# AdaGlimpse: Active Visual Exploration with Arbitrary Glimpse Position and Scale

Adam Pardyl[1,2,3], Michał Wronka[2], Maciej Wołczyk[1], Kamil Adamczewski[1], Tomasz Trzciński[1,4,5], and Bartosz Zieliński[1,2]

[1] IDEAS NCBR
{adam.pardyl, maciej.wolczyk, kamil.adamczewski,
tomasz.trzcinski, bartosz.zielinski}@ideas-ncbr.pl
[2] Jagiellonian University, Faculty of Mathematics and Computer Science
michal.wronka@student.uj.edu.pl
[3] Jagiellonian University, Doctoral School of Exact and Natural Sciences
[4] Warsaw University of Technology
[5] Tooploox

**Abstract.** Active Visual Exploration (AVE) is a task that involves dynamically selecting observations (glimpses), which is critical to facilitate comprehension and navigation within an environment. While modern AVE methods have demonstrated impressive performance, they are constrained to fixed-scale glimpses from rigid grids. In contrast, existing mobile platforms equipped with optical zoom capabilities can capture glimpses of arbitrary positions and scales. To address this gap between software and hardware capabilities, we introduce AdaGlimpse. It uses Soft Actor-Critic, a reinforcement learning algorithm tailored for exploration tasks, to select glimpses of arbitrary position and scale. This approach enables our model to rapidly establish a general awareness of the environment before zooming in for detailed analysis. Experimental results demonstrate that AdaGlimpse surpasses previous methods across various visual tasks while maintaining greater applicability in realistic AVE scenarios.

**Keywords:** Active visual exploration · Vision transformers · Reinforcement learning

## 1 Introduction

Common machine learning solutions for computer vision tasks, such as classification, segmentation, or scene understanding, usually presume access to complete input data [13]. However, this assumption does not apply to embodied agents functioning in the real world. Agents such as robots and UAVs face constraints on their data-gathering capabilities, such as a restricted field of view and limited operational time, caused by dynamically changing environments [48]. Moreover, capturing and analyzing high-resolution images of the entire visible area is inefficient, as not every part of an image contains the same amount of details.

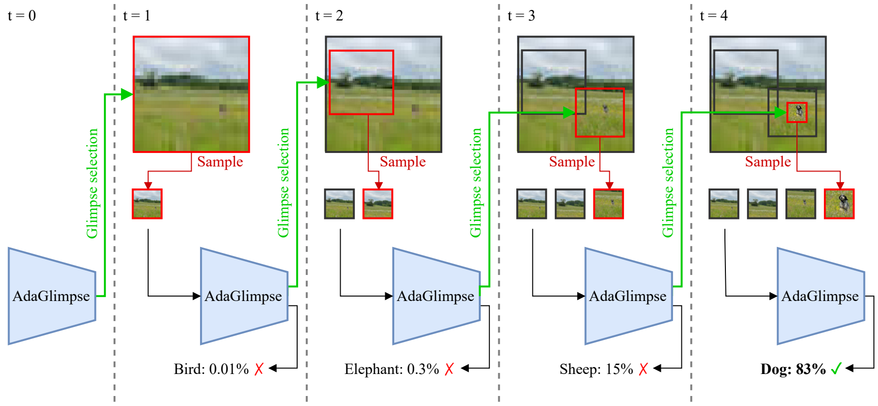

**Fig. 1: Adaptive Glimpse (AdaGlimpse)**: Our approach selects and processes glimpses of arbitrary position and scale, fully exploiting the capabilities of modern hardware. In this example, AdaGlimpse selects a low-resolution glimpse of the whole environment. Based on this glimpse, it predicts a bird with probability 0.01, too low to make the final decision. Instead, it selects the second glimpse by zooming in to the upper left corner. The process repeats four times until the probability of the predicted class is higher than a specified threshold.

Active Visual Exploration (AVE) addresses the challenge of how an agent should select visual information from its environment to achieve a particular objective. Instead of systematically sampling and analyzing the entire available environment at the highest resolution, an agent dynamically chooses the location for sampling subsequent observations, informed by insights from prior exploration steps [33]. This process of selecting visual samples, often referred to as *glimpses*, is inspired by the natural way humans explore their surroundings by instinctively moving their heads and eyes [17].

Current research in active visual exploration can be categorized into two groups. The first group of approaches divides the image into a regular grid of fixed-sized glimpses from which the model tries to pick the most informative ones [32, 35, 39, 41]. The second group starts by capturing a low-resolution image of the entire environment, and then it again selects glimpses from regular grids [12, 30, 46]. Relying solely on regular grids fails to fully exploit the capabilities of modern hardware, which can provide a glimpse of any position and scale. For example, in a pan-tilt-zoom camera, we can achieve it by using optical zoom [11], while in a UAV, we can alter its altitude [22].

In this paper, we overcome current limitations by introducing **AdaGlimpse** (Adaptive Glimpse)[6], an active visual exploration method that selects glimpses of arbitrary scale and position, significantly reducing the number of observations required to understand the environment. Drawing inspiration from [31], we build our network on an input-elastic vision transformer. In each exploration step, our

[6] Source code is available at `https://github.com/apardyl/AdaGlimpse`

model predicts the optimal position and scale for the next glimpse as a value in a continuous space. Since the patch-sampling operation is not differentiable, we train the model using a reinforcement learning algorithm. In particular, we use the Soft Actor-Critic algorithm [16] since it excels in exploration tasks.

Through exhaustive experiments, we show that AdaGlimpse outperforms state-of-the-art methods on common benchmarks for reconstruction, classification, and segmentation tasks. As such, our method enables more effective utilization of embodied platform capabilities, leading to faster environmental awareness. Our contributions can be summarized as follows:

- We introduce a novel approach to Active Visual Exploration (AVE) that selects and processes glimpses of arbitrary position and scale.
- We present a task-agnostic architecture based on a visual transformer.
- We formulate AVE as a Markov Decision Process with a carefully designed observation space, and leverage the Soft Actor-Critic reinforcement learning algorithm that excels in exploration.

## 2 Related work

**Missing data.** The problem of missing data in context of images has been addressed in a variety of ways, such as inferring remaining information from the input distribution using a fully connected network [42], or more commonly by image reconstruction. In particular, MAT [25] performs inpainting using a transformer network with local attention masking, a solution that additionally reduces computation by processing only informative parts of the image. A similar principle can be found in ViT based Masked Autoencoder (MAE) [18] where the encoder network operates only on visible patches, while the decoder processes all patches, including the masked ones.

**Region selection.** Numerous methods exist for selecting the most informative regions from an image, including expectation maximization [36, 51], majority voting [2], wake-sleep algorithm [4], sampling from self-attention or certainty maps [32, 39, 41], and Bayesian optimal experiment design [34]; yet, recently the most predominant solution is using reinforcement learning algorithms, such as variants of the Policy Search [28, 29, 50] Deep Q-Learning [6, 7] or Actor-Critic [35].

**Variable scale transformers.** A number of studies have been conducted to overcome the constraint of Vision Transformers (ViTs) of working only with rigid grid of fixed size patches, be it by modifying grid scale sampling during training phase [24,45,49] or with position and patch encoding rescaling tricks [5]. Beyond Grids [31] interests us in particular, as it equips ViT with the ability to use any square present in an image as a patch, removing both grid and size limitation.

**Active visual exploration.** The SLAM (simultaneous localization and mapping) challenge is often described in the context of active exploration [33]. A popular approach seen in many models [3, 12, 30, 46] is to feed the model a low-resolution version of the image and use a variation of a policy gradient algorithm to choose parts of an image to focus on. Many notable works in domain of AVE are using CNN-based attention maps for glimpse selection [39–41, 47]. Simglim [21] introduces a MAE-based model with an additional glimpse decision neural network to solve image reconstruction tasks. AME [32] similarly uses MAE as a backbone, but makes decisions solely on the basis of attention maps without added loss or modules. STAM [35] uses a visual transformer and a one-step actor-critic for choosing glimpse locations in the classification task.

## 3 AdaGlimpse

The key idea of Adaptive Glimpse (AdaGlimpse) is to let the agent select both the scale and position of each successive observation (*glimpse*) from a continuous action space. This way, the agent can learn to decide whether, at a given step of the exploration process, it is preferable to sample a wider view field at a lower relative resolution, zoom in on detail to capture a small high-resolution glimpse, or choose a midway solution.

In this section, we start by formalizing the concept of adaptive glimpse sampling (see Sec. 3.1), and then we discuss the main two components of AdaGlimpse presented in Fig. 2. In Sec. 3.2, we describe a vision transformer encoder with variable scale sampling and a task-specific head. In Sec. 3.3, we present the reinforcement learning agent based on the Soft Actor-Critic (SAC) algorithm.

### 3.1 Adaptive glimpse sampling

**Glimpses.** Let $X$ be an unobserved scene to explore. We assume $X$ to be a rectangle within the Cartesian coordinate system. A *glimpse* $G$ is a square region within $X$ that can be observed by a camera, specified by the position of its top-left corner $(x, y)$ and its size $d$ (camera field of view). Furthermore, we define glimpse scale as: $z = (d - d_{\text{min}})/(d_{\text{max}} - d_{\text{min}})$, where $d_{\text{max}}$ and $d_{\text{min}}$ are constants denoting the maximum and minimum field of view of an agent camera. Intuitively, a scale of 0 corresponds to the maximum camera zoom level and a scale of 1 to the widest view possible. Finally, let $C = (x, y, z)$ be the coordinates of glimpse $G$ and constant $d_{\text{cam}} \times d_{\text{cam}}$ denote the sampling resolution of $G$, i.e., the resolution glimpses obtained from the camera sensor.

**AVE process.** Now, we can define the active visual exploration process as a sequence of glimpse selections. Let $T$ be the maximum number of exploration steps. The process starts at time $t = 0$ with an empty sequence of observations. At time $t \in \{1, ..., T\}$, the camera captures a glimpse $G_t$ at coordinates $C_t$ proposed by the model, generating a patch of resolution $d_{\text{cam}} \times d_{\text{cam}}$. We simulate the process of glimpse capturing by cropping a patch from a large image representing

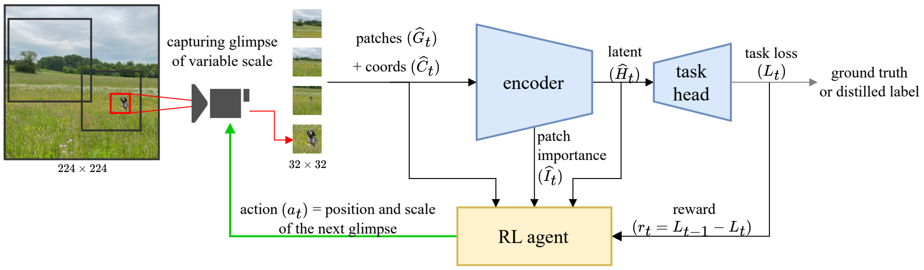

**Fig. 2: Architecture**: AdaGlimpse consists of two parts: a vision transformer-based encoder with a task-specific head (see Sec. 3.2) and a Soft Actor-Critic RL agent (see Sec. 3.3). At each exploration step, the RL agent selects the position and scale of the next glimpse based on the information about previous patches, their coordinates, importance, and latent representations.

the environment $X$ and scaling it to $d_{\text{cam}} \times d_{\text{cam}}$. The model stores information about glimpses; therefore, at step $t$, it can access glimpses $G_1, G_2, ..., G_t$ and their corresponding coordinates. The exploration process is stopped when a set confidence level or a maximum number of glimpses is reached. As we will show in Sec. 3.3, this process can be formulated as a Markov Decision Process (MDP) to leverage RL methods [44].

### 3.2 Vision transformer with variable scale sampling

The architecture of our backbone network consists of two parts: an encoder based on a modified version of ViT [10] and a task-specific head (decoder). The goal is to perform the main task, e.g. classification or reconstruction, based on already observed glimpses while providing information for the RL agent network.

**Glimpse encoder.** At each step $t$, the ViT encoder is provided with a sequence of glimpses $G_1, G_2, ..., G_{t-1}$ and their coordinates $C_1, C_2, ..., C_{t-1}$. Depending on the $d_{\text{cam}}$ resolution relative to the ViT native patch size $d_{\text{patch}}$, each glimpse $G_i$ is divided with a standard sampling grid into a sequence of patches $G'_i = g'_{i,1}, ..., g'_{i,k}$ with coordinates $C'_i = c'_{i,1}, ..., c'_{i,k}$, where $k = (\lceil d_{\text{cam}}/d_{\text{patch}} \rceil)^2$. The resulting sequences of patches and coordinates from all previous glimpses are concatenated into $\widehat{G}_t$ and $\widehat{C}_t$, respectively.

The standard ViT positional embeddings assume that all patches are sampled from a regular grid. As our method relaxes this constraint, we apply ElasticViT [31] positional encoding calculated according to the patch coordinates. Finally, a trainable class token is appended to the sequence, which is then passed through ViT transformer blocks. The encoder outputs a sequence of latent tokens $\widehat{H}_t$, one per patch, and the class token. Furthermore, we estimate the importance of each input patch, calculating a transformer attention rollout [1], which results in a sequence $\widehat{I}_t$.

**Task-specific decoder.** The head of the classification model is a simple linear layer taking as an input the class token, as in standard ViT. However, for dense prediction tasks (e.g., reconstruction, segmentation), a MAE-like [18] transformer decoder is used. Then, the decoder receives a sequence of all tokens from the encoder and a full grid of mask tokens as input. The mask tokens consist of a positional embedding for each position in the decoder grid and a shared learnable query embedding, indicating that the token value is to be predicted. This is in contrast to MAE, which only uses mask tokens for unknown areas of the image. However, using tokens for all positions is essential in our case because with variable scale sampling we must predict the entire image rather than solely focus on the absent portions.

The decoder consists of a series of transformer blocks, generating output tokens projected through a linear layer for reconstruction or a progressive upscale module [52] of 4 convolutional layers and interpolations for segmentation. Finally, the output mask tokens are arranged according to a grid, and the remaining ones are discarded.

**Training objectives.** The entire backbone network is trained using a task-specific loss function. Optimization is performed only on the last exploration step $t = T$ after seeing all glimpses. The loss function for reconstruction is the root mean squared error (RMSE). For classification and segmentation, we use distilled soft targets computed by a teacher model and the Kullback-Leibler divergence as the loss function. The teacher model is a pre-trained ViT from [45] for classification and a DeepLabV3 [8] with a ResNet-101 backbone for segmentation. In both cases, the teacher model is provided with the entire scene $X$, as in STAM [35].

### 3.3 Soft Actor-Critic agent

We consider AVE as a Markov Decision Process (MDP), where at timestep $t$, the agent observing state $s_t$ (information about previous glimpses) takes action $a_t$ (coordinates of the next glimpse). It leads to state $s_{t+1}$ (information about previous glimpses and the next glimpse) as well as the reward $r_{t+1}$ (scalar estimating how much the glimpse helped in refining the prediction). Presenting AVE as MDP allows us to leverage reinforcement learning algorithms.

**Preliminaries.** The focal point of reinforcement learning is the policy $\pi_\theta$, which chooses the next action based on the current state, i.e., $a_t \sim \pi_\theta(s_t)$. The goal is to find the policy $\pi_\theta$ that maximizes the value function, corresponding to the expected discounted sum of rewards: $V^{\pi_\theta}(s) := \mathbb{E}_{\pi_\theta}\left[\sum_{t=k}^{T} \gamma^t r_t | s_k = s\right]$, where $\gamma = 0.99$ is the discount factor. The expectation is taken under the policy $\pi_\theta$, i.e., we use $\pi_\theta$ to take actions in the environment. As such, we aim to find $\theta^* = \arg\max_\theta V^{\pi_\theta}$. Additionally, in order to evaluate actions and facilitate the learning of the policy, we define the state-action value function

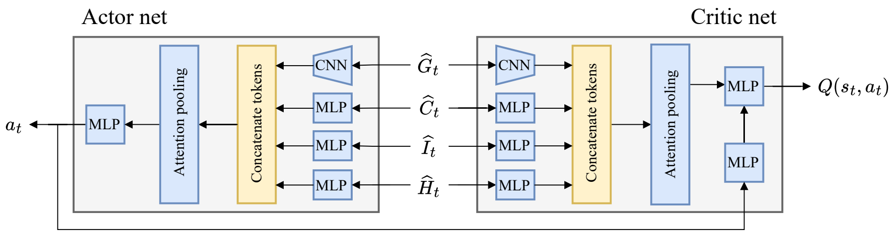

**Fig. 3: RL agent**: RL module of AdaGlimpse uses two networks: the actor and the critic. The actor predicts the action $a_t$ (position and scale of the next glimpse) based on state $s_t = (\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$. The critic estimates the $Q(s_t, a_t)$, corresponding to the expected cumulative reward for taking this action.

$Q^{\pi_\theta}(s,a) := r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi_\theta}(s')$, where $r(s,a)$ is the reward received in state $s$ when executing action $a$, and state $s'$ represents the next sampled state. Intuitively, it represents the expected return of our policy, that is, the value function of executing the action $a$ in the first step and choosing the subsequent actions according to the policy $\pi$. Below, we define observations, actions, rewards, the maximum-entropy objective central to the Soft Actor-Critic algorithm, and the design of our actor and critic architectures.

**State, action and reward.** To create the *state* for the RL agent, we supplement the glimpses $G_t$ with additional information to make the inference easier. In particular, the state $s_t$ consists of sequences $(\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$, where (as defined in Sec. 3.1 and 3.2):

- $\widehat{G}_t$ are all patches of the previously sampled glimpses,
- $\widehat{C}_t$ are coordinates of these patches,
- $\widehat{I}_t$ are their importance (one importance value per patch),
- $\widehat{H}_t$ are latent representations of these patches (one token per patch).

Notice that the starting state $s_0$ is an empty sequence corresponding to the fact that we do not know anything about the environment.

AdaGlimpse proposes continuous-valued *actions* that describe the arbitrary position and scale of an image. Therefore, the action is a tuple $(x, y, z) \in [0,1]^3$, where $x, y$ represents the normalized coordinates of the top-left glimpse corner and $z$ is its scale.

Finally, we define the *reward* as the difference between the loss in the successive timesteps, i.e. $r_t = L_{t-1} - L_t$.

**Soft Actor-Critic.** RL objective can be optimized with various approaches [44]. However, since exploration is crucial to solving AVE, we decided to use Soft Actor-Critic [16] (SAC) reinforcement learning algorithm. SAC operates in the maximum entropy framework, meaning that besides maximizing the expected sum of rewards, it also takes into account the entropy of the action distribution. That is, the goal is to optimize $V^{\pi_\theta}(s) := \mathbb{E}_\pi \left[\sum_{t=k}^{T} \gamma^t r_t + \alpha \mathcal{H}(\pi(s_t)) | s_k = s\right]$, where $\mathcal{H}$ is the entropy, and $\alpha$ is used to weigh its importance. Higher $\alpha$ values encourage the RL algorithm to find more exploratory policies, resulting in more diverse actions.

**Actor and Critic architectures.** AdaGlimpse requires two networks: the actor that encodes the policy $\pi$ and the critic that encodes the state-action value function $Q$. For this purpose, we build a custom-crafted architecture, see Fig. 3. In particular, we create separate token encoders for each part of the input $s_t$: a small convolutional network that processes patches in $\widehat{G}_t$, and a small MLP for each of the other inputs $\widehat{C}_t, \widehat{I}_t, \widehat{H}_t$. As a result, we obtain four embedding vectors for each patch. We concatenate and process them using an attention pooling [20] layer to combine information across patches. Finally, we use another MLP to obtain the action $a_t$ for the actor and the value prediction $Q(s_t, a_t)$ for the critic. Despite the similarity in the actor and critic architectures, we do not share any parameters between them as it destabilizes the training process.

## 4 Experimental Setup

**Architecture.** In all our experiments we use an encoder of the same size as standard ViT-B [10], i.e., 12 transformer blocks and embedding size of 768. The decoder consists of 8 blocks with the embedding size of 512. For the RL networks the hidden dimension size is 256, the number of attention heads is 8 and MLPs have 3 layers. All networks use GELU activation functions [19].

**Training.** We adopt the AdamW optimization algorithm [27], setting the weight decay value to $10^{-4}$ and the initial learning rate to $10^{-5}$ for classification and $10^{-4}$ for other tasks. The learning rate is then decayed using a half-cycle cosine rate to $10^{-8}$ for the remainder of the training. The model is trained for 100 epochs with early stopping. During training, we alternate between optimizing the backbone and the RL agent each epoch, except for the first 30 epochs when we train the RL agent only. We augment the training data using the 3-Augment regime proposed in [45] extended with a random affine transform for the segmentation target. Additionally, we pre-train the model for 600 epochs with 196 random glimpses per image with sizes and positions sampled from a uniform distribution. In segmentation experiments we fine-tune a model trained for reconstruction to accommodate for the relatively small size of the dataset.

**Datasets and metrics.** We assess our method on several publicly available vision datasets. ImageNet-1k [9] is used for classification and reconstruction tasks. The performance of zero-shot reconstruction is evaluated on MS COCO 2014 [26], ADE20K [53] and SUN360 [43]. Semantic segmentation is evaluated

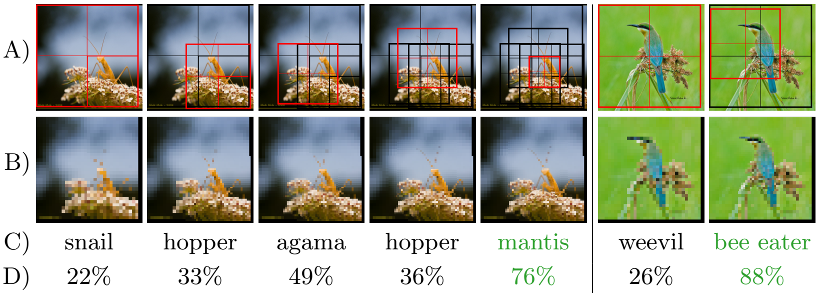

**Fig. 4: Glimpse selection step-by-step:** AdaGlimpse explores 224×224 images from ImageNet with $32 \times 32$ glimpses of variable scale, zooming in on objects of interest and stopping the process after reaching 75% predicted probability. The rows correspond to: A) glimpse locations, B) pixels visible to the model (interpolated from glimpses for preview), C) predicted label, D) prediction probability.

on the ADE20K dataset (the MIT scene parsing benchmark subset) [53]. Since the SUN360 dataset does not have a predetermined train-test split, we use a 9:1 train-test split according to an index provided by the authors of [39]. We report accuracy for classification tasks, root mean squared error (RMSE) for reconstruction, and pixel average precision (AP), class-mean average precision (mAP) and class-mean intersection over union score (mIoU) for segmentation.

**Glimpse regimes** We compare our model to baselines with different glimpse regimes, that can be categorized as follows: (a) simple - square glimpses with a fixed and constant resolution [21,32,35], (b) retinal - retina-like glimpses [38] with more pixels in the center than on the edges [32,39,41], (c) full+simple - one low-resolution glimpse of the entire scene followed by simple glimpses [3,12,30,37,46,47], and (d) adaptive - our variable scale glimpse regime.

For easy comparison of different glimpse regimes, we provide a *pixel percentage* metric representing the percentage of image pixels known to the model, as defined in [32], which is calculated as the number of pixels captured in all glimpses divided by the number of pixels in the full scene image.

## 5 Results

In this section, we present an evaluation of AdaGlimpse compared to competitive methods, followed by an analysis of our approach. Both quantitative and qualitative results for all baseline methods were taken from [32] for reconstruction and segmentation, and [35] for classification. Further results are provided in the supplementary materials.

An overview of glimpse selection performed by our model is portrayed in Fig. 4. The visualization demonstrates the arbitrary glimpse position and scale capabilities of our method. AdaGlimpse detects objects of interest and zooms in on them to extract fine details.

**Table 1: Reconstruction results:** RMSE (lower is better) obtained by our model for reconstruction task against AttSeg [41], GlAtEx [39], SimGlim [21], and AME [32] on ImageNet-1k, SUN360, ADE20K and MS COCO datasets. Regardless of the number of glimpses, as well as their resolution and regime (see Sec. 4), our method outperforms competitive solutions. Note that Pixel % denotes the percentage of image pixels known to the model, † a reproduced result not published in the relevant paper, and * zero-shot performance.

| Method | IMNET | SUN360 | ADE20k | COCO | Image res. | Glimpses | Regime | Pixel % |
|---|---|---|---|---|---|---|---|---|
| AME | 30.3† | 29.8 | 30.8 | 32.5 | 128 × 256 | $8 \times 32^2$ | simple | 25.00 |
| **Ours** | **14.5** | **11.1***  | **14.0*** | **14.5*** | 224 × 224 | $12 \times 32^2$ | adaptive | 24.49 |
| AttSeg | – | 37.6 | 36.6 | 41.8 | 128 × 256 | $8 \times 48^2$ | retinal | 18.75 |
| GlAtEx | – | 33.8 | 41.9 | 40.3 | 128 × 256 | $8 \times 48^2$ | retinal | 18.75 |
| AME | – | 23.6 | 23.8 | 25.2 | 128 × 256 | $8 \times 48^2$ | retinal | 18.75 |
| SimGlim | – | 26.2 | 27.2 | 29.8 | 224 × 224 | $37 \times 16^2$ | simple | 18.75 |
| AME | – | 23.4 | 26.2 | 28.6 | 224 × 224 | $37 \times 16^2$ | simple | 18.75 |
| **Ours** | **14.7** | **11.1*** | **14.2*** | **14.7*** | 224 × 224 | $9 \times 32^2$ | adaptive | 18.36 |
| AME | – | 37.9 | 40.7 | 43.2 | 128 × 256 | $8 \times 16^2$ | simple | 6.25 |
| **Ours** | **20.9** | **17.6*** | **20.5*** | **21.5*** | 224 × 224 | $12 \times 16^2$ | adaptive | 6.12 |
| **Ours** | **20.7** | **17.2*** | **20.7*** | **21.4*** | 224 × 224 | $3 \times 32^2$ | adaptive | 6.12 |

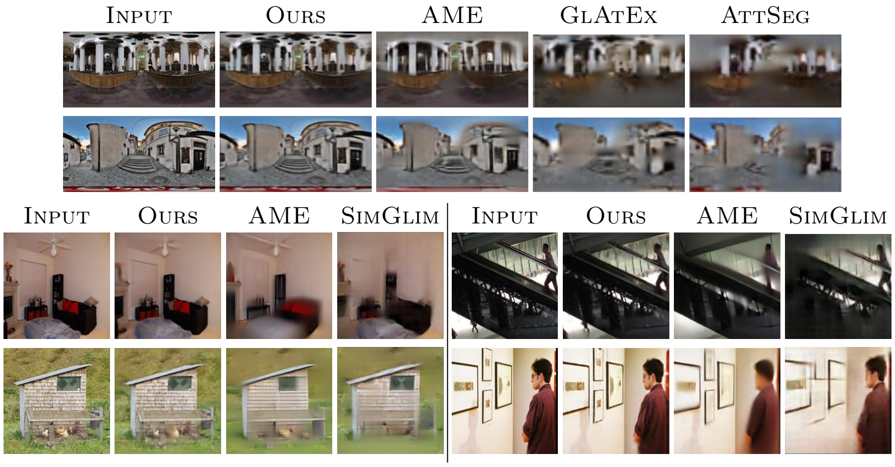

**Fig. 5: Reconstruction quality for SUN360 (top) and ADE20K (bottom):** Sample reconstructions of our method compared with AME [32], AttSeg [41], GlAtEx [39] and SimGlim [21] on the SUN360 and ADE20K datasets. Reconstructions done with our method are visibly more detailed and less blurry than those obtained by baseline methods. Notice that images for comparison were taken from the baseline publications (we did not select them).

**Table 2: Classification results**: Accuracy obtained by our model for classification task against DRAM [3], GFNet [47], Saccader [12], STN [37], TNet [30], PatchDrop [46] and STAM [35] on ImageNet-1k dataset. Our AdaGlimpse needs 40% less pixels to match the performance of the best baseline method. Note that Pixel % denotes the percentage of image pixels known to the model, while regimes are described in Sec. 4.

| Method | Accuracy % | Glimpses | Regime | Pixel % |
|---|---|---|---|---|
| DRAM | 67.50 | $8 \times 77^2$ | full+simple | 94.53 |
| GFNet | 75.93 | $5 \times 96^2$ | full+simple | 91.84 |
| Saccader | 70.31 | $6 \times 77^2$ | full+simple | 70.90 |
| TNet | 74.62 | $6 \times 77^2$ | full+simple | 70.90 |
| STN | 71.40 | $9 \times 56^2$ | full+simple | 56.25 |
| PatchDrop | 76.00 | $\sim 8.9 \times 56^2$ | full+simple+stopping | ~55.63 |
| STAM | 76.13 | $14 \times 32^2$ | simple | 28.57 |
| **Ours** | **77.54** | $14 \times 32^2$ | adaptive | 28.57 |
| **Ours** | **76.30** | $\sim 8.3 \times 32^2$ | adaptive+stopping | ~**16.94** |

### 5.1 Tasks

**Reconstruction.** Reconstructing the entire scene from observed glimpses verifies comprehensive scene understanding. In Tab. 1 we group the results by pixel percentage (fraction of pixels of the original input image known to the model). Our approach outperforms existing methods by a large margin. In particular, with only 6% of the pixels seen, AdaGlimpse performs better than other methods that had over 18% of the image visible to them. Note that unlike baseline methods, we train our model only on ImageNet-1k without fine-tuning for each evaluated dataset. Qualitative outcomes in Fig. 5 showcase AdaGlimpse producing reconstructions with a higher level of detail, reproducing more objects compared to baseline methods. All models except AttSeg have backbone networks pre-trained on ImageNet-1k; AttSeg's pre-training details were not disclosed.

**Classification.** Results for multi-class classification on the ImageNet-1k dataset can be found in Tab. 2. AdaGlimpse outperforms all prior methods, achieving 77.54% (±0.18) accuracy compared to 76.13% of the best baseline method STAM [35]. With early exploration termination after reaching 85% probability of predicted class, it requires over 40% less pixels to match STAM accuracy. Visualizations of glimpse selection for classification are presented in Fig. 4. With an early stopping probability threshold of 75%, it is able to classify 224 × 224 images using only a few 32 × 32 glimpses of 4 patches each.

**Segmentation.** The goal of the semantic segmentation task is to classify each pixel of the full scene based on the captured glimpses. The numerical results are presented in Tab. 3. AdaGlimpse outperforms both AttSeg [41] and GlAtEx [39] by a large margin. It performs on par with AME [32] in terms of accuracy;

**Table 3: Segmentation results:** Comparison of our model against AttSeg [41], GlAtEx [39], and AME [32] on the ADE20K dataset. Our method performs on pair with AME, but requires 35% less pixels, while outperforming other competitive methods on all considered metrics: Pixel-wise Accuracy (mPA, higher is better), Pixel-Accuracy (PA, higher is better), and Intersection over Union (IoU, higher is better).

| Method | PA % | mPA % | IoU % | Image res. | Glimpses | Regime | Pixel % |
|---|---|---|---|---|---|---|---|
| AttSeg | 47.9 | – | – | 128 × 256 | $8 \times 48^2$ | retinal | 18.75 |
| GlAtEx | 52.4 | – | – | 128 × 256 | $8 \times 48^2$ | retinal | 18.75 |
| Ours | **67.4** | **29.4** | **22.7** | 224 × 224 | $4 \times 48^2$ | adaptive | 18.36 |
| AME | 70.3 | 32.2 | 24.4 | 128 × 256 | $8 \times 48^2$ | simple | 56.25 |
| Ours | 70.0 | 32.8 | 25.7 | 224 × 224 | $8 \times 48^2$ | adaptive | **36.73** |

**Table 4: Importance of state components**: The RL state consists of the sequence $(\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$ as described in Sec. 3.3. For this study, we replaced each element with its mean value (averaged over the entire dataset) to see how important it is for the model. As a result, we observe that the transformer latent is the most informative part of the state, followed by glimpse coordinates.

| patches $\widehat{G}_t$ | coordinates $\widehat{C}_t$ | importance $\widehat{I}_t$ | latent $\widehat{H}_t$ | Accuracy % |
|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | 77.54 |
| ✗ | ✓ | ✓ | ✓ | 76.99 |
| ✓ | ✗ | ✓ | ✓ | 68.25 |
| ✓ | ✓ | ✗ | ✓ | 77.36 |
| ✓ | ✓ | ✓ | ✗ | 61.82 |

however, it requires 35% less information to achieve this result. Consistently, visualizations in Sec. 5 confirm the quality of the produced segmentation maps.

### 5.2 Analysis

**Percentage of image pixels.** The relationship between the percentage of the full image pixels known to the model (pixel %, see Sec. 4) and performance is plotted in Fig. 7. AdaGlimpse requires fewer pixels to perform better than the baseline methods. In particular for reconstruction, with only 5% of pixels it produces superior results to those achieved by competitive approaches when provided with 25% of scene pixels.

**Importance of RL state elements.** The key component of AdaGlimpse reinforcement learning algorithm is the state, which consists of the sequence $(\widehat{G}_t, \widehat{C}_t, \widehat{I}_t, \widehat{H}_t)$ as described in Sec. 3.3. In Tab. 4, we present the ImageNet-1k classification performance when omitting and replacing each component with

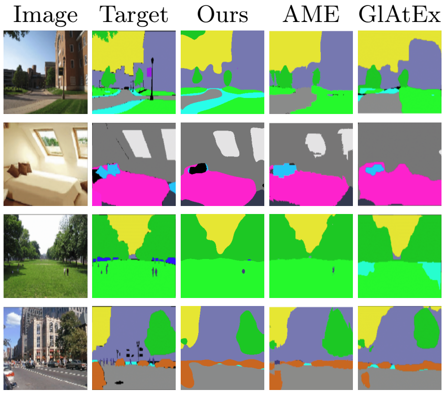

**Fig. 6: Segmentation qualitative results.** Sample semantic segmentation of our method compared with AME [32] and GlAtEx [39] on the ADE20k dataset. In terms of quality, the segmentation maps generated by our approach are at least comparable to those produced by competing methods.

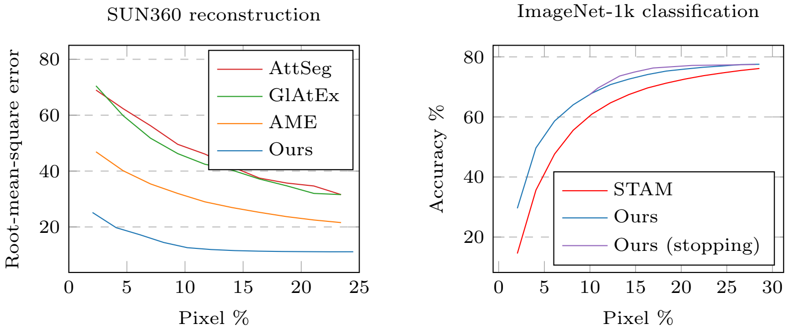

**Fig. 7: Percentage of image pixels observed**: Figures present the relationship between the amount of pixels observed by the model relative to the full scene resolution (pixel %), and its performance. AdaGlimpse outperforms competitive solutions, requiring significantly less information to achieve the same performance level.

its mean value during inference. The resulting lower accuracy proves the significance of each component. The transformer latent $\widehat{H}_t$ and glimpse positions $\widehat{C}_t$ are especially crucial for this task, highlighting both the importance of the glimpse location within the original scene and the benefit of using the processed input over the original image.

**Glimpse location.** In Fig. 8, we illustrate the average location of each subsequent glimpse, revealing a notable distinction between reconstruction and classification tasks explored by AdaGlimpse. In the reconstruction task, attention

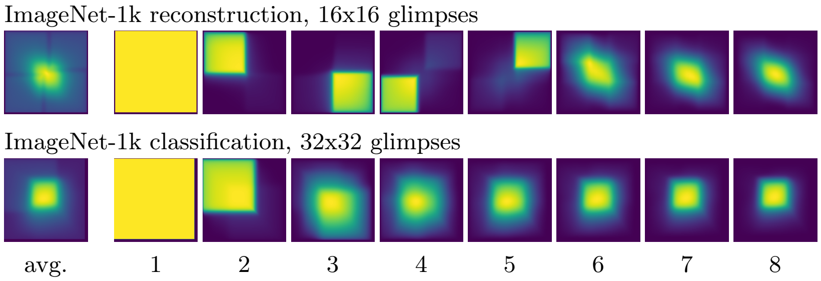

**Fig. 8: Average glimpse image:** Mean glimpse maps for models trained for reconstruction (top) and classification (bottom) averaged over all test images. On the left, an average map for all glimpses is presented, followed by maps for successive glimpses $t = 1, ..., 8$. One can observe that AdaGlimpse learns to select the entire image as the first glimpse for both tasks, but subsequent glimpse maps differ. Four successive glimpses in reconstruction concentrate on four parts of the image, while for classification, they mostly explore the center.

spans all image regions initially and later focuses on key elements. In contrast, in the classification task our model swiftly identifies crucial class-specific elements in the image center, directing attention those regions early on. Notably, both tasks uncover the well-known fact about ImageNet-1k, where key objects are concentrated around the image center [23].

## 6 Discussion and Conclusions

This paper presents AdaGlimpse, a novel approach to Active Visual Exploration, which enables the selection and processing of glimpses at arbitrary positions and scales. We formulate the glimpse selection problem as a Markov Decision Process with continuous action space and leverage the Soft Actor-Critic reinforcement learning algorithm, which specializes in exploration problems. Our task-agnostic architecture allows for a more efficient exploration and understanding of environments, significantly reducing the number of observations needed. AdaGlimpse can quickly analyze the scene with large low-resolution glimpses before zooming in on details for a closer inspection. Its success across multiple benchmarks suggests a broad applicability and potential for further development in embodied AI and robotics.

While excelling in exploration, AdaGlimpse is limited in performance by the underlying transformer architecture, which incurs quadratic computational cost relative to the number of sampled patches. A possible way to overcome this limitation is to replace it with a selective structured state-space model [14, 15]. Finally, although AdaGlimpse perform well on current benchmarks, they do not fully reflect the complexity of Active Visual Exploration, as they do not incorporate dynamic scenes that change over time. As such, further evaluations are required before real-life deployment.

## Acknowledgments

This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237. This research was funded by National Science Centre, Poland (grant no. 2023/49/N/ST6/02465, 2022/47/B/ST6/03397, 2022/45/B/ST6/02817, and 2023/50/E/ST6/00469). The research was supported by a grant from the Faculty of Mathematics and Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016558.

## References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. arXiv preprint arXiv:2005.00928 (2020)
2. Alexe, B., Heess, N., Teh, Y., Ferrari, V.: Searching for objects driven by context. Advances in Neural Information Processing Systems **25** (2012)
3. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. In: ICLR (2015)
4. Ba, J., Salakhutdinov, R.R., Grosse, R.B., Frey, B.J.: Learning wake-sleep recurrent attention models. Advances in Neural Information Processing Systems **28** (2015)
5. Beyer, L., Izmailov, P., Kolesnikov, A., et al.: Flexivit: One model for all patch sizes. arXiv:2212.08013 (2022)
6. Caicedo, J.C., Lazebnik, S.: Active object localization with deep reinforcement learning. In: Proceedings of the IEEE international conference on computer vision. pp. 2488–2496 (2015)
7. Chai, Y.: Patchwork: A patch-wise attention network for efficient object detection and segmentation in video streams. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3415–3424 (2019)
8. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE (2009)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
11. Double Robotics, Inc.: Double 3 - telepresence robot for the hybrid office. `https://www.doublerobotics.com/` (2024), accessed: 2024-02-24
12. Elsayed, G., Kornblith, S., Le, Q.V.: Saccader: Improving accuracy of hard attention models for vision. Advances in Neural Information Processing Systems **32** (2019)
13. Girdhar, R., Singh, M., Ravi, N., van der Maaten, L., Joulin, A., Misra, I.: Omnivore: A Single Model for Many Visual Modalities. In: ICCV (2022)
14. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
15. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021)

16. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. pp. 1861–1870. PMLR (2018)
17. Hayhoe, M., Ballard, D.: Eye movements in natural behavior. Trends in cognitive sciences **9**(4), 188–194 (2005)
18. He, K., Chen, X., Xie, S., et al.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022)
19. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
20. Ilse, M., Tomczak, J., Welling, M.: Attention-based deep multiple instance learning. In: International conference on machine learning. pp. 2127–2136. PMLR (2018)
21. Jha, A., Seifi, S., Tuytelaars, T.: Simglim: Simplifying glimpse based active visual reconstruction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 269–278 (2023)
22. Krotenok, A.Y., Yu, A.S., Yu, V.A.: The change in the altitude of an unmanned aerial vehicle, depending on the height difference of the area taken. In: IOP Conference Series: Earth and Environmental Science. vol. 272, p. 022165. IOP Publishing (2019)
23. Kümmerer, M., Theis, L., Bethge, M.: Deep gaze I: boosting saliency prediction with feature maps trained on imagenet. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings (2015)
24. Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., Gao, J.: Efficient self-supervised vision transformers for representation learning. arXiv preprint arXiv:2106.09785 (2021)
25. Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., Jia, J.: Mat: Mask-aware transformer for large hole image inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10758–10768 (2022)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
27. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
28. Mathe, S., Pirinen, A., Sminchisescu, C.: Reinforcement learning for visual object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2894–2902 (2016)
29. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. Advances in neural information processing systems **27** (2014)
30. Papadopoulos, A., Korus, P., Memon, N.: Hard-attention for scalable image classification. Advances in Neural Information Processing Systems **34**, 14694–14707 (2021)
31. Pardyl, A., Kurzejamski, G., Olszewski, J., Trzciński, T., Zieliński, B.: Beyond grids: Exploring elastic input sampling for vision transformers. arXiv preprint arXiv:2309.13353 (2023)
32. Pardyl, A., Rypeść, G., Kurzejamski, G., Zieliński, B., Trzciński, T.: Active visual exploration based on attention-map entropy. In: Elkind, E. (ed.) Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. pp. 1303–1311 (8 2023), main Track
33. Ramakrishnan, S.K., Jayaraman, D., Grauman, K.: An exploration of embodied visual exploration. International Journal of Computer Vision **129**, 1616–1649 (2021)

34. Rangrej, S.B., Clark, J.J.: A probabilistic hard attention model for sequentially observed scenes. arXiv preprint arXiv:2111.07534 (2021)
35. Rangrej, S.B., Srinidhi, C.L., Clark, J.J.: Consistency driven sequential transformers attention model for partially observable scenes. In: CVRP. pp. 2518–2527 (2022)
36. Ranzato, M.: On learning where to look. arXiv preprint arXiv:1405.5488 (2014)
37. Recasens, A., Kellnhofer, P., Stent, S., Matusik, W., Torralba, A.: Learning to zoom: a saliency-based sampling layer for neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 51–66 (2018)
38. Sandini, G., Metta, G.: Retina-like sensors: motivations, technology and applications. In: Sensors and sensing in biology and engineering, pp. 251–262. Springer (2003)
39. Seifi, S., Jha, A., Tuytelaars, T.: Glimpse-attend-and-explore: Self-attention for active visual exploration. In: ICCV. pp. 16137–16146 (2021)
40. Seifi, S., Tuytelaars, T.: Where to look next: Unsupervised active visual exploration on 360° input. CoRR **abs/1909.10304** (2019), `http://arxiv.org/abs/1909.10304`
41. Seifi, S., Tuytelaars, T.: Attend and segment: Attention guided active semantic segmentation. In: European Conference on Computer Vision. pp. 305–321. Springer (2020)
42. Śmieja, M., Struski, Ł., Tabor, J., Zieliński, B., Spurek, P.: Processing of missing data by neural networks. Advances in neural information processing systems **31** (2018)
43. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015)
44. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
45. Touvron, H., Cord, M., Jégou, H.: Deit iii: Revenge of the vit. In: European Conference on Computer Vision. pp. 516–533. Springer (2022)
46. Uzkent, B., Ermon, S.: Learning when and where to zoom with deep reinforcement learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12345–12354 (2020)
47. Wang, Y., Lv, K., Huang, R., Song, S., Yang, L., Huang, G.: Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. Advances in Neural Information Processing Systems **33**, 2432–2444 (2020)
48. Wenzel, P., Wang, R., Yang, N., Cheng, Q., Khan, Q., von Stumberg, L., Zeller, N., Cremers, D.: 4seasons: A cross-season dataset for multi-weather slam in autonomous driving. In: Akata, Z., Geiger, A., Sattler, T. (eds.) Pattern Recognition. pp. 404–417. Springer International Publishing, Cham (2021)
49. Wu, C.Y., Girshick, R., He, K., Feichtenhofer, C., Krahenbuhl, P.: A multigrid method for efficiently training video models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 153–162 (2020)
50. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057. PMLR (2015)
51. Yoo, D., Park, S., Lee, J.Y., Paek, A.S., So Kweon, I.: Attentionnet: Aggregating weak directions for accurate object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2659–2667 (2015)

52. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6881–6890 (2021)
53. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)