Visual Hard Attention Models Under Partial Observability

Samrudhdhi B Rangrej

Doctor of Philosophy

Department of Electrical & Computer Engineering
McGill University
Montreal, Quebec, Canada

March 2023

A thesis submitted to McGill University in partial
fulfillment of the requirements of the degree of
Doctor of Philosophy

©Samrudhdhi B Rangrej, 2023

Abstract

Existing state-of-the-art recognition models achieve impressive performance but require a complete scene which may not always be available. For example, sensing a complete scene at once is infeasible in applications such as aerial imaging. Further, in applications such as disaster recovery, imaging devices should be light, inexpensive, and energy-efficient; thus, they are often built using small field-of-view cameras that capture only a part of a scene at a time. In the above cases, the imaging devices must scan the area sequentially. Moreover, they must also prioritize the scanning of informative subregions for timely recognition.

Many developed attention models that recognize a scene by observing it through small informative subregions called glimpses. However, most models locate informative glimpses by glancing at a low-resolution gist of a complete scene, which is unavailable in practice. In this thesis, we develop sequential recognition models that locate and attend to informative glimpses without assessing a complete scene. Our sequential attention models predict the location of the next glimpse based solely on past glimpses. Our models achieve effective attention policies under partial observability by selecting subsequent glimpses that, combined with past glimpses, help the most in reasoning about the complete scene.

We present three attention models, two for spatial and one for spatiotemporal recognition. The first is a *Probabilistic Attention Model (PAM)*. PAM uses Bayesian Optimal Experiment Design to attend to a glimpse with maximum expected information gain (EIG). It synthesizes features of the complete scene from past glimpses to estimate the EIG for yet unobserved regions. The second is a *Sequential Transformers Attention Model (STAM)*, which employs the one-step actor-critic algorithm to attend to a sequence of glimpses that produce class distribution consistent with the one produced using a complete scene. The third is a *Glimpse Transformer (GliTr)*. GliTr learns an effective attention mechanism for online action recognition by selecting glimpses with features and class distribution consistent with the corresponding complete video frames.

Throughout the thesis, we evaluate our models on multiple datasets and compare them with existing models. Our two key findings are as follows. First, reasoning about the complete scene from partial observations helps in learning an effective attention policy under partial observability. Second, while reducing the amount of sensing required for recognition, our glimpse-based models achieve comparable or higher performance than the existing models that require complete scenes. The key takeaway is that one can attain good performance even using low-cost sensing devices and non-ideal imaging by automating the sensing process and compelling the recognition model to fill in the missing information.

Abrégé

L'état de l'art sur les modèles de reconnaissance a atteint une performance impressionnante. Ces modèles ont besoin d'une vue complète de la scène pour qu'ils puissent bien fonctionner, ce qui n'est pas toujours disponible. Par exemple, la détection et l'acquisition d'une scène complète est impossible dans les applications telles que l'imagerie aérienne. De plus, dans des applications telles que la reprise après sinistre, les dispositifs d'imagerie doivent être légers, peu coûteux et économes en énergie. C'est pour ces raisons que ces matériels sont souvent construits avec des caméras à petit champ de vision, qui ne capturent qu'une partie de la scène, à la fois et d'une manière séquentielle. En outre, ces dispositifs doivent donner la priorité à la numérisation et capture des sous-régions plus informatives pour une reconnaissance efficace et plus rapide.

Plusieurs méthodes d'attention ont été développées qui reconnaissent une scène tout en observant des petites sous-régions informatives appelées aperçus (glimpses en anglais). Cependant, la plupart de ces modèles localisent des aperçus informatifs en regardant à l'essentiel à basse résolution d'une scène complète, qui n'est pas disponible dans la pratique. Dans cette thèse, nous développons des modèles de reconnaissance séquentielle qui localisent et assistent à des aperçus informatifs sans évaluer une scène complète.

Nos modèles d'attention séquentielle prédisent l'emplacement du prochain aperçu en se basant uniquement sur les aperçus passés. Nos modèles réalisent des politiques d'attention efficaces sous observabilité partielle en sélectionnant des aperçus ultérieurs qui, combinés à des aperçus passés, aident le plus à raisonner sur la scène complète.

Nous présentons trois modèles d'attention: deux pour la reconnaissance spatiale et un pour la reconnaissance spatio-temporelle. Le premier est le modèle d'attention probabiliste (Probabilistic Attention Model PAM, en Anglais). PAM utilise la conception d'expérience optimale bayésienne pour assister à un aperçu avec un gain d'information maximal attendu (Expected Information Gain, EIG, en anglais). Ceci synthétise les caractéristiques de la scène complète à partir d'aperçus passés pour pouvoir estimer l'EIG pour les régions encore non observées. Le second est le modèle d'attention des transformateurs séquentiels (Sequential Transformers Attention Model, STAM, en anglais), qui utilise l'algorithme acteur-critique en une étape pour assister à une séquence d'aperçus qui produisent une distribution de classe cohérente avec celle produite à l'aide d'une scène complète. Le troisième est appelé Glimpse Transformer (GliTr). GliTr apprend un mécanisme d'attention efficace pour la reconnaissance d'action en ligne en sélectionnant des aperçus avec des caractéristiques et une distribution de classe cohérentes avec les images vidéo complètes correspondantes.

Tout au long de la thèse, nous évaluons nos modèles sur plusieurs jeux de données et nous les comparons avec des modèles existants. Nos conclusions principales sont les suivantes: Premièrement, le raisonnement sur la scène complète à partir d'observations partielles aide à apprendre une politique d'attention efficace sous observabilité partielle. Deuxièmement, tout en réduisant la quantité de détection requise pour la reconnaissance, nos modèles basés sur l'aperçu atteignent des performances comparables ou même supérieures à celles des modèles existants qui nécessitent des scènes complètes. Enfin,

on peut atteindre de bonnes performances même en utilisant des dispositifs de détection à faible coût et non-optimisés pour l'imagerie grâce à l'automatisation du processus de détection et la capacité du modèle à remplir les informations manquantes.

Acknowledgements

I am enormously grateful to my doctoral supervisor Prof. James Clark for taking an unproven student like me under his wings. I cannot begin to imagine or thank him enough for the silent efforts he had to make to support me while I was following my research interests. I am also indebted to Prof. Tal Arbel for her support through these years and for generously introducing me to Tal Hassner. Immense thanks to Tal Hassner for his mentorship. Also, my deepest gratitude to my co-authors and excellent friends, Kevin Liang and Chetan Srinidhi, who taught me much about research and helped me achieve clarity.

I also wish to express my gratitude to Prof. James Clark, Prof. Tal Arbel, and Prof. Dave Meger for serving on my supervisory committee. I also thank Prof. Kaleem Siddiqi and Prof. Neil Bruce for serving as my thesis examiners. I thank Prof. Kaleem Siddiqi, Prof. Tal Arbel, and Prof. Aaron Courville for serving on my defense committee. I thank the above and the anonymous reviewers for their time and effort in reviewing my work and providing valuable feedback.

I cannot thank Ibtihel and Zahra enough for spending countless hours discussing ideas, publishing our research, and lending an ear in challenging times. Special thanks to

Ibtihel for graciously translating the abstract of this thesis from English to French. I am also grateful to my fellow labmates for joining numerous group meetings and hearing me speak about the work I find interesting. I wish to convey my thanks to Karthik, Raghav, Sukesh, Tabish, Nehal, Shruti, Rutuja, Sumedh, Jay, Karen, Dhruv, and Riya for filling these years with fun and memories. Also, thanks to the CIM staff, Nick, Jan, Chelsea, and Marlene, for system and administrative support.

Finally, I want to thank my entire family. It was their vision that brought me to where I am today. I am grateful to my loving parents, Bharat and Shilpa, and my dearest sister, Ridhdhi, whose unwavering support helped me pursue research. I am thankful to Kavita, Pradeep, and Krishna for wholeheartedly welcoming me into their family and supporting my endeavors. I am also grateful to Dhruv and Nisarg for joining our family and sharing this collective pursuit. Lastly, I am forever indebted to Harshal, who has remained my dominant source of inspiration for the last fifteen years, first as my dearest friend and now as my incredible husband.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

BOED Bayesian Optimal Experiment Design

EIG Expected Information Gain

ELBO Evidence Lower Bound

FOV field of view

GliTr Glimpse Transformers

MHA Multi-Head Attention

MLP Multi-Layer Perceptron

NF Normalizing Flows

PAM Probabilistic Attention Model

POMDP Partially Observable Markov Decision Process

STAM Sequential Transformers Attention Model

STN Spatial Transformer Network

VAE Variational Autoencoder

ViT Vision Transformers

Introduction & Background

1

Introduction

Have you seen a blowfly escape a predator approaching it from behind? *Many times.* Right? Blowflies have immobile eyes with a nearly 360° field of view (FOV), allowing them to see predators approaching them from almost any direction and plan an escape covertly. Processing such a large amount of visual stimuli also comes with a high cost. Vision consumes nearly 8% of the blowfly's total resting metabolism [109, 129]. High energy costs led many insects and animals to develop eyes with acute vision in a small region. To compensate for the limited field of view, these organisms use eye movements

and head movements to orient their gaze in a specific direction [179, 66]. Eye and head movements consume less energy than moving the entire body [106]. Further, the low FOV eyes are also lightweight. While blowfly's eyes weigh nearly 6% of their total mass [108], human eyes consume only $\sim 0.2\%$ of the body mass [11].

Synthetic cameras follow the same principle as biological eyes. The small FOV cameras with low-resolution image sensors are lightweight, energy efficient, and require fewer computations [84]. However, the small FOV and low resolution become limiting factors in many applications. In this thesis, we explore the approach used in nature and design models that emulate eye movements to scan the scene by deploying these cameras serially.

1.1 Problem Statement

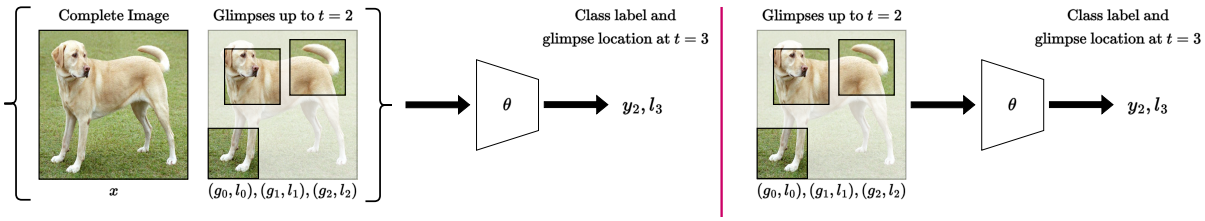
In applications such as monitoring at-risk animals and performing rescue operations, we often need mobile devices to explore remote and obscure locations. Building these devices using small FOV cameras keeps them light and energy efficient. The low cost of these cameras is an additional advantage since there is a significant risk of losing a mobile device in such operations. Moreover, when these devices communicate with the control unit on the ground, small and low-resolution images reduce the amount of information that needs to be transmitted. While we purposefully invite a limited field of view in the above situation, in some applications, mobile devices naturally have access to only a narrow view due to the vastness and physical constraints or occlusion. Examples include aerial imaging for disaster recovery and indoor robotics for human assistance.

The above mobile devices compensate for the limited or partial view by shifting their attention to various regions according to some strategy. One such strategy would be to observe neighboring areas in raster scan order. While this strategy is simple and easy to implement, it is lengthy and thus unsuitable for time-sensitive applications. Also, not

all views are equally crucial for the task. It is rather prudent to follow a strategy that prioritizes the most informative regions. To implement this strategy, we need to sequentially predict the location of the next most informative area based on past observations. Further, we must learn to perform a task from incomplete or partially observed scenes.

Existing image recognition models such as EfficientNet [166], ResNet [71], and Vision Transformers (ViT) [49] assume that we will have access to the complete scene when deployed in the real world. Even the state of the art video action recognition models such as TSM [116], Swin-B [119], or VideoMAE [171] assume we have a complete video at the test time. The performance of these models degrades significantly when presented with spatially or temporally incomplete information. For example, the performance of Swin-B drops by ∼30% on the Something-Something-v2 (SSv2) dataset [62] when only the first 70% frames are observed [163]. Similarly, the accuracy of DeiT-S[172] drops by around 10% on ImageNet [149] when 50% of the image regions are unavailable at random. Now, suppose we pass the top 50% *useful* patches to DeiT-S. In that case, the accuracy is hardly affected [127]. The experiment suggests that the model could perform well even with partial scenes if we observe the scenes tactfully. But how can we come up with an appropriate observation strategy? The above models cannot autonomously predict the locations of the useful patches and cannot provide a good strategy.

Many researchers developed agents that autonomously acquire a series of narrow but most informative subregions from a scene [125, 12, 51, 135, 17, 75, 122, 186, 188]. They call these narrow subregions 'glimpses'. Most existing scalable approaches initially glance at an entire scene to locate useful glimpses (see Figure 1.1(a)). While these models process the complete scene at low resolution or using a lightweight model, they still require a global view. As mentioned earlier, a global view may not be available or comes at a high cost in certain applications. Further, in the case of videos, many glimpse-based attention



(a) (b)

Figure 1.1: *Glimpse-based sequential attention models.* At time t, models observe a new (*i.e.*, previously unseen) glimpse-location pair (g_t, l_t) and use the observation history to predict label y_t and next glimpse location l_{t+1}. (a) Previous attention models also observe complete global view x to locate informative glimpses. (b) Our attention models never observe complete scenes; they rely exclusively on local views from past glimpses to make the above predictions. Unlike previous models, our models assess the informativeness of a glimpse *before* observing them.

models are offline. They wait until all video frames are available and assess the complete video to locate informative glimpses after the action has concluded. Thus, the existing models are unhelpful for applications requiring online decision-making based on partial information.

In this thesis, we develop autonomous agents that predict informative glimpse locations without observing the entire scene, therefore obviating the need for high-resolution, large FOV cameras. Starting from a glimpse at a given location, the autonomous agents decide which location to attend to next solely based on the previously observed glimpses. Consequently, our models predict labels using only local information. They never observe the complete global view (see Figure 1.1(b)). Further, the models predict labels online; they start predicting labels based on one glimpse and update their predictions as new glimpses become available. Due to their online nature, our models are capable of early exit; they can stop sensing further glimpses when they are sufficiently confident in their predictions.

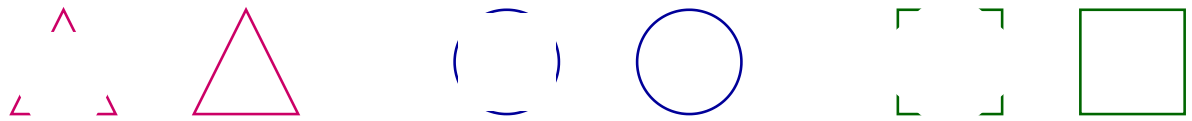


Figure 1.2: *Examples illustrating Law of Prägnanz.* We inherently connect salient fragments (left) and perceive them as complete shapes (right).

1.2 Approach

The key difference between our problem and previous works is *partial observability*. In this section, we discuss our approach to overcome the lack of visual evidence in decision-making.

Humans observe a fragmented scene as a whole. In Gestalt psychology, this phenomenon is known as the *Law of Prägnanz* [97] (see Figure 1.2). Further, we assume a *simplest* explanation when we perceive a scene as a whole, and often the simplest explanation is the most plausible one. In probability theory, Occam's razor also favors a simple explanation over a complex one. But what is considered 'simple'? In this context, uniform or perfectly articulated patterns are considered simple. These are the patterns we encounter routinely and are most familiar to us.

Motivated by this phenomenon, we develop methods that compel our attention models to observe glimpses and reason about the complete scene. Since we train our models on natural image datasets, they implicitly gain the ability to incorporate well-articulated and frequent patterns in the reasoning process. However, note that our end goal is to perform a given task and not the scene completion. Thus it is not necessary to reason about every detail of the scene. In fact, we let our models figure out only the task-specific information from the complete scene.

To further explain our motivation for task-specific reasoning, let us discuss an interesting eye-tracking study by Alfred L. Yarbus [197]. Yarbus presented human subjects with an image of 'An Unexpected Visitor' and asked them to assess the material circumstances of a family. In response, the subjects mainly examined clothing and furniture. When the same subjects were asked to predict the ages of people in the picture, they focused on people's faces and hardly looked at their clothing or furniture. The study suggests that the importance of various regions changes with the task and depends on the amount of task-relevant information they carry.

Therefore, we mainly focus on task-relevant details when compelling our models to reason about a complete scene. We ask our models to reproduce a task-specific latent representation of the scene given the glimpses. We consider a latent representation of the complete scene predicted by deep layers of a pre-trained neural network as the target for the above task. Note, while shallow layers of a network extract task-agnostic features such as color and texture, deep layers identify high-level concepts such as dog, wheel, etc [201]. This improved reasoning allows our models to understand the relationship between the seen and the unseen objects in the context of a given task and not distress over the exact appearances of these objects. To understand this further, let us consider an example task of identifying indoor vs. outdoor scenes. When our models observe a glimpse from the grass area, it is sufficient for them to acknowledge the co-occurrence of clouds in the complete scene. We do not require the models to generate pixels depicting the exact color, shape, and texture of the clouds.

Reasoning about the complete scene from glimpses is helpful for our models in two ways. First, it enables our models to recognize the scene better under partial observability. Second, it encourages our models to observe glimpses that are most informative of the complete scene, thus learning an effective attention policy. The intuition is that

our models implicitly learn to relate the observed but ambiguous regions with the unobserved yet unambiguous regions, and they incorporate the learned semantic and spatial relationship in decision-making and attention, respectively.

1.3 Thesis Overview

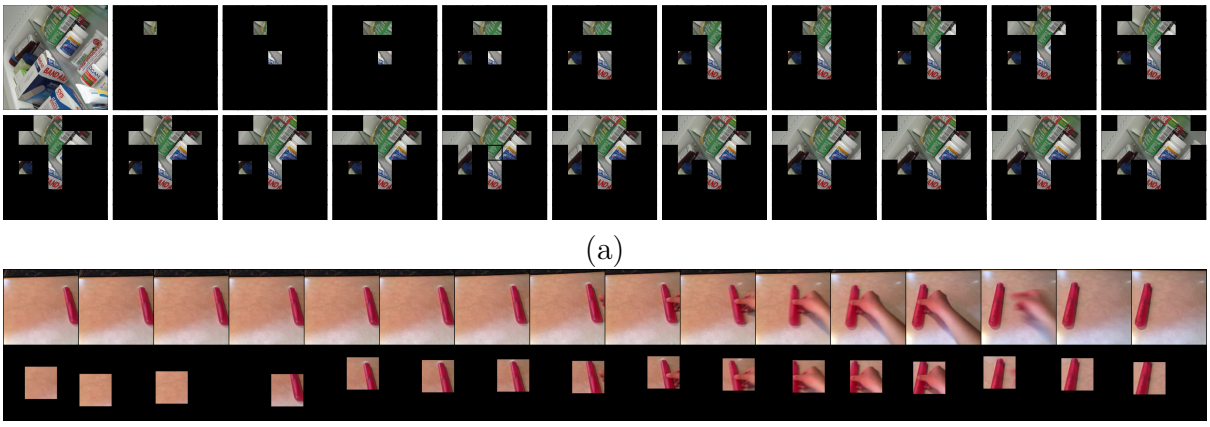
Our goal is to learn attention patterns akin to biological eye movements. Thus, we consider a stationary camera with *limited* pan and tilt motion. To simplify the task further and use existing open-source vision datasets, we mimic the limited camera movement by shifting the glimpse location in a two-dimensional plane within the bounds of images and video frames. This approach also models cases where an agent with an immobile camera moves in two dimensions to explore the scene. One example is an aerial device flying at a specific height and capturing a series of narrow top views to recognize a landmark.

The objective of this thesis is to develop attention models for scenes that are observed sequentially, either due to hardware or physical constraints. This is a lofty goal. Attention depends on the task; recall findings from Yarbus's experiments. Investigating attention for numerous vision tasks is beyond the scope of this thesis. Rather, we focus on studying attention for the visual recognition tasks.

We develop separate attention models for static and dynamic scenes. We focus on image classification for the static case and online action prediction for the dynamic case.

Attention behaves differently in the two cases. When a scene is static (*i.e.*, a single image), an agent uses attention to explore objects and entities in the scene and to assess their relationships (see Figure 1.3(a)). On the other hand, when the scene consists of moving objects or dynamic entities (*i.e.*, a video), an agent uses attention to locate a moving object and then maintain a steady gaze as it moves (see Figure 1.3(b)).

We develop three attention models, two for image recognition and one for online action



(a)

(b)

Figure 1.3: *Attention for static and dynamic scenes.* We show glimpses observed by our models. Our models do not observe complete scenes; complete scenes are shown for reference only. (a) Our STAM operates on a static scene. STAM explores the scene to assess various objects and their relationships. Time increases in raster scan order. (b) Our GliTr operates on a dynamic scene. GliTr explores the scene to locate an object of interest (first four glimpses), then maintains a steady gaze on the moving object (remaining glimpses). Time increases from left to right.

recognition. The two models for image recognition are the *Probabilistic Attention Model (PAM)* and the *Sequential Transformers Attention Model (STAM)*. The model for online action recognition is *Glimpse Transformers (GliTr)*. PAM and STAM observe multiple glimpses from a single image sequentially. GliTr observes one glimpse per frame from a live-streaming video. PAM is designed using Convolution Neural Networks (CNN) and Recurrent Neural Networks (RNN), whereas STAM and GliTr are designed using transformers.

The above attention models use different techniques to explain the complete scene from glimpses. Given a set of past glimpses, PAM and STAM reproduce features (a 2D map) and class distribution of a complete scene, respectively. GliTr observes glimpses from a live video and reconstructs frame features (1D vectors) and class distribution of a

spatially complete preliminary video. Since learning a generative model to produce large 2D feature maps requires massive computing and intensive training, we train PAM on datasets with small images of size 32×32 or 64×64. On the other hand, STAM and GliTr generate low dimensional quantities (*i.e.*, class distributions and feature vectors); hence, they scale to datasets with a standard image size of 224×224.

We train all models using gradient backpropagation for optimizing training objectives related to i) explaining the complete scene and ii) classification. Moreover, since PAM and STAM select glimpses using non-differentiable cropping, they also use separate mechanisms to learn attention policies. PAM decides on an attention policy for a given scene by performing Bayesian Optimal Experiment Design (BOED) on the generated feature maps; it selects glimpses with maximum expected information gain. STAM uses reinforcement learning to attend to the glimpses that help the most in reproducing the class distribution of the complete scene. Different from the others, GliTr uses a Spatial Transformer Network (STN) to crop glimpses in a differentiable manner and learns attention through backpropagation. It learns to attend to glimpses that are most informative of the current state of the scene during an ongoing action.

1.4 Contributions

This thesis looks at *a unique* problem of developing attention for partially and sequentially observable scenes. The work presented in this thesis has undergone peer review and has been published at prominent venues. We list the publications related to the three attention models and their respective contributions below.

[142] *A Probabilistic Hard Attention Model for Sequentially Observed Scenes.*
Samrudhdhi B Rangrej and James J Clark.
British Machine Vision Conference, 2021.

(The author of this thesis led the conceptualization, implementation, and evaluation of the method under the guidance of James J Clark. The author has open-sourced the corresponding codebase. The author also created the illustrations and played a major role in writing the paper.)

Contributions:

- We develop an attention model called PAM to classify images using a series of partial observations. Unlike previous approaches, PAM never observes a complete scene. At each time step, it estimates the EIG of the yet unobserved locations and attends to a location with maximum EIG.
- To estimate the EIG of unobserved regions, PAM *synthesizes* their content from the observed regions. We improve the efficiency of our PAM by synthesizing the content in the feature space. Further, we use normalizing flows to capture the multi-modality in inferring complete scenes from partial observations.
- When tested on five public datasets, PAM achieves 2-10% higher accuracy than the baseline methods when both have seen only a couple of glimpses. While a CNN achieves $\sim$90% accuracy on CIFAR-10 with complete images, our PAM achieves $\sim$80% accuracy after observing only $<$50% of the total image area.

[141] *Consistency driven sequential transformers attention model for partially observable scenes.*
Samrudhdhi B Rangrej, Chetan L Srinidhi, and James J Clark.
IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

(Under the supervision of James J Clark, the author of this thesis conceptualized the method jointly with Chetan L Srinidhi. The author implemented and evaluated the method under the guidance of Chetan L Srinidhi and James J Clark. The author has open-sourced the project's codebase. The author also played a significant role in creating the illustrations and writing the paper.)

Contributions:

- We develop a transformers-based RL agent called STAM, which actively senses glimpses from a scene and predicts class-label based on partial observations. Instead of locating informative glimpses by observing an entire image, our STAM sequentially predicts the next most informative glimpse location based on past glimpses.

- We propose a consistency-based training objective where STAM must predict a class distribution consistent with the complete image using only partial observations. With only 4% of the total image area observed, our proposed objective yields ∼3% and ∼8% gain in accuracy on ImageNet and fMoW, respectively.

- Our STAM that never observes a complete image outperforms previous methods that initially glance at an entire image to locate informative glimpses. It starts exceeding the previous state-of-the-art while sensing 27% and 42% fewer pixels in glimpses on ImageNet and fMoW, respectively.

[140] *GliTr: Glimpse Transformers with Spatiotemporal Consistency for Online Action Prediction.*
Samrudhdhi B Rangrej, Kevin J Liang, Tal Hassner and James J Clark.
IEEE/CVF Winter Conference on Applications of Computer Vision, 2023.

(The author of this thesis led the conceptualization, implementation, and evaluation of the method under the guidance of Kevin J Liang, Tal Hassner, and James J Clark. The author open-sourced the corresponding codebase. The author also created the illustrations and played a key role in writing the paper.)

Contributions:

- We develop GliTr, an online action prediction model that observes only glimpses and predicts ongoing action based on partial spatiotemporal observations. While previous work locates glimpses by first observing full frames, GliTr predicts the next informative glimpse location solely based on the past glimpses.

- We propose a novel spatiotemporal consistency objective to train GliTr without the ground truth for glimpse location. Under this objective, GliTr must select glimpses that summarize features and class distribution predicted from the entire frames. Our proposed consistency yields ∼10% gain in accuracy on SSv2 compared to the baseline cross-entropy objective.

- Our GliTr that never observes complete frames and recognizes action solely based on local information gathered through glimpses achieves nearly 53% and 94% accuracy on the SSv2 and Jester datasets, respectively, while reducing the total area observed per frame by nearly 67%.

Contributions of the Author: We present the details of the above contributions in Chapters 4, 5, and 6. The material presented in the remaining chapters, including the tables and the figures, is prepared by the thesis author under the guidance of Prof. James J Clark.

1.5 Materials from Published Works

While this is not a manuscript-based thesis, it contains considerable material from the following papers. Note that the author of this thesis is the first author of these works.

[142] *A Probabilistic Hard Attention Model for Sequentially Observed Scenes.*
Samrudhdhi B Rangrej and James J Clark.
British Machine Vision Conference, 2021.

[141] *Consistency driven sequential transformers attention model for partially observable scenes.*
Samrudhdhi B Rangrej, Chetan L Srinidhi, and James J Clark.
IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

[140] *GliTr: Glimpse Transformers with Spatiotemporal Consistency for Online Action Prediction.*
Samrudhdhi B Rangrej, Kevin J Liang, Tal Hassner and James J Clark.
IEEE/CVF Winter Conference on Applications of Computer Vision, 2023.

Permissions

BMVA (for [142]): The authors hold the copyright in BMVC papers in every instance [169]. While BMVA (the publisher of the proceedings) holds copyright over the collection, the authors may use the papers they have authored.

IEEE (for [141] and [140]): The IEEE does not require individuals working on a thesis to obtain a formal reuse license [57]. However, it requires that the thesis author follows these requirements:

- The author must give full credit to the source with proper referencing.

- The author must include the IEEE copyright notice for all figures and tables.

We respect the above requirements and cite the source articles in the thesis at the beginning of the specific chapters. We also include the copyright notice in all figures and tables.

1.6 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we provide a brief overview of a few techniques used in this thesis. In Chapter 3, we review the existing literature. Chapters 4, 5 and 6 present our attention models, which cover the original contribution of the thesis. We divide these chapters into two parts. Part I consists of spatial attention models: our Probabilistic Attention Model (PAM) (Chapter 4) and our Sequential Transformers Attention Model (STAM) (Chapter 5). Part II focuses on a spatiotemporal attention model: our Glimpse Transformers (GliTr) (Chapter 6). We conclude with a discussion of our results and directions for future work in Chapter 7.

2

Background

Let us discuss several architectures and algorithms employed in developing our attention models. With the aim of reviving the theoretical understanding of these frameworks, here we study them without going into much detail about their application in attention models. During the discussion, we only briefly mention how we use the introduced models and techniques in the rest of the thesis. We will extend them in the context of attention in Chapters 4, 5, and 6. When discussing our attention models later in the thesis, we shall utilize the concepts covered in this background chapter. We provide a list of topics

Topic	Section	Model	Our usecase
Variational Autoencdoer (VAE)	2.1	PAM (Chapter 4)	generates feature map of complete scene given glimpses
Normalizing Flows (NF)	2.2	PAM (Chapter 4)	captures multi-modal posterior in VAE
Bayesian Optimal Experiment Design (BOED)	2.3	PAM (Chapter 4)	decides the location of the next glimpse
One-step Actor-Critic	2.4	STAM (Chapter 5)	learns attention policy
Student-Teacher training paradigm	2.5	STAM (Chapter 5) GliTr (Chapter 6)	compels models to produce features and class distributions of complete scenes from glimpses
Spatial Transformer Networks (STN)	2.6	GliTr (Chapter 6)	extracts glimpses and corresponding position embeddings in a differentiable manner
Transformers	2.7	STAM (Chapter 5) GliTr (Chapter 6)	forms the base architecture of our attention models

Table 2.1: *Topics covered in Chapter 2.* We discuss the above-listed topics in this chapter. We mention their usecase in our attention models.

covered in this chapter in Table 2.1.

2.1 Variational Autoencoder

A Variational Autoencoder (VAE) [93] is a generative latent-variable model. It operates under the premise that data x is generated from a random process involving an unobserved random variable z. Specifically, x is generated from a conditional distribution $p(x|z)$ where z follows a prior $p(z)$. Often, $p(z)$ is assumed to have a known parametric form such as that of a Gaussian or Uniform distribution. The goal of a VAE is to learn a data distribution $p(x) = \int p(x|z)p(z)dz$, which poses two challenges. First, we cannot directly

learn the likelihood $p(x|z)$ as the ground-truth distribution representing an association between x and z is unknown. Second, the integration with respect to z is intractable.

In response to the first challenge, VAE estimates the likelihood $p(x|z)$ using a decoder (also known as a recognition network). To learn $p(x|z)$, we infer $z \sim p(z|x)$ and use the pair (x, z) to train the decoder. Let's see how we can compute the posterior $p(z|x)$ using Bayes' rule.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}. \tag{2.1}$$

Unfortunately, the integration in the denominator is intractable; thus, we cannot compute $p(z|x)$ analytically. Instead, we introduce an encoder (also called an inference network) to approximate the posterior $p(z|x)$ using a $q(z|x)$. Further, we assume $q(z|x)$ has a known parametric form and let the encoder emit parameters of this distribution.

Yet, we cannot optimize $p(x) = \int p(x|z)p(z)dz$ due to intractable integration *i.e.*, the second challenge. Instead, we maximize the evidence lower bound (ELBO) on $\log p(x)$ to learn the parameters of the encoder and the decoder. Let us derive the ELBO.

$$\begin{aligned}
\log p(x) &= \log \int \frac{p(x, z)q(z|x)}{q(z|x)}dz && (2.2)\\
&= \log \mathbb{E}_{q(z|x)} \frac{p(x, z)}{q(z|x)} && (2.3)\\
&\geq \mathbb{E}_{q(z|x)} \log \frac{p(x, z)}{q(z|x)} \qquad (\because \text{Jensen's inequality}) && (2.4)\\
&= \mathbb{E}_{q(z|x)} \log p(x|z) + \mathbb{E}_{q(z|x)} \log \frac{p(z)}{q(z|x)} && (2.5)\\
&= \mathbb{E}_{q(z|x)} \log p(x|z) - D_{\mathrm{KL}}(q(z|x)||p(z)) && (2.6)
\end{aligned}$$

The right-hand side of the above equation represents the ELBO. We estimate the ex-

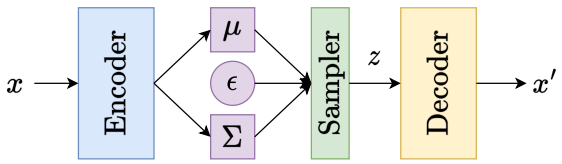


Figure 2.1: *Variational Autoencoder framework.* An encoder observes data x and generates μ and Σ of a multivariate Gaussian posterior. A decoder observes a sample z from the above posterior and reconstructs x.

pectation of the likelihood (*i.e.*, the first term in equation 2.6) using Monte-Carlo sampling: we draw multiple zs from $q(z|x)$ and compute an average $\log p(x|z)$. In practice, due to mini-batch gradient descent, it is sufficient to draw as low as one z per x for each parameter update. To backpropagate gradients through the sampling process, we use a reparameterization trick. Let's assume that the posterior $q(z|x)$ is a multivariate Gaussian distribution, and an encoder emits mean ($\mu = [\mu_1, \ldots, \mu_n]$) and co-variance ($\Sigma = diag(\sigma_1^2, \ldots, \sigma_n^2)$) of the Gaussian. Now we draw z in a differentiable manner as follows.

$$\forall i: \quad z_i = \mu_i + \sigma_i \odot \epsilon \; ; \quad \text{where } \epsilon \sim \mathcal{N}(0, 1).$$

Here $\mathcal{N}(0, 1)$ is a univariate Gaussian distribution with zero mean and unit variance. We illustrates the elements of a VAE framework in Figure 2.1.

In Chapter 4, we use a VAE to generate feature maps of complete scenes from glimpses. Note this is a multimodal problem; we may find multiple plausible complete scenes congruous with the given glimpses. For instance, given that the model observed the sky region, complete scenes with and without clouds are equally valid. While we assume a unimodal Gaussian form for the posterior $q(z|\cdot)$ in the above discussion, our scene gener-

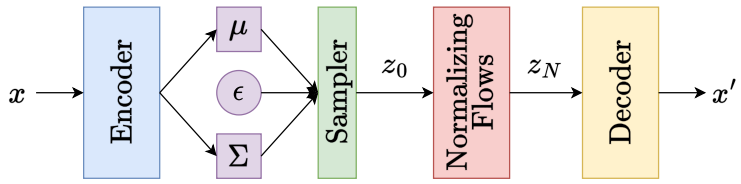


Figure 2.2: *Variational Autoencoder with Normalizing Flows.* Normalizing flows transform unimodal posterior $\mathcal{N}(\mu, \Sigma)$ into a complex posterior of unknown form. We achieve sample z_N from this unknown posterior by transforming a sample $z_0 \sim \mathcal{N}(\mu, \Sigma)$ to z_N using normalizing flows.

ation task follows a complex multi-modal posterior of unknown form. To learn complex $q(z|\cdot)$, we use normalizing flows [95] as described next.

2.2 Normalizing Flows

Normalizing flows map a simple distribution, such as Gaussian, to a complex multi-modal distribution. Thus, one can achieve a sample from the complex distribution by drawing a sample from the Gaussian distribution and transforming it using normalizing flows.

In the case of VAEs, normalizing flows map the predicted unimodal Gaussian posterior $q(z_0|x)$ to a complex multimodal posterior $q(z_N|x)$ (see Figure 2.2). Normalizing flows map z_0 to z_N using a series of differentiable bijections, $\{f_1, f_2, \ldots, f_N\}$.

$$z_N = f_N \circ \cdots \circ f_2 \circ f_1(z_0); \text{ where } z_0 \sim q(z_0|x). \tag{2.7}$$

Examples of neural architectures for normalizing flows are IAF [95], MAF [136], NICE [46], Real NVP [47], Glow [94], NAF [74], BNAF [42], RAD [48], NSF [50], etc. Figure 2.3 illustrates transformations through *Glow* when $q(z_N|\cdot)$ is a mixture of Gaussian. Let

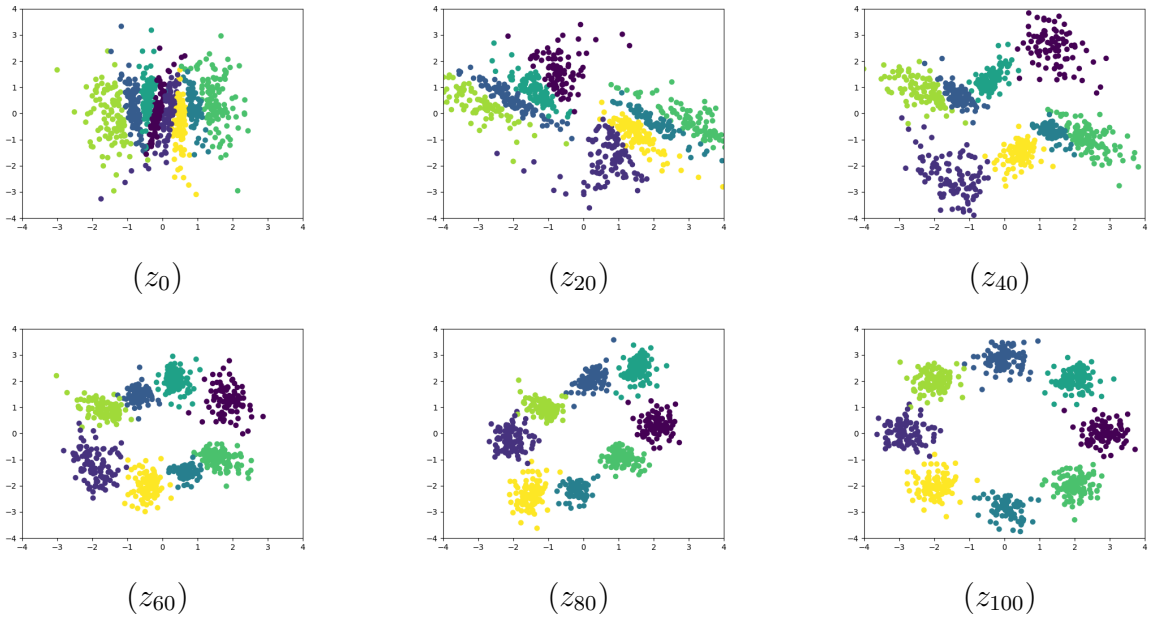


Figure 2.3: *Invertible transformations through normalizing flows.* Normalizing flows (here, the Glow model [94] with $N = 100$) transform samples from a Gaussian distribution to samples from a mixture of Gaussians distribution.

us use a change of variable formula to establish a relationship between $q(z_0|\cdot)$ and $q(z_N|\cdot)$.

$$q(z_N|\cdot) = q(z_0|\cdot) \prod_{n=1}^{N} |det(J_{f_n})|^{-1}, \tag{2.8}$$

where J_{f_n} is a Jacobian of f_n. An important property of the normalizing flows is that we can use the law of the unconscious statistician (LOTUS) to compute expectations of any quantity $g(z_N)$ with respect to $q(z_N|\cdot)$ without knowing its explicit formula.

$$\mathbb{E}_{q(z_N|\cdot)}[g(z_N)] = \mathbb{E}_{q(z_0|\cdot)}[g(f_N \circ \cdots \circ f_1(z_0))] = \mathbb{E}_{q(z_0|\cdot)}[g(f_{1:N}(z_0))]. \tag{2.9}$$

Let us rewrite the ELBO using LOTUS. Below we incorporate equation 2.8 in equa-

tion 2.5.

$$\text{ELBO} = \mathbb{E}_{q(z_N|x)}\Big\{ \log p(x|z_N) + \log p(z_N) - \log q(z_N|x)\} \tag{2.10}$$

$$= \mathbb{E}_{q(z_0|x)}\Big\{ \log p(x|f_{1:N}(z_0)) + \log p(f_{1:N}(z_0)) - \log q(z_0|x) + \sum_{n=1}^{N} \log |det(J_{f_n})|\Big\} \tag{2.11}$$

Recall, the prior $p(z_N) = p(f_{1:N}(z_0)) = \mathcal{N}(0, I)$. The encoder and the decoder of a VAE predict $q(z_0|x) = \mathcal{N}(\mu, \Sigma)$ and $p(x|z_N)$, respectively.

2.3 Bayesian Optimal Experiment Design

In Chapter 4, we use Bayesian Optimal Experiment Design (BOED) to decide which glimpse location should be attended to next. Let us review the general framework of BOED in this section. We will discuss BOED in the context of attention in Chapter 4.

BOED helps us decide which experiment should we conduct to collect the most informative data. Specifically, given a set of experiments Δ, BOED finds an optimal experiment $\delta^* \in \Delta$ with maximum utility U [31]. A utility function $U(\delta, \alpha, o)$ defines the gain from choosing to perform experiment δ resulting in observation o with the interest parameter value α. Thus, optimizing U with respect to δ requires knowledge of $p(\alpha, o|\delta) = p(\alpha|o, \delta)p(o|\delta)$.

$$\delta^* = \underset{\delta\in\Delta}{\text{argmax}}\, \mathbb{E}_{\alpha,o}\{U(\delta, \alpha, o)\} \tag{2.12}$$

$$= \underset{\delta\in\Delta}{\text{argmax}} \int_o \int_\alpha U(\delta, \alpha, o)p(\alpha, o|\delta)\ do\ d\alpha \tag{2.13}$$

$$= \underset{\delta\in\Delta}{\text{argmax}} \int_o \Big\{ \int_\alpha U(\delta, \alpha, o)p(\alpha|o, \delta)d\alpha\Big\}p(o|\delta)do \tag{2.14}$$

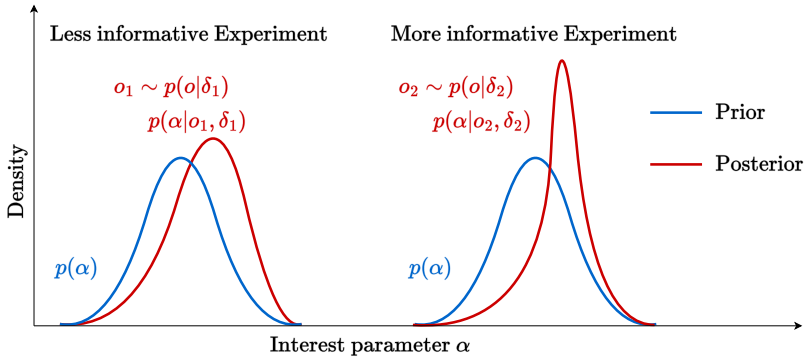


Figure 2.4: *Information Gain as a utility function for Bayesian Optimal Experiment Design (BOED).* We compare two experiments δ_1 and δ_2 by assessing their respective information gain (*i.e.* divergence between posterior $p(\alpha_o, \delta)$ and prior $p(\alpha)$. We favor δ_2 over δ_1 due to higher information gain.

The expected utility function can be defined in various ways depending on the task. A few examples of expected utility functions are feature variance [76], uncertainty in the prediction [124], and expected Shannon information [117]. In Chapter 4, we will use expected information gain (EIG) as our utility function.

2.3.1 Expected Information Gain

We measure the informativeness of an experiment δ resulting in observation o using information gain,

$$IG(o, \delta) = D_{\mathrm{KL}}[p(\alpha|o, \delta)||p(\alpha)]. \tag{2.15}$$

To understand how information gain works as a utility function, let's consider two experiments δ_1 and δ_2 resulting in observations o_1 and o_2. As shown in Figure 2.4, we compute posterior $p(\alpha|o, \delta)$ for both experiments and assess their respective information gain. Since δ_2 achieves higher information gain than δ_1, we consider δ_2 to be a more

useful experiment than δ_1 [1].

Note that IG (equation 2.15) requires knowledge of observation o, which is not available prior to an experiment. Thus, we compute the expectation of IG over all possible observations that experiment δ can yield. This gives us the Expected Information Gain,

$$EIG(\delta) = \mathbb{E}_{p(o|\delta)} D_{\mathrm{KL}}[p(\alpha|o,\delta)||p(\alpha)]. \tag{2.16}$$

We can also reach EIG as the expected utility function by using $U(\delta,\alpha,o) = [\log p(\alpha|o,\delta) - \log p(\alpha)]$ in equation 2.14.

$$\begin{aligned}
\delta^* &= \underset{\delta\in\Delta}{\operatorname{argmax}}\, \mathbb{E}_{p(o|\delta)}\Big\{\int_\alpha U(\delta,\alpha,o)p(\alpha|o,\delta)d\alpha\Big\} && (2.17)\\
&= \underset{\delta\in\Delta}{\operatorname{argmax}}\, \mathbb{E}_{p(o|\delta)}\Big\{\int_\alpha [\log p(\alpha|o,\delta) - \log p(\alpha)]p(\alpha|o,\delta)d\alpha\Big\} && (2.18)\\
&= \underset{\delta\in\Delta}{\operatorname{argmax}}\, \mathbb{E}_{p(o|\delta)} D_{\mathrm{KL}}[p(\alpha|o,\delta)||p(\alpha)] && (2.19)\\
&= \underset{\delta\in\Delta}{\operatorname{argmax}}\, EIG(\delta) && (2.20)
\end{aligned}$$

The above equation 2.20 presents the formula for BOED with EIG as the expected utility function. In practice, we use a neural network f_θ to model $p(\alpha|o,\delta)$. Moreover, we learn a generative model g_ϕ, such as a VAE, to model $p(o|\delta)$. Finally, we compute the expectation in equation 2.16 by probing $p(\alpha|o,\delta) = f_\theta(o,\delta)$ at Monte Carlo samples $o \sim p(o|\delta) = g_\phi(\delta)$.

[1] An example based on [83].

2.4 One-step Actor-Critic

In Chapter 5, we consider glimpse-based attention to be a Partially Observable Markov Decision Process (POMDP). In POMDP, an autonomous agent perceives a world via a series of partial observations collected through task-aware interactions. At time t, the sequential agent summarizes its perception of the world in a Markov state s_t. At $t+1$, the agent uses s_t to decide and take an action a_{t+1} yielding observation o_{t+1}. It uses (a_{t+1}, o_{t+1}) to update the state to s_{t+1}.

We teach the agent to interact with the world using reinforcement learning. Specifically, we use a module called an 'actor' to learn a parameterized policy $\pi_\theta(a_{t+1}|s_t)$, representing a probability distribution over actions a_{t+1}. During training, when an agent takes an action $a_{t+1} \sim \pi_\theta(a_{t+1}|s_t)$, we award it a reward R_{t+1} indicating the utility of action a_{t+1}. Let us define return G_t to be the sum of the future rewards, *i.e.*, $G_t = \sum_{t'=t+1}^{T} R_{t'}$. We can learn parameters θ by following gradients for a criterion maximizing G_t. However, it updates the parameters only once at the end of the episode since the calculation of G_t requires rewards from $t' > t$. To improve efficiency in the learning process, we use the *One-step Actor-Critic Method* [164].

One-step Actor-Critic provides the actor immediate feedback using a separate module called a 'critic'. A critic learns a parameterized value function $v_\omega(s)$ estimating the expected return G_t given the current state s_t. Since we require

$$v_\omega(s_t) \approx \mathbb{E}_\pi[G_t] = \mathbb{E}_\pi[R_{t+1} + G_{t+1}] \approx \mathbb{E}_\pi[R_{t+1} + v_\omega(s_{t+1})], \tag{2.21}$$

we learn ω by minimizing difference between $\mathbb{E}_\pi[R_{t+1} + v_\omega(s_{t+1})]$ and $v_\omega(s_t)$. For mini-

batch training, this results in the following update rule.

$$\delta = (R_{t+1} + v_\omega(s_{t+1})) - v_\omega(s_t), \tag{2.22}$$

$$\omega = \omega + \alpha_\omega \delta \nabla v_\omega(s_t); \tag{2.23}$$

where α_ω is the learning rate for ω. Note that $v_\omega(s_{t+1})$ is only used to compute targets for $v_\omega(s_t)$; we do not compute gradients with respect to $v_\omega(s_{t+1})$.

An actor must learn a policy that achieves maximum return. Thus, when a_{t+1} achieves lower return than $\mathbb{E}_\pi[G_t] \approx v_\omega(s_t)$, we penalize $\pi_\theta(a_{t+1}|s_t)$. We estimate the deficit in return as $(R_{t+1} + \mathbb{E}_\pi[G_{t+1}]) - \mathbb{E}_\pi[G_{t+1}] \approx (R_{t+1} + v_\omega(s_{t+1})) - v_\omega(s_t)$. Recall we already defined this quantity as δ in equation 2.22. From this point of view, δ is also known as (lost) *advantage*. Penalizing policy by δ gives us an update rule,

$$\theta = \theta + \alpha_\theta \delta \nabla \log \pi_\theta(a_{t+1}|s_t); \tag{2.24}$$

where α_θ is a learning rate for θ. Note that δ is only used as a penalty; we do not update ω through δ. We outline the process of training through the one-step actor-critic method in Algorithm 1.

2.5 Student-Teacher Training Paradigm

The student-teacher training paradigm is a prevalent technique in machine learning. The idea is to let a *Teacher* model predict the class distribution $p(y|\cdot)$ (or features) for a given image x and use the predicted distribution (or features) as a 'soft' target for the *Student* model. The student model replicates $p(y|\cdot)$ (or features) from the same or the augmented image x'.

Algorithm 1 One-step Actor-Critic Algorithm

Initialize parameters θ and ω;
Define learning rates α_θ and α_ω;
repeat
    Initialize s_0;
    for $t \in \{0, \ldots, T-1\}$ **do**
        $a_{t+1} \sim \pi_\theta(a_{t+1}|s_t)$;
        Take action a_{t+1}, observe o_{t+1} and R_{t+1}, predict state s_{t+1};
        $\delta = (R_{t+1} + v_\omega(s_{t+1})) - v_\omega(s_t)$; $(v_\omega(s_{t+1}) = 0$ if $t = T-1)$
        $\omega = \omega + \alpha_\omega \delta \nabla v_\omega(s_t)$;
        $\theta = \theta + \alpha_\theta \delta \nabla \log \pi_\theta(a_{t+1}|s_t)$;
    end for
until Convergence

Let us refer to the prediction from the student model as $q(y|\cdot)$. We train the student model by minimizing the forward or the reverse KL divergence between $p(y|\cdot)$ and $q(y|\cdot)$,

$$\mathcal{L}_{forward} = D_{\mathrm{KL}}[p(y|x)||q(y|x')], \tag{2.25}$$

$$\mathcal{L}_{reverse} = D_{\mathrm{KL}}[q(y|x')||p(y|x)]. \tag{2.26}$$

The forward KL divergence favors mean-seeking behavior, whereas the reverse KL divergence favors mode-seeking behavior. (We use mean squared error for the image features predicted using the student and the teacher.)

The success of this training paradigm relies upon the fact that the soft targets are often more informative than their hard counterparts. Unlike hard targets (*i.e.*, one-hot labels), soft targets (*i.e.*, distributions) carry information regarding incorrect but more likely categories for the given image. However, achieving soft annotation is an extremely challenging and error-prone process. Therefore, we use a teacher model to learn the soft labels from the hard labels. Consider an example from Figure 2.5. While the hard ground truth label (y) only communicates that the semantic category of the given image (x) is 'Dog', a soft target from the teacher ($p(y|x)$) informs the student that the image of a dog

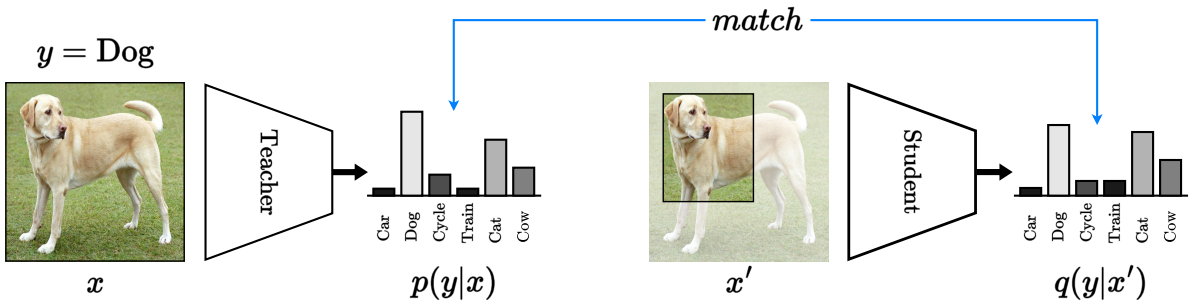


Figure 2.5: *Student-Teacher Training Paradigm.* A teacher model predicts class distribution $p(y|x)$ for a given image x. We use $p(y|x)$ as soft targets for the student model. The student model reproduces the soft targets from an augmentation of x.

is semantically more similar to an image of a cat than an image of a car.

Since its inception [72], the student-teacher training paradigm has been widely used in different but related methods. Let us discuss a few of these methods below.

Knowledge Distillation [72]. While optimizing for hard labels, we penalize a model uniformly for incorrect prediction. Thus a model has no incentive to move away from an incorrect and less likely prediction to an incorrect yet more likely prediction. This phenomenon hurts a small model more than a large model. While large models are more accurate, they require more memory and computations. Instead, we consider a large model as a teacher model and distill its knowledge into a compact student network. A student model learns better representations through soft targets and achieves improved performance, often on par with a large teacher model.

Unsupervised representation learning. So far we discussed supervised training with labeled data. However, the student-teacher training paradigm is also used for unsupervised learning. Among many, one unsupervised representation learning objective is to bring the latent representations of two augmented versions of the same image closer to each other. Usually, this entails creating a teacher model by copying either the latest or the moving average weights of a student model. Then we input random augmentations

of the same image, say x' and x'', to the teacher (f_T) and the student (f_S) and minimize the distance between $f_S(x')$ and $f_T(x'')$. Once trained, we discard the teacher model. Examples of this method are BYOL[64], SimSiam[33], and DINO[29].

Consistency Learning. Sometimes we have only limited labeled data but abundant unlabeled data. In consistency learning, we train a model on these two datasets simultaneously. We use labeled data to train a model in supervised fashion. For the unlabeled data, we create a copy of the model and consider it as a teacher model. We input a weakly augmented unlabeled image (x') to a teacher model and compute $p(y|x')$. Next, we input a heavy augmentation of the same image (x'') to the original (student) model and produce $q(y|x'')$. Finally, we minimize the distance between the predicted $q(y|x'')$ and soft targets $p(y|x')$ using equation 2.25 or equation 2.26. Once again, we discard the teacher model at the end. Examples of this approach are FixMatch [159] and RemixMatch [19].

We use the above ideas in Chapters 4 and 6 to explain the complete scene from the partial observations. We consider partial observations as a type of image augmentation. Then, we input the complete image to a pretrained teacher model to predict soft targets: latent representations and class distributions. Different from above, we use a separate pretrained teacher model to achieve stable targets. Our attention models act as students and reproduce the above targets from partial observations. Further, in Chapter 6, we improve performance of our model using knowledge distillation.

2.6 Spatial Transformers Network

A Spatial Transformers Network (STN) [85] computes an affine transformation of an input in a differentiable manner. Consider an image (or a feature map) $P \in \mathbb{R}^{H \times W \times C}$ with pixels sampled at regular sampling grid $G^p = \{(x_i^p, y_i^p) | i \in \{1, \ldots, HW\}\}$. STN predicts a 2D affine transformation matrix A_θ and transforms G^p to G^q by applying

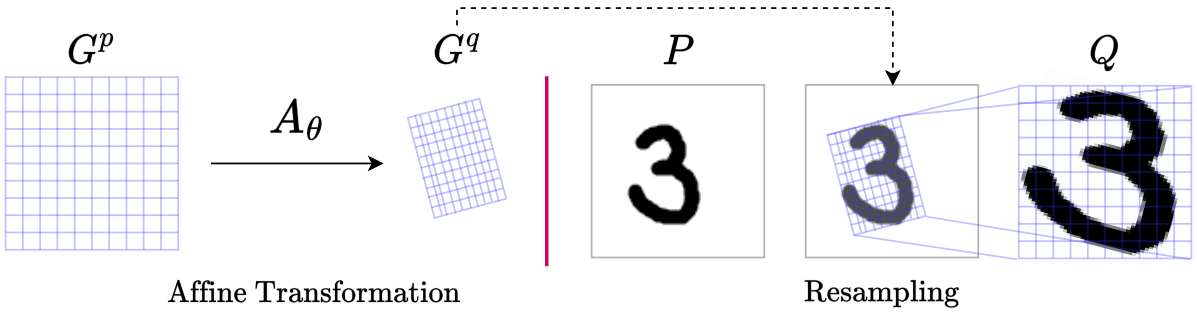


Figure 2.6: *Spatial Transformers Network (STN).* STN transforms the sampling grid G^p to G^q using the affine transformation matrix A_θ and resamples pixels on the transformed grid in a differentiable manner.

element-wise transformation on each coordinate (x_i^p, y_i^p) as follows.

$$\begin{bmatrix} x_i^q \\ y_i^q \end{bmatrix} = A_\theta \begin{bmatrix} x_i^p \\ y_i^p \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{bmatrix} x_i^p \\ y_i^p \\ 1 \end{bmatrix}. \tag{2.27}$$

Then, it resamples P on grid $G^q = \{(x_i^q, y_i^q) | i \in \{1, \ldots, H'W'\}\}$ using a differentiable sampling kernel k to achieve spatial transformation $Q \in \mathbb{R}^{H' \times W' \times C}$.

$$Q_i^c = \sum_n^H \sum_m^W P_{nm}^c k(x_i^q - m) k(y_i^q - n); \quad \forall i \in \{1, \ldots, H'W'\}; \quad \forall c \in \{1, \ldots, C\}. \tag{2.28}$$

A popular choice for k is the bilinear sampling kernel. We illustrate affine transformation (equation 2.27) and resampling (equation 2.28) in Figure 2.6. In Chapter 6, we use spatial transformer networks (STN) to extract square glimpses from a square video frame. Our

affine transformation matrix is

$$A_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}, \tag{2.29}$$

which allows cropping, scaling, and translation of the glimpse region. We set s to a predetermined value based on our model's field of view.

2.7 Vision Transformers

In its simplest form, a Vision Transformer (ViT) [49] predicts class label y for an image x. ViT is a stack of transformer encoder layers [178], and therefore, a sequence-to-sequence model. It views an image x as a sequence of non-overlapping patches. We append a class token to this sequence and let ViT process it to pool information from image patches to the class token. A linear classifier uses the resultant class token to predict label y. Below, we provide more details on the transformer encoder layers and the input formation.

Transformer Encoder Layers. We show the architecture of a transformer encoder layer in Figure 2.7 (right). Each encoder layer processes a sequence of vectors $\nu = [\nu_1^T; \nu_2^T; \ldots; \nu_N^T] \in \mathbb{R}^{N \times D}$. The functional representation of an encoder layer is as follows.

$$\nu = \nu + MHA(LN(\nu)), \tag{2.30}$$

$$\nu = \nu + MLP(LN(\nu)); \tag{2.31}$$

where LN is a LayerNorm [14], MLP is a multi-layer perceptron and MHA is a multi-head self-attention layer[178]. MHA is a collection of multiple parallel single-head self-attention (SHA) layers, $MHA(\cdot) = [SHA_1(\cdot), SHA_2(\cdot), \ldots]$. A single self-attention head

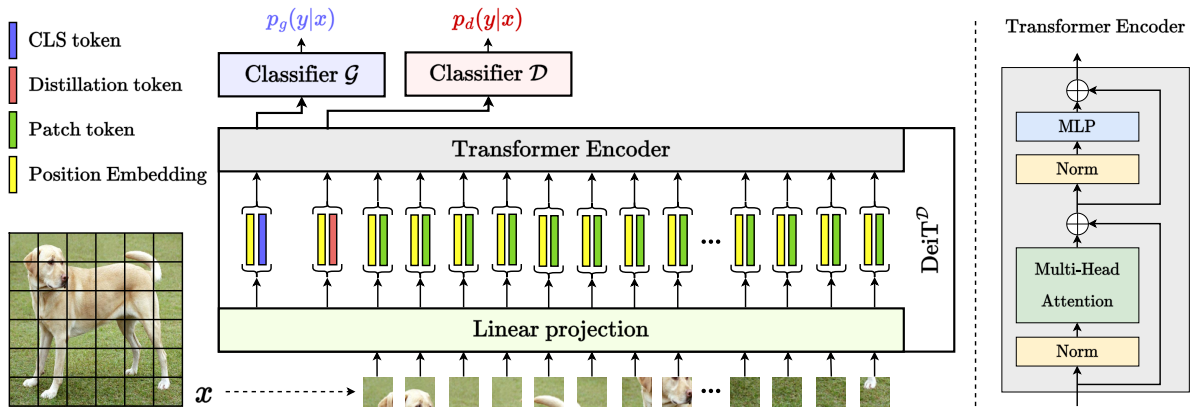


Figure 2.7: *Vision Transformers (ViT).* (Left) ViT processes a sequence of image patches to predict label y. Given the linear projection of the image patches, class token and optional distillation token, ViT predicts $p_g(y|x)$ and optionally $p_d(y|x)$. Following Touvron *et al.* [172], we refer to a ViT with distillation as DeiT$^{\mathcal{D}}$. (Right) ViT consists of transformer encoder layers. Each encoder layer is a stack of two residual blocks, one with a multi-head attention and one with a multi-layer perceptron.

computes the following functions.

$$Q, K, V = f_Q(\nu), f_K(\nu), f_V(\nu), \tag{2.32}$$

$$Attn(Q, K, V) = softmax\Big(\frac{QK^T}{\sqrt{d_K}}\Big)V; \tag{2.33}$$

where f_Q, f_K and f_V are linear projection layers. Recall that ν is a series of vectors. As a result, the above operations yield a series of vectors where each vector is a weighted average of all vectors in ν. Note that self-attention is 'soft' attention, fundamentally different from the 'hard' attention we seek to pursue in this thesis. While hard attention only attends to a limited region of a scene, soft attention attends to all regions but with varying emphasis [125, 195]. We further discuss soft and hard attention in Section 3.1.2.

Input and Output Formulation. To let a ViT process an image of size $H \times W$,

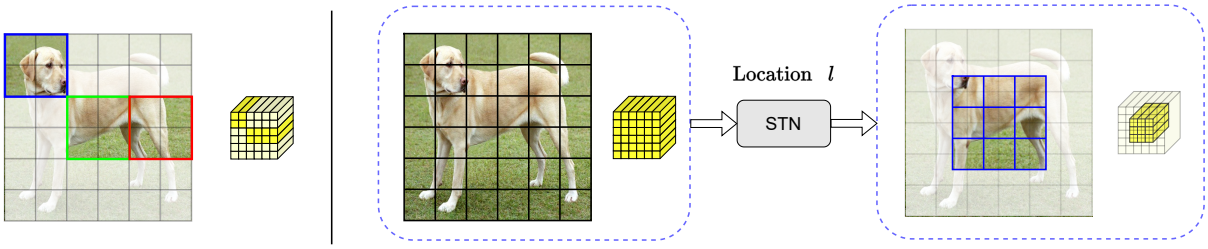


Figure 2.8: *Patches and position embeddings (in yellow) for a glimpse-based ViT.* A glimpse-based ViT processes patches from the observed glimpses only. (left) We extract non-operating glimpses from an image. We select position embeddings for the image patches covered by the glimpses and discard the rest. (right) We extract a glimpse on real coordinates within the bounds of a video frame. We pass a frame and the matrix of position embeddings to an STN. The STN returns a glimpse and a set of interpolated position embeddings associated with the glimpse-patches.

we first divide an image into N patches of size $P \times P$ (*i.e.*, $N = HW/P^2$). We input flattened patches to a learnable *Linear Projection* layer and compute patch tokens, each of size D. Next, we concatenate a classification token, an optional distillation token (for DeiT$^{\mathcal{D}}$ [172]), and the above patch tokens to form a sequence. Note that the classification and the distillation tokens are learnable vectors of size D. Since the transformer encoder is invariant to input permutations, we inject knowledge about spatial order by adding learnable positional embeddings to the above sequence of tokens. We process the resultant sequence (ν) using transformer encoder layers, as discussed above. We separate output corresponding to the classification and distillation tokens and pass them to two independent linear classifiers to predict $p_g(y|x)$ and $p_d(y|x)$, respectively. We evaluate supervised and distillation criteria on $p_g(y|x)$ and $p_d(y|x)$, respectively. We show a complete overview of Vision Transformers (ViT) in Figure 2.7 (left).

In Chapters 5 and 6, we use a ViT for glimpse-based attention. To process glimpses, we form an input sequence using patches from the glimpse regions only. Therefore, we need to extract position embeddings corresponding to the glimpse patches. We discuss

the process of glimpse and position embedding extraction below. The class and the distillation tokens and their position embeddings are the same as discussed above.

Recall, the position embeddings for the image patches are of size $HW/P^2 \times D$, where each D-length embedding learns a representation of the corresponding patch location. Let us reshape the embeddings into a matrix of size $\frac{H}{P} \times \frac{W}{P} \times D$ and keep them aside. In Chapter 5, we extract non-overlapping glimpses from an image. Further, the glimpses are of size $mP \times mP$. To process glimpses using a ViT, we only input the patches from the glimpse region to the linear projection layer. We identify the location of these patches in the image and extract corresponding position embeddings from the embedding matrix that we previously kept aside (see Figure 2.8 (left)). Further, in Chapter 6, we extract an $nP \times nP$ glimpse centered at location $l \in \mathbb{R}^2$ in the given video frame. We extract a glimpse and the corresponding position embeddings by passing a frame and the above matrix of embeddings to the STN (described in Section 2.6). The STN outputs a glimpse of size $nP \times nP$ and a matrix of $n \times n$ *interpolated* position embeddings. We extract n^2 glimpse patches of size $P \times P$ and their corresponding n^2 position embeddings from the above output. We illustrate this process in Figure 2.8 (right).

3

Literature Review

This chapter reviews literature related to attention, recognition, and reasoning from partial observations. We aim to provide a reader with a starting point and to assist in developing a basic understanding of the field. We also shed light on the aspects where our research builds on previous works versus when we make an original contribution. We do not rely on a deep understanding of the presented material in the remaining chapters. If essential, we review a few works again in Chapters 4, 5, and 6 and compare and contrast them with our attention models at greater length.

3.1 Attention

Vision scientists have been studying attention for more than half a century. In this section, we provide an overview of seminal works in the area of discrete glimpse-based attention. We start with early models and move on to more recent work. We group recent works into several categories and signal the categories that apply to our models.

3.1.1 Early Work

Early traces of visual attention can be found in multi-scale edge detection algorithms. Kelly [91] proposes locating edge regions on a coarser scale and attending to them on a finer scale. Fukushima [58] proposes neural networks that attend to and recognize patterns in an image sequentially. At a time, the network facilitates signals from the most active pattern and inhibits remaining signals. The process continues for the following most active pattern until all patterns are attended. Koch and Ullman [96] propose a winner-take-all algorithm to select the most conspicuous regions sequentially. They inhibit previously attended areas to shift attention to the next most salient region. Later, Itti *et al.*[82, 81] extend the winner-take-all method by employing image saliency as a conspicuous map. They proposed to fuse elementary features such as colors, intensity, and orientation to model image sliency. Clark and Ferrier [35] develop a control system for a binocular image acquisition device based on the above methods. Ahmad [2] proposes a SWIFT strategy to search for an object of interest among many distractors. Since the feature channels corresponding to numerous distractors should be highly active, their idea is to prioritize searching an object in the feature channel with minimal activity. On the other hand, Olshausen *et al.*[131] develop a biologically plausible attention model to form position and scale-invariant object representations based on the dynamic routing

of information. Swain and Ballard [165] develop a histogram-based method to attend to the instances of known objects. They detect the presence of a known object in the scene using histogram intersection and locate the object using the histogram back-projection method. For spatiotemporal attention, Burt [22] proposes to generate pyramids and to fixate on those locations on a finer scale where an event is detected on a coarser scale. For the tracking problem, they compute motion vectors from coarser scales of two frames and update them after attending to finer scales.

3.1.2 Recent Work

Graphical Models for Attention

Several researchers use graphical models to describe visual attention. Rimey and Brown [147, 148] model context-independent eye movements using the Hidden Markov Model (HMM). To adapt eye movements for the given scene, they incorporate visual feedback in an HMM and call it the Augmented Hidden Markov Model (AHMM). Larochelle and Hinton [107] introduce Fixation Restricted Boltzmann Machine (Fixation RBM) with third-order connections to learn a relationship between the context, the location of fixations, and the internal representation of a scene. The Fixation Neural Autoregressive Distribution Estimator (Fixation NADE) [207] is an extension of above method. Unlike Fixation RBMs, Fixation NADE learns better representation due to exact gradients and achieves higher performance. Denil *et al.*[44] develop an attention model for simultaneous object detection and tracking. Their model consists of two interacting pathways - identity and control - implemented using RBM and particle filters, respectively. Tang *et al.*[168] develop an attention-based generative model. They proposed to learn a generative Gaussian Deep Belief Network (GDBN) for human faces by attending to the faces in natural images using dynamic routing [131].

Reinforcement Learning for Attention

Many use reinforcement learning to train attention models. Paletta *et al.*[132] model attention-based object recognition as a Markov Decision Process (MDP). They use Q-learning with the reward that measures information gain in the posterior of the object hypothesis. Butko and Movellan [23] model eye movements as Information-gathering Partially Observable Markov decision process (I-POMDP) and use a policy gradient algorithm with the infomax reward function to learn attention policy. Later, they extend the I-POMDP method to locate multiple targets in a large image [24]. They replace classical raster-scan windowing in the Viola-Jones face detector with attention-based windowing and report improved runtime.

Deep Reinforcement Learning for Attention

Recurrent Attention Model (RAM) [125] is the first glimpse-based *deep* attention model. RAM sequentially observes glimpses from images and videos. At each time step, it predicts task-specific labels and the location of the next glimpse. RAM learns attention policy using REINFORCE algorithm, with the reward being the accuracy of a model. Later, Xu *et al.* [195] use the above method for the caption generation task. They also present another attention method where a model observes all regions of a scene, not just glimpses, but with varying degrees of importance; they refer to this method as 'soft attention'. Recall Vision Transformers (ViT) from Section 2.7 use soft attention. To differentiate the discrete glimpse-based attention from soft attention, Xu *et al.* call the former 'hard attention'. Figure 3.1 characterizes the difference between soft and hard attention. Since its inception, hard attention has been employed for many vision tasks such as image captioning [195], multiple object recognition [189, 12], localization and recognition [118], active object localization [27], 3D scene classification and 3D object

Figure 3.1: *Soft and Hard Attention.* Soft attention observes all regions of a scene with varying weights. Hard attention observes only discrete regions.

recognition [87], playing Atari games [161], autonomous driving [152], unsupervised scene exploration [86], etc. We discuss hard attention for recognition tasks at length in Section 3.3.

Other Approaches

A few recent works propose alternate approaches to attention. Ranzato [143] considers the location of the next glimpse as a latent variable and optimizes the model using an EM-like algorithm. Ba *et al.* [13] introduce weighted Wake-Sleep algorithm to RAM [125] and develop techniques to improve convergence rate. Alexe *et al.*[3] propose a voting-based nearest neighbor region selection. A few works [63, 52] use Spatial Transformer Networks [85] (described in Section 2.6) to make glimpse-based attention differentiable. Another line of works [156, 155] suggest sampling attention-worthy regions from self-attention or certainty maps.

The models presented in this thesis follow different techniques for attention. Our PAM (Chapter 4) uses Bayesian Optimal Experiment Design (BOED) to plan attention. Our STAM (Chapter 5) uses reinforcement learning. Our GliTr (Chapter 6), on the other

hand, uses differentiable STN and learns attention through task-specific objectives.

3.2 Active perception

Attention is closely related to the problem of 'learning where to look', also known as 'view-planning, 'active selection' or 'active vision' [192, 43, 45, 154]. The problems of active vision and visual attention are different in the following ways. In active vision, an agent has more degrees of freedom, and it continuously relocates to evaluate a scene from multiple viewpoints. In visual attention, the agent is temporarily static and only makes eye movements to sense small parts of a given view sequentially. A large body of work focuses on actively perceiving a 3D environment for scene recognition and localization [27], manipulating a 3D object for recognition [4], active vision-based view-planning [192], navigation and localization [41], etc. Moreover, a few works focus on fast and timely active agents that start recognizing objects, though imperfectly, from the very first viewpoint and update the decision as time progresses [88, 8, 9].

3.3 Recognition with Partial Observations

This thesis presents attention models for image and action recognition. In this section, we review these two problems. Note that scene recognition has the most extended presence in the history of computer vision. Our goal is not to provide an in-depth literature review of these problems. Here, we only provide an overview of some milestone works related to the methods used in the thesis. Specifically, we focus on works related to glimpse-based attention. We contrast these works with our models and highlight our original contributions.

3.3.1 Image Recognition

Mnih *et al.* [125] apply RAM to the MNIST image classification task. Ba *et al.* [12] extend RAM for multiple object localization and recognition. Since they also increase the depth of RAM, they call their model Deep RAM or DRAM. They apply DRAM to the task of recognizing sequences of numbers. Later, Sermanet *et al.* [157] evaluate DRAM on ImageNet – a natural image dataset. With the sole purpose of improving the efficiency of recent state-of-the-art image recognition models, Glance and Focus Net (GFNet) [187] adapts ideas from RAM and DRAM in recent models and devises a confidence-based early exit strategy. Recently, a few ideas from classic attention models are reintroduced in hard attention models. First, to improve the interpretability of image recognition, 'what' and 'where' attention pathways are reintroduced in a hard attention model called Saccader [51]. Second, a hard-attention model called Traversal Network (TNet) [135] visits informative glimpses from a multiscale image pyramid in a top-down fashion.

Most previous hard attention models initially glance at a complete image to locate the most informative glimpses. For instance, Saccader [51] analyses the complete image at its original resolution; whereas DRAM [13], TNet [135], and GFNet [187] observe the complete image at low resolution. Furthermore, TNet [135], and GFNet [187] use the low-resolution gist of an image to predict the class label. In contrast, our models do not look at the entire image at low resolution or otherwise. We predict the attention-worthy glimpse locations and the class of the complete image solely based on partial observations. From this perspective, RAM [125], which also operates under partial observability, is the closest related approach. While RAM is trained using REINFORCE, we train our PAM model using differentiable objectives. While RAM and our PAM perform well on small datasets, our ViT-based STAM model scales to large-scale real-world datasets

(*e.g.*, ImageNet). Note, due to the transformers architecture, STAM *uniquely* uses soft attention for the glimpses selected using hard attention.

We note that the hard attention models differ from another related set of approaches that observe an entire image to select all informative image patches simultaneously. Examples of these approaches are region proposal networks [146], top-K patch selection [5, 37], multiple-instance learning [78], attention sampling [89] and PatchDrop [175]. The above approaches observe an entire image and find all attention-worthy patches simultaneously. Unlike these approaches, our hard attention models do not observe the entire image and predict the location of informative sub-regions sequentially.

Further, in the space of Vision Transformers, methods such as PS-ViT [199], Dynamic-ViT [144], and IA-RED^2 [133] start with observing a complete image and progressively (re-)sample most discriminative patches in each successive transformer block. Unlike the above approaches, our STAM samples and inputs only informative patches to the ViT. Moreover, as mentioned earlier, our STAM is sequential in nature; it senses only one additional glimpse at each step.

3.3.2 Action Recognition

Many state-of-the-art methods perform offline action recognition once the entire video is available [53, 55, 30, 90, 173, 174, 181, 183]. Recently, hard attention has been applied to offline video action recognition. Wang *et al.*[180] propose a DRAM-style action recognition model. However, they locate multiple glimpses in the feature space. Huang *et al.*[75] extend their image-based Glance and Focus Net (GFNet) for videos. They locate useful glimpses by processing complete videos using a global model, then process these glimpses using a local model to recognize the action. Baradel *et al.*[17] present Glimpse Cloud, an attention model for human activity recognition. Glimpse Cloud tracks an unstructured

cloud of glimpses over time using multiple trackers, soft assigned using memory. Mac *et al.*[122] propose a spatiotemporal sampling scheme where they observe a complete video at low resolution using transformers and sample useful locations from the attention maps. They observe the sampled regions at high resolution and use them to perform recognition. Note the above models locate and observe multiple informative glimpses per frame. On the other hand, Chen *et al.*[32] develop a 3D Attention (A3D) model to sequentially locate the most informative spatiotemporal glimpses from the 3D features extracted from a complete video. Another line of approaches leverages pose information and focuses only on the relevant body parts [16, 40]. Note these works access full frames to locate informative regions. In contrast, our GliTr model never observes complete frames; it only observes a narrow glimpse from each frame. Further, our GliTr does not assume access to a temporally complete video.

The offline models discussed above are not optimized for the case where the entire video is yet to be available, and the models have to predict the action based on a preliminary, incomplete video. Performing an online or early action recognition based on a temporally incomplete video is a challenging task. A partially observed video may associate with multiple possible actions, leading to inherent uncertainty in the prediction task. Several methods focus on predicting actions from partial videos. Zhao and Wildes [204], Wu *et al.* [191], and Pang *et al.* [134] anticipate future actions based on the motion and object relations in the past frames. Many analyze micro-motions in the available early frames [163, 105, 100, 98]. Other approaches such as dynamic bag-of-words [150], global-local saliency [103], memorizing hard-to-predict samples [99], soft regression with multiple soft labels [73], and probabilistic modeling [112, 28] are also used. There exist only a few glimpse-based online action prediction models. Recent Adafocus and Adafocusv2 models [186, 188] include spatial glimpse-based attention to online ac-

tion recognition. Similar to DRAM [12] for image recognition, these models observe the recent frame completely to locate the most informative glimpse. Although, Adafocus, Adafocusv2, and other previous models only consider partial observability in the temporal dimension. They assume that the recent frames are spatially complete. In contrast, our GliTr operates with spatially and temporally incomplete information. Further, the glimpse-based Adafocus and Adafocusv2 operate in offline mode for the datasets with long actions covering a wide temporal horizon. On the other hand, our GliTr operates in online mode for these datasets well.

3.4 Explaining a Scene from Partial Observations

Our attention models perform recognition from incomplete information. To overcome partial observability, we force our models to reason about the complete scene from glimpses, which achieves two goals. First, it compels our models to observe glimpses that are most descriptive of the complete scene. Second, it improves recognition by forcing the model to view glimpses as part of a holistic scene.

In this section, we review the literature in two categories. First, we review the literature for scene completion from partial observation. Recall we require our PAM model to generate a feature map of a complete scene from glimpses. Second, we discuss consistency learning and related methods. Recollect, we ask our STAM and GliTr to produce features and class distribution consistent with the complete scene but using glimpses.

3.4.1 Image Completion

Image completion methods aim at synthesizing unobserved or missing image pixels conditioned on the observed pixels [198, 137, 77, 182, 196, 185]. Often, depending on whether the number of observed pixels is greater than or less than the number of unobserved

pixels, the image completion task is known as 'inpainting' or 'outpaining'. In this thesis, we are interested in outpainting. Image outpainting is an ill-posed problem with multiple possible solutions for the missing image regions. Zhang *et al.* [202] use generative adversarial networks (GAN) with a feature pyramid discriminator to generate multiple high-quality images given only a small observed region. Cai and Wei [25] develop PI-IGAN, which combines different image styles with the observed regions for pluralistic image completion. Zhao *et al.* [205] propose UCTGAN, where they learn probabilistic manifold mapping between partial and complete images and use the map to translate the former to the latter. Zheng *et al.* [206] propose a model with reconstructive and generative branches. The latter infers the posterior of the former to generate images based on partial observations.

While the above approaches use either GAN or a combination of GAN and VAE for multiple predictions, few researchers propose purely probabilistic methods. Sohn *et al.* [160] develop the seminal Conditional Variational Autoencoder or CVAE for a conditional generation. Garnelo *et al.* [60, 59] develop neural processes, a neural counterpart of Gaussian processes, and utilize them for a conditional generation. In our PAM model, we use Partial VAE (PVAE) [121] to predict the content of the complete image given only a few glimpses. PVAE posits that the seen and unseen regions are conditionally independent, given that they share a common latent space. Thus, we infer a latent code from seen regions and use them to generate unseen regions. However, unlike the original PVAE that infers image content in the pixel space, we predict content in the feature space.

3.4.2 Consistency Learning

Consistency learning, an idea initially proposed by Sajjadi *et al.*[151], has become an essential component in many recent semi-supervised learning (SSL) algorithms [159, 193, 19, 104]. Consistency learning acts as a regularizer that enforces the model output to be invariant to different augmentations of the same input [159, 193, 19, 114], or variations in the internal representations [15, 151], or the model parameters at different training epochs [104]. Consistency is achieved by using the predictions made from one perturbation as pseudo-targets for the predictions made from another perturbation. Refer to Section 2.5 for more details.

Another closely related idea in SSL is that of pseudo-labeling [110, 7, 138], where a trained model, known as 'teacher model', generates soft (continuous distributions) or hard (one-hot distributions) pseudo-labels for the unlabeled data under no perturbations. These pseudo-labels are later used as targets while training a student model with unlabeled examples under some perturbations [194, 20]. This approach is similar to Knowledge Distillation [72], where the student is trained to reconstruct the output or internal representation [1] of the teacher, but for labeled data.

Image consistency

In Chapter 5, we develop consistency training objectives based on the above concepts. We train our models to be invariant to specific type of input perturbations, i.e., the partial and the complete observations. Furthermore, we use a teacher model to produce soft pseudo-labels from an entire image and use them as targets while training our student models using partial observations.

Video consistency

Many early action recognition models learn to predict the class distribution consistent with the complete video using only a subset of early frames [26, 56, 101, 139, 184]. Others have also leveraged spatiotemporal consistency for complete frames [171, 54]. Unlike previous work, in Chapter 6, we use a teacher model that predicts class distribution in an online fashion, *i.e.*, using complete frames from a preliminary video. Our student model reproduces the above class distribution in an online manner but using only glimpses from the corresponding frames. Since this objective forces our student model to be consistent in the temporal dimension, we call it temporal consistency.

Moreover, our teacher model also provides per-frame features from spatially complete frames. We let our student model reproduce these features from glimpses extracted from the corresponding frames. Since this objective makes our student model consistent in the spatial dimension, we refer to it as spatial consistency. Note our spatial consistency is similar to the unsupervised SSL methods such as BYOL [64], SimSiam [33], and DINO [29]. Like these methods, we enforce consistency between features of two augmentations of the same frames. However, unlike these SSL methods, we predict features of the two augmentations using two different models. Since we aim to learn a model that performs recognition from glimpses, we never show our student model complete frames. Instead, we use a separate teacher model to predict targets from complete frames.

Part I

Spatial Attention Models

4

Probabilistic Attention Model

Note: Significant parts of this chapter are adapted, with permission, from:

[142] *A Probabilistic Hard Attention Model for Sequentially Observed Scenes.*
Samrudhdhi B Rangrej and James J Clark.
British Machine Vision Conference, 2021.

The authors hold the copyright in BMVC papers they have authored. While BMVA (the publisher of the proceedings) holds copyright over the collection, the authors may use the papers they have authored [169].

In this chapter, we present our Probabilistic Attention Model (PAM) for glimpse-based image classification. We emphasize that PAM never observes an image entirely and classifies it solely based on glimpses. We formulate the problem of finding informative glimpse locations under partial observability as Bayesian Optimal Experiment Design (BOED). Starting from a random location, our sequential PAM uses BOED to determine the following optimal location. PAM estimates the expected information gain (EIG) obtained from yet-to-be-observed glimpses and selects a location with maximum EIG. As the computation of EIG requires the content of unseen regions, PAM *synthesizes* the unknown content conditioned on the observed glimpses.

Our PAM consists of three modules, a recurrent feature aggregator, a linear classifier, and a Partial VAE [121]. Partial VAE synthesizes the content of various glimpses in the scene based on partial observations. We predict the content in the feature space instead of the pixel space to improve efficiency. Since there are multiple possibilities for the content of the unobserved regions, we use normalizing flows in Partial VAE to capture the multi-modality in the posterior. We train the model using a combination of discriminative and generative objectives. When tested on five datasets, our PAM gains 2-10% higher accuracy than the baseline models when both have seen only a couple of glimpses.

4.1 Model

Our PAM sequentially captures glimpses from an image x and predicts label y. It maintains a hidden state h_t that summarizes glimpses observed up to time t. At time t, PAM predicts coordinates l_t based on the hidden state h_{t-1} and captures a square glimpse g_t centered at l_t in an image x, i.e. $g_t = g(x, l_t)$. Based on g_t and l_t, it updates the hidden state to h_t and predicts label y from the updated h_t.

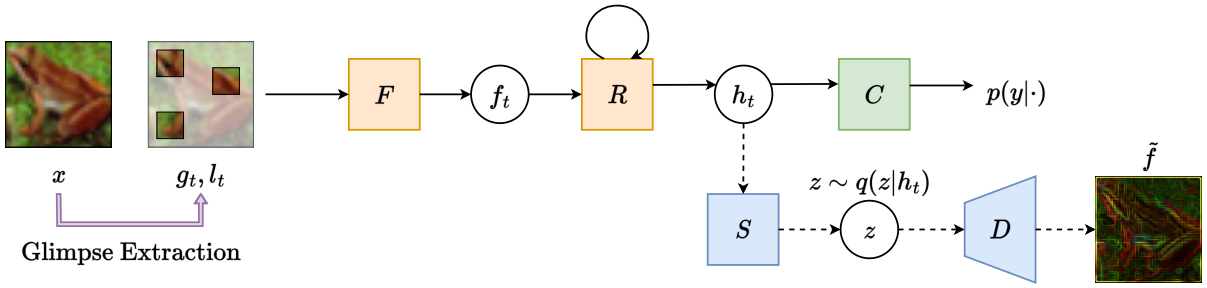
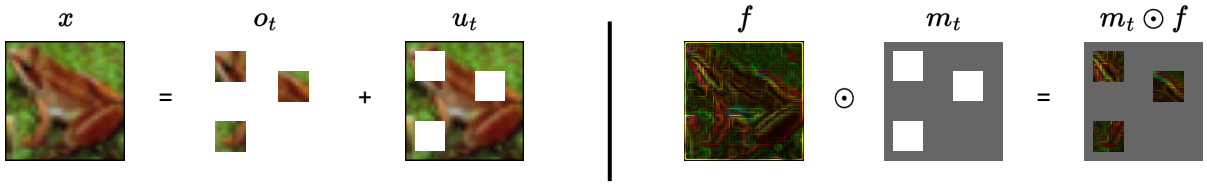


Figure 4.1: *Architecture of our PAM.* Our PAM consists of a recurrent feature aggregator (F and R), a linear classifier (C), and a Partial VAE (S and D). At the time t, PAM actively observes a glimpse g_t and its coordinates l_t. Given g_t and l_t, the feed-forward module F extracts features f_t, and the recurrent module R updates a hidden state to h_t. Using an updated hidden state h_t, the linear classifier C predicts the class distribution $p(y|h_t)$. Simultaneously, the normalizing flow-based encoder S predicts the approximate posterior $q(z|h_t)$. The decoder D uses a sample $z \sim q(z|h_t)$ to synthesize a feature map $\tilde{f}$ containing features of all glimpses. We use $\tilde{f}$ in BOED (section 4.1.2) to decide the next glimpse location.

4.1.1 Building Blocks

As shown in Figure 4.1, our proposed PAM is comprised of three building blocks. A recurrent feature aggregator (F and R) maintains a hidden state h_t. A classifier (C) predicts the class probabilities $p(y|h_t)$. A normalizing flow-based variational autoencoder (S and D) synthesizes a feature map of a complete image from h_t. We may view a feature map of a complete image as a map containing features of all glimpses. The BOED, as discussed in section 4.1.2, uses the synthesized feature map to find an optimal location to attend at the next time step. To distinguish the synthesized feature map from an actual one, let us call the former $\tilde{f}$ and the latter f. Henceforth, we crown any quantity derived from the synthesized feature map with a ($\tilde{}$). Next, we provide details about the three building blocks of the model, followed by a discussion of the BOED in the context of hard attention.

Recurrent Feature Aggregator. Given a glimpse g_t at location l_t, a feed-forward



(a) (b)

Figure 4.2: *Conditional independence in PVAE.* (a) seen and unseen regions (o_t and u_t) are conditionally independent given they share a common latent space. (b) We create a binary mask m_t based on the regions observed so far and multiply it with the features (f and $\tilde{f}$) to compute likelihood in PVAE.

module extracts features $f_t = F(g_t, l_t)$, and a recurrent network updates a hidden state to $h_t = R(h_{t-1}, f_t)$. We define $F(g, l) = \mathcal{F}_g(g) + \mathcal{F}_l(l)$ and $R(h, f) = LeakyReLU(\mathcal{F}_h(h) + \mathcal{F}_f(f))$. $\mathcal{F}_g$ is a small CNN with receptive-field equal to size of g. $\mathcal{F}_l, \mathcal{F}_h, \mathcal{F}_f$ are shallow networks with one linear layer.

Classifier. At each time step t, a linear classifier predicts the distribution $p(y|h_t) = C(h_t)$ from a hidden state h_t. As the goal is to predict label y for an image x, we learn a distribution $p(y|h_t)$ by minimizing $D_{\mathrm{KL}}[p(y|x)||p(y|h_t)]$. Optimization of this KL divergence is equivalent to minimizing the following cross-entropy loss,

$$\mathcal{L}_{\mathrm{CE}}(t) = -p(y|x)\log(p(y|h_t)). \tag{4.1}$$

Partial Variational Autoencoder. We adapt a variational autoencoder (VAE) to synthesize the feature map of a complete image from the hidden state h_t. A VAE learns a joint distribution between the feature map f and the latent variable z given h_t, $p(f, z|h_t) = p(f|z)p(z|h_t)$. An encoder approximates the posterior $q(z|f, h_t)$, and a decoder infers the likelihood $p(f|z)$. Refer to Section 2.1 for a brief introduction to VAE. The optimization of VAE requires calculation of $D_{\mathrm{KL}}[q(z|f, h_t)||p(z|h_t)]$ [93]. As the hard attention model does not observe the complete image, it cannot estimate $q(z|f, h_t)$.

Hence, we cannot incorporate the standard VAE directly into a hard attention framework and instead use the following approach.

At the time t, let us separate an image x into two parts, o_t — the set of regions observed up to t, and u_t — the set of regions as yet unobserved (see Figure 4.2(a)). Ma *et al.* [121] observed that in a VAE, o_t and u_t are conditionally independent given z, i.e. $p(x|z) = p(u_t|z)p(o_t|z)$. They synthesize u_t independently from the sample $z \sim q(z|o_t)$, while learning the posterior $q(z|o_t)$ by optimizing ELBO on $\log(p(o_t))$. They refer to the resultant VAE as a Partial VAE.

The BOED, as discussed in section 4.1.2, requires *features* of o_t and u_t. Hence, without loss of generality, we consider o_t and u_t in the feature space. Synthesizing o_t and u_t in the feature space serves two purposes. First, the model does not have to extract features of o_t and u_t for the BOED as they are readily available. Second, Partial VAE does not have to produce unnecessary details, such as the exact pixel color, that the feature extractor may disregard later. Recall that the features $f_{1:t}$ and the hidden state h_t calculated by our attention model correspond to the glimpses observed up to t, which is equivalent to o_t, the set of observed regions. Hence, we write $q(z|o_t)$ as $q(z|h_t)$ and $p(o_t|z)$ as $p(f_{1:t}|z)$ in the ELBO of Partial VAE, giving us

$$\mathcal{L}_{\text{PVAE}}(t) = -ELBO = -\Big\{\mathbb{E}_{q(z|o_t)} \log(p(o_t|z)) - D_{\text{KL}}[q(z|o_t)||p(z)]\Big\} \tag{4.2}$$

$$= -\Big\{\mathbb{E}_{q(z|h_t)} \log(p(f_{1:t}|z)) - D_{\text{KL}}[q(z|h_t)||p(z)]\Big\}. \tag{4.3}$$

In equation 4.3, the prior $p(z)$ is a Gaussian distribution with zero mean and unit variance. To obtain expressive posterior $q(z|h_t)$, we use normalizing flows in Partial VAE [95]. We explain how normalizing flows capture more complex and expressive posterior in Section 2.2. Here, we specifically use auto-regressive Neural Spline Flows (NSF) [50]. Be-

tween the two flow layers, we flip the input [47] and normalize it using ActNorm [94]. In Figure 4.1, the flow-based encoder S infers the posterior $q(z|h_t) = S(h_t)$.

In a Partial VAE, $p(f|z) = p(f_{1:t}|z)p(f^c_{1:t}|z)$; where $f^c_{1:t}$ are the features of the glimpses other than the ones observed up to t and $f = f_{1:t} \cup f^c_{1:t}$. We implement a decoder D that synthesizes a feature map containing features of all glimpses in an image given the sample $z \sim q(z|h_t)$, *i.e.*, $\tilde{f} = D(z)$. Let m_t be a binary mask with value $\mathbf{1}$ for the glimpses observed by the model up to t and $\mathbf{0}$ otherwise; hence, $f_{1:t} = m_t \odot f$, where $\odot$ is an element-wise multiplication (see Figure 4.2(b)). We assume a Gaussian likelihood and evaluate the log-likelihood in equation 4.3 using the mask m_t as follows.

$$-\log(p(f_{1:t}|z)) \propto \frac{1}{2}\sum \frac{|m_t \odot \tilde{f} - m_t \odot f|^2}{\sigma^2} + \log(\sigma^2), \tag{4.4}$$

where σ is a model parameter. The BOED uses $\tilde{f}$ to find an optimal location to attend.

4.1.2 Bayesian Optimal Experiment Design (BOED)

The BOED evaluates the optimality of a set of experiments by measuring information gain in the interest parameter due to the experimental outcome [31]. In hard attention, an experiment is to attend a location l and observe a corresponding glimpse $g = g(x, l)$. An experiment of attending to a location l is optimal if it gains maximum information about the class label y. We can evaluate the optimality of attending to a specific location by measuring several metrics such as feature variance [76], uncertainty in the prediction [124], expected Shannon information [117]. For a sequential model, information gain $D_{\mathrm{KL}}[p(y|g, l, h_{t-1})||p(y|h_{t-1})]$ is an ideal metric. It measures the change in the entropy of the class distribution from one time step to the next due to the observation of a glimpse g at location l [18, 121]. Since PAM has to find an optimal location to attend at time

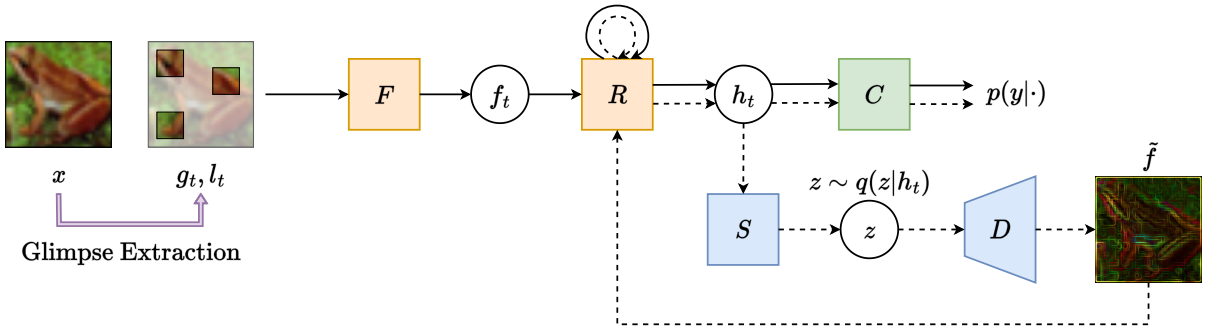


Figure 4.3: *Overview of our PAM.* At time t, PAM actively observes a glimpse-location pair (g_t, l_t), updates hidden state h_t, and predicts the class distribution $p(y|h_t)$. At time $t+1$, PAM assesses various candidate locations l before attending an optimal one. It predicts $p(y|g, l, h_t)$ ahead of time and selects the candidate l that maximizes $D_{\mathrm{KL}}[p(y|g, l, h_t)||p(y|h_t)]$. PAM synthesizes features of g using a Partial VAE to approximate $p(y|g, l, h_t)$ without attending to the glimpse g. The normalizing flow-based encoder S predicts the approximate posterior $q(z|h_t)$. The decoder D uses a sample $z \sim q(z|h_t)$ to synthesize a feature map $\tilde{f}$ containing features of all glimpses. PAM uses $\tilde{f}(l)$ as features of a glimpse at location l and evaluates $p(y|g, l, h_t) \approx p(y|\tilde{f}(l), h_t)$. Dashed arrows show a path to compute the 'lookahead' class distribution $p(y|\tilde{f}(l), h_t)$.

t before observing the corresponding glimpse, we consider an expected information gain (EIG) over the generating distribution of g. Note, EIG is also a measure of Bayesian surprise [80, 153]. We briefly introduced BOED and EIG in Section 2.3.

The EIG over $p(g|\cdot)$ is given by

$$EIG(l) = \mathbb{E}_{p(g|l,h_{t-1})} D_{\mathrm{KL}}\big[p(y|g, l, h_{t-1})||p(y|h_{t-1})\big] \tag{4.5}$$

$$= \mathbb{E}_{p(f(l)|h_{t-1})} D_{\mathrm{KL}}\big[p(y|f(l), h_{t-1})||p(y|h_{t-1})\big], \tag{4.6}$$

where $f(l)$ are features of a glimpse located at l, i.e. $f(l) = F(g, l)$. Inspired by [67], we

define $p(f(l)|h_{t-1})$ as follows.

$$p(f(l)|h_{t-1}) = \mathbb{E}_{q(z|h_{t-1})} p(f(l)|z) \tag{4.7}$$

$$= \mathbb{E}_{q(z|h_{t-1})} \delta(D(z)(l)) \tag{4.8}$$

$$= \mathbb{E}_{q(z|h_{t-1})} \delta(\tilde{f}(l)), \tag{4.9}$$

where $\delta(\cdot)$ is a delta distribution. As discussed in the section 4.1.1, flow-based encoder S predicts the posterior $q(z|h_{t-1})$ and the decoder D predicts the feature map containing features of all glimpses in an image, $\tilde{f} = D(z)$. Combining equation 4.6 and equation 4.9 yields,

$$EIG(l) = \mathbb{E}_{q(z|h_{t-1})} D_{\mathrm{KL}}\big[p(y|\tilde{f}(l), h_{t-1})||p(y|h_{t-1})\big]. \tag{4.10}$$

To find an optimal location to attend at time t, the model compares various candidates for l_t. It predicts $EIG(l)$ for each candidate l and selects an optimal candidate as l_t, i.e. $l_t = \mathrm{argmax}_l\, EIG(l)$. When the model is considering a candidate l, it uses $\tilde{f}(l)$ to calculate $\tilde{h}_t = R(h_{t-1}, \tilde{f}(l))$ and $p(y|\tilde{h}_t) = C(\tilde{h}_t)$. It uses the distribution $p(y|\tilde{h}_t) = p(y|\tilde{f}(l), h_{t-1})$ to calculate EIG in equation 4.10. We refer to $p(y|\tilde{h}_t)$ as the *lookahead* class distribution computed by anticipating the content at the location l ahead of time. The dashed arrows in Figure 4.3 show a 'lookahead' step. Furthermore, to compute EIG for all locations simultaneously, we implement all modules of our model with convolution layers. The model calculates EIG for all locations as a single activation map in a single forward pass. An optimal location is equal to the pixel coordinates with maximum value in the EIG map.

4.2 Experiment Setup

4.2.1 Datasets

We evaluate our PAM on SVHN [128], CINIC-10 [39], CIFAR-10 [102], CIFAR-100 [102], and TinyImageNet [111] datasets. These datasets consist of real-world images categorized into 10, 10, 10, 100, and 200 classes, respectively. Images in TinyImageNet are of size 64×64, and images in the remaining dataset are of size 32×32.

Glimpses. PAM runs for $T = 7$ time steps. It senses glimpses of size 16×16 overlapping with stride $n = 8$ for TinyImageNet and glimpses of size 8×8 overlapping with stride $n = 4$ for the remaining datasets. The Partial VAE predicts $\tilde{f}$ and EIG for a set of glimpses separated with stride equal to n. We do not allow our model to revisit glimpses attended in the past.

4.2.2 Implementation

Table 4.1 shows the architecture of various components of our model. Note that all modules use a small number of layers. F_g uses the least possible layers to achieve the effective receptive field equal to the area of a single glimpse. The decoder D uses the smallest number of $ConvTranspose$ and $Conv$ layers to generate the feature maps of required spatial dimensions and refine them based on the global context. The encoder S uses flow layers according to the complexity of the dataset. All other modules use a single linear layer. We implement linear layers using 1×1 convolution layers. Table 4.2 lists the dimensionality of features f, hidden state h, and latent representation z for various datasets.

Building Block	Module	Architecture
Recurrent feature aggregator	$\mathcal{F}_l$	$Conv_{k=1}(\cdot)$
	$\mathcal{F}_g$	$n_g \times \{BN(LeakyReLU(Conv_{k=3}(\cdot))\}$ $Conv_{k=2}(\cdot)$
	$F(g,l)$	$\mathcal{F}_g(g) + \mathcal{F}_l(l)$
	$\mathcal{F}_h$	$Conv_{k=1}(\cdot)$
	$\mathcal{F}_f$	$Conv_{k=1}(BN(LeakyReLU(\cdot)))$
	$R(h,f)$	$LN(LeakyReLU(\mathcal{F}_h(h) + \mathcal{F}_f(f)))$
Classifier	C	$Softmax(Conv_{k=1}(Dropout_{p=0.5}(\cdot)))$
Partial VAE	S	$n_s \times (ActNorm(Flip(NSF(\cdot))))$
	D	$3 \times \{LN(LeakyReLU(ConvTranspose_{k=3}(\cdot)\}$ $5 \times \{LN(LeakyReLU(Conv^{p=1}_{k=3}(\cdot)\}$ $Conv^{p=1}_{k=3}(\cdot)$

Table 4.1: *Architecture of our PAM.* $k = kernel\_size$, $p = padding$. n_g is set to 3 for SVHN, CINIC-10, CIFAR-10 and CIFAR-100, and set to 7 for TinyImageNet. n_s is set to 4 for SVHN, CINIC-10 and CIFAR-10, and set to 6 for CIFAR-100 and TinyImageNet. BN = Batch Normalization [79]. LN = Layer Normalization [14]. NSF = Neural Spline Flows [50]. ActNorm layer is presented in [94]. Reprinted, with permission, from [142].

4.2.3 Optimization

We train our PAM in three stages.

- **Stage I.** We pre-train modules F, R, and C with a random sequence of T glimpses using $\sum_{t=0}^{T-1} \mathcal{L}_{CE}(t)$ as a training objective.

- **Stage II.** We introduce S and D. We pre-train S and D while keeping F, R, C frozen. Again, we use a random sequence of T glimpses and train S and D using $\sum_{t=0}^{T-1} \mathcal{L}_{\text{PVAE}}(t)$ training criterion. To produce the target f used in equation 4.4, we feed a complete image and a grid of all locations to the pretrained F, which computes features of all glimpses as a single feature map. Pre-training (F, R, C) and (S, D) separately ensures that the Partial VAE receives a stable target feature

	f	h	z
SVHN	128	512	256
CINIC-10	128	512	256
CIFAR-10	128	512	256
CIFAR-100	512	2048	1024
TinyImageNet	512	2048	1024

Table 4.2: *Dimensionality of features f, hidden state h and latent representation z.* Reprinted, with permission, from [142].

map f in equation 4.4.

- **Stage III.** We fine-tune all modules end-to-end using the training objective $\mathcal{L} = \sum_{t=0}^{T-1} \alpha \mathcal{L}_{\text{PVAE}}(t) + \beta \mathcal{L}_{CE}(t)$, where α and β are hyperparameters. We choose the hyperparameters α and β such that the two loss terms contribute equally. We set α to the inverse of dimensionality of the latent representation z. Further, we set β to 32, 16, 16, 8, and 8 for SVHN, CINIC-10, CIFAR-10, CIFAR-100, and TinyImageNet, respectively. In the finetuning stage, we sample an optimal sequence of glimpses using the BOED framework.

We trained our models on a single Tesla P100 GPU with 12GB of memory or a single Tesla V100 GPU with 16GB of memory. Next, we describe our training setup. We use the same setting for all three stages unless stated otherwise.

Data Preparation. For all datasets, we augment training images using random crop, scale, horizontal flip, and color jitter transformations and map pixel values in the range $[-1, 1]$. We use a batch size of 64.

Optimizer. We use Adam optimizer [92] with the default setting of $(\beta_1, \beta_2) = (0.9, 0.999)$. We use a learning rate of 0.001 for all datasets in the first training stage. In the second and third training stages, we use a learning rate of 0.001 for SVHN, CIFAR-10, and CINIC-10 and a learning rate of 0.0001 for CIFAR-100 and TinyImageNet. We divide

Algorithm 2 Inference using PAM

Randomly sample l_0; Capture g_0 at l_0, compute f_0, h_0 and $p(y|h_0)$
for $t \in \{1, \ldots, T-1\}$ **do** ▷ T is the time budget
    Sample $z_i \sim q(z|h_{t-1})$ and predict $\tilde{f}^i; i \in \{0, \ldots, P-1\}$ ▷ P is the sample budget
    Compute $\tilde{h}_t^i$, $p(y|\tilde{h}_t^i)$ and $EIG = \frac{1}{P}\sum_i D_{\mathrm{KL}}[p(y|\tilde{h}_t^i)||p(y|h_{t-1})]$ ▷ (4.10)
    $l_t = \mathrm{argmax}\{EIG\}$
    Capture g_t at l_t; Compute f_t, h_t and $p(y|h_t)$
end for

the learning rate by 0.5 at a plateau.

EIG estimation. We use only one sample $z \sim q(z|h_t)$ to estimate the EIG map during the training, which leads to exploration. In the test phase, we achieve exploitation by using $P = 20$ samples of z to estimate the EIG accurately.

The test procedure is shown in Algorithm 2.

4.3 Results

4.3.1 Baseline Comparison

We compare our PAM with four baselines in Figure 4.4. RAM is a state-of-the-art hard attention model that observes images partially and sequentially to predict the class labels [125]. We implement RAM using the same structure as our model. Instead of the Partial VAE, RAM has a controller that learns a Gaussian attention policy. Mnih *et al.*[125] minimize $\mathcal{L}_{CE}$ at the end of T steps. Following Li *et al.* [113], we improve RAM by minimizing $\mathcal{L}_{CE}$ at all T steps. We refer to this baseline as RAM+. We also consider a baseline model that attends to glimpses at random locations. The Random baseline does not have a controller or a Partial VAE. Our PAM and the three baselines described so far observe the image only partially through glimpses. Additionally, we train a feed-forward CNN that observes the entire image to predict the class label.

For the SVHN dataset, the Random baseline outperforms RAM for initial time steps.

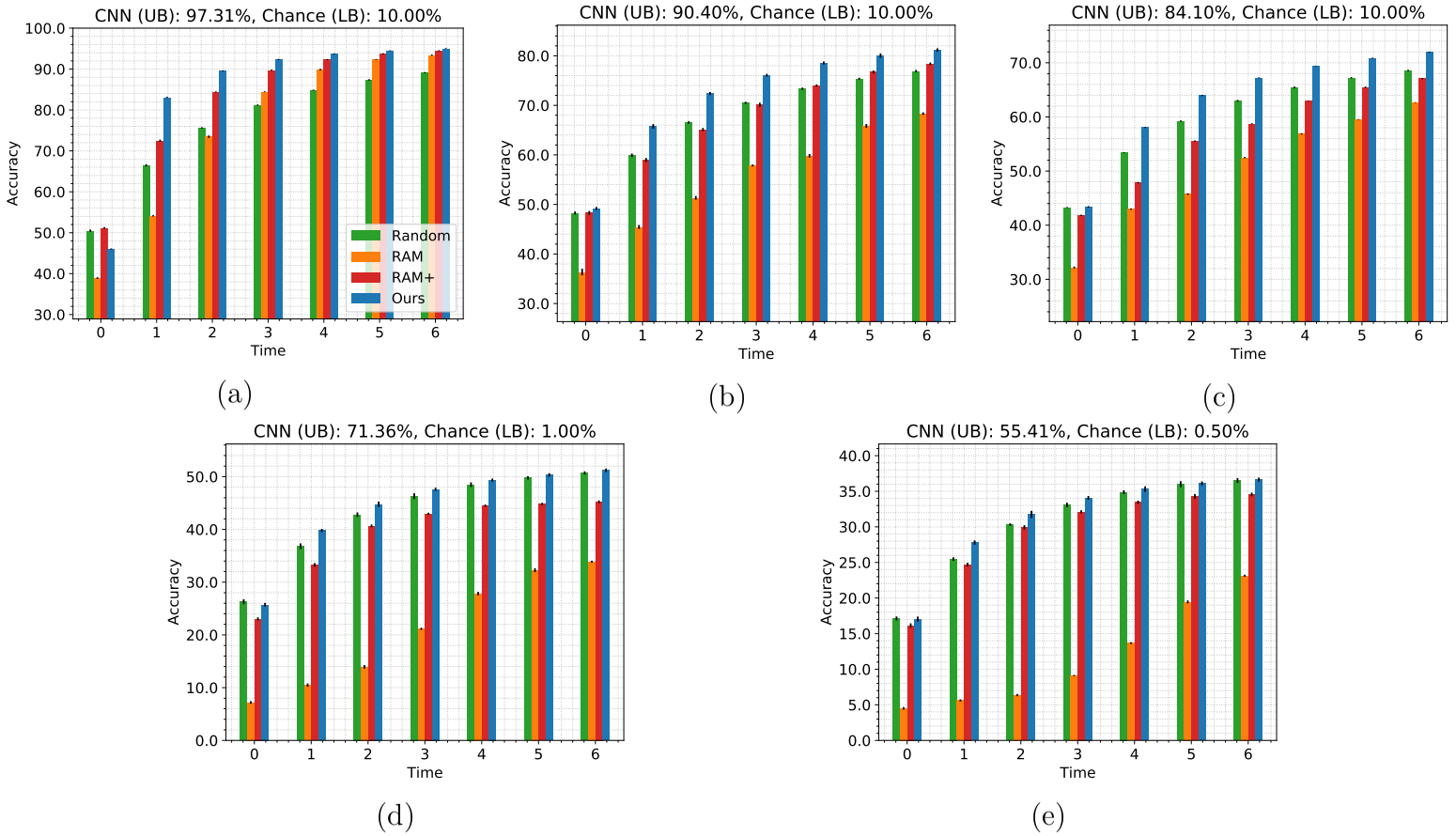


Figure 4.4: *Baseline Comparison on (a) SVHN (b) CIFAR-10 (c) CINIC-10 (d) CIFAR-100 (e) TinyImageNet. Ablation study on normalizing flows on (f) TinyImageNet.* (a-e) We compare various methods for $t = 0$ to 6. All results are averaged over ten different runs. Accuracy of a CNN and chance accuracy, displayed on top of each plot, serve as upper and lower bounds for the accuracy of glimpse-based methods. A CNN observes an entire image, whereas glimpse-based methods observe $< 43.75\%$ area of the image by $t = 6$. When compared with baseline methods, our method achieves the highest accuracy. Reprinted, with permission, from [142].

However, with time, RAM outperforms the Random baseline by attending to more useful glimpses. RAM+ outperforms RAM and the Random baselines at all time steps. We observe a different trend for non-digit datasets. RAM+ consistently outperforms RAM on non-digit datasets; however, RAM+ falls behind the Random baseline. In RAM and RAM+, the classifier shares latent space h_t with the controller, while the Random baseline dedicates an entire latent space to the classifier. We speculate that the dedicated latent space in the Random baseline is one of the reasons for its superior performance on complex datasets.

Our PAM consistently outperforms all attention baselines on all datasets. As one can expect, the performance gap between the highest-performing baseline and our PAM reduces with many glimpses. Predicting an optimal glimpse location is difficult for early time steps as the models have access to minimal information about the scene so far. Compared to the highest performing baseline at $t = 1$, our PAM achieves around 10% higher accuracy on SVHN, around 5-6% higher accuracy on CIFAR-10 and CINIC-10, and around 2-3% higher accuracy on CIFAR-100 and TinyImageNet. Note that CIFAR-100 and TinyImageNet are more challenging datasets compared to SVHN, CINIC-10, and CIFAR-10. Similar to RAM and RAM+, the classifier and the Partial VAE share a common latent space in our model. Hence, our model achieves a lower gain over the Random baseline for complex datasets. We attribute the small but definite improvement in the accuracy of our model to a better selection of glimpses.

CNN has the highest accuracy as it observes a complete image. The accuracy of the CNN serves as the upper bound for the hard attention methods. Unlike CNN, the hard attention methods observe less than half of the image through small glimpses, each uncovering only 6.25% area of an image. On the CIFAR-10 dataset, the CNN predicts correct class labels for approximately 9 out of 10 images after observing each image

completely. Remarkably, our model predicts correct class labels for 8 out of 10 images after observing less than half of the total area in each image.

Comparison of Attention Policies using a common CNN

Above, we compared the attention policies of various methods using their respective classifiers. However, each model attains different discriminative power due to different training objectives. While RAM, RAM+, and our PAM are trained jointly for two different tasks, i.e., classification and glimpse-location prediction or feature synthesis, the Random baseline is trained for only one task, i.e., classification. Consequently, the Random baseline attains higher discriminative power than others and achieves high accuracy despite using a sub-optimal attention policy. To make a fair comparison of the attention policies irrespective of the discriminative power of the models, we perform the following experiment.

We mask all image regions except those observed by the attention model so far and let the baseline CNN predict a class label from this masked image (see Figure 4.5). RAM+ consistently outperforms RAM, suggesting that the former has learned a better attention policy than the latter. As RAM+ is trained using $\mathcal{L}_{CE}$ at all time steps, it achieves higher accuracy and, ultimately, higher reward during training with REINFORCE [125]. RAM and RAM+ outperform the Random baseline for the SVHN dataset. However, they fall short on natural image datasets. In contrast, our PAM outperforms all baselines with a significant margin on all datasets, suggesting that the glimpses selected by our PAM are more informative about the image class than the ones chosen by the baselines.

RAM and RAM+ struggle on images with many objects and repeated structure [157], as is often the case with natural image datasets. For example, TinyImageNet includes many images with multiple entities (*e.g.*, beer bottle), repeated patterns (*e.g.*, spider

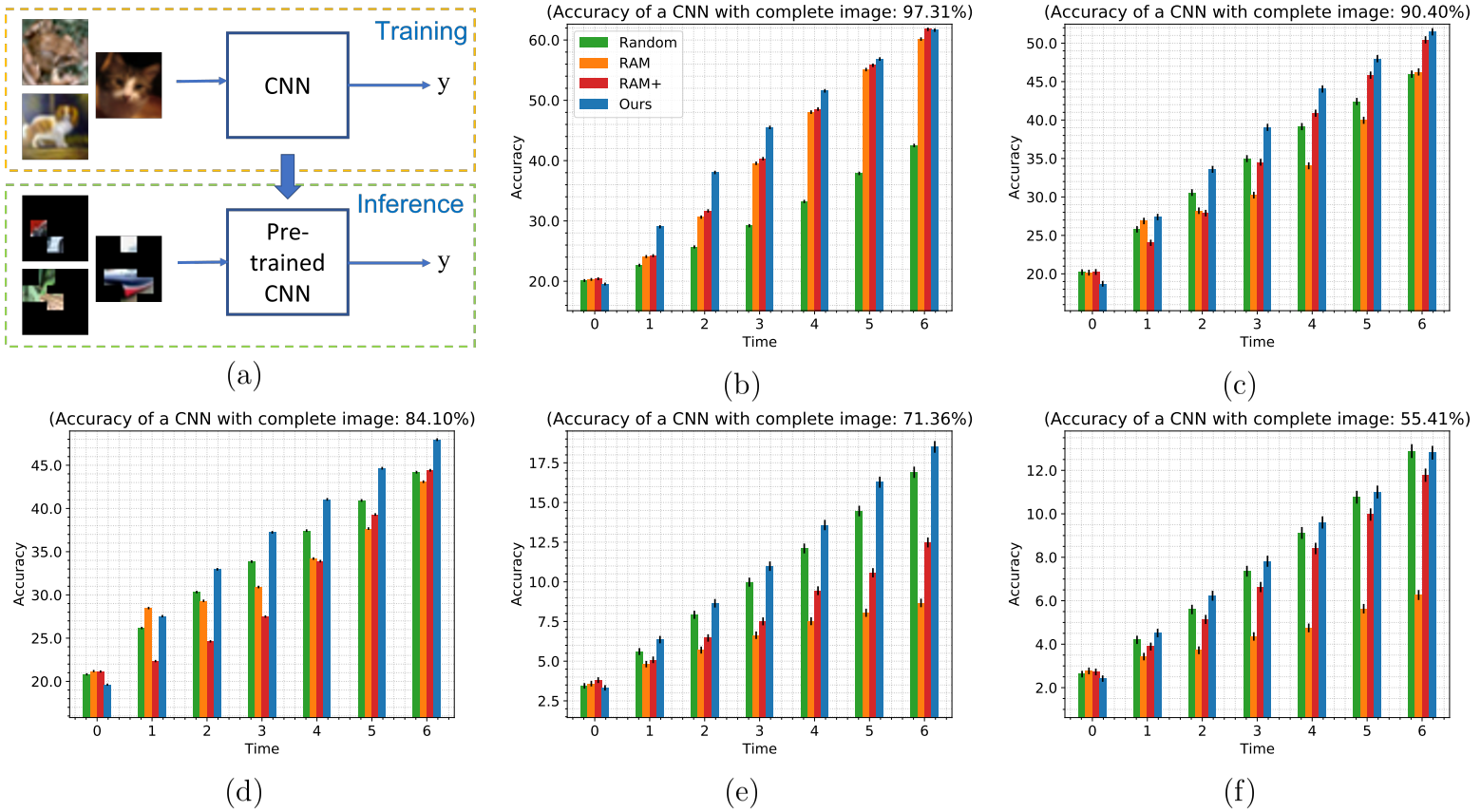


Figure 4.5: *Accuracy of a CNN when only the attended glimpses are observed.* (a) Experiment setup: we train a CNN using complete images and test it on masked images with only the glimpses attended by various methods made visible. Results on (b) SVHN (c) CIFAR-10 (d) CINIC-10 (e) CIFAR-100 (f) TinyImageNet. All results are averaged over three runs. Reprinted, with permission, from [142].

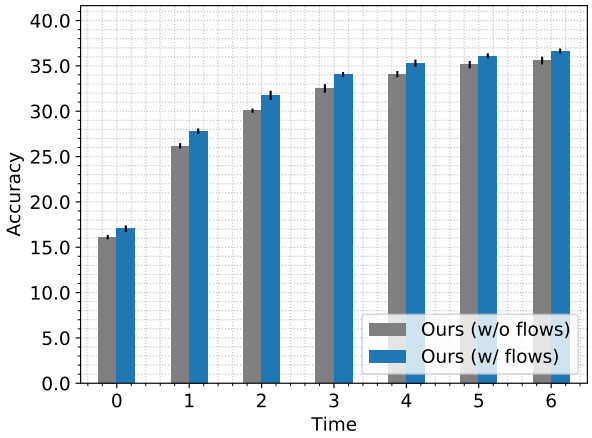


Figure 4.6: *Normalizing flows improve accuracy on TinyImageNet.* We evaluate our PAM with and without normalizing flows. Without normalizing flows, the encoder predicts unimodal Gaussian posterior. The normalizing flows-based encoder predicts multimodal posterior and achieves higher performance. Reprinted, with permission, from [142].

web), and dispersed items (*e.g.*, altar). Note that a random policy can perform competitively in such scenarios, especially when the location of various objects in an image is unknown due to partial observability. Yet, our method learns policies superior to the Random baseline.

4.3.2 Ablation study on Normalizing Flows

Synthesizing a feature map of a complete image using only partial observations is an ill-posed problem with many solutions. We use normalizing flows in the encoder S to capture a complex multimodal posterior $q(z|h_t)$ that helps the decoder predict multiple plausible feature maps for a complete image. Here we inspect the need for such a flexible posterior $q(z|h_t)$ and, therefore, the necessity of normalizing flows in the encoder S. To this end, we model the posterior with a unimodal Gaussian distribution and let S output the mean and diagonal covariance of a Gaussian. We do not use flow layers in this case. Figure 4.6 shows the result for the TinyImageNet dataset. We observe that modeling a

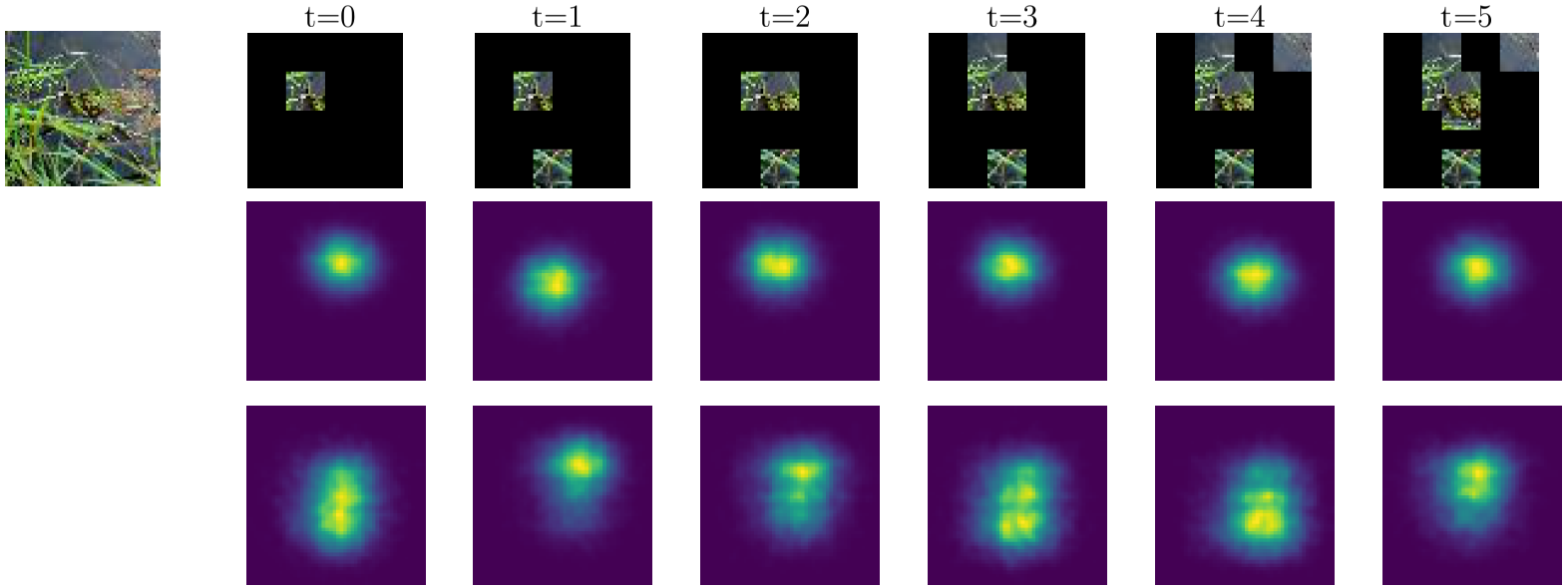


Figure 4.7: *TSNE projection of $q(z|h_t)$ estimated with and without normalizing flows for an example image from TinyImageNet dataset.* A complete image is shown in the first column for reference. Our model never observes a complete image. In columns two to seven: (top) glimpses observed by the models to compute h_t (middle) TSNE projection of $q(z|h_t)$ estimated without using normalizing flows. (bottom) TSNE projection of $q(z|h_t)$ estimated using normalizing flows. Normalizing flows capture a complex multimodal posterior. Reprinted, with permission, from [142].

complex posterior using normalizing flows improves accuracy. Ideally, the Partial VAE should predict all possibilities of $\tilde{f}$ consistent with the observed region o_t. When the model observes a small region, a complex posterior helps determine multiple plausible feature maps. In Figure 4.7, we present TSNE [176] projections of $q(z|h_t)$ estimated with and without the use of normalizing flows. We observe that the normalizing flows capture a complex multimodal posterior. A unimodal posterior fails to cover all possibilities. Since normalizing flows capture multiple modes, they facilitate accurate estimation of EIG and, consequently, higher performance.

4.3.3 Visualization

We visualize a few interesting examples of sequential recognition from the CIFAR-10 dataset in Figure 4.8. In Figure 4.8(a), activity in the EIG map reduces as the model settles on a class 'Bird'. In Figure 4.8(b), PAM requires a few glimpses to decide the true class of the image. Though incorrect, it predicts classes that are types of vehicles. After discovering parts like the headlight and rear-view mirror, it predicts the true class 'Automobile' at $t = 6$. Figure 4.8(c) shows a difficult example. PAM decides the true class 'Bird' after covering parts of the bird at $t = 5$. Notice a high amount of activity in the EIG maps up to $t = 5$ and reduced activity at $t = 6$. Figure 4.8(d) shows a failure case with enduring activity in EIG maps. Finally, observe that the EIG maps are often multimodal.

4.4 Conclusions

We presented a Probabilistic Attention Model (PAM). PAM is a hard attention model and uses BOED to find the optimal locations to attend to when the image is observed only partially. To find an optimal location without observing the corresponding glimpse,

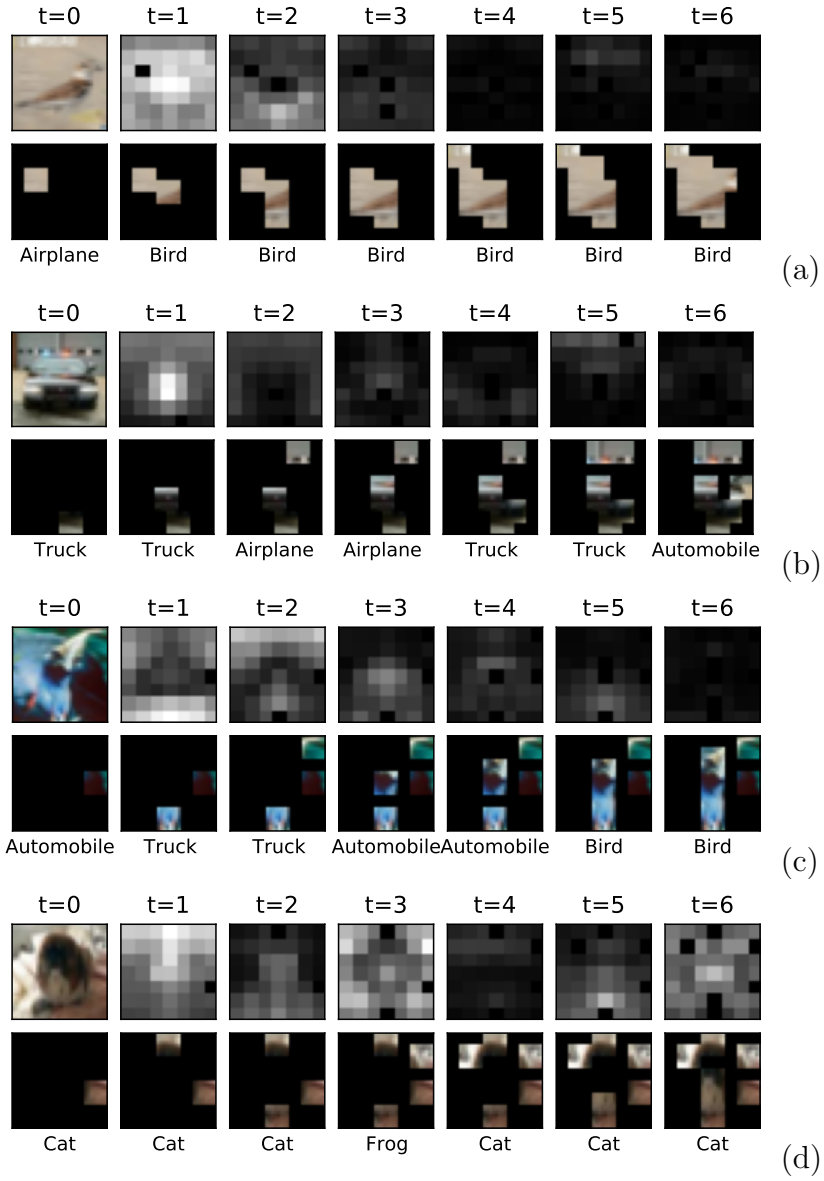


Figure 4.8: *Visualization of the EIG maps and the glimpses observed by our PAM on CIFAR-10 images.* The top rows in each plot show the entire image and the *EIG* maps for $t = 1$ to 6. The bottom rows in each plot show glimpses attended by our PAM. PAM observes the first glimpse at a random location. It observes a glimpse of size 8×8. The glimpses overlap with the stride of 4, resulting in a 7×7 grid of glimpses. The EIG maps are of size 7×7 and are upsampled for the display. We display the entire image for reference; our PAM never observes the whole image. (a-c) success cases (d) failure case. Reprinted, with permission, from [142].

PAM uses Partial VAE to synthesize the content of the glimpse in the feature space. Synthesizing features of unobserved regions is an ill-posed problem with multiple solutions. We use normalizing flows in Partial VAE to capture a complex distribution of unobserved glimpse features, which leads to improved performance. The synthesized features enable our PAM to evaluate and compare the expected information gain (EIG) of various candidate locations, from which it selects a candidate with optimal EIG. The predicted EIG maps are often multimodal. Consequentially, the attention policy used by our PAM is multimodal. Our PAM achieves superior performance compared to the baseline methods that use unimodal attention policy, proving the effectiveness of multimodal policies in hard attention. When all models have seen only a couple of glimpses, our PAM achieves 2-10% higher accuracy than the baselines.

5

Sequential Transformers Attention Model

Note: Significant parts of this chapter are adapted, with permission, from:

[141] *Consistency driven sequential transformers attention model for partially observable scenes.*
Samrudhdhi B Rangrej, Chetan L Srinidhi, and James J Clark.
IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

The IEEE does not require individuals working on a thesis to obtain a formal reuse

license. However, it requires that the thesis author cite the source and include the IEEE copyright notice for all figures and tables [57].

In this chapter, we discuss our Sequential Transformers Attention Model (STAM) for glimpse-based image classification. Similar to PAM, our STAM never observes a complete image. It classifies an image solely based on glimpses. Unlike PAM, STAM is constructed using transformers and scales to handle large datasets. Specifically, we design our STAM agent using DeiT-distilled[172]. Transformers [178, 49] efficiently model long-range dependencies and are ideal for aggregating information from distant glimpses.

Starting from observing a glimpse at a random location, our STAM predicts an optimal location for the next glimpse and class-label of an image based on the glimpses collected so far. As glimpse acquisition is a discrete and non-differentiable process, we train our STAM using reinforcement learning (RL). Further, we propose an additional training objective where STAM is required to predict a class distribution from a set of glimpses consistent with the class distribution predicted from a complete image. To do so, we employ a teacher transformers model to predict the class distribution from a complete image and our STAM (a student model) tries to reproduce this distribution using partial observations. We perform experiments on two large-scale real-world datasets, namely, ImageNet [149] and fMoW [34]. With only 4% of the total image area observed, our proposed consistency objective yields $\sim$3% and $\sim$8% gain in accuracy on ImageNet and fMoW, respectively. Further, our STAM outperforms the previous state-of-the-art on ImageNet and fMoW by observing nearly 27% and 42% fewer pixels in glimpses, respectively.

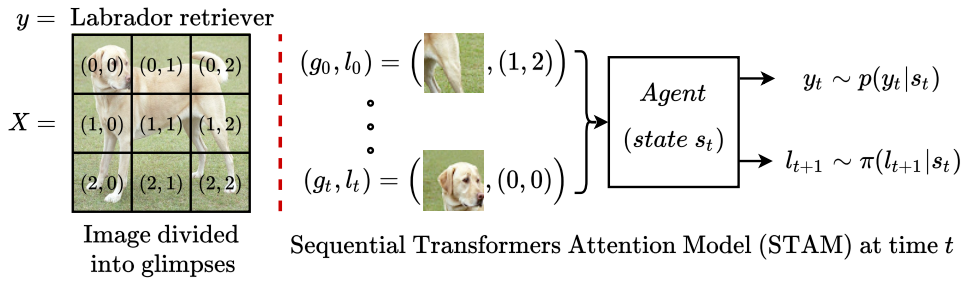


Figure 5.1: *Schematic diagram of Sequential Transformers Attention Model (STAM).* We divide an image (X) into equally-sized non-overlapping glimpses. STAM sequentially observes informative glimpses (g_t) from an image. While never observing an image entirely, STAM predicts the class-label of an image (y) based on glimpses. At each t, our agent encodes past glimpses and their locations ($g_{0:t}, l_{0:t}$) into a Markov state s_t. It uses state s_t to predict class distribution $p(y_t|s_t)$ and attention policy $\pi(l_{t+1}|s_t)$. We sample the next glimpse location l_{t+1} from $\pi(l_{t+1}|s_t)$. © [2022] IEEE. Reprinted, with permission, from [141].

5.1 Model

Given an *unobserved* scene X, STAM actively captures a series of non-overlapping glimpses and, while *never* observing X completely, it predicts the class-label y of X based on glimpses. A schematic diagram of our agent is shown in Figure 5.1. At time t, the agent senses a glimpse g_t at location l_t from an image X. Using the glimpses observed up to time t, our agent predicts: i) y_t, an approximation of label y, and ii) l_{t+1}, the location of the next glimpse.

We model the sequential attention mechanism of our agent as a Partially Observable Markov Decision Process (POMDP). In the POMDP, our agent encodes a history of partial observations, $\{(g_{t'}, l_{t'})|\ t' \in \{0, \ldots, t\}\}$, in a Markov state s_t and maps it to: i) a class distribution $p(y_t|s_t)$ and ii) an attention policy $\pi(l_{t+1}|s_t)$ – a distribution over candidate glimpse locations for $t+1$.

We build our agent using DeiT-distilled[172], referred to as DeiT$^{\mathcal{D}}$ in the rest of the

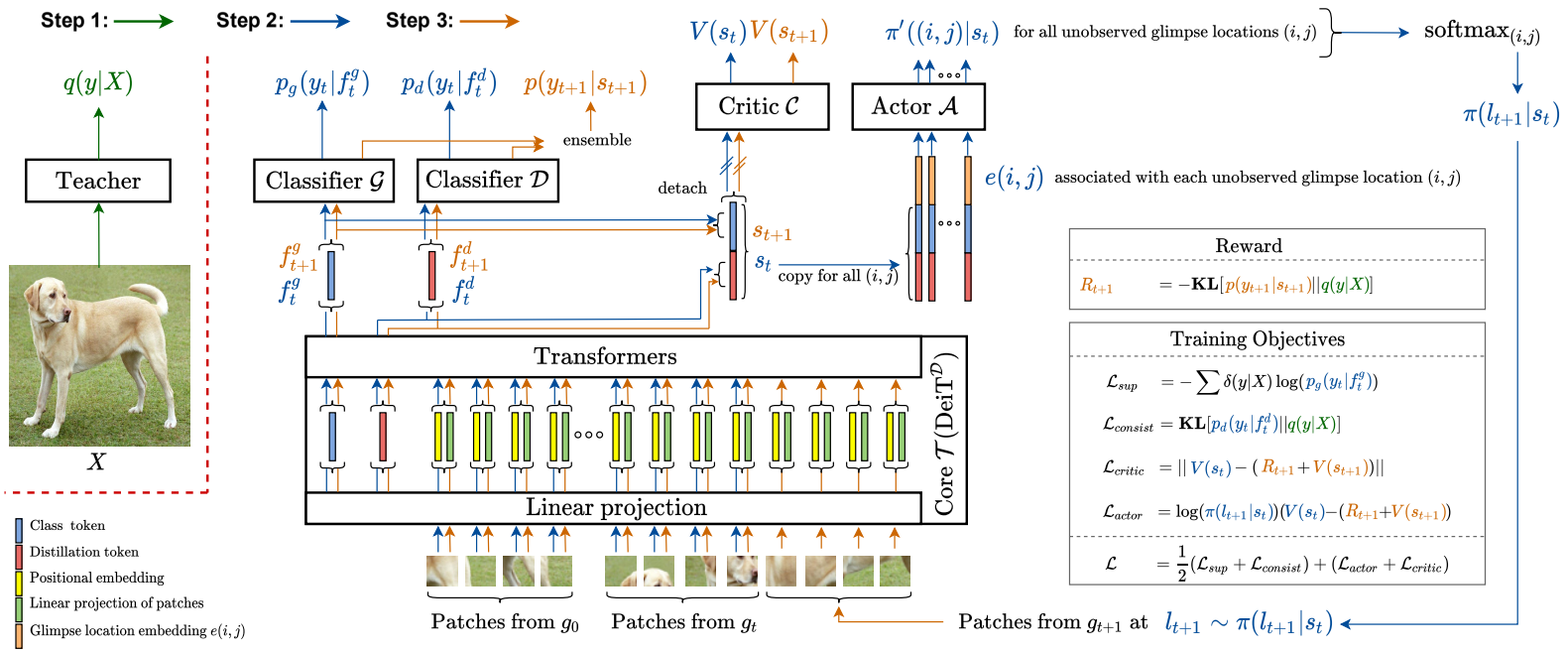


Figure 5.2: *An overview of our Sequential Transformers Attention Model (STAM).* The STAM consists of a core $\mathcal{T}$, classifiers $\mathcal{G}$ and $\mathcal{D}$, an actor $\mathcal{A}$, and a critic $\mathcal{C}$ (only used during training). We discuss the working principles of these modules in Section 5.1, except for the critic $\mathcal{C}$, which we discuss in Section 5.2. We update model parameters T times per batch using the objectives shown on the right and discussed in Section 5.2. Each training iteration consists of three steps: **Step 1** (**green path**): Given a complete image X, the teacher model predicts a soft pseudo-label $q(y|X)$. **Step 2** (**blue path**): Given glimpses $g_{0:t}$, the core $\mathcal{T}$ predicts features f_t^g and f_t^d. The classifiers $\mathcal{G}$ and $\mathcal{D}$ predict class distributions $p_g(y_t|f_t^g)$ and $p_d(y_t|f_t^d)$ from features f_t^g and f_t^d, respectively. Given a state $s_t = [f_t^g; f_t^d]$, the critic $\mathcal{C}$ predicts value $V(s_t)$ and the actor $\mathcal{A}$ predicts attention policy $\pi(l_{t+1}|s_t)$. The actor predicts logits $\pi'((i,j)|s_t)$ for all unobserved glimpse locations (i,j) in a conditionally independent manner and applies softmax to the logits resulting in $\pi(l_{t+1}|s_t)$. **Step 3** (**orange path**): A glimpse g_{t+1} at $l_{t+1} \sim \pi(l_{t+1}|s_t)$ is sensed. Using the glimpses $g_{0:t+1}$, the ensemble class distribution $p(y_{t+1}|s_{t+1})$ and value $V(s_{t+1})$ are computed following the same path as Step 2. The model parameters are updated using the gradients from Step 2. In practice, Step 1 is performed once per batch at $t = 0$, whereas, Steps 2-3 are performed T times per batch. © [2022] IEEE. Reprinted, with permission, from [141].

chapter. Briefly, DeiT$^{\mathcal{D}}$ is a type of ViT trained using knowledge distillation. The transformers in DeiT$^{\mathcal{D}}$ transform an input sequence of a class token, a distillation token, and patch tokens (linear projections of the image patches added to the positional embeddings) to an output sequence; with the outputs corresponding to the class and the distillation tokens, the two classifiers predict the ground truth and the teacher's prediction, respectively. In our approach, we adapt the DeiT$^{\mathcal{D}}$ to predict labels from glimpses and use the distillation token to impose consistency. We discuss Vision Transformers, including DeiT$^{\mathcal{D}}$, at greater length in Section 2.7.

Figure 5.2 shows the model of our agent. Our agent is composed of the following components.

Sensor. We consider a sensor that captures non-overlapping glimpses from a scene. To model this sensor, we divide the image X into $N \times N$ equally sized non-overlapping blocks, $X = \{X(i,j)|\ i,j \in \{1,\ldots,N\}\}$. Given a location $l_t = (i,j)$, a sensor senses a glimpse $g_t = X(i,j)$, as shown in Figure 5.1.

Core ($\mathcal{T}$). At time t, we extract $M \times M$ patches from each glimpse observed up to t, forming a set of $t \times M \times M$ patches. We feed these patches, the positional embeddings, the class token, and the distillation token to the DeiT$^{\mathcal{D}}$ model. The positional embedding represents the position of a patch in an image. We derive the position of a patch from the location of a parent glimpse in an image (see Section 2.7 and Figure 2.8). Among the outputs of the final transformer block, let us define the ones corresponding to the class token as f_t^g and the distillation token as f_t^d. We then form a Markov state s_t by concatenating f_t^g and f_t^d, which an actor module will later use to predict an attention policy.

Classifiers ($\mathcal{G}$ and $\mathcal{D}$). As in DeiT$^{\mathcal{D}}$, we use two linear classifiers to predict two class distributions $p_g(y_t|f_t^g)$ and $p_d(y_t|f_t^d)$ from f_t^g and f_t^d, respectively. We treat the predicted

Algorithm 3 Inference using STAM

Initialize l_0 randomly;
for $t \in \{0, \ldots, T-1\}$ **do**
    Sample g_t at l_t from an image ▷ **Sensor**
    $f_t^g, f_t^d = \mathcal{T}(g_{0:t}, l_{0:t})$; $s_t = [f_t^g; f_t^d]$ ▷ **Core**
    $p_g(y_t|\cdot) = \mathcal{G}(f_t^g)$; $p_d(y_t|\cdot) = \mathcal{D}(f_t^d)$ ▷ **Classifiers**
    $\pi'(l'|\cdot) = \mathcal{A}(s_t, l')$, $\forall l' \in \{\{1, .., N\}^2 - l_{0:t}\}$ ▷ **Actor**
    $y_t = argmax(p_g(y_t|\cdot) + p_d(y_t|\cdot))$
    $l_{t+1} = argmax(\pi'(l'|\cdot))$
end for

distributions independently during training and average them to form an ensemble distribution during inference [172]:

$$p(y_t|s_t) = \frac{1}{2}(p_g(y_t|f_t^g) + p_d(y_t|f_t^d)). \tag{5.1}$$

Actor ($\mathcal{A}$)**.** An actor MLP predicts attention policy $\pi(l_{t+1}|s_t)$. The distribution $\pi(l_{t+1}|s_t)$ is computed by applying softmax over logits $\{\pi'((i,j)|s_t)\}$, where the (i,j)s are unobserved glimpse locations. An actor predicts $\pi'((i,j)|\cdot)$ for each (i,j) in a conditionally independent manner [167, 6]. For any (i,j), the actor accepts a concatenation of glimpse location embeddings $e(i,j)$ and a Markov state s_t, and outputs $\pi'((i,j)|s_t)$. Here, the $e(i,j)$s are learnable embeddings initialized by interpolating positional embeddings of a pretrained DeiT$^{\mathcal{D}}$. We use $l_{t+1} \sim \pi(l_{t+1}|s_t)$ during training and $l_{t+1} = argmax(\pi(l_{t+1}|s_t))$ during inference.

We provide the inference steps in Algorithm 3.

5.2 Training Objectives

We train the parameters of the core ($\theta_{\mathcal{T}}$), the classifiers ($\theta_{\mathcal{G}}$ and $\theta_{\mathcal{D}}$) and the actor ($\theta_{\mathcal{A}}$) using the training objectives discussed next. Figure 5.2 illustrates the training steps of our model.

5.2.1 Learning Classification

Our agent predicts two class distributions based on input glimpses, namely, p_g and p_d, where p_g is an estimation of the ground truth class distribution associated with a complete image and p_d is an approximation of the class distribution predicted by a teacher model from a complete image. We learn p_g and p_d using the following two objectives:

Supervised Loss. As our goal is to predict y from partial observations, we learn the parameters $\{\theta_{\mathcal{T}}, \theta_{\mathcal{G}}\}$ by minimizing a cross-entropy between $p_g(y_t|s_t)$ and $\delta(y|X)$ as given by

$$\mathcal{L}_{sup} = -\sum \delta(y|X) \log(p_g(y_t|s_t)), \tag{5.2}$$

where $\delta(y|X)$ is a delta distribution indicating the ground truth label of a complete image.

Consistency Loss. To improve the performance of our agent, we enforce that the predictions made from the glimpses are consistent with the predictions made from a complete image. Furthermore, the above predictions should also be the same irrespective of the number and location of the glimpses observed so far. *Ideally*, for each t, we require our agent to produce $p_d(y_t|s_t)$ that minimize $D_{\mathrm{KL}}[p_d(y_t|s_t)||p(y|X)]$; where $p(y|X)$ is the agent's prediction after observing all glimpses from an image.

The direct optimization of the above KL divergence is difficult as the target $p(y|X)$ keeps shifting during training. To circumvent this issue, we rely on a separate teacher model to provide a stable target. Our teacher model predicts the class distribution $q(y|X)$ from a complete image; where $q(y|X)$ is commonly referred to as soft pseudo-label for X in the literature [194, 72]. The resultant consistency objective to train $\{\theta_{\mathcal{T}}, \theta_{\mathcal{D}}\}$ is given

by

$$\mathcal{L}_{consist} = D_{\mathrm{KL}}[p_d(y_t|s_t)||q(y|X)]. \tag{5.3}$$

For more details on the student-teacher training paradigm, refer to Section 2.5.

5.2.2 Learning Attention Policy

We consider attention to be modeled by a POMDP. After observing a glimpse at location $l_{t+1} \sim \pi(l_{t+1}|s_t)$, we award our agent a reward R_{t+1} indicating the utility of the observed glimpse. Our training objective is to learn $\pi(l_{t+1}|s_t)$ that maximizes the sum of future rewards, also known as return, $G_t = \sum_{t'=t+1}^{T}(R'_t)$. A majority of the existing works [125, 12, 51, 135] use the REINFORCE algorithm [190] to learn an attention policy. These methods run an agent for $t = 0$ to $T - 1$ steps to achieve R_1 to R_T and compute G_0 to G_{T-1}. At the end, the parameters of the agent are updated once to maximize the returns. Due to the quadratic complexity of the transformers, running our agent for T steps and updating the parameters just once at the end is expensive. Instead, we adopt the one-step actor-critic algorithm [164] to update the parameters at each time step. We briefly introduce one-step actor-critic in Section 2.4. Below we discuss the key elements of the algorithm in the context of hard attention.

Critic loss. To train our agent using the one-step actor-critic algorithm, we introduce a critic MLP ($\mathcal{C}$) with parameters υ. A critic learns a value function $V(s_t)$ that estimates the expected return given the current state of the agent, i.e., $\mathbb{E}_\pi[G_t]$. As $\mathbb{E}_\pi[G_t] = \mathbb{E}_\pi[R_{t+1}+G_{t+1}]$, $V(s_t)$ should be equal to $\mathbb{E}_\pi[R_{t+1}+V(s_{t+1})]$. Hence, the critic parameters υ are learned by minimizing the difference between the two quantities. In practice, we

estimate the expectation with respect to π using a single Monte-Carlo sample, yielding

$$\mathcal{L}_{critic} = ||V(s_t) - (R_{t+1} + V(s_{t+1}))||. \tag{5.4}$$

We run our agent for one additional time step to compute $V(s_{t+1})$. Note that the quantity $(R_{t+1} + V(s_{t+1}))$ acts as a target and does not contribute to the parameter update. We use the critic MLP only during training and discard it once the training is over.

Actor loss. The goal of an agent is to learn a policy that achieves the maximum return. When the agent achieves lower than the expected return by sensing a glimpse at location l_{t+1}, $\pi(l_{t+1}|s_t)$ must reduce proportional to the deficit. In other words, $\pi(l_{t+1}|s_t)$ must reduce by the factor of $(V(s_t) - (R_{t+1} + V(s_{t+1}))$; where $V(s_t)$ is an estimation of the expected return for s_t, and $(R_{t+1} + V(s_{t+1}))$ is the estimation of the expected return following glimpse at l_{t+1}. We optimize the parameters $\{\theta_{\mathcal{T}}, \theta_{\mathcal{A}}\}$ by minimizing

$$\mathcal{L}_{actor} = \log(\pi(l_{t+1}|s_t))(V(s_t) - (R_{t+1} + V(s_{t+1}))). \tag{5.5}$$

Note that $(V(s_t) - (R_{t+1} + V(s_{t+1})))$ acts as a scaling factor and does not contribute to the parameter update.

Reward. We use a reward that incentivizes the agent to predict y_t that is consistent with the label predicted by the teacher model based on a complete image. Our reward is

$$R_t = -D_{\mathrm{KL}}[p(y_t|s_t)||q(y|X)], \tag{5.6}$$

where $p(y_t|s_t)$ is computed using Equation 5.1. We expect the accuracy of the predictions made from a complete image to provide an upper bound for the accuracy of the predictions made from partial observations. The above reward encourages the agent to reach for the

upper bound.

Our overall final training objective is as follows.

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_{sup} + \mathcal{L}_{consist}) + (\mathcal{L}_{actor} + \mathcal{L}_{critic}) \tag{5.7}$$

5.3 Experiment Setup

5.3.1 Datasets

We experiment with two large-scale real-world datasets, namely, ImageNet [149] and fMoW [34]. ImageNet consists of natural images from 1000 categories. It includes ∼1.3M training images and 50K validation images. We resize the images to size 224×224. The fMoW contains satellite images from 62 categories. It holds ∼0.36M, ∼53K, and ∼64K images for training, validation, and test, respectively. We crop the images based on the bounding boxes provided with the dataset and resize the cropped images to 224×224. Unless stated otherwise, we implement and optimize STAM with the same default setting on both datasets.

5.3.2 Implementation

We divide the images into non-overlapping glimpses of size 32×32, yielding a 7×7 grid of glimpses. As required by DeiT$^{\mathcal{D}}$, we further divide each glimpse into four non-overlapping patches of size 16×16.

We use DeiT$^{\mathcal{D}}$-Small architecture for our agent unless stated otherwise. The actor and the critic MLPs are of the form {3×{FC-BN-ReLU}-FC} with hidden dimensions of 2048 and 512, respectively. We initialize the core and the classifiers using a pretrained

DeiT$^{\mathcal{D}}$[1] and initialize the actor and the critic at random. We normalize the logits $\pi'(\cdot)$ with l_2 norm for training stability and multiply them with τ before applying a softmax. We stop the gradients from the critic to our agent. We normalize the rewards to have zero mean and unit variance in each training iteration. The magnitude of the value $V(\cdot)$ varies from one time-step to the next, as it approximates the expected sum of future rewards. To learn $V(\cdot)$ of varying magnitude, we apply PopArt-style normalization [68] to the predicted values.

We use DeiT$^{\mathcal{D}}$and DeiT as teacher models for ImageNet and fMoW, respectively. For the ImageNet teacher model, we use publicly available weights. The teacher model for fMoW is first initialized with the DeiT model pretrained on ImageNet, followed by fine-tuning on the fMoW dataset for 100 epochs using the default hyperparameter setting with an additional vertical flip augmentation.

5.3.3 Optimization

Our agent runs for $T = 21$ time-steps per image, capturing one glimpse at a time. We update the model parameters T times per batch, once after each time step. To account for T updates per batch, we allow the agent to see only $1/T^{th}$ of the data during one epoch. We train our agents with the batch size (B) of 4096 for 200 epochs on ImageNet and B of 600 for 400 epochs on fMoW. The hyperparameter τ is increased linearly from 1 to 4 for the first 100 epochs and fixed to 4 for the remaining training.

We augment the training images using the Rand-Augment scheme [38] and follow the same setting as Touvron *et al.*[172]. Additionally, for fMoW, we also use random vertical flip augmentation. We train our agents using an AdamW optimizer [120] with a weight decay of 0.05. We adapt a cosine learning schedule with an initial learning rate of

[1] https://github.com/facebookresearch/deit

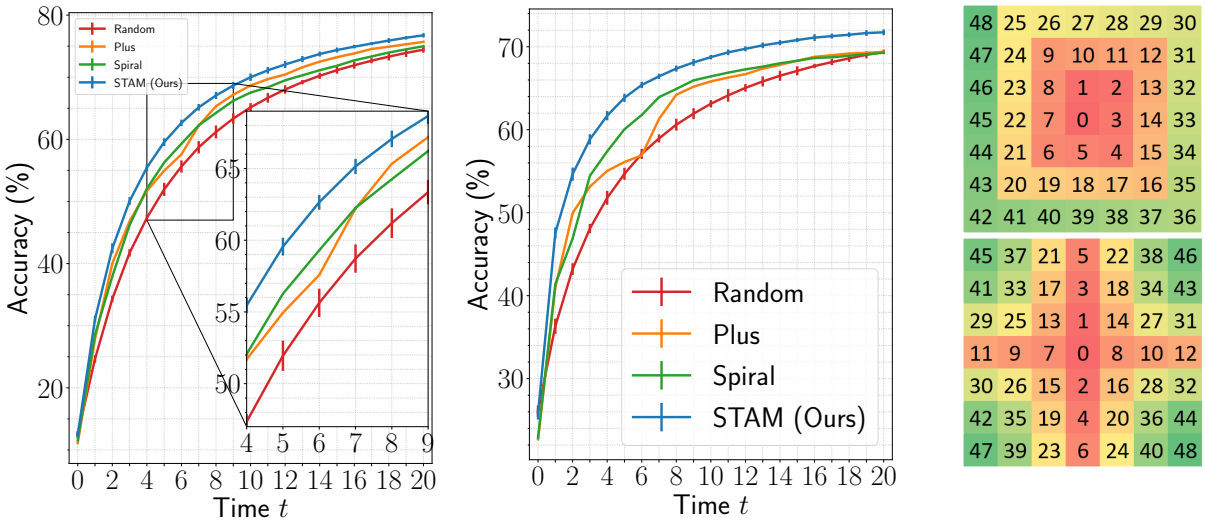
$lr_{base} \times B/512$ and a minimum learning rate of 1e-6. The base learning rate lr_{base} is set to 1e-3 for the critic module. For the remaining modules, lr_{base} is set to 1e-6 for ImageNet and 1e-5 for fMoW. We train our agents on four V100 GPUs in less than a day, using 32GB memory per GPU for ImageNet and 16GB for fMoW.

5.4 Results

5.4.1 Comparison with Baseline Attention Policies

We compare the policy learned by our agent with three baseline policies, namely, the Random, the Plus, and the Spiral. The Random agent selects the next glimpse randomly from a set of unobserved glimpses. In contrast, the Plus and the Spiral agents account for the object-centric nature of vision datasets and select glimpses in the order shown in Figure 5.3 (c). For a fair comparison, all baseline agents begin with the first glimpse at a random location. The model architecture of the baseline agents is similar to our proposed agent, except that the baseline agents do not have an actor module. We train the baseline agents following the same procedure as our agent using the losses from equation 5.2 and 5.3.

Results are shown in Figure 5.3. Among the three baselines, the Spiral and the Plus agents outperform the Random agent. For $t \geq 8$, the Plus achieves higher accuracy than the Spiral on ImageNet, whereas, on fMoW, the Spiral outperforms the Plus. This inconsistent behavior is mainly due to the different orientations of objects in the two datasets. While the objects are mainly aligned vertically or horizontally in ImageNet, the landmarks in fMoW have no specific orientation. Finally, our agent outperforms all baseline agents across the two datasets both at initial ($t < 8$) and later ($t \geq 8$) time-steps. At $t = 8$, it achieves 1.8% higher accuracy on ImageNet and 2.3% higher accuracy on



(a) (b) (c)

Figure 5.3: *Baseline comparison of various attention policies.* (a) ImageNet; (b) fMoW. The Random selects glimpses in random order. The Plus and the Spiral select glimpses in the order shown in (c). Starting from a random glimpse location, our STAM uses an RL agent to predict next glimpse location. Results for the Random and STAM are presented as mean±5×std computed from ten independent runs. © [2022] IEEE. Reprinted, with permission, from [141].

fMoW than the top-performing baselines for the respective datasets.

5.4.2 Analysis of Consistency Loss

To quantify the gain achieved with a consistency loss from equation 5.3, we compare our agents trained with and without this loss. For a fair comparison, when training our agent without the consistency loss, we evaluate the cross-entropy loss between the ensemble distribution $p(y_t|s_t)$ from equation 5.1 and the ground truth. The remaining training setup is the same for both agents.

In Figure 5.4 (blue curve), we display the *difference* in the accuracy of STAM when trained by including and excluding the consistency loss in the training objectives (equation 5.7). With only two glimpses (i.e., one random and one selected by the agent at

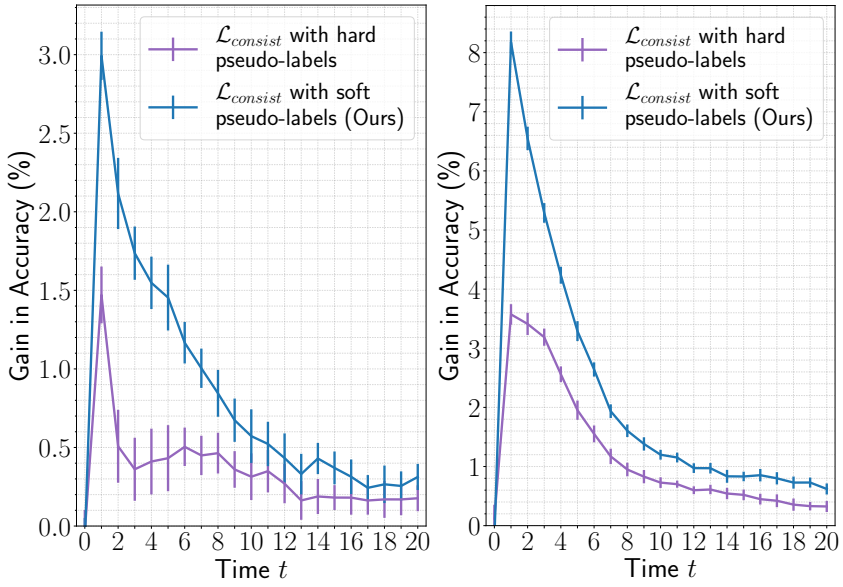


Figure 5.4: *Gain in the accuracy of STAM due to the consistency loss.* We display gain in accuracy of our STAM when trained using consistency loss (with soft vs hard pseudo-labels) over the STAM trained without consistency loss. (a) ImageNet; (b) fMoW. Results are presented as mean ± std computed using ten different runs. © [2022] IEEE. Reprinted, with permission, from [141].

$t = 1$), the proposed consistency loss results in significant improvement in accuracy with a gain of $\sim 3\%$ on ImageNet and $\sim 8\%$ on fMoW datasets.

To benchmark the improvement offered by our proposed consistency loss based on soft pseudo-labels, we study the effect of an alternate consistency loss using hard pseudo-labels: $\mathcal{L}_{consist} = -\sum \delta(\hat{y}|X) \log(p_d(y_t|s_t))$ where $\hat{y}$ are the hard pseudo-labels predicted by the teacher model from a complete image, i.e., $\hat{y} = \text{argmax}(q(y|X))$, and $\delta(\hat{y}|X)$ is a delta distribution. The results are shown in Figure 5.4 (purple curve). An agent trained using a consistency loss with hard pseudo-labels achieves a gain of $\sim 1.5\%$ on ImageNet and $\sim 3.5\%$ on fMoW for the first two glimpses. Overall, the consistency loss improves the performance of STAM. The gain in accuracy with consistency loss evaluated using soft pseudo-labels is greater than that with hard pseudo-labels.

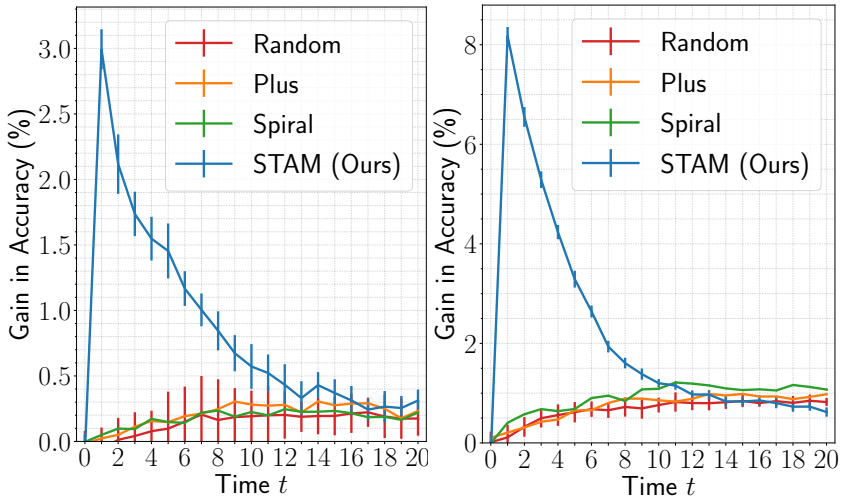


Figure 5.5: *Comparison of gain in accuracy of baseline agents due to the consistency loss.* We display gain in accuracy of a specific baseline when trained using consistency loss over the same baseline trained without consistency loss. (a) ImageNet; (b) fMoW. Results for the Random and STAM are presented as mean ± std computed across ten independent runs. © [2022] IEEE. Reprinted, with permission, from [141].

Next, we analyze the performance gain in the baseline agents (*i.e.*, the Random, the Plus, and the Spiral) vs STAM due to the proposed consistency loss. We train baseline agents with and without the proposed consistency objective and plot the difference in their accuracy in Figure 5.5. We observe that the consistency training objective yields a positive gain in the accuracy for all baseline agents.

Furthermore, the gain achieved with learned policy (i.e., STAM) is higher than the heuristics-based baseline policies. The gain in accuracy is highest for STAM as it learns to attend to the most discriminative glimpses early in time. These results align with the recent findings showing that minimizing the distance between the predictions made from two views of the same image improves model performance the most when the views optimally share the task-specific information [170].

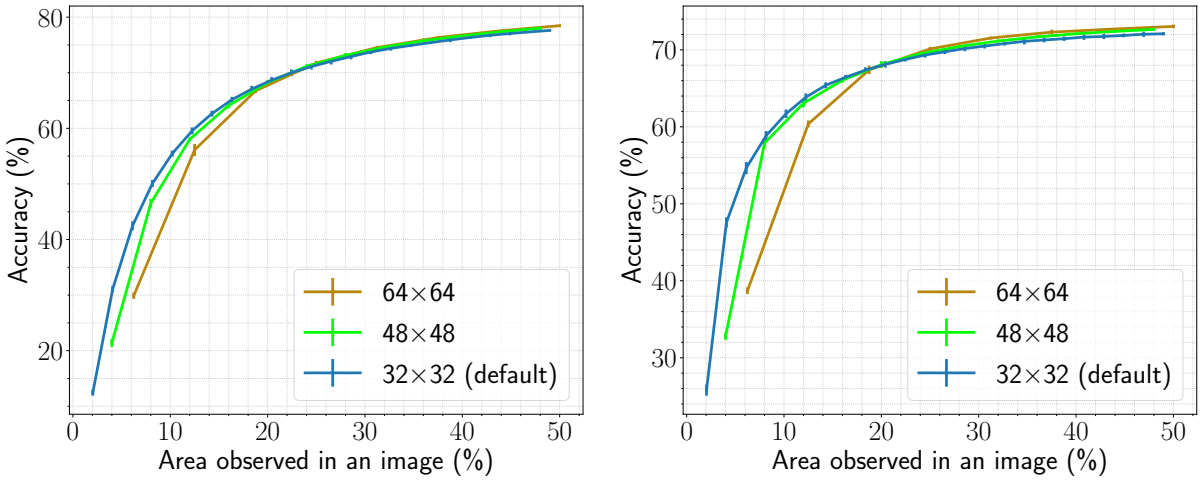


Figure 5.6: *Effect of glimpse size on accuracy.* We display accuracy of STAM with different glimpse sizes presented as a function of %area observed in an image (a) ImageNet; (b) fMoW. The results are presented as mean±5×std computed across ten independent runs. © [2022] IEEE. Reprinted, with permission, from [141].

5.4.3 Effect of Glimpse Size

We compare the performance of our STAM agents with glimpses of sizes 32×32, 48×48, and 64×64. To extract the non-overlapping glimpses, we resize the image to 224×224, 240×240, and 256×256 for the three glimpse sizes stated above, respectively. For the image-size 224×224, we use the teacher models as discussed in Section 5.3. To train teacher models for images of sizes 240×240 and 256×256, we finetune the pretrained DeiT on images of respective sizes, following the procedure suggested by Touvron *et al.*[172]. The agents for image sizes 240×240 and 256×256 observe a maximum of 16 and 7 glimpses per image.

As the glimpse and the image sizes are different, we compare the accuracy of the three agents as a function of the area observed in the image (see Figure 5.6). Initially, when an area observed in an image is less than 20%, the agent with smaller glimpses

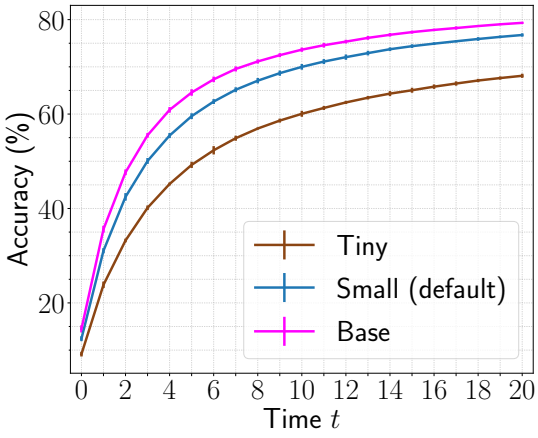
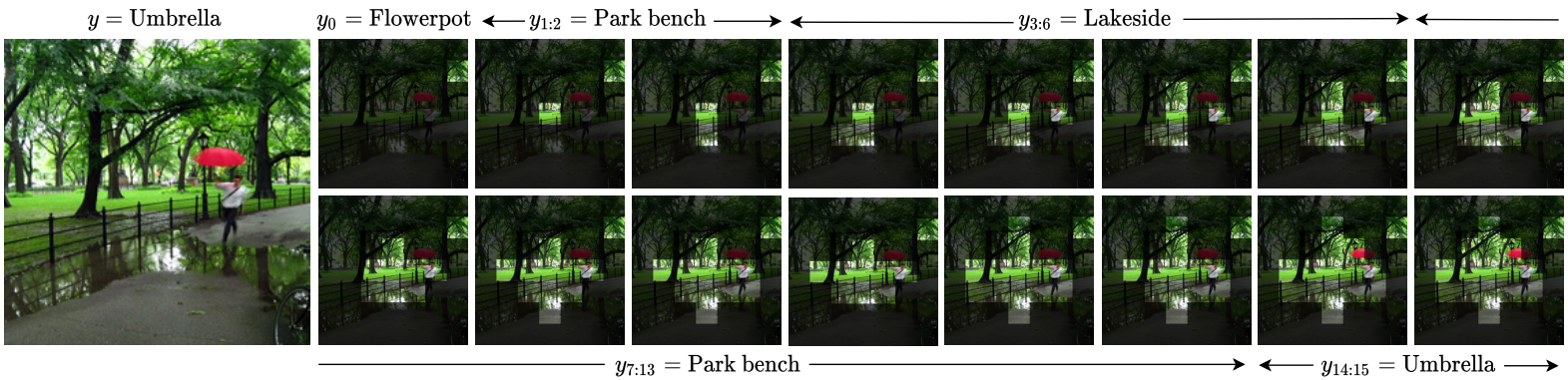


Figure 5.7: *Accuracy of STAM with core of different capacity.* We compare DeiT$^{\mathcal{D}}$-Tiny, DeiT$^{\mathcal{D}}$-Small, DeiT$^{\mathcal{D}}$-Base architectures for the core module. The results are presented as mean±5×std computed across ten independent runs. © [2022] IEEE. Reprinted, with permission, from [141].

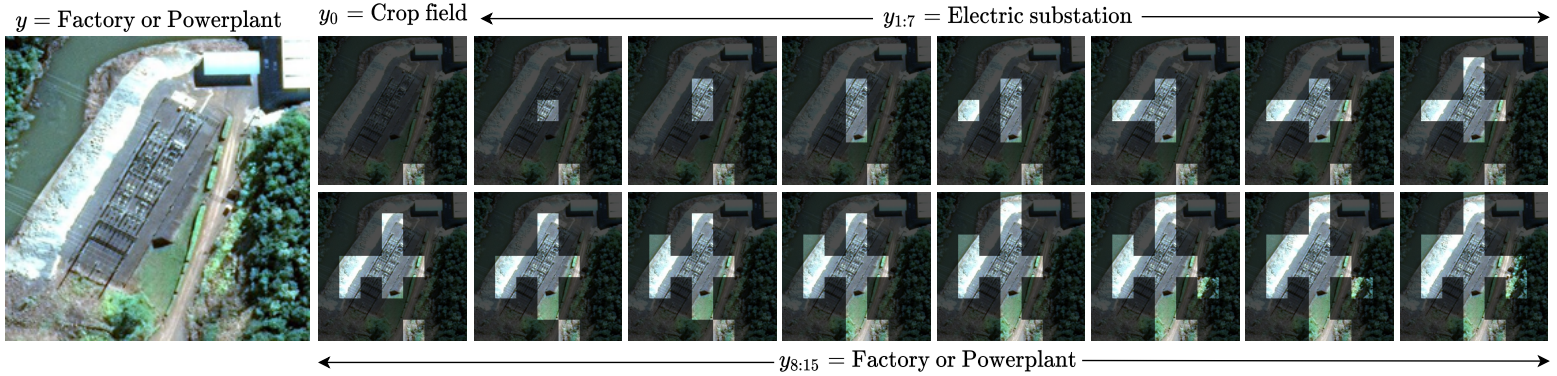
achieves higher accuracy than the agent with larger glimpses. The reason is that the agent explores more regions using smaller glimpses than the larger ones while sensing the same amount of area. Once the agents have observed sufficient informative regions (nearly 20% of the total image area), their performance converges. We use glimpse size 32×32 with image size 224×224 as our default setting.

5.4.4 Effect of Model Capacity

To study the effect of model capacity on the performance, we compare DeiT$^{\mathcal{D}}$-Tiny, DeiT$^{\mathcal{D}}$-Small, and DeiT$^{\mathcal{D}}$-Base architectures as the core of our STAM agent. We use pretrained DeiT$^{\mathcal{D}}$of respective capacity as the teacher model. Results for ImageNet are presented in Figure 5.7. We observe increasing accuracy with increasing model capacity. However, training an agent with DeiT$^{\mathcal{D}}$-Base is computationally expensive. To achieve a good trade-off between efficiency and accuracy, we use DeiT$^{\mathcal{D}}$-Small as a default architecture for our agent.



(a)



(b)

Figure 5.8: *Glimpse Visualization for STAM.* We visualize of glimpses selected by STAM on example images from $t = 0$ to 15. (a) ImageNet; (b) fMoW. Complete images are shown for reference only. STAM does not observe a complete image. © [2022] IEEE. Reprinted, with permission, from [141].

5.4.5 Glimpse Visualization

In Figure 5.8, we display the glimpses selected by STAM and the predicted labels on example images from ImageNet and fMoW. In the ImageNet example, STAM locates and identifies the umbrella at $t = 14$. In the fMoW example, STAM locates the transmission line and identifies the powerplant at $t = 8$. Note that, while not observing a complete image, STAM predicts the location of informative glimpses solely based on past glimpses.

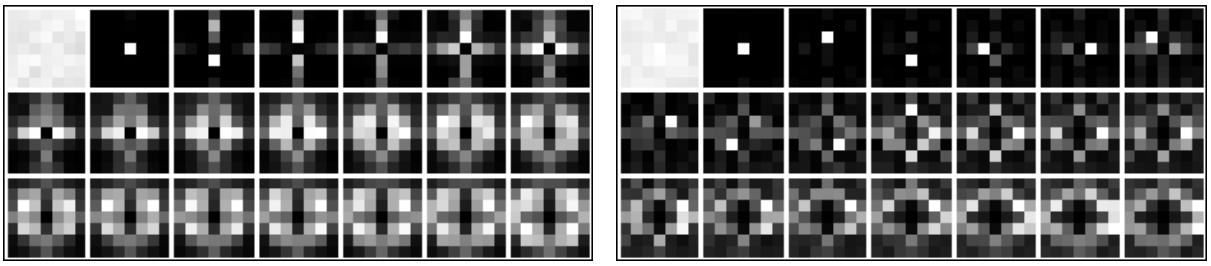
In Figure 5.9, we display the histograms of glimpse locations selected by our agent with increasing t. At $t = 0$, the agent observes a glimpse at a random location, and at $t = 1$, the agent learns to observe mainly a glimpse centered on an image, perhaps due to the object-centric nature of the dataset. For the subsequent glimpses, the agent prefers to attend vertically and horizontally centered glimpses in ImageNet. While for fMoW, it attends to glimpses with minimum distance from the center. Note in ImageNet the object of interest frequently appears at the center and is aligned vertically or horizontally; whereas in fMoW, the object of interest appears at the center but with no specific orientation. With time, the agent attends to different locations away from the center based on the content observed through previous glimpses as shown in Figure 5.8.

5.4.6 State-of-the-Art Comparison

A fair comparison between our method and the previous works is challenging due to the following reasons. Most previous works observe an entire image, occasionally at low resolution, to locate the most informative glimpses [187, 12, 51, 135, 175]. Furthermore, if they observe an entire image at low resolution, they optionally use the image along with the glimpses to predict the class-label [187, 175, 135]. In contrast, our agent operates only under partial observability. We present a comparison against state-of-the-art methods in

				ImageNet		fMoW	
Method	Note	Is complete image used for Attention?	Classification?	#pixels for classification	Accuracy (%)	#pixels for classification	Accuracy (%)
DRAM[12]	results from [135]	Yes	No	47.4K	67.50	—	—
GFNet[187]		Yes	Yes	46.1K	75.93	—	—
Saccader[51]	results from [135]	Yes	No	35.6K	70.31	—	—
TNet[135]		Yes	Yes	35.6K	74.62	—†	—
STN [175, 145]	results from [175]	Yes	Yes	28.2K	71.40	22.0K	64.8
PatchDrop[175]		Yes	Yes	27.9K	76.00	19.4K	68.3
STAM (DeiT$^{\mathcal{D}}$-Small)$^{\triangle}$	*equivalent	No	No	**20.5K**$(t = 19)$	**76.35**	**11.3K**$(t = 10)$	**68.8**
STAM (DeiT$^{\mathcal{D}}$-Base)	accuracy to [175]	No	No	**14.3K**$(t = 13)$	**76.13**	—	—
STAM (DeiT$^{\mathcal{D}}$-Small)$^{\triangle}$	$^{\diamond}$equivalent	No	No	**27.7K**$(t = 26)$	**78.25**	**19.5K**$(t = 18)$	**71.5**
STAM (DeiT$^{\mathcal{D}}$-Base)	sensing to [175]	No	No	**27.7K**$(t = 26)$	**80.78**	—	—

Table 5.1: *State-of-the-Art comparison.* We report the number of pixels sensed per image for classification and the resultant accuracy. If a method uses a low-resolution gist of a complete image for classification, we include the pixels of the gist in the above count. Our results are presented as an average computed over ten runs. We present our results at two different time steps, (*first two rows) when the accuracy achieved by our method is equivalent to PatchDrop, and ($^{\diamond}$last two rows) when the number of pixels sensed by our agent for classification is equivalent to PatchDrop. †TNet [135] use 896×896 resolution images; hence, we do not include their results in the above comparison. $^{\triangle}$Our default models. © [2022] IEEE. Reprinted, with permission, from [141].

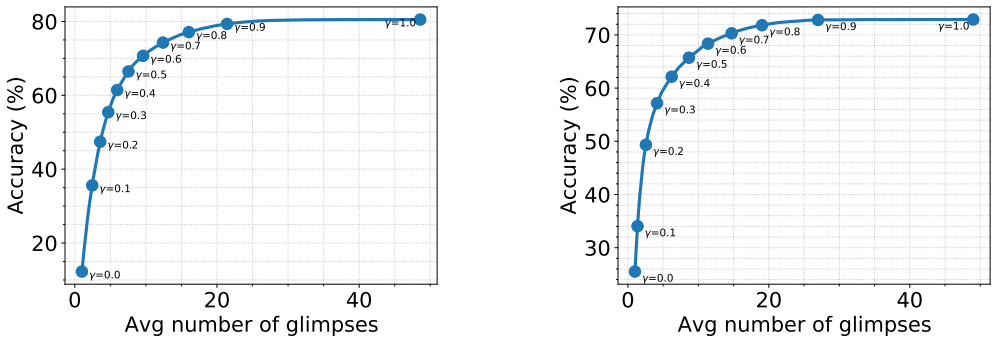


(a) (b)

Figure 5.9: *Histograms of glimpse locations sensed by STAM.* (a) ImageNet and (b) fMoW. The first, second, and third rows of each panel display histograms for $t = 0$ to 6, 7 to 13, and 14 to 20. At $t = 0$, STAM observes a glimpse at a random location. At $t > 0$, STAM senses glimpses at the locations predicted by an RL agent. © [2022] IEEE. Reprinted, with permission, from [141].

Table 5.1 and indicate which method uses an entire image and for what reasons. As different methods use different glimpse sizes, we compare them based on the number of pixels sensed per image for classification. If the method senses a complete image but does not use it for classification [12, 51], we do not include the pixels of the complete image in the above count.

Among the previous methods PatchDrop [175] is the best performing method. To achieve the same level of accuracy as PatchDrop, our default agent observes 7.4K fewer pixels per image from ImageNet and 8.1K fewer pixels per image from fMoW. Moreover, while observing a similar number of pixels as PatchDrop, our default agent achieves 2.25% higher accuracy on ImageNet and 3.2% higher accuracy on fMoW. We also train our agent with DeiT$^{\mathcal{D}}$-Base as the core module on ImageNet. This agent requires 13.6K fewer pixels per image to achieve the same accuracy as PatchDrop, and achieves 4.78% higher accuracy than PatchDrop while sensing the same number of pixels per image. We emphasize that our agent does not observe a complete image to locate informative glimpses or to perform classification, whereas PatchDrop does.



(a) (b)

Figure 5.10: *Accuracy due to early termination.* We display an average number of glimpses observed per image vs the accuracy achieved by STAM on (a) ImageNet and (b) fMoW datasets with an early termination scheme. STAM suspends sensing once the probability of the predicted class is higher than threshold γ. © [2022] IEEE. Reprinted, with permission, from [141].

5.4.7 Early Termination

In practice, we can save time and resources by terminating sensing when the agent confidently concludes a class for the image. To this end, we devise a simple mechanism to decide when to stop sensing. Let us define a scoring function based on the probability of the predicted class, $\mathcal{S}_t = max(p(y_t|s_t))$. The agent stops sensing more glimpses if $\mathcal{S}_t$ is greater than threshold γ. In Figure 5.10, we show the average number of glimpses observed per image vs. the accuracy of our agent for γ varying from 0 to 1. Remarkably, with $\gamma = 0.5$, STAM suspends sensing on ImageNet after observing on an average 7.5 glimpses per image and achieves 66.47% accuracy. Similarly, on fMoW with $\gamma = 0.5$, STAM suspends sensing after observing on an average 8.7 glimpses per image and achieves 65.71% accuracy.

5.5 Conclusions

In this chapter, we introduced a novel Sequential Transformers Attention Model (STAM) that progressively observes a scene only partially using glimpses to predict its label. It predicts future informative glimpse locations solely based on past glimpses. We trained STAM using a one-step actor-critic algorithm; and proposed a novel consistency training objective which further improves its accuracy by 3% on ImageNet and 8% on fMoW with only two glimpses.

While never sensing a complete image, our agent outperforms the previous state-of-the-art [175] that observes an entire image by sensing nearly 27% and 42% fewer pixels in glimpses per image on ImageNet and fMoW, respectively. Finally, to save the inference time and resources, we devise a simple scheme to terminate sensing when STAM has predicted a label with sufficient confidence. With a confidence score > 0.5, STAM correctly classifies nearly 65% images on both datasets by observing on an average < 9 glimpses per image (i.e., $< 18\%$ of the total image area). However, STAM is limited by its quadratic computational cost. One way to overcome this limitation is to replace DeiT$^{\mathcal{D}}$with sparse transformers [133, 144].

Part II

Spatiotemporal Attention Model

6

Glimpse Transformers

Note: Significant parts of this chapter are adapted, with permission, from:

[140] *GliTr: Glimpse Transformers with Spatiotemporal Consistency for Online Action Prediction.*
Samrudhdhi B Rangrej, Kevin J Liang, Tal Hassner and James J Clark.
IEEE/CVF Winter Conference on Applications of Computer Vision, 2023.

The IEEE does not require individuals working on a thesis to obtain a formal reuse license. However, it requires that the thesis author cite the source and include IEEE copyright notice for all figures and tables [57].

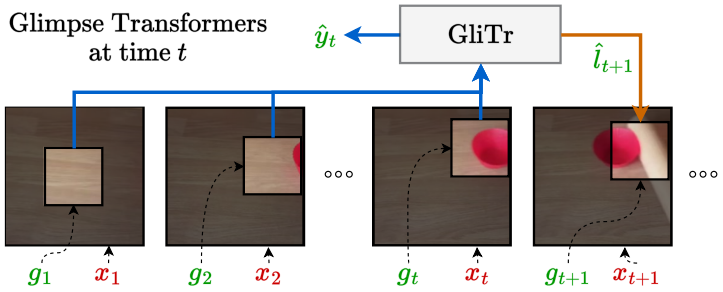


Figure 6.1: *Schematics of Glimpse Transformers (GliTr).* Our GliTr is an online action prediction model that only attends to the most informative glimpses (g_t) in the frames (x_t). While never observing frames completely, GliTr predicts label $\hat{y}_t$ (*i.e.* an estimate of ongoing action at time t) and the next glimpse location $\hat{l}_{t+1}$ based solely on the glimpses observed up to t. © [2023] IEEE. Reprinted, with permission, from [140].

In this chapter, we discuss our Glimpse Transformers (GliTr) for glimpse-based online action prediction. Our GliTr observes only a narrow glimpse from each video frame. Starting from a glimpse at a given location, our model decides which location to attend to in the subsequent frames solely based on previously observed glimpses. Note that GliTr never observes complete video frames. Consequently, it predicts an ongoing action using partial spatiotemporal information. We display schematics of GliTr in Figure 6.1.

As transformers [178] can efficiently encode the relations between spatially and temporally distant glimpses, we choose them to learn a glimpse-based attention mechanism and action prediction. Following a *factorized encoder* architecture [10], we use a) a spatial encoder that solely models relations between the patches from a single glimpse to predict spatial features, and b) two temporal encoders that model interactions between various glimpse features across time to predict the class label and the next glimpse location, respectively.

Since the ground truth for optimal glimpse locations is unavailable, we propose a novel spatiotemporal teacher-student consistency objective to incentivize GliTr to learn glimpse location in a weakly supervised manner. With only glimpses, GliTr (as the student model)

is trained to reproduce the spatial features and class distribution of a teacher model ingesting the complete frames of the video. As the teacher learns to produce predictive features and logits for the downstream task of online action recognition from the full frames, enforcing this consistency loss on the student model implicitly requires focusing attention on the most informative regions, leading to learning a glimpse mechanism.

We demonstrate GliTr's effectiveness on Something-Something-v2 [62] and Jester [123] datasets. Our proposed consistency yields $\sim$ 10% gain in accuracy on SSv2 compared to the baseline cross-entropy objective. While observing only 32% of the total area per frame, our GliTr achieves nearly 53% and 94% accuracy on SSv2 and Jester dataset, respectively.

6.1 Models

We use a teacher model to i) initialize our GliTr - a student model and ii) compute targets for the spatiotemporal consistency objective used for training GliTr. We discuss our teacher model in Section 6.1.1 followed by GliTr in Section 6.1.2. We crown the quantities computed by our models using complete frames and glimpses with ($\tilde{}$) and ($\hat{}$), respectively.

6.1.1 Teacher

Given spatially complete frames $x_{1:t}$ from a preliminary video at time $t \leq T$, our online teacher model predicts $\tilde{y}_t$, an early estimate of true action y_{GT}. We adapt the *factorized transformers encoder* architecture [10] for our teacher model, and aggregate spatial and temporal information sequentially. It includes the following components.

Feature Extraction ($\mathcal{T}_f$). We use a spatial transformer $\mathcal{T}_f$ to extract features $\tilde{f}_t$ from each individual frame x_t for all t. We use the ViT architecture [178, 172] without the

final classification head, and collect features from the output corresponding to the input class token. Refer to Section 2.7 for a brief introduction to ViT.

Early Action Prediction ($\mathcal{T}_c$). We use a temporal transformer $\mathcal{T}_c$ to aggregate features $\tilde{f}_{1:t}$ and predict label $\tilde{y}_t$. Since transformers are permutation invariant, we enforce order in the input sequence using temporal position embeddings. Moreover, we do not use a separate class token and pass the output corresponding to $\tilde{f}_t$ to the linear classifier to predict $\tilde{y}_t$. Further, to reduce training time, we use causal attention masking [61, 36]. Hence, during training, $\mathcal{T}_c$ observes $\tilde{f}_{1:T}$ and produces $\tilde{y}_{1:T}$ in a single forward pass while aggregating features in an online progressive manner, referencing only $\tilde{f}_{1:t}$ to produce output $\tilde{y}_t$ at index t.

Glimpse Location Prediction ($\mathcal{T}_l$). We include temporal transformer $\mathcal{T}_l$ to predict glimpse location $\tilde{l}_{t+1}$ from $\tilde{f}_{1:t}$. $\mathcal{T}_l$ has the same architecture as $\mathcal{T}_c$, except the final linear classifier is replaced by a linear regression head to predict coordinates $\tilde{l}_{t+1}$. Though not required for online action prediction from full frames, we train $\mathcal{T}_l$ to initialize the corresponding module in our student model. Once the student model is initialized, we discard $\mathcal{T}_l$ from the teacher model.

We illustrate a teacher with $\mathcal{T}_f$ and $\mathcal{T}_c$ in Figure 6.2 (left).

6.1.2 Glimpse Transformer (GliTr) — Student

Our Glimpse Transformer (GliTr) is derived and initialized from the teacher model discussed in Section 6.1.1. It is an iterative model that actively locates and attends to narrow glimpses in a scene and predicts an ongoing action early based on spatially and temporally incomplete observations. At time t, GliTr senses a new glimpse g_t at location $\hat{l}_t$ from frame x_t. Using glimpses $g_{1:t}$, it predicts i) $\hat{y}_t$, an early approximation of label y_{GT} and ii) $\hat{l}_{t+1}$, location of the next glimpse. We illustrate GliTr's operation in Algorithm 4

Algorithm 4 Inference using GliTr

$\hat{l}_1$ is predefined.
for $t \in \{1, \ldots, T\}$ **do**
    Sample g_t at $\hat{l}_t$ from x_t. ▷ **Glimpse Extraction**
    $\hat{f}_t = \mathcal{T}_f(g_t, \hat{l}_t)$ ▷ **Feature Extraction**
    $\hat{y}_t = \mathcal{T}_c(\hat{f}_{1:t})$ ▷ **Early Action Prediction**
    $\hat{l}_{t+1} = \mathcal{T}_l(\hat{f}_{1:t})$ ▷ **Glimpse Location Prediction**
    Save $\hat{f}_t$.
end for

and Figure 6.2. It consists of the following components.

Glimpse Extraction. Given a location $\hat{l}_t = (i, j)$, we crop a glimpse g_t centered at location $\hat{l}_t$ in frame x_t. To maintain differentiability through the cropping operation, we use a spatial transformer network (STN) [85][1]. Refer to Chapter 2, Section 2.6 for an introduction to STN.

Feature Extraction ($\mathcal{T}_f$). Similar to the teacher model, we use $\mathcal{T}_f$ to extract features $\hat{f}_t$ from glimpse g_t. We derive position embeddings for patches in g_t using STN (See Section 2.7 and Figure 2.8).

Early Action Prediction ($\mathcal{T}_c$). We input glimpse features $\hat{f}_{1:t}$ to $\mathcal{T}_c$ which in turn predicts class label $\hat{y}_t$.

Glimpse Location Prediction ($\mathcal{T}_l$). Similarly, we pass the features $\hat{f}_{1:t}$ to $\mathcal{T}_l$ which predicts the next glimpse location $\hat{l}_{t+1}$.

6.2 Training Objectives

We discuss training objectives for GliTr in Section 6.2.1. Considering GliTr as the downstream model, we design training objectives suitable for our teacher model in Section 6.2.2. We crown training objectives of GliTr and the teacher model with ($\hat{}$) and ($\tilde{}$), respectively.

[1]Not to be confused with (spatial) Vision Transformers (ViT) [49].

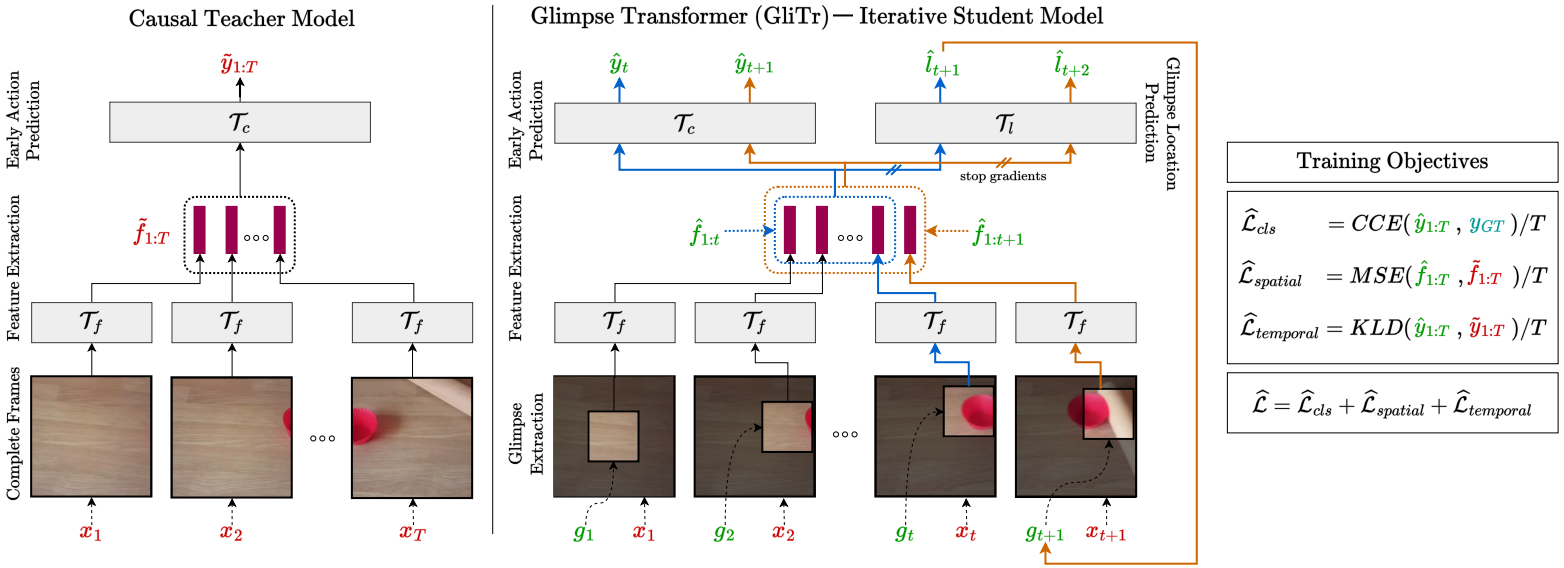


Figure 6.2: *An overview of our GliTr.* GliTr consists of a frame-level spatial transformer $\mathcal{T}_f$ and causal temporal transformers $\mathcal{T}_c$ and $\mathcal{T}_l$. One training iteration requires T forward passes through our model. Above, we show two consecutive forward passes at time $t \leq T-1$ and $t+1 \leq T$. **Forward pass t** (blue path): Given a new glimpse g_t, $\mathcal{T}_f$ extracts glimpse-features $\hat{f}_t$. We append $\hat{f}_t$ to $\hat{f}_{1:t-1}$, *i.e.*features extracted from $g_{1:t-1}$ during previous passes. Next, $\mathcal{T}_c$ predicts label $\hat{y}_t$ from $\hat{f}_{1:t}$. Simultaneously, $\mathcal{T}_l$ predicts next glimpse location $\hat{l}_{t+1}$ from $\hat{f}_{1:t}$. **Forward pass t+1** (orange path): Given a predicted location $\hat{l}_{t+1}$, we extract a glimpse g_{t+1} at $\hat{l}_{t+1}$ from a frame x_{t+1}. Then, we follow the same steps as the blue path. After T forward passes, we compute the losses shown in the right. To find targets $\tilde{y}_{1:T}$ and $\tilde{f}_{1:T}$ for spatial and temporal consistency, we use a separate pre-trained and fixed teacher model (shown on the left and explained in Figure 6.3) that observes complete frames $x_{1:T}$. To maintain stability, we stop gradients from $\mathcal{T}_l$ to $\mathcal{T}_f$. © [2023] IEEE. Reprinted, with permission, from [140].

6.2.1 Glimpse Transformer (GliTr) — Student

Classification Loss. Since our goal is to predict action label y_{GT} early using the spatially and temporally incomplete video, we minimize the cross-entropy loss given by

$$\widehat{\mathcal{L}}_{cls} = CCE(\hat{y}_{1:T}, y_{GT})/T. \tag{6.1}$$

Spatial Consistency Loss. We require GliTr to attend to the glimpses that produce features as predictive of the action as the ones predicted using complete frames by our teacher model. Hence, we minimize the mean squared error (MSE) between the glimpse features $\hat{f}_t$ predicted by GliTr and the frame features $\tilde{f}_t$ predicted by our teacher model, which is

$$\widehat{\mathcal{L}}_{spatial} = MSE(\hat{f}_{1:T}, \tilde{f}_{1:T})/T. \tag{6.2}$$

Temporal Consistency Loss. While the teacher model has all instantaneous spatial information available in a complete frame, GliTr must rely on past glimpses to reason about the unobserved yet informative regions in the current frame. To incentivize GliTr to aggregate spatial information from the past to mitigate partial observability, we minimize the KL-divergence between the class logits predicted by GliTr using glimpses ($\hat{y}_t$) and the teacher using complete frames ($\tilde{y}_t$), yielding

$$\widehat{\mathcal{L}}_{temporal} = KLD(\hat{y}_{1:T}, \tilde{y}_{1:T})/T. \tag{6.3}$$

Our final training objective for GliTr is the following:

$$\widehat{\mathcal{L}} = \widehat{\mathcal{L}}_{cls} + \widehat{\mathcal{L}}_{spatial} + \widehat{\mathcal{L}}_{temporal} \tag{6.4}$$

For more details on the student-teacher training paradigm, refer to Section 2.5.

6.2.2 Teacher

Classification loss. For all t, we minimize cross-entropy loss between the prediction $\tilde{y}_t$ and the ground-truth label y_{GT} of the action,

$$\widetilde{\mathcal{L}}_{cls} = CCE(\tilde{y}_{1:T}, y_{GT})/T. \tag{6.5}$$

Distillation loss. When available, we also use a more powerful offline action recognition model such as VideoMAE [171] to predict action $y_T^{offline}$ from a complete video, *i.e.*$x_{1:T}$. Then, we minimize the KL-divergence between the final prediction $\tilde{y}_T$ and the above $y_T^{offline}$ given by

$$\widetilde{\mathcal{L}}_{dist} = KLD(\tilde{y}_T, y_T^{offline}). \tag{6.6}$$

For more details on distillation, refer to Section 2.5.

Spatiotemporal Consistency losses. Note that the above two losses train only $\mathcal{T}_f$ and $\mathcal{T}_c$. We use the following strategy to train $\mathcal{T}_l$. First, we use the locations $\tilde{l}_1$ (learnable parameter) and $\tilde{l}_{2:T}$ predicted by $\mathcal{T}_l$, to extract glimpses $g_{1:T}$ from frames $x_{1:T}$. Next, we create copies of $\mathcal{T}_f$ and $\mathcal{T}_c$ denoted as $\mathcal{T}_f'$ and $\mathcal{T}_c'$. We input $g_{1:T}$ and the corresponding position embeddings to $\mathcal{T}_f'$ and predict glimpse features $\hat{f}_{1:T}$. Given $\hat{f}_{1:T}$, $\mathcal{T}_c'$ predicts

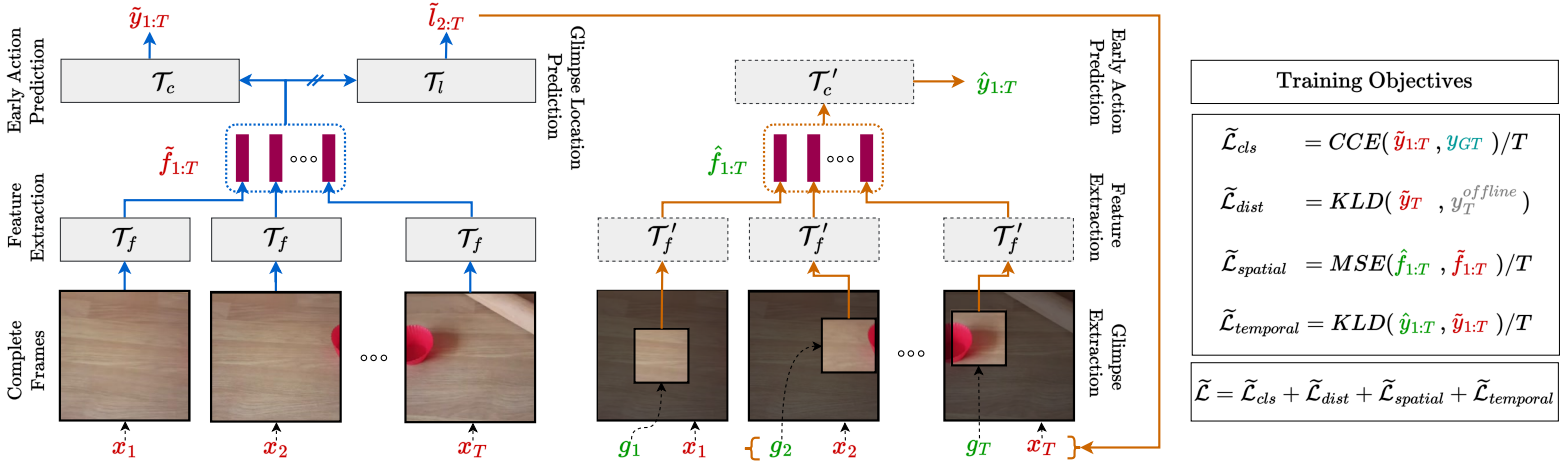


Figure 6.3: *An overview of our teacher model.* Our teacher model consists of a spatial transformer $\mathcal{T}_f$ and *causal* temporal transformers $\mathcal{T}_c$ and $\mathcal{T}_l$. Each training iteration of the teacher model consists of two steps. **Step 1** (blue path): Given complete video frames $x_{1:T}$, $\mathcal{T}_f$ extracts frame features $\tilde{f}_{1:T}$. Next, $\mathcal{T}_c$ and $\mathcal{T}_l$ predict class labels $\tilde{y}_{1:T}$ and glimpse locations $\tilde{l}_{2:T+1}$ from $\tilde{f}_{1:T}$, respectively. We discard $\tilde{l}_{T+1}$. **Step 2** (orange path): Given $\tilde{l}_1$ (learnable parameter) and $l_{2:T}$ (predicted in step 1), we extract glimpses $g_{1:T}$ from $x_{1:T}$. Then, we create non-learnable copies of $\mathcal{T}_f$ and $\mathcal{T}_c$ denoted as $\mathcal{T}'_f$ and $\mathcal{T}'_c$. $\mathcal{T}'_f$ extracts glimpse-features $\hat{f}_{1:T}$ from $g_{1:T}$ and $\mathcal{T}'_c$ predicts labels $\hat{y}_{1:T}$ from $\hat{f}_{1:T}$. We compute losses shown on the right and update model parameters. To achieve stability during training, we stop gradients from $\mathcal{T}_l$ to $\mathcal{T}_f$. © [2023] IEEE. Reprinted, with permission, from [140].

actions $\hat{y}_{1:T}$ in an online fashion. Then we minimize,

$$\widetilde{\mathcal{L}}_{spatial} = MSE(\hat{f}_{1:T}, \tilde{f}_{1:T})/T, \tag{6.7}$$

$$\widetilde{\mathcal{L}}_{temporal} = KLD(\hat{y}_{1:T}, \tilde{y}_{1:T})/T. \tag{6.8}$$

We use the above two losses to update parameters of $\mathcal{T}_l$ only. We design these consistency objectives based on the spatiotemporal consistency objectives of GliTr (equations 6.2 and 6.3). As discussed in Section 6.2.1, they encourage $\mathcal{T}_l$ to locate glimpses covering the most useful task-relevant regions in the frames, but based on complete frames observed in the past. We demonstrate the training procedure in Figure 6.3.

The final objective for our teacher model is as follows.

$$\widetilde{\mathcal{L}} = \widetilde{\mathcal{L}}_{cls} + \widetilde{\mathcal{L}}_{dist} + \widetilde{\mathcal{L}}_{spatial} + \widetilde{\mathcal{L}}_{temporal} \tag{6.9}$$

6.3 Experiment Setup

6.3.1 Datasets

We experiment with two publicly available large-scale real-world datasets, namely, Something-Something-v2 (SSv2) [62] and Jester [123]. We adopt the official training-validation splits. SSv2 dataset contains videos recording 174 human actions using everyday objects. There are ∼170K videos for training and ∼25K for validation. Jester dataset is a collection of videos capturing 27 basic hand gestures, consisting of ∼120K videos for training and ∼15K videos for validation.

6.3.2 Implementation

We sample a sequence of 16 and 8 frames per video from SSv2 and Jester, respectively. We resize each frame to size 224×224 and use glimpses of size 96×96, unless stated otherwise. We use ViT-Small [172] architecture for $\mathcal{T}_f$. For $\mathcal{T}_c$ and $\mathcal{T}_l$, we use a custom transformers architecture with 768 embedding dimensions, 6 heads, and a depth of 4.

6.3.3 Optimization

First, we discuss the common setting followed by a model-specific setting. For all models and datasets, we use the same data augmentation scheme as the one used for VideoMAE [171]. Similar to Wang *et al.*[188], we stop gradients from $\mathcal{T}_l$ to $\mathcal{T}_f$ to maintain stability during training. We use the AdamW optimizer [120] with weight decay of 5e-2 and cosine learning rate schedule with no warmup unless stated otherwise. We run experiments for SSv2 and Jester on 4 A100 GPUs with 40 GB of memory and 4 V100-SXM2 GPUs with 32 GB of memory, respectively.

To train a teacher model on SSv2 dataset, we initialize $\mathcal{T}_f$ using an open-source ViT-S model [209] pretrained on the ImageNet dataset [149], and initialize $\mathcal{T}_c$ and $\mathcal{T}_l$ randomly. We form a mini-batch using $b = 60$ videos and use an initial learning rate of $\frac{\alpha b}{128}$, with base learning rate α being 1e-5, 1e-4 and 1e-4 for $\mathcal{T}_f$, $\mathcal{T}_c$ and $\mathcal{T}_l$, respectively. We train the teacher model for 40 epochs with a warmup of 15 epochs for $\mathcal{T}_l$. For the Jester dataset, we initialize the teacher model with the teacher model trained on the SSv2 dataset. We do not use distillation loss $\widetilde{\mathcal{L}}_{dist}$ for the Jester dataset. We use a batch size b of 100 and α of 1e-5 for all modules. The model is trained for 50 epochs.

Each student model (GliTr) is initialized from a teacher model trained on the corresponding dataset. We use a base learning rate $\alpha =$ 1e-5 for all modules and train them

for 100 and 150 epochs with a batch-size b of 360 and 800 videos from SSv2 and Jester, respectively.

6.4 Results

6.4.1 Empirical Comparisons

Glimpse Mechanisms Under Partial Observability

We compare the glimpse attention strategy learned by GliTr with four baselines and an approximate upper bound:

- *Uniform random*: Glimpse locations are independently drawn from a uniform distribution for each t.
- *Gaussian random*: Similar to *uniform random* but instead, the glimpse locations are sampled from a Gaussian distribution with zero mean and unit variance and passed through a tanh() function to constrain locations to remain within the bounds of the frame.
- *Center*: The model observes glimpses from a constant location at the center of each frame.
- *Bottom Left*: The model attends to the glimpses in the bottom left corner of the frames.
- *Teacher (an upper bound)*: Glimpse locations are chosen as predicted by the teacher model which looks at the full frames. In the absence of ground truth glimpse locations, this provides an approximate upper bound.

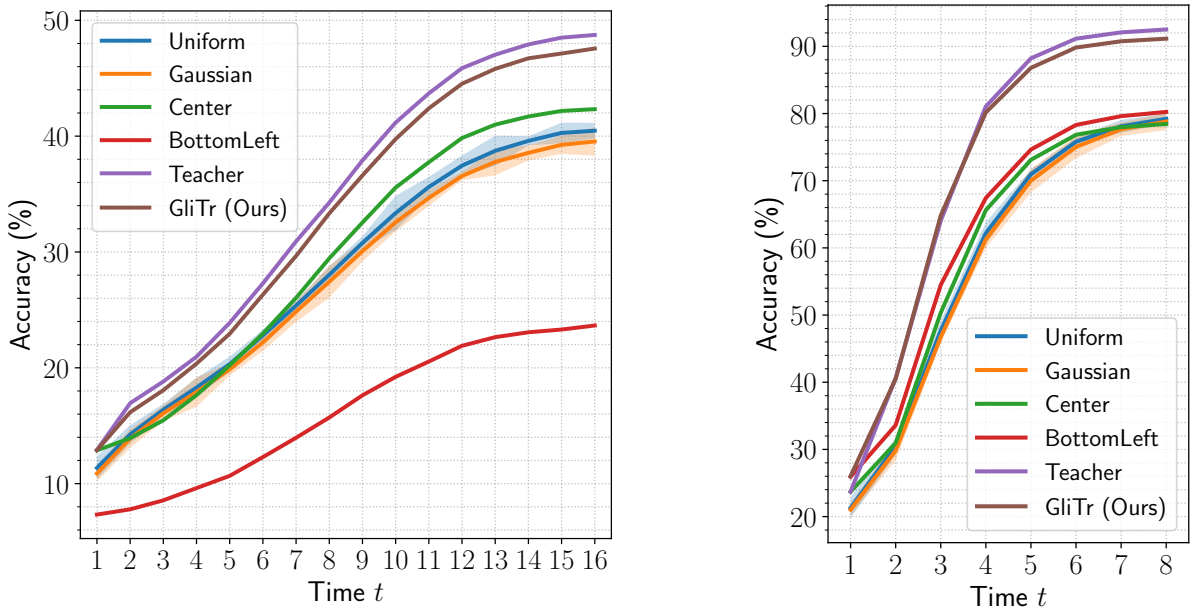
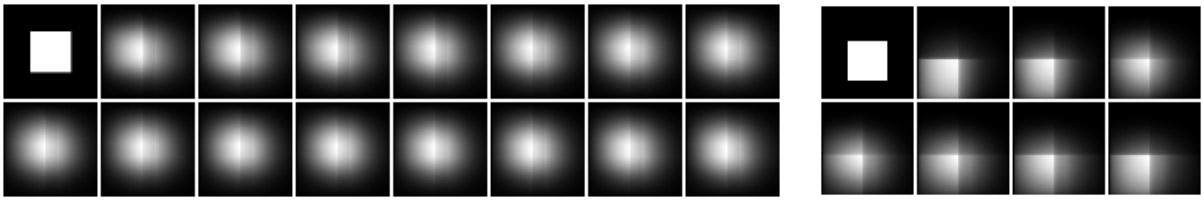


Figure 6.4: *Comparison of online action prediction accuracy using different glimpse mechanisms.* (a) SSv2 and (b) Jester. The *Uniform* and the *Gaussian* strategies sample locations from the respective distributions. We display mean±5×std computed using five independent runs. The *Center* and the *Bottom Left* strategies always observe glimpses at the constant locations. The *Teacher* (an approximate upper bound) and our GliTr locate informative glimpses based on past frames and glimpses, respectively. © [2023] IEEE. Reprinted, with permission, from [140].

To isolate the glimpse strategy's effect on performance, we evaluate the glimpses selected by various strategies using the same model *i.e.*GliTr. While assessing the baselines and the upper bound, we ignore predictions from $\mathcal{T}_l$ and instead use locations given by the specific strategies described above. We show results in Figure 6.4, plotting online action prediction accuracy after each t. As expected, the prediction accuracy for all strategies increases as the model observes more glimpses. The *Center* and the *Bottom Left* strategies outperform other baselines on SSv2 and Jester datasets, respectively. We suspect this is because the object of interest frequently appears in the center in SSv2; while in most examples from Jester, hand movements begin and end in the region near the bottom left



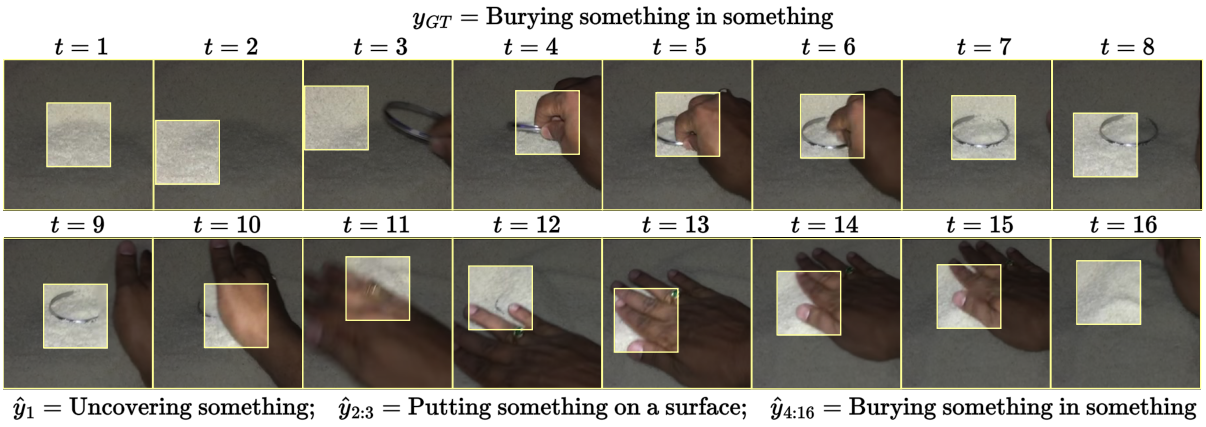
(a) (b)

Figure 6.5: *Histograms of the glimpse regions selected by GliTr.* We display histograms with increasing time (raster scan order) on (a) SSv2 and (b) Jester. Recall that GliTr observes the first glimpse at a predetermined location followed by active selection. © [2023] IEEE. Reprinted, with permission, from [140].

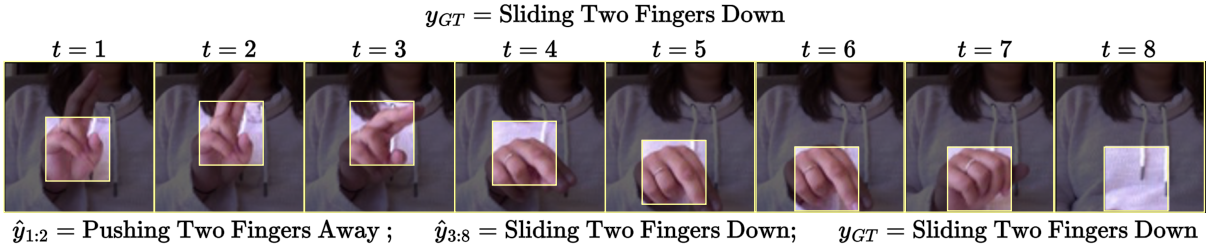
corner of the frames. On the other hand, GliTr outperforms all baselines and achieves performance closest to the upper bound (*i.e.*the *Teacher* strategy). We plot a histogram of glimpse regions selected by GliTr in Figure 6.5. We observe that not only does GliTr successfully capture different biases (center vs. bottom left) in the two datasets, but it also ignores the bias if necessary. Notice the spread in the histograms for $t > 1$, suggesting GliTr observes various regions in different videos. Consequently, GliTr achieves better accuracy faster than the baselines, and at time T, outperforms the best performing baselines with the respective margins of nearly 5% and 11% on SSv2 and Jester. We visualize glimpses selected by GliTr on example videos from SSv2 and Jester in Figure 6.6.

Models with Complete Spatial Observability

Glimpse-based offline models. We compare our GliTr with previous glimpse-based offline action recognition models in Table 6.1. We note that a direct comparison between these approaches is unfair since previous models also observe complete frames. Further, unlike offline approaches that initially observe a complete video and select an informative glimpse at t based on the current, past, and future frames, our GliTr - an online model - relies only on the past information to locate glimpses in the current frame. Moreover, previous methods use global information gathered from complete frames to locate



(a)



(b)

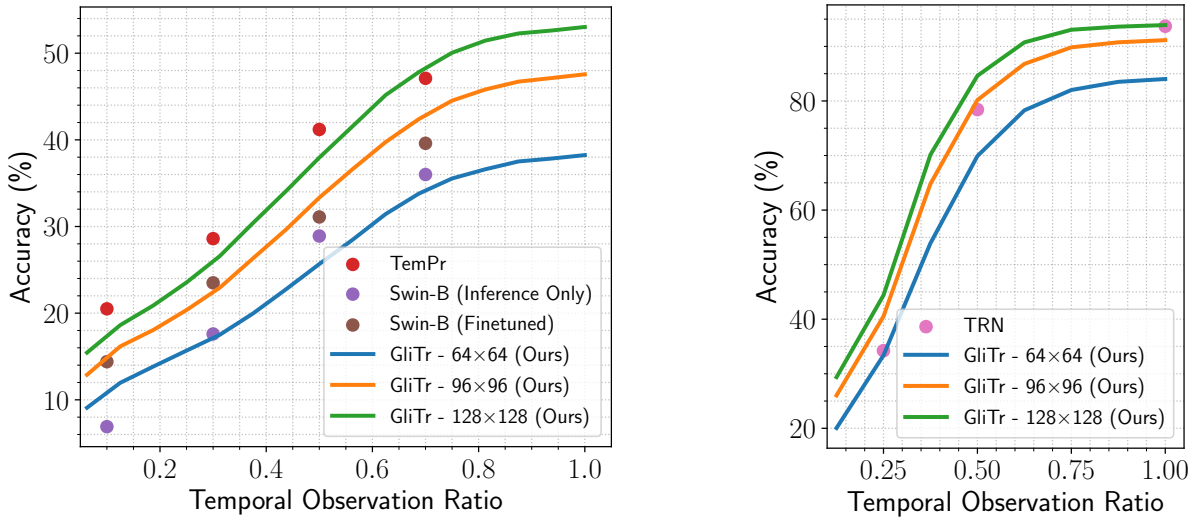
Figure 6.6: *Glimpses selected by GliTr.* (a) SSv2 and (b) Jester. The complete frames are shown for reference only. GliTr does not observe full frames. It only observes glimpses. © [2023] IEEE. Reprinted, with permission, from [140].

glimpses and predict actions; however, GliTr only uses local information. Nevertheless, we include this analysis to highlight the savings achieved by GliTr in terms of the amount of area observed for recognition while still achieving competitive performance with partial observations.

We calculate and compare the number of pixels sensed by various methods to perform action recognition. AdaFocus [186] and AdaFocusV2 [188] uniformly sample 8 frames from a complete video to predict glimpse locations, followed by uniform sampling of another 12 frames to extract glimpses. In total, they require sensing 20 complete frames ($20\times(224\times224)\approx$1M pixels) in advance due to their offline nature. GFNet [75], on the

Method	Online/ Offline?	Observes full frames?	SSv2[62] Glimpse size	SSv2[62] No. of frames	SSv2[62] No. of pixels	SSv2[62] Accuracy (%)	Jester[123] Glimpse size	Jester[123] No. of frames	Jester[123] No. of pixels	Jester[123] Accuracy (%)
AdaFocus [186]$^{\diamond}$	Offline	Yes	144×144	8+12	1M	59.70	-	-	-	-
			160×160	8+12	1M	60.20	-	-	-	-
			176×176	8+12	1M	60.70	-	-	-	-
AdaFocusV2 [188]$^{\diamond}$	Offline	Yes	128×128	8+12	1M	59.60	128×128	8+12	1M	96.60
			144×144	8+12	1M	60.50	176×176	8+12	1M	96.90
			160×160	8+12	1M	60.80	-	-	-	-
			176×176	8+12	1M	61.30	-	-	-	-
GFNet [75]§	Offline	Yes	96×96 (×2)*	8	401K	59.50	80×80 (×2)*	8	401K	95.50
			96×96 (×2)*	12	602K	61.00	96×96 (×2)*	12	602K	95.80
			96×96 (×2)*	16	803K	62.00	128×128 (×2)*	16	803K	96.10
GliTr (Ours)	**Online**	**No**	64×64	16	**66K**	38.24	64×64	8	**33K**	84.03
			96×96	16	**147K**	47.56	96×96	8	**74K**	91.15
			128×128	16	**262K**	53.02	128×128	8	**131K**	93.91

Table 6.1: *Comparison with glimpse-based action recognition models.* We count the number of pixels sensed by different approaches to perform recognition. Previous approaches are offline and use complete frames to locate informative glimpses and to recognize actions. GliTr is an online model and only observes glimpses, not complete frames. GliTr achieves competitive performance with a significant saving in the total area observed. $^{\diamond}$AdaFocus [186] and AdaFocusV2 [188] first observe 8 frames to locate useful glimpses and then sample additional 12 frames to extract glimpses, which requires sensing total 20 frames in advance due to their offline nature. §Results are based on Figure 13 from [75]. *GFNet observes two glimpses per frame. For comparison with online methods, refer to Figure 6.7. © [2023] IEEE. Reprinted, with permission, from [140].



(a) (b)

Figure 6.7: *Comparison with early action prediction models.* (a) SSv2 and (b) Jester. While Swin-B [119], TemPr [163] and TRN [208] predict action early based on complete frames, GliTr predicts action based on early glimpses. © [2023] IEEE. Reprinted, with permission, from [140].

other hand, locates and extracts glimpses from the same set of complete frames. When compared to AdaFocusV2 with glimpses of size 128×128, our GliTr reduces the amount of sensing by nearly 74% and 87% while compromising only around 6% and 3% accuracy on SSv2 and Jester, respectively. Further, while GFNet outperforms GliTr by nearly 14.4% and 4.7% with glimpses of size 96×96 on SSv2 and Jester, GliTr (with 16 and 8 glimpses, respectively) reduces the amount of sensing by nearly 82% and 88% compared to GFNet (with 16 and 12 frames, respectively) on these datasets. We emphasize that GFNet observes full frames and *two* glimpses per frame in an offline manner, while GliTr observes only one glimpse per frame in an online fashion.

Early action prediction models. We additionally compare GliTr with early action prediction models in Figure 6.7. We emphasize that these approaches observe entire frames (*i.e.*global information) from a preliminary video; whereas, GliTr observes frames

only partially through glimpses (*i.e.*local information). For SSv2 dataset, we consider Swin-B [119] and TemPr [163]. We cite Swin-B results from [163], who evaluate Swin-B for early action prediction before (*i.e.*direct inference with pretrained model) and after finetuning on preliminary videos. Notice that, with glimpses of size 96×96 and higher, GliTr outperforms Swin-B finetuned for early action prediction. Further, GliTr also outperforms TemPr with the glimpses of size 128×128 when both have observed early 70% video. For the Jester dataset, GliTr outperforms TRN [208] for early action prediction with glimpses of size 96×96 and higher. The results demonstrate the efficiency of GliTr for early action prediction using only local information.

6.4.2 Ablation on Spatiotemporal Consistency for GliTr

To demonstrate the value of the proposed spatiotemporal training objectives, we perform an ablation study for each on the SSv2 dataset. We train four variants of GliTr using the following combinations of the training objectives: i) $\text{GliTr}_{\text{baseline}}$ using $\widehat{\mathcal{L}}_{cls}$, ii) $\text{GliTr}_{\text{spatial}}$ using $\widehat{\mathcal{L}}_{cls} + \widehat{\mathcal{L}}_{spatial}$, iii) $\text{GliTr}_{\text{temporal}}$ using $\widehat{\mathcal{L}}_{cls} + \widehat{\mathcal{L}}_{temporal}$, and iv) our default variant $\text{GliTr}_{\text{spatiotemporal}}$ using $\widehat{\mathcal{L}}_{cls} + \widehat{\mathcal{L}}_{spatial} + \widehat{\mathcal{L}}_{temporal}$. Note that the above variants have the same architecture and operation; only their training objectives are different. Figure 6.8(a) shows results. We observe that including only spatial or only temporal consistency in the training objectives boosts GliTr's accuracy by nearly 6% at t=16. Moreover, including both spatial and temporal consistency provides the highest improvement of around 10%.

To understand the sources of improvements provided by the two consistency losses, we perform two more experiments. First, we evaluate glimpse selection strategies learnt by the above versions of GliTr using an impartial teacher model in Figure 6.8(b). We observe better performance for GliTr when spatial consistency is included in the training objectives, indicating that spatial consistency helps GliTr learn *better glimpse selection*

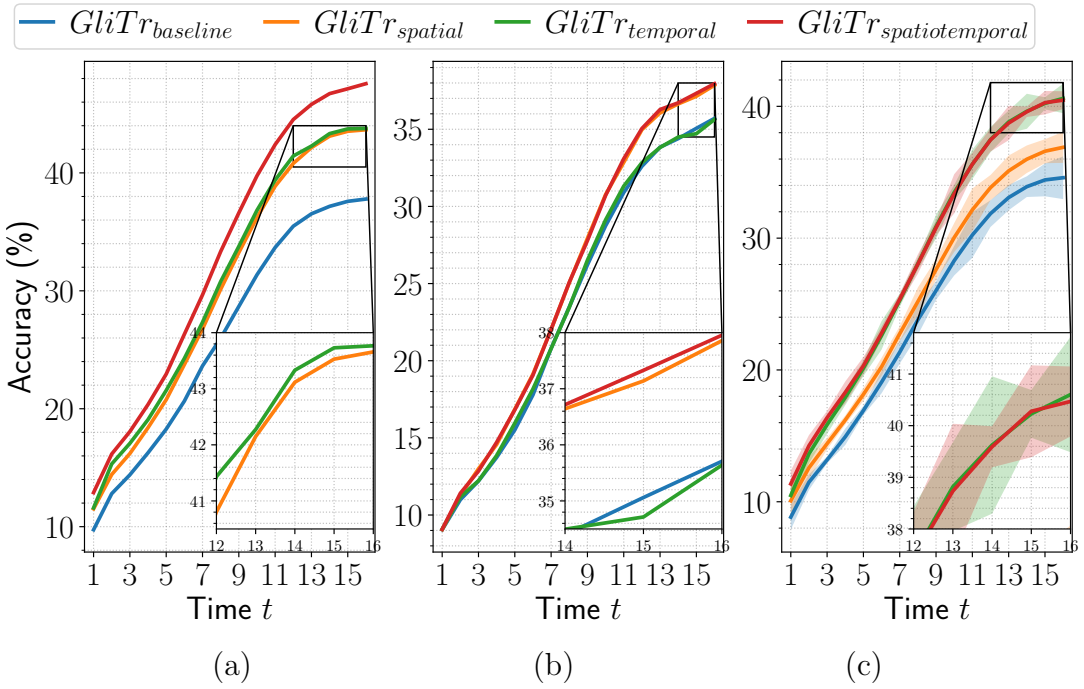
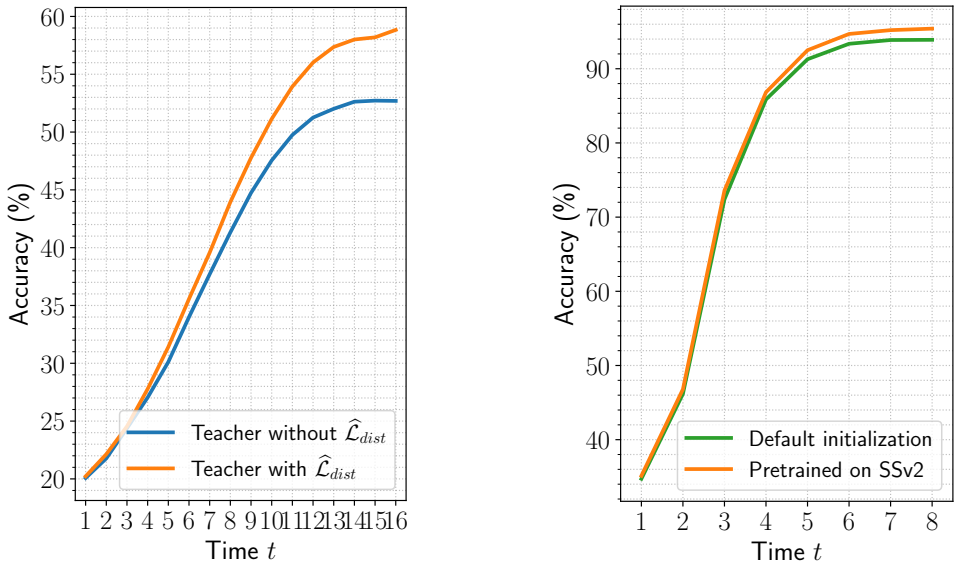


Figure 6.8: *Ablation study on the spatiotemporal consistency objective* on SSv2 dataset. (a) accuracy of GliTr when trained using different combinations of the training objectives (b) accuracy of the teacher with the glimpses selected by the above variants. (c) accuracy of the above variants of GliTr when tested with the *Uniform random* strategy. We display mean±5×std from five independent runs. © [2023] IEEE. Reprinted, with permission, from [140].

strategy and in turn improves its performance. Second, we evaluate above four versions of GliTr using an impartial *Uniform random* strategy in Figure 6.8(c). We observe that GliTr provides the highest performance for the *Uniform random* strategy when we include temporal consistency in the training objective, suggesting that temporal consistency improves GliTr's performance by learning *a better classifier under partial observability.*

6.4.3 Ablation on Teacher model

Ablation on $\widetilde{\mathcal{L}}_{dist}$. We distill VideoMAE [171] (a transformers-based offline action recognition model) to our teacher model on SSv2 dataset. To do so, we minimize $\widetilde{\mathcal{L}}_{dist}$



(a) (b)

Figure 6.9: *Ablation on Teacher model* (a) Ablation on $\widetilde{\mathcal{L}}_{dist}$ objective for the teacher trained on SSv2 dataset. (b) Ablation on initialization scheme for the teacher trained on Jester dataset. © [2023] IEEE. Reprinted, with permission, from [140].

i.e. KL-divergence between the class distributions predicted by our teacher model and VideoMAE based on complete video (equation 6.6). To assess importance of this objective, we train our teacher model with and without $\widetilde{\mathcal{L}}_{dist}$ and display results in Figure 6.9(a). We observe improvement of approximately 6% in accuracy at $t = 16$ when $\widetilde{\mathcal{L}}_{dist}$ is included in the training objectives. Note, since a pretrained VideoMAE [171] in unavailable for Jester, we do not use $\widetilde{\mathcal{L}}_{dist}$ for training the teacher model on this dataset.

Ablation on Initialization Scheme. To improve the performance of the teacher model on the Jester dataset, we initialize its parameters using the parameters of the teacher model pretrained on SSv2 with a complete set of training objectives (equation 6.9), including $\widetilde{\mathcal{L}}_{dist}$. We compare the performance of the above model with the performance of the teacher initialized using default scheme *i.e.*$\mathcal{T}_f$ initialized using an open-source ViT-S model [209] pretrained on the ImageNet, and $\mathcal{T}_c$ and $\mathcal{T}_l$ initialized randomly. The re-

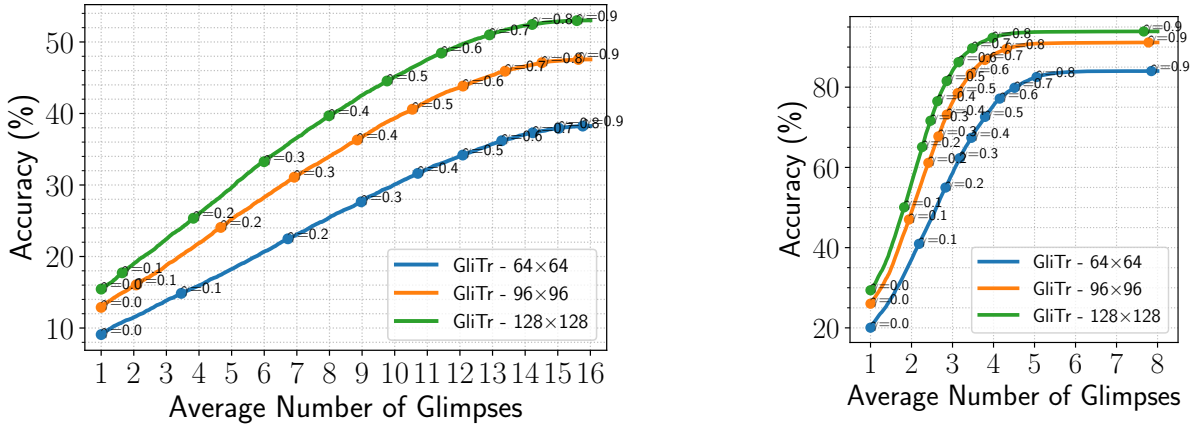


Figure 6.10: *GliTr with early exit.* We display accuracy vs an average number of glimpses seen by GliTr per video to predict a class with probability $> \gamma$. (a) SSv2 and (b) Jester. © [2023] IEEE. Reprinted, with permission, from [140].

sult shown in Figure 6.9(b) indicates that once finetuned on Jester dataset, the teacher pretrained on SSv2 achieves higher performance than the teacher initialized with default scheme, especially for $t > 4$. Finally, at $t = 8$, the teacher with pretrained weights achieves nearly 1.5% higher accuracy.

6.4.4 Early Exit

We extend GliTr for applications that require timely decision-making. We terminate sensing and conclude a class when GliTr makes a sufficiently confident prediction. We evaluate confidence using the maximum value in the predicted class logits, $\mathcal{C}_t = \max(p(\hat{y}_t))$ and exit when GliTr achieves confidence $\mathcal{C}_t > \gamma$. We show the performance of GliTr for varying γ in Figure 6.10. We observe a trade-off between the glimpse size and the average number of glimpses required for confident prediction. GliTr achieves higher confidence early with larger glimpse sizes and thus requires fewer glimpses to achieve certain performance. While continued sensing improves GliTr's performance on SSv2, the performance

saturates on Jester after the initial 50% of the glimpses, rendering further sensing unnecessary.

6.5 Conclusions

In this chapter, we have described a novel online action prediction model called Glimpse Transformer (GliTr) that observes video frames only partially through glimpses and predicts an ongoing action solely based on spatially and temporally incomplete observations. It predicts an informative glimpse location for a current frame based on the glimpses observed in the past. Without any ground truth for the glimpse locations, we train GliTr using a novel spatiotemporal consistency objective. On the Something-Something-v2 (SSv2) dataset, the proposed consistency objective yields around 10% higher accuracy than the cross-entropy-based baseline objective. Further, we establish that spatial consistency helps GliTr learn a better glimpse selection strategy, whereas temporal consistency improves classification performance under partial observability. While never observing frames completely, GliTr achieves 53.02% and 93.91% accuracy on SSv2 and Jester datasets and reduces the sensing area per frame by $\sim 67\%$. Finally, we also showcase a trade-off between the glimpse size and the number of glimpses required for early action prediction. GliTr is useful for lightweight, low-cost devices with small field-of-view cameras.

Summary & Future Directions

7

Summary & Future Directions

7.1 Summary

In this thesis, we presented attention models for image and video recognition tasks. We learn attention mechanisms that are akin to biological eye movements. We consider agents equipped with limited field of view cameras. Our agents observe only a small glimpse – spatially narrow window – from a scene at a time. We develop attention models that allow these agents to explore the scene intelligently by focusing their cameras on the

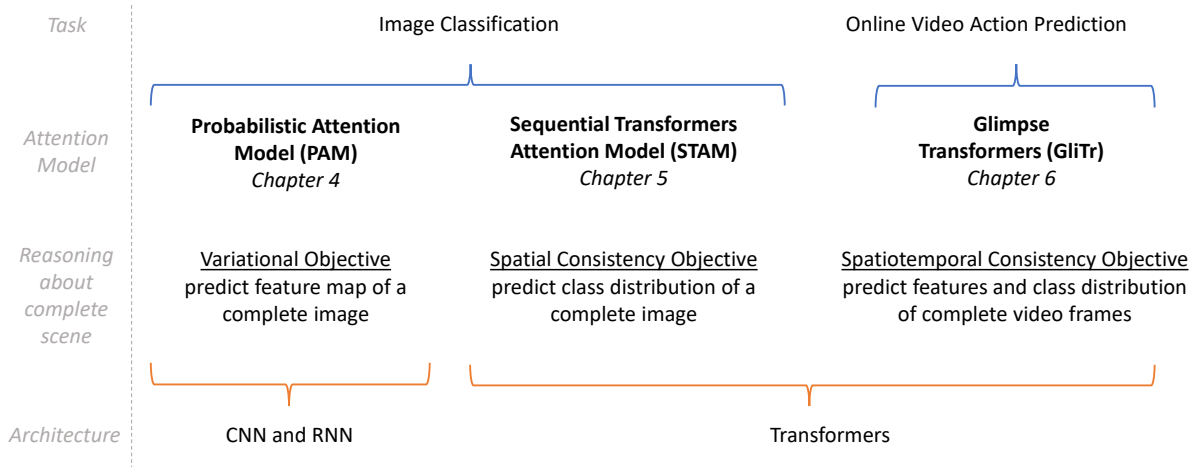


Figure 7.1: *Summary of our attention models.* We develop attention models for image classification and online action recognition. Our attention models never observe the entire scene. To improve performance and learn effective attention policy under partial observability, we compel our attention models to reason about the complete scene from glimpses. To do so, we design variational and consistency learning objectives.

most informative regions. Our sequential agents start performing recognition from the first glimpse and update their predictions as more glimpses become available. As a result, agents perform recognition using spatially and temporally incomplete information. To overcome partial observability, we compel the models to reason about the complete scene based on glimpses. We achieve this goal using variational (Chapter 4) and consistency (Chapter 5 and 6) objectives. These objectives encourage our models to observe the most informative glimpses in a scene, thus learning an effective glimpse mechanism. Further, they compel our attention models to consider glimpses as part of a holistic scene, thus improving recognition. We experimentally show that while never observing complete scenes, our attention models achieve better or comparable performance with state-of-the-art methods that require entire scenes. We summarize the attention models in Figure 7.1.

7.2 Future Directions

The methods presented in this thesis mainly cover three research problems: i) learning to look, ii) learning from incomplete data iii) learning from weak supervision. Below we discuss why these problems are essential in achieving artificial intelligence. Further, we connect our research with some open research questions and discuss future research directions in the respective areas.

Learning to look

After achieving tremendous success in solving passive tasks, researchers are now tending to embodied vision problems where an active agent engages with an environment to observe, interpret and interact with the world. Eye-movement-like attention can improve the efficiency of these agents tremendously. It paves the way for multi-tasking by allowing agents to continue observing the world while performing embodied tasks that do not require visual senses. For example, consider an agent making a lemonade. While waiting for the container to be filled with water, an agent can look around and spot lemons, sugar, cup, etc. Further, it allows agents to observe the world from various viewpoints without moving the whole body, thus improving energy efficiency.

Further, embodied agents only have access to an ego-centric, thus partial, view of the world. For most tasks, agents have to move around to interact with the world. Therefore the agents need to learn which surrounding areas are attention-worthy and prioritize exploring them. The approach of reasoning about the complete scene from glimpses can be applied in this situation. Consider an example where an agent facing a couch is asked to switch on the television. Though yet to be seen, the television should most likely be in front of the couch. Thus the agent should explore the area opposite the couch first.

Learning from incomplete data

In this thesis, we looked at recognizing a scene based on incomplete or partial observations. Current high-performing vision solutions assume we have captured data with the entire object of interest visible. The object may be occluded, but it is contained within the bounds of the image or video frames. Often, the scene is sensed such that an object of interest lies in the center. Note that capturing the scene in such a manner already requires intelligence. Current vision systems rely on human intelligence for this task. However, in the future, with the increasing application of embodied agents, the agents will have to capture the scene on their own without human guidance. In the process, the agent will commonly encounter a situation where the object of interest is out of the visible field of the agent, and the agent has to perform a given task anyways. Thus, learning to perform tasks from incomplete data is crucial for the embodied agents. The above issue calls for innovations on two fronts. First, we develop architectures that handle incomplete data naturally. For instance, architectures such as BagNet [21], and Transformers [49] can handle missing data well due to patch-based processing. Second, we develop methods that make the model robust to missing data. One such method is consistency learning, as leverage in the thesis. Recent Transformers-based Masked Autoencoders (MAE) [69, 171] are patch-based models that are robust to missing data.

Learning from weak supervision

A less obvious yet interesting future direction of our work is learning from weak supervision. In this thesis, we learn attention policy without explicit annotations using a surrogate learning signal from the recognition task. Rich annotations have enabled us to train large-scale models. However, it is notoriously challenging and expensive to attain

quality labels. In case of limited and potentially noisy labels, we use methods such as transfer learning[200], self-supervised learning [70, 177], and few-shot learning [158, 115]. However, gaining even a few annotations is exceptionally challenging for many tasks, such as attention. The issue encourages us to think about the following two questions.

First, how can we solve vision tasks with weak or surrogate labels? Many researchers are now focusing on answering this question. Some examples include distilling classifiers into object detectors [65], procedure planning using instructional videos [203], learning affordance maps from action labels [126], etc. Note, these approaches learn a task directly using surrogate labels, which is different from another interesting technique of pretraining a model on surrogate pretext tasks such as learning to solve jigsaw puzzles for object detection, recognition, and image retrieval tasks [130].

Second, which tasks can we learn using annotations from a different task? or which tasks can provide a surrogate learning signal for the given task? To answer these questions, we need to understand inter-task relationships. While some tasks agree with each other, some tasks are in direct competition. For instance, one can quickly learn edge detection from instance segmentation. However, detecting key points using the same labels would take much work. Researchers are studying inter-task relationships in the context of transfer learning [200] and multi-task learning [162]. Such understanding will help us streamline the annotation process. We can focus our labeling efforts on the tasks that are easy to annotate and provide us with learning signals for many other tasks.

The methods discussed in this thesis bring us one step closer to achieving human-level cognition in artificial agents. Further, this chapter discusses a few future directions to achieve truly intelligent and autonomous agents.

Bibliography

[1] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. "Knowledge distillation from internal representations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020.

[2] Subutai Ahmad. "VISIT: a neural model of covert visual attention". In: *Advances in neural information processing systems*. 1992, pp. 420–427.

[3] Bogdan Alexe, Nicolas Heess, Yee W Teh, and Vittorio Ferrari. "Searching for objects driven by context". In: *Advances in Neural Information Processing Systems*. 2012, pp. 881–889.

[4] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Košecká, and Alexander C Berg. "A dataset for developing and benchmarking active vision". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1378–1385.

[5] Baptiste Angles, Simon Kornblith, Shahram Izadi, Andrea Tagliasacchi, and Kwang Moo Yi. "MIST: Multiple Instance Spatial Transformer Network". In: *arXiv preprint arXiv:1811.10725* (2018).

[6] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. "Image generators with conditionally-independent pixel synthesis". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14278–14287.

[7] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. "Pseudo-labeling and confirmation bias in deep semi-supervised learning". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8.

[8] Tal Arbel and Frank P Ferrie. "Entropy-based gaze planning". In: *Image and vision computing* 19.11 (2001), pp. 779–786.

[9] Tal Arbel and Frank P Ferrie. "On the sequential accumulation of evidence". In: *International Journal of Computer Vision* 43.3 (2001), pp. 205–230.

[10] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. "Vivit: A video vision transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6836–6846.

[11] Robert C Augusteyn. "Growth of the human eye lens". In: *Molecular vision* 13 (2007), p. 252.

[12] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. "Multiple Object Recognition with Visual Attention". In: *ICLR*. 2015.

[13] Jimmy Ba, Russ R Salakhutdinov, Roger B Grosse, and Brendan J Frey. "Learning wake-sleep recurrent attention models". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2593–2601.

[14] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[15] Philip Bachman, Ouais Alsharif, and Doina Precup. "Learning with pseudo-ensembles". In: *Advances in Neural Information Processing Systems* 27 (2014), pp. 3365–3373.

[16] Fabien Baradel, Christian Wolf, and Julien Mille. "Human action recognition: Pose-based attention draws focus to hands". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 604–613.

[17] Fabien Baradel, Christian Wolf, Julien Mille, and Graham W Taylor. "Glimpse clouds: Human activity recognition from unstructured feature points". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 469–478.

[18] José M Bernardo. "Expected information as expected utility". In: *the Annals of Statistics* (1979), pp. 686–690.

[19] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. "ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring". In: *International Conference on Learning Representations*. 2019.

[20] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. "Knowledge distillation: A good teacher is patient and consistent". In: *arXiv preprint arXiv:2106.05237* (2021).

[21] Wieland Brendel and Matthias Bethge. "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet". In: *arXiv preprint arXiv:1904.00760* (2019).

[22] Peter J Burt. "Attention mechanisms for vision in a dynamic world". In: *9th international conference on pattern recognition*. IEEE Computer Society. 1988, pp. 977–978.

[23] Nicholas J Butko and Javier R Movellan. "I-POMDP: An infomax model of eye movement". In: *2008 7th IEEE International Conference on Development and Learning*. IEEE. 2008, pp. 139–144.

[24] Nicholas J Butko and Javier R Movellan. "Optimal scanning for faster object detection". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 2751–2758.

[25] Weiwei Cai and Zhanguo Wei. "PiiGAN: Generative adversarial networks for pluralistic image inpainting". In: *IEEE Access* 8 (2020), pp. 48451–48463.

[26] Yijun Cai, Haoxin Li, Jian-Fang Hu, and Wei-Shi Zheng. "Action knowledge transfer for action prediction with partial videos". In: *Proceedings of the AAAI conference on artificial intelligence*. 2019.

[27] Juan C Caicedo and Svetlana Lazebnik. "Active object localization with deep reinforcement learning". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2488–2496.

[28] Yu Cao, Daniel Barrett, Andrei Barbu, Siddharth Narayanaswamy, Haonan Yu, Aaron Michaux, Yuewei Lin, Sven Dickinson, Jeffrey Mark Siskind, and Song Wang. "Recognize human activities from partially observed videos". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 2658–2665.

[29] 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 International Conference on Computer Vision (ICCV)*. 2021.

[30] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6299–6308.

[31] Kathryn Chaloner and Isabella Verdinelli. "Bayesian experimental design: A review". In: *Statistical Science* (1995), pp. 273–304.

[32] Guangyi Chen, Jiwen Lu, Ming Yang, and Jie Zhou. "Learning recurrent 3D attention for video-based person re-identification". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 6963–6976.

[33] Xinlei Chen and Kaiming He. "Exploring simple siamese representation learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15750–15758.

[34] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. "Functional map of the world". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6172–6180.

[35] James J Clark and Nicola J Ferrier. "Modal Control Of An Attentive Vision System." In: *ICCV*. 1988, pp. 514–523.

[36] Alexis Conneau and Guillaume Lample. "Cross-lingual language model pretraining". In: *Advances in neural information processing systems* 32 (2019).

[37] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. "Differentiable Patch Selection for Image Recognition". In: *arXiv preprint arXiv:2104.03059* (2021).

[38] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 702–703.

[39] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. "CINIC-10 is not ImageNet or CIFAR-10". In: *arXiv preprint arXiv:1810.03505* (2018).

[40] Srijan Das, Arpit Chaudhary, Francois Bremond, and Monique Thonnat. "Where to focus on for human action recognition?" In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 71–80.

[41] Andrew J Davison and David W. Murray. "Simultaneous localization and map-building using active vision". In: *IEEE transactions on pattern analysis and machine intelligence* 24.7 (2002), pp. 865–880.

[42] Nicola De Cao, Ivan Titov, and Wilker Aziz. "Block neural autoregressive flow". In: *arXiv preprint arXiv:1904.04676* (2019).

[43] Frank Deinzer, Joachim Denzler, and Heinrich Niemann. "Viewpoint selection–planning optimal sequences of views for object recognition". In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2003, pp. 65–73.

[44] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. "Learning where to attend with deep architectures for image tracking". In: *Neural computation* 24.8 (2012), pp. 2151–2184.

[45] Joachim Denzler and Christopher M Brown. "Information theoretic sensor data selection for active object recognition and state estimation". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.2 (2002), pp. 145–157.

[46] Laurent Dinh, David Krueger, and Yoshua Bengio. "Nice: Non-linear independent components estimation". In: *arXiv preprint arXiv:1410.8516* (2014).

[47] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using real nvp". In: *arXiv preprint arXiv:1605.08803* (2016).

[48] Laurent Dinh, Jascha Sohl-Dickstein, Hugo Larochelle, and Razvan Pascanu. "A RAD approach to deep mixture models". In: *arXiv preprint arXiv:1903.07714* (2019).

[49] 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*. 2020.

[50] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. "Neural spline flows". In: *Advances in Neural Information Processing Systems*. 2019, pp. 7511–7522.

[51] Gamaleldin Elsayed, Simon Kornblith, and Quoc V Le. "Saccader: improving accuracy of hard attention models for vision". In: *Advances in Neural Information Processing Systems*. 2019, pp. 702–714.

[52] SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. "Attend, infer, repeat: Fast scene understanding with generative models". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3225–3233.

[53] Christoph Feichtenhofer. "X3d: Expanding architectures for efficient video recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 203–213.

[54] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. "Masked Autoencoders As Spatiotemporal Learners". In: *arXiv preprint arXiv:2205.09113* (2022).

[55] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. "Slowfast networks for video recognition". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6202–6211.

[56] Basura Fernando and Samitha Herath. "Anticipating human actions by correlating past with the future with jaccard similarity measures". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13224–13233.

[57] "Frequently Asked Questions Regarding IEEE Permissions". In: *ieee.com* (2013). Accessed on 29/03/2023. URL: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/pubs/permissions_faq.pdf.

[58] Kunihiko Fukushima. "A neural network model for selective attention in visual pattern recognition". In: *Biological Cybernetics* 55.1 (1986), pp. 5–15.

[59] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. "Conditional Neural Processes". In: *International Conference on Machine Learning*. 2018, pp. 1704–1713.

[60] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. "Neural processes". In: *arXiv preprint arXiv:1807.01622* (2018).

[61] Rohit Girdhar and Kristen Grauman. "Anticipative video transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13505–13515.

[62] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. "The "something something" video database for learning and evaluating visual common sense". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5842–5850.

[63] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. "DRAW: A Recurrent Neural Network For Image Generation". In: *ICML*. 2015.

[64] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and

Michal Valko. “Bootstrap your own latent: A new approach to self-supervised learning”. In: *Advances in neural information processing systems* 33 (2020), pp. 21271–21284.

[65] Shuxuan Guo, Jose M Alvarez, and Mathieu Salzmann. “Distilling Image Classifiers in Object Detectors”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 1036–1047.

[66] Ben J Hardcastle and Holger G Krapp. “Evolution of biological image stabilization”. In: *Current Biology* 26.20 (2016), R1010–R1021.

[67] William Harvey, Michael Teng, and Frank Wood. “Near-Optimal Glimpse Sequences for Improved Hard Attention Neural Network Training”. In: *arXiv preprint arXiv:1906.05462* (2019).

[68] Hado P van Hasselt, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. “Learning values across many orders of magnitude”. In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 4287–4295.

[69] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. “Masked autoencoders are scalable vision learners”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16000–16009.

[70] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[71] 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*. 2016, pp. 770–778.

[72] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[73] Jian-Fang Hu, Wei-Shi Zheng, Lianyang Ma, Gang Wang, Jianhuang Lai, and Jianguo Zhang. "Early action prediction by soft regression". In: *IEEE transactions on pattern analysis and machine intelligence* 41.11 (2018), pp. 2568–2583.

[74] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. "Neural autoregressive flows". In: *arXiv preprint arXiv:1804.00779* (2018).

[75] Gao Huang, Yulin Wang, Kangchen Lv, Haojun Jiang, Wenhui Huang, Pengfei Qi, and Shiji Song. "Glance and Focus Networks for Dynamic Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), pp. 1–16. DOI: 10.1109/TPAMI.2022.3196959.

[76] Sheng-Jun Huang, Miao Xu, Ming-Kun Xie, Masashi Sugiyama, Gang Niu, and Songcan Chen. "Active feature acquisition with supervised matrix completion". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1571–1579.

[77] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and locally consistent image completion". In: *ACM Transactions on Graphics (ToG)* 36.4 (2017), pp. 1–14.

[78] Maximilian Ilse, Jakub Tomczak, and Max Welling. "Attention-based deep multiple instance learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2127–2136.

[79] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[80] Laurent Itti and Pierre F Baldi. "Bayesian surprise attracts human attention". In: *Advances in neural information processing systems*. 2006, pp. 547–554.

[81] Laurent Itti and Christof Koch. "A saliency-based search mechanism for overt and covert shifts of visual attention". In: *Vision research* 40.10-12 (2000), pp. 1489–1506.

[82] Laurent Itti, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis". In: *IEEE Transactions on pattern analysis and machine intelligence* 20.11 (1998), pp. 1254–1259.

[83] Desi R. Ivanova. "Introduction to Bayesian Optimal Experimental Design". In: *desirivanova.com* (2021). URL: https://desirivanova.com/post/boed-intro/.

[84] Vikram Iyer, Ali Najafi, Johannes James, Sawyer Fuller, and Shyamnath Gollakota. "Wireless steerable vision for live insects and insect-scale robots". In: *Science robotics* 5.44 (2020), eabb0839.

[85] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. "Spatial transformer networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2017–2025.

[86] Dinesh Jayaraman and Kristen Grauman. "Learning to look around: Intelligently exploring unseen environments for unknown tasks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1238–1247.

[87] Dinesh Jayaraman and Kristen Grauman. "Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion". In: *European Conference on Computer Vision*. 2016, pp. 489–505.

[88] Sergey Karayev, Tobias Baumgartner, Mario Fritz, and Trevor Darrell. "Timely object recognition". In: *Advances in Neural Information Processing Systems* 25 (2012).

[89] Angelos Katharopoulos and Francois Fleuret. "Processing Megapixel Images with Deep Attention-Sampling Models". In: *International Conference on Machine Learning*. 2019, pp. 3282–3291.

[90] Dotan Kaufman, Gil Levi, Tal Hassner, and Lior Wolf. "Temporal tessellation: A unified approach for video analysis". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 94–104.

[91] Michael D Kelly. *Edge detection in pictures by computer using planning*. Tech. rep. STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE, 1970.

[92] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[93] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[94] Durk P Kingma and Prafulla Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions". In: *Advances in neural information processing systems*. 2018, pp. 10215–10224.

[95] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. "Improved variational inference with inverse autoregressive flow". In: *Advances in neural information processing systems*. 2016, pp. 4743–4751.

[96] Christof Koch and Shimon Ullman. "Shifts in selective visual attention: towards the underlying neural circuitry". In: *Matters of intelligence*. Springer, 1987, pp. 115–141.

[97] Kurt Koffka. *Principles of Gestalt psychology*. Routledge, 2013.

[98] Yu Kong and Yun Fu. "Max-margin action prediction machine". In: *IEEE transactions on pattern analysis and machine intelligence* 38.9 (2015), pp. 1844–1858.

[99] Yu Kong, Shangqian Gao, Bin Sun, and Yun Fu. "Action prediction from videos via memorizing hard-to-predict samples". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018.

[100] Yu Kong, Dmitry Kit, and Yun Fu. "A discriminative model with multiple temporal scales for action prediction". In: *European conference on computer vision*. Springer. 2014, pp. 596–611.

[101] Yu Kong, Zhiqiang Tao, and Yun Fu. "Deep sequential context networks for action prediction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1473–1481.

[102] Alex Krizhevsky. "Learning multiple layers of features from tiny images". In: (2009).

[103] Shaofan Lai, Wei-Shi Zheng, Jian-Fang Hu, and Jianguo Zhang. "Global-local temporal saliency action prediction". In: *IEEE Transactions on Image Processing* 27.5 (2017), pp. 2272–2285.

[104] Samuli Laine and Timo Aila. "Temporal ensembling for semi-supervised learning". In: *International Conference on Learning Representations*. 2017.

[105] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. "A hierarchical representation for future action prediction". In: *European conference on computer vision*. Springer. 2014, pp. 689–704.

[106] Michael F Land and Dan-Eric Nilsson. *Animal eyes*. OUP Oxford, 2012.

[107] Hugo Larochelle and Geoffrey E Hinton. "Learning to combine foveal glimpses with a third-order Boltzmann machine". In: *Advances in neural information processing systems*. 2010, pp. 1243–1251.

[108] Simon B Laughlin. "The metabolic cost of information-a fundamental factor in visual ecology". In: *Ecology of sensing*. Springer, 2001, pp. 169–185.

[109] Simon B Laughlin, Rob R de Ruyter van Steveninck, and John C Anderson. "The metabolic cost of neural information". In: *Nature neuroscience* 1.1 (1998), pp. 36–41.

[110] Dong-Hyun Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Workshop on Challenges in Representation Learning, ICML*. Vol. 3. 2013, p. 896.

[111] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. *Tiny ImageNet Challenge*. http://cs231n.stanford.edu/2016/project.html. 2016.

[112] Kang Li and Yun Fu. "Prediction of human activity by discovering temporal sequence patterns". In: *IEEE transactions on pattern analysis and machine intelligence* 36.8 (2014), pp. 1644–1657.

[113] Zhichao Li, Yi Yang, Xiao Liu, Feng Zhou, Shilei Wen, and Wei Xu. "Dynamic computational time for visual attention". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 1199–1209.

[114] Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. "MixKD: Towards Efficient Distillation of Large-scale Language Models". In: *International Conference on Learning Representations*. 2021.

[115] Kevin J Liang, Samrudhdhi B Rangrej, Vladan Petrovic, and Tal Hassner. "Few-shot learning with noisy labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9089–9098.

[116] Ji Lin, Chuang Gan, and Song Han. "Tsm: Temporal shift module for efficient video understanding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7083–7093.

[117] Dennis V Lindley. "On a measure of the information provided by an experiment". In: *The Annals of Mathematical Statistics* (1956), pp. 986–1005.

[118] Xiao Liu, Jiang Wang, Shilei Wen, Errui Ding, and Yuanqing Lin. "Localizing by describing: Attribute-guided attention localization for fine-grained recognition". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[119] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. "Video swin transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3202–3211.

[120] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2018.

[121] Chao Ma, Sebastian Tschiatschek, Konstantina Palla, José Miguel Hernández-Lobato, Sebastian Nowozin, and Cheng Zhang. "Eddi: Efficient dynamic discovery of high-value information with partial vae". In: *arXiv preprint arXiv:1809.11142* (2018).

[122] Khoi-Nguyen C Mac, Minh N Do, and Minh P Vo. "Efficient Human Vision Inspired Action Recognition using Adaptive Spatiotemporal Sampling". In: *arXiv preprint arXiv:2207.05249* (2022).

[123] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. "The jester dataset: A large-scale video dataset of human gestures". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 2874–2882.

[124] Prem Melville, Maytal Saar-Tsechansky, Foster Provost, and Raymond Mooney. "Active feature-value acquisition for classifier induction". In: *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE. 2004, pp. 483–486.

[125] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. "Recurrent models of visual attention". In: *Advances in neural information processing systems*. 2014, pp. 2204–2212.

[126] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. "Grounded human-object interaction hotspots from video". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 8688–8697.

[127] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. "Intriguing Properties of Vision Transformers". In: *arXiv preprint arXiv:2105.10497* (2021).

[128] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. "Reading digits in natural images with unsupervised feature learning". In: *Advances in neural information processing systems* (2011).

[129] Jeremy E Niven, John C Anderson, and Simon B Laughlin. "Fly photoreceptors demonstrate energy-information trade-offs in neural coding". In: *PLoS biology* 5.4 (2007), e116.

[130] Mehdi Noroozi and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles". In: *European conference on computer vision*. Springer. 2016, pp. 69–84.

[131] Bruno A Olshausen, Charles H Anderson, and David C Van Essen. "A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information". In: *Journal of Neuroscience* 13.11 (1993), pp. 4700–4719.

[132] Lucas Paletta, Gerald Fritz, and Christin Seifert. "Q-learning of sequential attention for visual object recognition from informative local descriptors". In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 649–656.

[133] Bowen Pan, Yifan Jiang, Rameswar Panda, Zhangyang Wang, Rogerio Feris, and Aude Oliva. "IA-RED ^{2}: Interpretability-Aware Redundancy Reduction for Vision Transformers". In: *arXiv preprint arXiv:2106.12620* (2021).

[134] Guoliang Pang, Xionghui Wang, Jianfang Hu, Qing Zhang, and Wei-Shi Zheng. "DBDNet: Learning Bi-directional Dynamics for Early Action Prediction." In: *IJCAI*. 2019, pp. 897–903.

[135] Athanasios Papadopoulos, Pawel Korus, and Nasir Memon. "Hard-attention for scalable image classification". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 14694–14707.

[136] George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked autoregressive flow for density estimation". In: *Advances in neural information processing systems* 30 (2017).

[137] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. "Context encoders: Feature learning by inpainting". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2536–2544.

[138] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. "Meta pseudo labels". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11557–11568.

[139] Jie Qin, Li Liu, Ling Shao, Bingbing Ni, Chen Chen, Fumin Shen, and Yunhong Wang. "Binary coding for partial action analysis with limited observation ratios". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 146–155.

[140] Samrudhdhi B Rangrej, Kevin J Liang, Tal Hassner, and James J Clark. "GliTr: Glimpse Transformers with Spatiotemporal Consistency for Online Action Prediction". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 3413–3423.

[141] Samrudhdhi B Rangrej, Chetan L Srinidhi, and James J Clark. "Consistency driven Sequential Transformers Attention Model for Partially Observable Scenes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2518–2527.

[142] Samrudhdhi B. Rangrej and James J. Clark. "A Probabilistic Hard Attention Model For Sequentially Observed Scenes". In: *32nd British Machine Vision Conference*. 2021.

[143] Marc'Aurelio Ranzato. "On learning where to look". In: *arXiv preprint arXiv:1405.5488* (2014).

[144] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. "DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification". In: *arXiv preprint arXiv:2106.02034* (2021).

[145] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. "Learning to zoom: a saliency-based sampling layer for neural networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 51–66.

[146] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: towards real-time object detection with region proposal networks". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*. 2015, pp. 91–99.

[147] Raymond Rimey and Christopher Brown. “Selective attention as sequential behavior: Modeling eye movements with an augmented hidden Markov model”. In: (1990).

[148] Raymond D Rimey and Christopher M Brown. “Controlling eye movements with hidden Markov models”. In: *International Journal of Computer Vision* 7.1 (1991), pp. 47–65.

[149] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. “Imagenet large scale visual recognition challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[150] Michael S Ryoo. “Human activity prediction: Early recognition of ongoing activities from streaming videos”. In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 1036–1043.

[151] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. “Regularization with stochastic transformations and perturbations for deep semi-supervised learning”. In: *Advances in neural information processing systems* 29 (2016), pp. 1163–1171.

[152] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. “Deep reinforcement learning framework for autonomous driving”. In: *Electronic Imaging* 2017.19 (2017), pp. 70–76.

[153] Philipp Schwartenbeck, Thomas FitzGerald, Ray Dolan, and Karl Friston. “Exploration, novelty, surprise, and free energy minimization”. In: *Frontiers in psychology* 4 (2013), p. 710.

[154] William R Scott, Gerhard Roth, and Jean-François Rivest. "View planning for automated three-dimensional object reconstruction and inspection". In: *ACM Computing Surveys (CSUR)* 35.1 (2003), pp. 64–96.

[155] Soroush Seifi, Abhishek Jha, and Tinne Tuytelaars. "Glimpse-Attend-and-Explore: Self-Attention for Active Visual Exploration". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2021, pp. 16137–16146.

[156] Soroush Seifi and Tinne Tuytelaars. "Attend and Segment: Attention Guided Active Semantic Segmentation". In: *Computer Vision–ECCV: 16th European Conference, Glasgow, UK, Proceedings, Part XXV 16*. 2020, pp. 305–321.

[157] Pierre Sermanet, Andrea Frome, and Esteban Real. "Attention for fine-grained categorization". In: *arXiv preprint arXiv:1412.7054* (2014).

[158] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[159] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. "Fixmatch: Simplifying semi-supervised learning with consistency and confidence". In: *Advances in neural information processing systems* 33 (2020), pp. 596–608.

[160] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. "Learning structured output representation using deep conditional generative models". In: *Advances in neural information processing systems* 28 (2015), pp. 3483–3491.

[161] Ivan Sorokin, Alexey Seleznev, Mikhail Pavlov, Aleksandr Fedorov, and Anastasiia Ignateva. "Deep attention recurrent Q-network". In: *arXiv preprint arXiv:1512.01693* (2015).

[162] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. "Which tasks should be learned together in multi-task learning?" In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9120–9132.

[163] Alexandros Stergiou and Dima Damen. "Temporal Progressive Attention for Early Action Prediction". In: *arXiv preprint arXiv:2204.13340* (2022).

[164] Richard Stuart Sutton. "Temporal credit assignment in reinforcement learning". PhD thesis. University of Massachusetts Amherst, 1984.

[165] Michael J Swain and Dana H Ballard. "Color indexing". In: *International journal of computer vision* 7.1 (1991), pp. 11–32.

[166] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International Conference on Machine Learning*. 2019, pp. 6105–6114.

[167] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. "Fourier features let networks learn high frequency functions in low dimensional domains". In: *arXiv preprint arXiv:2006.10739* (2020).

[168] Charlie Tang, Nitish Srivastava, and Russ R Salakhutdinov. "Learning generative models with visual attention". In: *Advances in Neural Information Processing Systems*. 2014, pp. 1808–1816.

[169] "The British Machine Vision Association and Society for Pattern Recognition". In: *https://britishmachinevisionassociation.github.io/* (2023). Accessed on 29/03/2023. URL: https://britishmachinevisionassociation.github.io/bmvc.

[170] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. "What Makes for Good Views for Contrastive Learning?" In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 6827–6839.

[171] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. "Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training". In: *arXiv preprint arXiv:2203.12602* (2022).

[172] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. 2021, pp. 10347–10357.

[173] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. "Video classification with channel-separated convolutional networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5552–5561.

[174] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 6450–6459.

[175] Burak Uzkent and Stefano Ermon. "Learning when and where to zoom with deep reinforcement learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12345–12354.

[176] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of Machine Learning Research* 9.11 (2008).

[177] Zahra Vaseqi, Ibtihel Amara, and Samrudhdhi Rangrej. "Label Noise Resiliency with Self-supervised Representations". In: *NeurIPS 2021 Workshop: Self-Supervised Learning - Theory and Practice* (2021).

[178] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[179] GL Walls. "The evolutionary history of eye movements". In: *Vision Research* 2.1-4 (1962), pp. 69–80.

[180] Gang Wang, Wenmin Wang, Jingzhuo Wang, and Yaohua Bu. "Better deep visual attention with reinforcement learning in action recognition". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2017, pp. 1–4.

[181] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. "Temporal segment networks: Towards good practices for deep action recognition". In: *European conference on computer vision*. Springer. 2016, pp. 20–36.

[182] Miao Wang, Yu-Kun Lai, Yuan Liang, Ralph R Martin, and Shi-Min Hu. "Biggerpicture: data-driven image extrapolation using graph matching". In: *ACM Transactions on Graphics* 33.6 (2014).

[183] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. "Non-local neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803.

[184] Xionghui Wang, Jian-Fang Hu, Jian-Huang Lai, Jianguo Zhang, and Wei-Shi Zheng. "Progressive teacher-student learning for early action prediction". In: *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 3556–3565.

[185] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. "Wide-context semantic image extrapolation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1399–1408.

[186] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. "Adaptive focus for efficient video recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16249–16258.

[187] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. "Glance and Focus: a Dynamic Approach to Reducing Spatial Redundancy in Image Classification". In: *Advances in Neural Information Processing Systems* 33 (2020).

[188] Yulin Wang, Yang Yue, Yuanze Lin, Haojun Jiang, Zihang Lai, Victor Kulikov, Nikita Orlov, Humphrey Shi, and Gao Huang. "AdaFocus V2: End-to-End Training of Spatial Dynamic Networks for Video Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20062–20072.

[189] Sean Welleck, Jialin Mao, Kyunghyun Cho, and Zheng Zhang. "Saliency-based sequential image attention with multiset prediction". In: *Advances in neural information processing systems* 30 (2017).

[190] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3-4 (1992), pp. 229–256.

[191] Xinxiao Wu, Jianwei Zhao, and Ruiqi Wang. "Anticipating Future Relations via Graph Growing for Action Prediction". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021.

[192] Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang, and Jianxiong Xiao. "3d shapenets for 2.5 d object recognition and next-best-view prediction". In: *arXiv preprint arXiv:1406.5670* 2.4 (2014).

[193] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. "Unsupervised data augmentation for consistency training". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6256–6268.

[194] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10687–10698.

[195] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*. 2015, pp. 2048–2057.

[196] Zongxin Yang, Jian Dong, Ping Liu, Yi Yang, and Shuicheng Yan. "Very long natural scenery image prediction by outpainting". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 10561–10570.

[197] Alfred L Yarbus. *Eye movements and vision*. Springer, 2013.

[198] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. "Generative image inpainting with contextual attention". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5505–5514.

[199] Xiaoyu Yue, Shuyang Sun, Zhanghui Kuang, Meng Wei, Philip HS Torr, Wayne Zhang, and Dahua Lin. "Vision transformer with progressive sampling". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2021, pp. 387–396.

[200] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. "Taskonomy: Disentangling task transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3712–3722.

[201] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[202] Lingzhi Zhang, Jiancong Wang, and Jianbo Shi. "Multimodal image outpainting with regularized normalized diversification". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 3433–3442.

[203] He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. "P3IV: Probabilistic Procedure Planning from Instructional Videos with Weak Supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2938–2948.

[204] He Zhao and Richard P Wildes. "Spatiotemporal feature residual propagation for action prediction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7003–7012.

[205] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. "UCTGAN: Diverse Image Inpainting Based on Unsu-

pervised Cross-Space Translation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5741–5750.

[206] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. "Pluralistic image completion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1438–1447.

[207] Yin Zheng, Richard S Zemel, Yu-Jin Zhang, and Hugo Larochelle. "A neural autoregressive approach to attention-based recognition". In: *International Journal of Computer Vision* 113.1 (2015), pp. 67–79.

[208] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. "Temporal relational reasoning in videos". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 803–818.

[209] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. "Image BERT Pre-training with Online Tokenizer". In: *International Conference on Learning Representations*. 2021.