated feature map, corresponding to the top-left coordinate (x, y) of a window in the original feature map z. Second, a standard Multi-Head Self-Attention (MSA) module is applied to the aggregated feature map $\bar{z}$ to perform *Window-level Global Attention*. Finally, the output of the global attention is upsampled back to the original resolution and used to update the original feature map via a residual connection:

$$z \leftarrow z + \text{Upsample}(\text{MSA}(\text{LN}(\bar{z}))) \tag{2}$$

Variance-based Scoring. After the feature map z is updated with global context, a lightweight `ScoreNet` is employed to differentiate salient regions. To do this, a coarse representation, $\hat{z}$, is first calculated by averaging features within each window of the updated map. The `ScoreNet`, which is an MLP, then processes the residual $z - \hat{z}$, feeding its output to a SoftMax function to generate an importance score for each window:

$$\text{ScoreNet}(z, \hat{z}) = \text{SoftMax}(\text{MLP}(z - \hat{z})) \tag{3}$$

Window Sparsification. Based on the scores from `ScoreNet`, we apply a top-k selection strategy to create a sparse set of windows. This selection can be formally expressed as:

$$Z_{sparse} = M_{select} \cdot Z \tag{4}$$

where $Z \in \mathbb{R}^{N \times D}$ represents the features of all N windows, $M_{select} \in \{0, 1\}^{K \times N}$ is a one-hot selection matrix that chooses the top K windows, and $Z_{sparse} \in \mathbb{R}^{K \times D}$ are the features for resulting sparse windows. This allows subsequent local operations to focus only on these salient regions.

GigaMoE Local Block for Adaptive Computation

As illustrated in the lower panel of Figure 2, the GigaMoE Local Block operates on the sparse set of windows identified

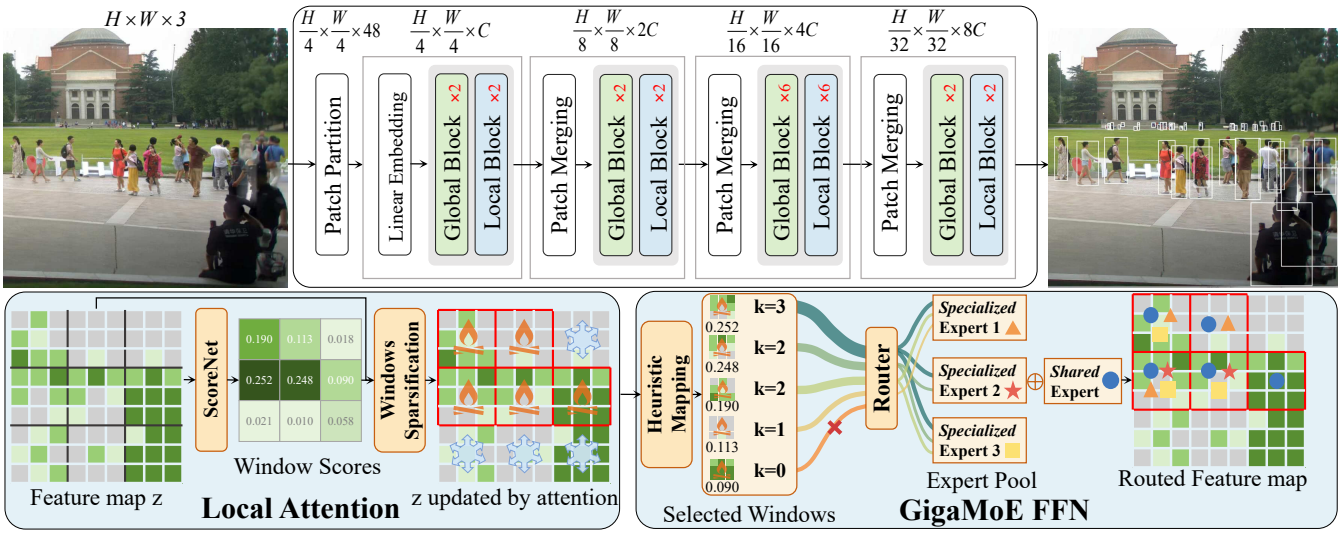

Figure 2: The overall architecture of GigaMoE, composed of four hierarchical stages.The lower panel details our core contribution. The `ScoreNet` first generates scores to select salient windows for local attention (highlighted with a red border). These scores then guide a **Heuristic Mapping** to determine the number of experts (k) for each window. Finally, a router selects the specific k experts. As detailed in the diagram's legend, simple geometric shapes are used to represent different experts.

in the previous stage (Z_{sparse}). It performs intensive local computation in two sequential steps: local window attention and our adaptive GigaMoE FFN.

Local Window Attention. The features corresponding to the selected sparse windows (Z_{sparse}) are first processed by a standard Multi-Head Self-Attention (MSA) module. We use the shifted-window attention mechanism from Swin Transformer (Liu et al. 2021) to enable interaction both within and across the most salient local regions, resulting in fine-grained feature refinement.

GigaMoE FFN with Sparsity-Guided Routing. The output of the attention module is then passed to our novel GigaMoE FFN, which replaces the standard, monolithic FFN. It comprises a single *shared expert* (E_{shared}) and a set of N_s *specialized experts* ($\{E_i\}_{i=1}^{N_s}$).

Each expert, both shared and specialized, is a standard two-layer Feed-Forward Network (FFN). Our core design principle is to partition the computational capacity of a single, large FFN from a baseline model (like SparseFormer) across our multiple, smaller experts. Specifically, if the baseline FFN has a hidden dimension of C_{ffn}, and our model is configured to use a shared expert and route to a maximum of k_{max} specialized experts, we set the hidden dimension of each of our experts to $C_{expert} = C_{ffn}/(k_{max} + 1)$. This design ensures that even in the most computationally intensive case (when a window is processed by the shared expert and all k_{max} specialized experts), the total FLOPs remain comparable to that of the original monolithic FFN, thereby enabling adaptive computation without a significant increase in theoretical peak cost.

The key innovation here is our *Sparsity-Guided Routing* mechanism, which repurposes the importance scores from the `ScoreNet` to dynamically control the computational budget. This process, depicted in Figure 2, involves two steps: a **Heuristic Mapping** to determine the number of experts, followed by a router to select them. Let $S = \{s_w\}_{w=1}^{K}$ be the set of importance scores for the K sparsely selected windows. We first rank these windows in descending order of their scores. The core of our heuristic is a predefined distribution vector $P = (p_0, p_1, \ldots, p_{k_{max}})$ where $\sum_{i=0}^{k_{max}} p_i = 1$, and p_i represents the desired proportion of windows to be assigned i specialized experts. Based on this, the number of experts k_w for a window w with rank $r_w \in \{1, \ldots, K\}$ is determined by the following assignment rule:

$$k_w = j \quad \text{if} \quad K \sum_{i=j+1}^{k_{max}} p_i < r_w \leq K \sum_{i=j}^{k_{max}} p_i \qquad (5)$$

For our best-performing model, we use $k_{max} = 3$ and a distribution of $P = (0.4, 0.3, 0.2, 0.1)$, which assigns 3 experts to the top 10% of windows, 2 experts to the next 20%, 1 expert to the next 30%, and no specialized experts to the bottom 40%. This creates a powerful synergy: the same signal used to decide *what* to process is also used to decide *how much* computation to allocate. This design allows the model to dedicate more powerful, specialized resources only to the areas that require deeper analysis, a behavior qualitatively confirmed by the visualizations in Figure 3.

While the number of experts k_w is determined by this heuristic mapping, the choice of *which* k_w experts to use is determined by a lightweight router G, implemented as a single linear layer. For each window's feature tensor z_w from the attention module, the router first computes its layer-normalized representation $z'_w = \text{LN}(z_w)$ and then calculates logits over all N_s specialized experts based on its mean-

Figure 3: Visualization of first-stage expert allocation for GigaMoE on the PANDA dataset. The original images (top) are shown with their corresponding expert allocation maps overlaid (bottom), where warmer colors (from blue to green) signify a higher computational budget. GigaMoE intelligently allocates computational resources to complex regions, such as dense crowds, vehicles, and richly-textured foliage. It also focuses on object edges, including building facades, road curbs, and traffic markings. In contrast, semantically simple areas like the sky and open ground receive minimal resources (deep blue).

pooled features $\bar{z}'_w = \text{mean}(z'_w)$:

$$g_w = G(\bar{z}'_w) \tag{6}$$

The final output for the window, y_w, is the sum of the shared expert's output and the weighted sum of the top k_w selected specialized experts' outputs, both of which take the normalized features z'_w as input:

$$y_w = E_{shared}(z'_w) + \sum_{i \in \text{Top-k}(g_w, k_w)} \text{SoftMax}(g_w)_i \cdot E_i(z'_w) \tag{7}$$

Online Expert Load Balancing

A common challenge in training MoE models is "expert collapse," where the router disproportionately favors a few experts, leaving others under-trained. The standard solution is an auxiliary load-balancing loss, which, however, introduces an extra hyperparameter that is often difficult to tune and can interfere with the primary training objective. To circumvent this, we employ a more direct and elegant online, bias-based load balancing mechanism, inspired by recent work (Wang et al. 2024b). Rather than modifying the loss function, this method introduces a learnable bias term b_i for each specialized expert E_i. This bias is dynamically adjusted during training based on real-time usage statistics. If an expert is underutilized, its bias is increased to make it a more attractive choice for the router, and vice-versa, thereby directly encouraging balanced utilization without complicating the loss landscape.

Building upon this mechanism, we introduce a simple yet effective enhancement: an *adaptive bias update rate*. We posit that a fixed update rate is suboptimal. In the early stages of training, a larger update rate is beneficial to quickly correct severe load imbalances. Conversely, in later stages, a smaller rate is preferable for fine-tuning the expert load without disrupting the learned routing patterns. To achieve this, we make the update rate u decay over the course of training, from an initial value u_{init} to zero. We implement two decay schedules based on the current training step t and total steps T_{max}: a linear decay, where the rate is scaled by $(1 - t/T_{max})$, and a cosine annealing schedule, where it is scaled by $\frac{1}{2}(1 + \cos(\frac{t \cdot \pi}{T_{max}}))$. The update process is detailed in Algorithm 1. This method leads to more stable and efficient training of the MoE module.

Algorithm 1: Online Expert Load Balancing with Adaptive Update Rate.

Input: MoE model θ, training batch iterator B, initial bias update rate u_{init}, total training steps T_{max}, number of specialized experts N_s.

Initialize per-expert bias $b_i = 0$ for $i = 1, \ldots, N_s$.
Initialize step counter $t = 0$.
for a batch $\{(x_w, y_w)\}_w$ in B **do**
  $t \leftarrow t + 1$.
  Calculate decay factor α_t based on a schedule (e.g., linear: $1 - t/T_{max}$, or cosine).
  Set current update rate $u_t = u_{init} \cdot \alpha_t$.
  Calculate gating scores g_w for each window w.
  Select top-k_w experts for each window w based on the biased scores $g_w + b$.
  Count the number of windows c_i assigned to each expert E_i, and the average number $\bar{c}$.
  Calculate the load violation error $e_i = \bar{c} - c_i$.
  Update bias $b_i \leftarrow b_i + u_t \cdot \text{sign}(e_i)$.
end forOutput: Trained model θ, updated bias b.

Experiments

Main Results

Datasets. Our evaluation is on the public PANDA benchmark (Wang et al. 2020), which is the first human-centric gigapixel-level dataset. It contains 18 scenes with over 15,974.6k bounding boxes annotated. Specifically, there are 13 scenes for training and 5 scenes for testing.

Evaluation Metrics. We report the FLOPs and standard COCO metrics, including AP_{total}, AP_S ($< 96^2$ pixels), AP_M (96^2 – 288^2 pixels), and AP_L ($> 288^2$ pixels). For a fair efficiency evaluation, the GFLOPs for all methods are calculated based on the average computational cost to process a 1280 × 800 image window. Furthermore, we report GFLOPs for both foreground (F) and background (B) regions to specifically demonstrate the efficiency of our adaptive computation approach in reducing redundant calculations on non-salient areas.

Implementation Details. We implement all detectors using the MMDetection toolbox (Chen et al. 2019). To ensure a fair comparison against baseline backbones such as SparseFormer and Swin Transformer, all models are configured with identical architectural hyperparameters (e.g., depths, embedding dimensions, number of multi-heads) when paired with the same detection head. Following standard practice, all models are trained from scratch for 36 epochs. The code for our model and experiments will be made available in the supplementary material.

Comparison with State-of-the-Art. Table 1 compares our model with state-of-the-art methods on the challenging PANDA benchmark, unequivocally demonstrating the superiority of our approach. When paired with DINO head, GigaMoE not only achieves a new state-of-the-art AP of **79.1%**, surpassing the previous best (SparseFormer) by a significant margin of **1.1%**, but also does so with vastly greater efficiency. Our model requires only **51.24 GFLOPs**, a remarkable **32.3%** reduction in computational cost compared to SparseFormer's 75.71 GFLOPs. This gain underscores the power of our adaptive computation. Notably, our approach also yields a considerable improvement in small object detection (AP_S), a critical challenge in this domain. A similar trend is observed with the Dynamic-Head, where GigaMoE improves performance while cutting GFLOPs by 18.1%. As illustrated in the Pareto frontier plot in Figure 4, our GigaMoE models, represented by the green and orange bubbles in the top-left, consistently establish a more favorable accuracy-efficiency trade-off, pushing the boundary towards higher accuracy at a significantly lower computational cost.

Ablation Studies

Expert Allocation Strategy. We ablate our dynamic expert allocation strategy in Table 2, which rigorously quantifies its benefits.

First, we establish baselines by forcing all windows to use a fixed number of specialized experts (from 0 to 3), in addition to the shared expert. As expected, using more experts consistently improves performance, but at a steep computational cost. The "Shared + 3 Specialized" configuration reaches a peak AP of 79.4%, but requires 74.03 GFLOPs. Conversely, relying solely on the shared expert ("Shared Expert Only") causes a severe performance drop to 74.3% AP, confirming that the specialized experts are essential for achieving high accuracy.

Against these baselines, our adaptive strategies demonstrate a far superior trade-off. Our main configuration, the

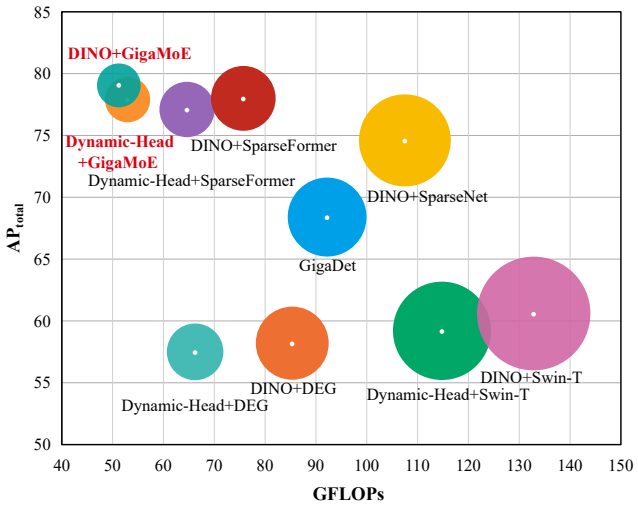

Figure 4: FLOPs vs. AP on PANDA. The center of bubbles represents the model's performance, while the radius is proportional to its GFLOPs. Our GigaMoE models, labeled in red, achieve a superior accuracy-efficiency trade-off.

"Inverted Pyramid," which allocates more experts to higher-scoring windows, achieves an AP of **79.1%**—statistically on par with the expensive "Shared + 3 Specialized" baseline—while requiring only **51.24 GFLOPs**. This represents a massive **30.8%** reduction in computation for a negligible 0.3% AP difference. This result substantiates our claim that GigaMoE achieves the performance of a much larger model at a fraction of the cost.

Effectiveness of GigaMoE Components. Table 3 dissects the contribution of each GigaMoE component. (a) We start with a strong sparse processing baseline (SparseFormer), which uses a standard FFN and achieves a competitive 78.0% AP. (b) Simply replacing the FFN with a traditional MoE module that routes every window to a fixed number of experts (top-2) yields a marginal performance gain (+0.1% AP) but at a significantly lower computational cost (-16.9% GFLOPs), demonstrating the inherent efficiency of the MoE structure. (c) The introduction of our core contribution, the **Sparsity-Guided Routing** mechanism, marks a significant leap. By dynamically allocating experts based on content complexity, our model achieves a new peak performance of **79.1%** AP (+1.0% over traditional MoE) while further reducing the computational cost to just **51.24 GFLOPs**. This clearly validates that our adaptive routing strategy is superior to fixed routing. (d) Finally, to verify the role of the shared expert, we remove it from our full model. This results in a drastic performance drop of 5.6% AP down to 73.5%, even though the FLOPs are lower. This confirms its crucial role in providing a feature baseline, which the specialized experts build upon.

Load Balancing Mechanism. To ensure a robust expert balance beyond the implicit balancing from our routing, we employ an online, bias-based strategy (Table 4). We evaluate the expert load balance using the Maximal Violation

Method	Backbone	GFLOPs-F	GFLOPs-B	GFLOPs-A	AP_{total}	AP_S	AP_M	AP_L
General Two-Stage Detectors								
FasterRCNN (Ren et al. 2015)	ResNet-101	14.15	268.98	283.14	-	19.0	55.2	74.4
FasterRCNN* (Fan et al. 2022)	ResNet-50	10.35	196.71	207.07	70.5	20.3	71.2	76.0
RetinaNet (Lin et al. 2017)	ResNet-101	15.77	299.62	315.39	-	22.1	56.1	74.0
CascadeRCNN (Cai and Vasconcelos 2018)	ResNet-101	15.54	295.24	310.78	-	22.7	57.9	76.5
Specialized Gigapixel Detectors								
ClusDet (Yang et al. 2019)	ResNet-50	10.35	196.71	207.07	71.8	21.9	69.6	78.2
DMNet (Li et al. 2020)	ResNet-50	10.35	196.71	207.07	54.0	11.9	37.1	71.4
GigaDet (Chen et al. 2022)	CSP-DarkNet-53	4.61	87.59	92.20	68.4	21.0	59.9	76.2
PAN (Fan et al. 2022)	ResNet-50	10.35	196.71	207.07	71.5	25.6	71.9	76.8
Sparse Processing Methods								
Dynamic-Head (Dai et al. 2021)	Swin-T	5.74	109.1	114.8	59.2	16.5	53.7	69.4
Dynamic-Head+DEG (Song et al. 2021)	PVT-DEG	6.12	60.11	66.23	57.5	15.4	50.8	69.5
Dynamic-Head+SparseFormer (Li et al. 2024d)	SparseFormer	6.29	58.35	64.64	77.1	36.4	74.0	**86.3**
Dynamic-Head+GigaMoE (Ours)	**GigaMoE**	**6.07**	**46.87**	**52.94**	**77.9**	**37.7**	**74.3**	85.5
DINO (Zhang et al. 2023)	ResNet-50	6.21	118.02	124.24	54.2	28.9	53.0	59.2
DINO+SparseNet (Li et al. 2024b)	SparseNet	6.53	100.97	107.50	74.6	38.1	75.4	79.7
DINO (Zhang et al. 2023)	Swin-T	6.64	126.19	132.84	60.6	36.7	61.2	64.9
DINO+DEG (Song et al. 2021)	PVT-DEG	6.77	78.57	85.34	58.2	33.9	57.8	62.4
DINO+SparseFormer (Li et al. 2024d)	SparseFormer	6.90	68.81	75.71	78.0	50.8	78.1	**82.3**
DINO+GigaMoE (Ours)	**GigaMoE**	**6.36**	**44.88**	**51.24**	**79.1**	**53.8**	**81.9**	82.1

Table 1: Comparison with the SoTAs on PANDA. "F" and "B" denote foreground and background, respectively (A=F+B). AP is reported in percentage. Our GigaMoE results are highlighted in bold.

Distribution Config	AP_{total}	GFLOPs-A
Fixed Number of Specialized Experts (Baselines)		
(1,0,0,0) - Shared Only (0 sp.)	0.743	**40.77**
(0,1,0,0) - Shared + 1 sp.	0.764	51.86
(0,0,1,0) - Shared + 2 sp.	0.781	62.94
(0,0,0,1) - Shared + 3 sp.	**0.794**	74.03
Variable Number of Specialized Experts (Ours)		
(0.25,0.25,0.25,0.25) - Uniform	0.789	57.00
(0.1,0.2,0.3,0.4) - Pyramid	0.792	62.32
(0.4,0.3,0.2,0.1) - Inverted Pyramid	0.791	51.24

Table 2: Ablation studies on the expert allocation distribution for our GigaMoE module.The distribution '(p0, p1, p2, p3)' defines the percentage of windows assigned 0, 1, 2, and 3 specialized experts, respectively. All experiments are conducted with the DINO head.

(`MaxVio`) metric, defined as:

$$\text{MaxVio} = \frac{\max_i \text{Load}_i - \overline{\text{Load}}}{\overline{\text{Load}}} \tag{8}$$

where a lower value indicates better balance. The results clearly demonstrate the efficacy of our approach. Without any explicit balancing mechanism, the model suffers from severe expert imbalance, resulting in a high `MaxVio` of 0.471 and suboptimal performance. Employing a fixed update rate for the bias already improves the situation substantially, reducing the `MaxVio` to 0.083. Most importantly, our proposed adaptive update rate strategies prove most effective. The linear decay schedule, in particular, achieves the highest AP of **79.1%** while simultaneously reducing the final `MaxVio` to just **0.061**, demonstrating its superiority.

Configuration	AP_{total}	GFLOPs-A
(a) Sparse Baseline (Standard FFN)	0.780	75.71
(b) Traditional MoE (Fixed top-2)	0.781	62.94
(c) Ours: Sparsity-Guided Routing	**0.791**	51.24
(d) (c) w/o Shared Expert	0.735	**43.64**

Table 3: Ablation study on the effectiveness of GigaMoE components.

Balancing Method	AP_{total}	Avg. Final MaxVio
No Balancing	0.769	0.471
Fixed Rate	0.788	0.083
Linear Decay	**0.791**	**0.061**
Cosine Decay	0.789	0.069

Table 4: Ablation study on the load balancing mechanism. We report the average `MaxVio` across all stages at the final validation epoch. For methods with a bias update, the initial rate is $u_{init} = 10^{-2}$.

Conclusion

In this paper, we introduced GigaMoE, a novel backbone that pioneers adaptive computation for gigapixel object detection. Addressing the limitation of uniform computational workload in sparse methods, we replace the standard FFN with a MoE module. Our core Sparsity-Guided Routing mechanism repurposes importance scores to dynamically allocate a "computational bonus," engaging a variable number of specialized experts based on content complexity. This creates an efficient synergy between deciding *what* to process and *how much* to process it. Experiments on the PANDA benchmark demonstrate that GigaMoE establishes a new state-of-the-art, significantly improving accuracy while reducing computational costs. We believe this work opens a promising new direction for designing more efficient and intelligent vision backbones for large-scale visual recognition.

Acknowledgments

This work was supported by National Science and Technology Major Project (No. 2022ZD0119402), "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C01142), National Natural Science Foundation of China (No. U21B2013), and the Key R&D Program of Xinjiang, China (No. 2022B01006).

References

Cai, Z.; and Vasconcelos, N. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *Proc. Eur. Conf. Comput. Vis.*

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. arXiv:1906.07155.

Chen, K.; Wang, Z.; Wang, X.; Gong, D.; Yu, L.; Guo, Y.; and Ding, G. 2022. Towards real-time object detection in GigaPixel-level video. *Neurocomputing*, 494: 251–259.

Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; and Zhang, L. 2021. Dynamic Head: Unifying Object Detection Heads With Attentions. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Ding, X.; Guo, Y.; Ding, G.; and Han, J. 2019. ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks. *CoRR*, abs/1908.03930.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. Int. Conf. Learn. Represent.*

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88: 303–338.

Fan, J.; Liu, H.; Yang, W.; See, J.; Zhang, A.; and Lin, W. 2022. Speed Up Object Detection on Gigapixel-Level Images With Patch Arrangement. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Gao, L.; Guo, Z.; Zhang, H.; Xu, X.; and Shen, H. T. 2017. Video Captioning With Attention-Based LSTM and Semantic Consistency. *IEEE Trans. Multimedia*, 19(9): 2045–2055.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. Yolox: Exceeding yolo series in 2021. In *Adv. Neural Inform. Process. Syst.*

Girshick, R. 2015. Fast R-CNN. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

He, M.; Lv, G.; He, W.; Fan, J.; and Zeng, G. 2021. DeepME: Deep Mixture Experts for Large-scale Image Classification. In *Proc. Int. Joint Conf. Artif. Intell.*

Huang, Q.; Misra, I.; Gollakota, A.; Cordonnier, J.-B.; Bhatt, G.; Miech, A.; Bordes, F.; Arnab, A.; Lu, W.-L. B.; LeCun, Y.; Leduc, L.; Joulin, A.; and Bojanowski, P. 2024. Harder Tasks Need More Experts: Dynamic Routing in MoE Models. arXiv:2403.07652.

Li, C.; Yang, T.; Zhu, S.; Chen, C.; and Guan, S. 2020. Density Map Guided Object Detection in Aerial Images. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. Worksh.*

Li, W.; Guo, Y.; Zheng, J.; Lin, H.; Ma, C.; Fang, L.; and Yang, X. 2024a. Bridging the gap between object detection in close-up and high-resolution wide shots. *Comput. Vis. Image Underst.*, 249: 104181.

Li, W.; Zhang, R.; Lin, H.; Guo, Y.; Ma, C.; and Yang, X. 2024b. SaccadeDet: A Novel Dual-Stage Architecture for Rapid and Accurate Detection in Gigapixel Images. In *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases*.

Li, W.; Zhang, R.; Lin, H.; Guo, Y.; Ma, C.; and Yang, X. 2024c. SaccadeMOT: Enhancing Object Detection and Tracking in Gigapixel Images via Scale-Aware Density Estimation. In *Proc. Eur. Conf. Artif. Intell.*

Li, W.; Zhang, R.; Lin, H.; Guo, Y.; Ma, C.; and Yang, X. 2024d. Sparseformer: Sparse-selected objects for efficient real-time giga-pixel image object detection. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Lin, B.; Tang, Z.; Ye, Y.; Huang, J.; Zhang, J.; Pang, Y.; Jin, P.; Ning, M.; Luo, J.; and Yuan, L. 2024a. MoE-LLaVA: Mixture of Experts for Large Vision-Language Models. arXiv:2401.15947.

Lin, H.; Wei, C.; He, L.; Guo, Y.; Zhao, Y.; Li, S.; and Fang, L. 2024b. GigaTraj: Predicting Long-term Trajectories of Hundreds of Pedestrians in Gigapixel Complex Scenes. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common objects in context. In *Proc. Eur. Conf. Comput. Vis.*

Liu, C.; Wei, H.; Yang, J.; Liu, J.; Li, W.; Guo, Y.; and Fang, L. 2024. Gigahumandet: Exploring full-body detection on gigapixel-level images. In *Proc. AAAI Conf. Artif. Intell.*

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. SSD: Single shot multibox detector. In *Proc. Eur. Conf. Comput. Vis.*

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*

Ma, T.; Bai, B.; Lin, H.; Wang, H.; Wang, Y.; Luo, L.; and Fang, L. 2024. When Visual Grounding Meets Gigapixel-level Large-scale Scenes: Benchmark and Approach. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; and Lim, S.-N. 2022. AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Nie, X.; Miao, X.; Zuo, S.; Lv, J.; Huang, B.; Yang, S.; Zhai, Y.; Deng, Z.-H.; Sun, Y.; Zong, Z.; and Li, H. 2021. EvoMoE: An Evolutional Mixture-of-Experts Training Framework via Dense-to-Sparse Gate. arXiv:2112.14397.

Oksuz, K.; Kuzucu, S.; Joy, T.; and Dokania, P. K. 2023. MoCaE: Mixture of Calibrated Experts Significantly Improves Object Detection. arXiv:2309.14976.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Adv. Neural Inform. Process. Syst.*

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Redmon, J.; and Farhadi, A. 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst.*

Riquelme, C.; Puigcerver, J.; Mustafa, B.; Neumann, M.; Jenatton, R.; Pinto, A. S.; Keysers, D.; and Houlsby, N. 2021. Scaling vision with sparse mixture of experts. In *Adv. Neural Inform. Process. Syst.*

Shen, F.; Shen, C.; Liu, W.; and Shen, H. T. 2015. Supervised Discrete Hashing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Shen, F.; Xu, Y.; Liu, L.; Yang, Y.; Huang, Z.; and Shen, H. T. 2018. Unsupervised Deep Hashing with Similarity-Adaptive and Discrete Optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12): 3034–3044.

Song, L.; Zhang, S.; Liu, S.; Li, Z.; He, X.; Sun, H.; Sun, J.; and Zheng, N. 2021. Dynamic Grained Encoder for Vision Transformers. In *Adv. Neural Inform. Process. Syst.*

Tian, Y.; Ye, Q.; and Doermann, D. 2025. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv preprint arXiv:2502.12524*.

Wang, A.; Chen, H.; Liu, L. Y.; Chen, K.; Lin, Z.; and Han, J. 2024a. YOLOv10: Real-Time End-to-End Object Detection. *arXiv preprint arXiv:2405.14458*.

Wang, B.; Yang, Y.; Xu, X.; Hanjalic, A.; and Shen, H. T. 2017. Adversarial Cross-Modal Retrieval. In *Proc. ACM Int. Conf. Multimedia*.

Wang, L.; Gao, H.; Zhao, C.; Sun, X.; and Dai, D. 2024b. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts. arXiv:2408.15664.

Wang, X.; Zhang, X.; Zhu, Y.; Guo, Y.; Yuan, X.; Xiang, L.; Wang, Z.; Ding, G.; Brady, D.; Dai, Q.; and Fang, L. 2020. PANDA: A Gigapixel-Level Human-Centric Video Dataset. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Yang, C.; Huang, Z.; and Wang, N. 2022. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*

Yang, F.; Fan, H.; Chu, P.; Blasch, E.; and Ling, H. 2019. Clustered Object Detection in Aerial Images. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*

Yang, Y.; Li, C.; Raffel, C.; Papadimitriou, P.; Feng, B.; Sachan, M.; and Dai, A. 2024. XMoE: Sparse Models with Fine-grained and Adaptive Expert Selection. In *Findings Assoc. Comput. Linguistics*.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L. M.; and Shum, H.-Y. 2023. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *Proc. Int. Conf. Learn. Represent.*

Zheng, H.; Xu, Z.; Wang, Z.; Tu, Y.; Chen, X.; Wang, S.; Wang, F.; Li, J.; Li, Y.; Jiang, X.; Li, C.; Zhang, Z.-Y.; and Sun, H. 2024. Learn to be Efficient: Build Structured Sparsity in Large Language Models. arXiv:2402.06126.