Hierarchical Clustering Attention for Unsupervised Object-Centric Representation Learning

by

Can Küçüksözen

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Sciences and Engineering

KOÇ ÜNİVERSİTESİ

September 1, 2022

Hierarchical Clustering Attention for Unsupervised Object-Centric Representation Learning

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Can Küçüksözen

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

_____

Prof. Yücel Yemez (Advisor)

_____

Assoc. Prof. Aykut Erdem

_____

Assoc. Prof. Yusuf Sahillioğlu

Date: _____

[Dedicated to my family, friends and loved ones for always being there for me. Without their endless support and encouragement, this would not have been possible.]

ABSTRACT

Hierarchical Clustering Attention for Unsupervised Object-Centric Representation Learning
Can Küçüksözen
Master of Science in Computer Sciences and Engineering
September 1, 2022

Extracting object-centric representations from a complex multi-object scene is indeed a crucial milestone for modern neural network architectures to achieve near human level cognition capabilities. Nevertheless, most of the contemporary neural networks that address object-centric representation learning problem require apriori initialization of a fixed set of object describing vectors or cannot manage to handle images of higher resolution. Contrary to long-standing paradigms in the literature, this work proposes Query Breaking Visual Attention (QBVA) module, an efficient and effective *building block* that introduces a "divide and conquer" strategy to object-centric representation learning while solving the unsupervised scene segmentation task. QBVA is essentially a stand-alone attention based clustering module that is capable of extracting object-centric representations from a multi-object scene when cascaded into a hierarchical network architecture. QBVA leverages a novel, fully differentiable and non-parametric clustering scheme named Query-Breaking Clustering (QBC) which eliminates the need for initializing a fixed set of clusters and holds the promise to provide dynamic representation for a variable number of objects. We demonstrate that QBVA-Net is indeed a competitive approach to address object-centric representation learning paradigm and prove to be advantageous compared to the state-of-the-art in the sense that it can provide better segmentation performance at the end of the encoder network and theoretically scale up to images of higher resolution and

ÖZETÇE

Obje Odaklı Temsil Öğrenimi için Hiyerarşik Kümeleyici Dikkat Yöntemleri

Can Küçüksözen

Bilgisayar Bilimleri ve Mühendisliği, Yüksek Lisans

1 Eylül 2022

Modern yapay sinir ağlarının insan seviyesine yakın bir görü kabiliyeti kazanması için karmaşık ve çoklu nesne içeren sahnelerden nesne-odaklı temsiller öğrenebilmeleri kritik bir öneme sahip. Fakat, günümüz yapay sinir ağları nesne-odaklı temsiller öğrenebilmek için önceden sabitlenmiş nesne sayısına göre çıkarımlar yapmaya çalışmakta veya yüksek çözünürlüklü görseller üzerinde bu görevi yerine getirememektedirler. İlgili çoğu yöntemin aksine, bu araştırma ile Sorgu Kümeleyici Görsel Dikkat (SKGD) modelini denetimsiz sahne bölütlemesi problemini çözmek ve bu esnada nesne-odaklı temsiller öğrenebilmek üzere sunuyoruz. SKGD yapısı itibari ile bahis edilen görevleri başarıyla tamamlayabilecek etkili ve verimli bir yapı taşıdır. SKGD denetimsiz sahne parçalandırması problemini "böl ve yönet" stratejisi kapsamında cevaplamaya çalışan bir kümeleyici dikkat modülüdür. Kendisinden birkaç adet ardarda sıralandığında ve hiyerarşik bir mimari inşa edildiğinde, SKGD hem yüksek çözünürlüklü resimlerde denetimsiz sahne parçalandırması yapabilir hem de daha önceden sabitlenmiş bir nesne sayısına bağımlı kalmadan ihtiyaç anında temsil kapasitesini düzenleyebilir. SKGD bu özelliklerini Sorgu Kümeleyici İşlem (SKİ) adını verdiğimiz özgün, türevlenebilir ve parametre içermeyen bir kümeleme süreci ile kazanır. Bu araştırma ile sadece SKGD katmanlarından oluşturulan bir kodlayıcının literatürdeki diğer modellere kıyasla nasıl rekabetçi bir performansa sahip olduğunu ve diğer avantajlı özelliklerini sahnelemiş olacağız.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Yücel Yemez for the endless support that he has provided to me during my M.Sc. research. I am more than grateful for his patience, motivation, enthusiasm and wisdom.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

QBVA	Query Breaking Visual Attention
SBC	Stick Breaking Clustering
QBC	Query Breaking Clustering
GNNs	Graph Neural Networks
TEB	Transformer Encoder Block
MHSA	Multi-Head Self-Attention
MLP	Multi Layer Perceptron
PN	Post-Norm
ARI	Adjusted Rand Index

Chapter 1

INTRODUCTION

Human perception excels at decomposing highly complex scenes into its constituent parts. We can effortlessly distinguish separate object instances and the background, successfully resolve occlusions and other visual perturbations, even imagine scenes where each object might be manipulated in endless ways. To facilitate such higher level cognition of our physical environment, our perception seamlessly processes the raw sensory input and organizes it in a compositional framework of basic building blocks; objects and spaces. In order to embed this intuition to the agents that we build, we often refer to neural network architectures that specialize in unsupervised learning of *object-centric representations*. Such a learning regime, where the representation of each object is kept *seperate* while being *comparable* to one another, indeed holds the promise for superior generalization capabilities for neural networks that try to explain a complex multi-object scene [Greff et al., 2020]. Despite the fact that similar methodologies can be adapted in the supervised learning regime, it is tedious and unfeasible to manually collect pixel accurate labels for each possible object category or scene composition. In addition, unsupervised object-centric models could enhance the capabilities of neural networks in a broad range of applications such as visual reasoning, modeling of structured environments, multi agent modeling and simulation of interacting physical systems [Locatello et al., 2020].

A variety of works on object-centric representation learning experimented with the task of unsupervised scene segmentation. This is mainly because image reconstruction objective in an autoencoder regime is a tempting choice for learning object-centric represenations in a fully unsupervised manner. These models try to learn an

encoder that summarizes a scene into distinct object representations and then train a coupled decoder to combine all of their information with the aim of reconstructing the original image back in the pixel space. By learning object-centric representation in such a way, the model can introduce an informational bottleneck specifically to learn discriminative features of the objects. In general, these models are developed around the following inductive bias; in order to achieve separate and comparable object representations, neural networks should reserve a separate "slot" for each object in the scene and adopt weight sharing between learnable parameters of the model to enforce comparability [Greff et al., 2020]. Roughly speaking, there are three main types of slots that are used in the related literature in order to achieve the desired properties of object-centric representation learning, these are: *Instance Slots*, *Sequential Slots* and *Spatial Slots*.

Networks that belong to the *Instance Slot* class aim at learning shared latent representations for each possible object in the dataset. They generally achieve this by iteratively refining their initial random representation. However, since all slots share a common representational format, *Instance Slots* are prone to what is known as the *routing problem*, the difficulty to break the symmetry between shared and similarly initialized slot representations. [Greff et al., 2020] Conversely, *Sequential Slots* are explicitly designed to break slot symmetries by enforcing an order on the construction of separate slot representations. Unfortunately, all known *Sequential Slot* models operate on images with full resolution, which significantly decreases their capacity to scale up to larger images. Finally, *Spatial Slots* assign a separate slot for each spatial window on the image, hence often easily break slot symmetries. Nonetheless, these kind of slot architectures struggle when presented with objects that are not well separated from each other or much bigger than the assigned slot window.

As it is discussed above, all three types of "slot" based object-centric representation schemes have their own advantages and disadvantages. With this work, we aim to *unify* the *favorable* characteristics presented by both the *Sequential* and *Spatial* Slots and *remedy* their *detrimental* qualities in the process. Stemmed from such a

motivation, we propose a novel modular building block called the Query Breaking Visual Attention Layer (QBVA) which is an *efficient and scalable* attention based clustering module that can address the unsupervised scene segmentation task when structured in a hierarchical pattern as an image encoder. QBVA is designed in order to address the lack of approaches that are both modular, following a "divide and conquer" strategy and dynamic, providing the ability to alter the number of objects that the model can represent given a variety of scenes. QBVA achieves this thanks to powerful and dynamic message passing capabilities of modern Self and Cross Attentional layers and effective clustering performance contributed by sequential masking inspired by the Stick-Breaking Clustering (SBC) process [Engelcke et al., 2021]. A single QBVA layer first facilitates message passing between all input node pairs with a traditional Self-Attention layer [Vaswani et al., 2017]. Next, QBVA projects all nodes into a shared embedding space and constructs pairwise distance maps for each node. Analyzing the freshly constructed distance maps according to a specific heuristic, which will be explained in Chapter 3, QBVA selects the top performing node in a non-parametric and differentiable way and assigns its distance map as the first output cluster. Following the initialization of the first cluster, QBVA updates all distance maps as to mask out the nodes that belong to the first cluster. After this scope update, all distance maps are analyzed again with their updated state according to the same heuristic in order to find the next cluster. This procedure is repeated until the stopping condition is met, which typically occurs when there are no nodes left to be explained. Finally, similar to [Engelcke et al., 2021, Locatello et al., 2020] the distance maps that are selected as output clusters are used to pool cluster features from the input.

In theory, being a Sequential and Spatial hybrid Slot model, QBVA can explicitly encode physical features of objects like position and size [Lin et al., 2020], scale to images of higher resolution, represent complex shaped objects or objects bigger than its local receptive field, alleviate the routing problem and dynamically adjust its representation capacity if the need emerges. QBVA Encoder, coupled with the Spatial Broadcast decoder [Watters et al., 2019], encompasses a bottom-up hierarchical

strategy for object discovery and trained end-to-end on the image reconstruction objective in a fully unsupervised fashion.

Our main contributions are as follows:

- We introduce the Query-Breaking Visual Attention module, an efficient, effective and scalable architectural component that can be used as a *basic building block* to address end-to-end unsupervised scene segmentation task while following the object-centric representation learning paradigm,
- We demonstrate that QBVA-Net can indeed perform on par with the state-of-the-art on the unsupervised scene segmentation task by analyzing pixel wise segmentation results obtained at the end of the decoder model,
- We show that QBCA-Net can also retrieve superior segmentation results at the end of the encoder model, which proves advantageous compared to the baselines.

Chapter 2

RELATED WORK

With this chapter, we review the most related work to ours, ranging from self attentional layers of recently dominant transformer architectures to set based pooling methods under the graph neural network literature. Beyond these, most essentially, we present a thorough analysis of models that are specifically designed to generate object-centric representations through the unsupervised scene segmentation problem.

2.1 Self-Attentional Layers and Transformers

The ability of "attention" to learn focusing on important regions within a context has made it a critical component in numerous neural network architectures in the recent past. Most influentially, Self-Attention and Cross-Attention layers embedded inside Transformer blocks gained enormous success spanning a diverse set of tasks, modalities and applications.[Vaswani et al., 2017, Carion et al., 2020, Dosovitskiy et al., 2020] In essence, attention based neural models are all capable because they can exchange information between any possible input element in an effective way. To refine a single node's representation, these attention methods first actively measure how all other nodes in the input set are related to the one at hand. Next, they determine the most similar elements and finally facilitate message-passing from these similar elements back to the original node. This capable information exchange that attention provides surely made its impact on various supervised and unsupervised computer vision tasks such as image classification, object detection, instance segmentation and so on.

Attention is able to achieve above mentioned capabilities thanks to its elegant design

for relating any two sets of elements and facilitate information exchange between similar element pairs. More concretely, let us consider two sets of inputs, $A \in \mathbb{R}^{N \times D}$ and $B \in \mathbb{R}^{M \times D}$ and imagine that we want to refine the representation of A by gathering related messages from B. To relate elements of B to A, determine the similar element pairs and conduct information exchange between them, Attention learns three distinct projections, namely the *queries* from A and the *keys* and *values* from B.

Thus, in a sense, queries are the handshake protocol for the message receiver, keys are the same for the message transmitter and the values are the actual message that is to be aggregated by the transmitter. It should be noted that if A an B happen to be the same data collection, this attention operation is called Self-Attention and it is termed Cross-Attention otherwise. The similarity between each query and key is scored using a scaled dot product operation. Next, for each single element in A, the softmax operator normalizes these similarity scores across all elements in B, actively measuring how important an element in B is to a specific node in A. Subsequently, based on these normalized similarity measures, the attention layer scales each message that the values hold and aggregate to the related element in A. The latter discussion can be easily put into a mathematical form as shared below where W_Q, W_K and W_V are the respective learnable projections for the queries, keys and values:

$$A_{attn} = \textbf{Softmax}\left(\frac{(W_Q \cdot A) \times (W_K \cdot B)^T}{\sqrt{D}}\right)(W_V \cdot B) \tag{2.1}$$

2.2 Graph Neural Networks and Set Pooling Methods

Another related part of the deep learning literature is the work conducted on set structured data and models that address functionalities such as set pooling, set node classification, set generation and set-input set-output mapping [Locatello et al., 2020, Zhang et al., 2019a, Bianchi et al., 2019]. Most of these models are designed as either Graph Neural Networks (GNNs) containing several message passing layers followed by a set pooling layer in series or Transformer architectures composed of a variety of Attentional layers. [Lee et al., 2019, Carion et al., 2020] A critical

requirement for any model that acts on set structured data is that it should only be composed of operations that are either *equivariant* or *invariant to permutations* in the input. [Zaheer et al., 2017]Consider the problem of object detection, if the objects in the image are translated at random around the image, the final detections that the model predicts should not be affected.

A variety of works explored different ways to conduct set pooling in recent years. The trend can be populated into three categories. First category can be generalized as models that try to initiate a fixed set of latent cluster centroids and iteratively refine them [Zhang et al., 2019a, Lee et al., 2019, Carion et al., 2020]. These models adopt a similar spirit to Instance Slots that we have mentioned in the Introduction section and recall that this iterative refinement often includes severely expensive operations such as calculating gradient descent during the forward pass. In addition, the usage of a fixed set of latent representations prevent these models to generalize to more set elements at test time. Second class of methods can be identified as the ones that try to directly predict cluster assignments scores for each input set element and use the softmax operator to push for a distinct cluster selection for each of them [Ying et al., 2018, Bianchi et al., 2019]. Even so, when used naively, this setting is ill-conditioned and these models are required to leverage a variety of auxiliary losses to enforce the cluster assignments to behave according to desired criteria. These auxiliary losses often attempt to make distinct cluster assignments orthogonal to each other, minimize the entropy of individual cluster assignments and infuse the min-cut objective principles that originate from spectral graph clustering literature. The third and last category of models attempt to somehow sort the input data and aggregate most important information [Gao and Ji, 2019, Zhang et al., 2019b]. Unfortunately, these types of methods generally suffer from the discontinuities arise due to the sorting operation and forced to invent ways to properly obtain gradients.

2.3 Unsupervised Object-Centric Representation Learning

Our model is essentially related to a series of recent work on learning object-centric representations that make use of the "slot" abstraction while addressing the task

of unsupervised scene segmentation with the image reconstruction objective. As discussed earlier, such models can be inspected under three main categories.

Instance Slots Neural networks that fall under the category of *Instance Slots* (e.g. [Greff et al., 2017, Greff et al., 2019, Locatello et al., 2020, Goyal et al., 2019]) model all objects as to share a common format, typically as a learnable multidimensional Gaussian distribution. Most often Instance Slots are initialised as a set containing a fixed number of samples from this Gaussian distribution and iteratively refined until they converge on distinct objects [Greff et al., 2020]. Unfortunately, enforcing comparability by weight sharing and initializing slots from the same distribution causes Instance Slots to suffer from a crucial challenge known as the *routing problem*: the model has to *break the symmetry* between the similarly initialized representations in order to assign distinct objects to separate slots [Greff et al., 2020]. It should also be noted that most iterative refinement techniques are expensive, some even involve gradient calculations only to refine the instance slots. Furthermore, Instance Slots operate on full image resolution, which hinders their capability to scale up to images of higher resolution.

Sequential Slots On the other hand, *Sequential Slots* are explicitly designed to break slot symmetries by enforcing an order on the construction of separate slot representations. Some successful Sequential Slot models such as [Burgess et al., 2019, Engelcke et al., 2019, Engelcke et al., 2021] use this recurrence to effectively route information. Generating a slot after another in a sequential manner enables these models to restrict the future slots from binding to an already explained object hence alleviate the problem of symmetry between the slots. One downside of these kind of models is that they also operate on full resolution images which substantially degrades their scaling up capabilities to larger images.

Spatial Slots Last but not least, *Spatial Slots* are slots that are linked with a specific spatial coordinate (e.g. in an image). Treating the input image as a grid of rectangular bounding boxes and defining a distinct slot for each such window, Spatial

Slots help to break slot symmetries and simplify information routing. Some recent successful approaches to the spatial slots can be listed as: [Crawford and Pineau, 2019, Lin et al., 2020, Jiang et al., 2019]. Nevertheless, an essential shortcoming of Spatial Slots is that their performance degrades substantially when presented with objects that are not well separated from each other or objects whose size is much larger than the corresponding bounding box of the Spatial Slot [Greff et al., 2020].

Chapter 3

METHODOLOGY

In this chapter, we present the Query-Breaking Visual Attention module and its hierarchical architecture QBVA-Net in detail and display their usage for the object-centric representation learning problem via the unsupervised scene segmentation task. Query-Breaking Visual Attention Layer takes inspiration from powerful message passing capabilities of modern Self-Attention Layers [Vaswani et al., 2017] as well as efficient and effective clustering performance provided by Stick-Breaking Process [Engelcke et al., 2021]. Despite its conceptual similarity to a more formal branch of non-parametric Bayesian methods covering Dirichlet Processes, [Nalisnick and Smyth, 2016, Yuan et al., 2019], the Query-Breaking Clustering procedure that is adapted in this work is closer in spirit to some popular Sequential Slot models such as MoNET, GENESIS and GENESIS-v2 [Burgess et al., 2019, Engelcke et al., 2019, Engelcke et al., 2021].

Being a modular layer, a number of QBVA layers can be cascaded in series to obtain a hierarchical clustering network called QBVA-Net. The overall processing pipeline for the QBVA-Net is shared with Figure 3.1. Following the retrieval of the initial features extracted by the convolutional backbone, at each layer, QBVA-Net partitions its input into sliding windows, generate clusters assignment maps for each window in parallel, pools the features corresponding to cluster assignment matrices and aggregates the final cluster features. This process repeats until QBVA-Net arrives at K separate vectors describing K distinct objects and the corresponding K cluster assignment maps that share the same resolution as the original image. After the image is encoded as described above, the Spatial Broadcast decoder [Watters et al., 2019] decodes each one of the K object representations back to the origi-

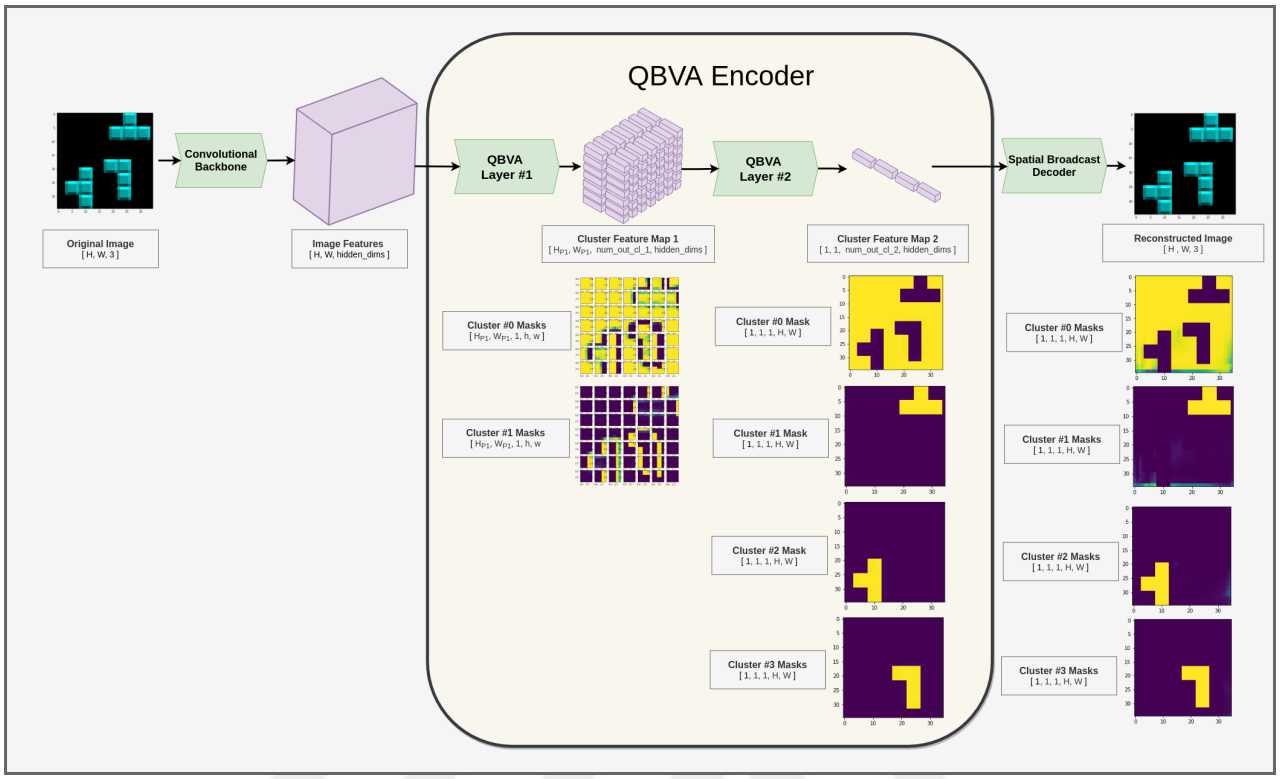

Figure 3.1: Overall computational pipeline for the proposed QBVA-Net method containing two QBVA Layers.

nal image space, combines them all to obtain the reconstruction of the image and generates cluster assignment maps for the decoder network. With such a processing pipeline, QBVA-Net can be considered as one of the first architectures that proposes a modular approach to addressing object-centric representation problem using the unsupervised scene segmentation task.

3.1 QBVA Layer

Query-Breaking Visual Attention module, displayed in Figure 3.2 is a *modular building block* that operates on sliding windows and maps an input set of N nodes into K distinct vectors that we call as output clusters. The intuition behind QBVA is that, in order to appropriately cluster an input set of nodes, one must identify pieces that are similar thus should be clustered together and distinguish pieces that are

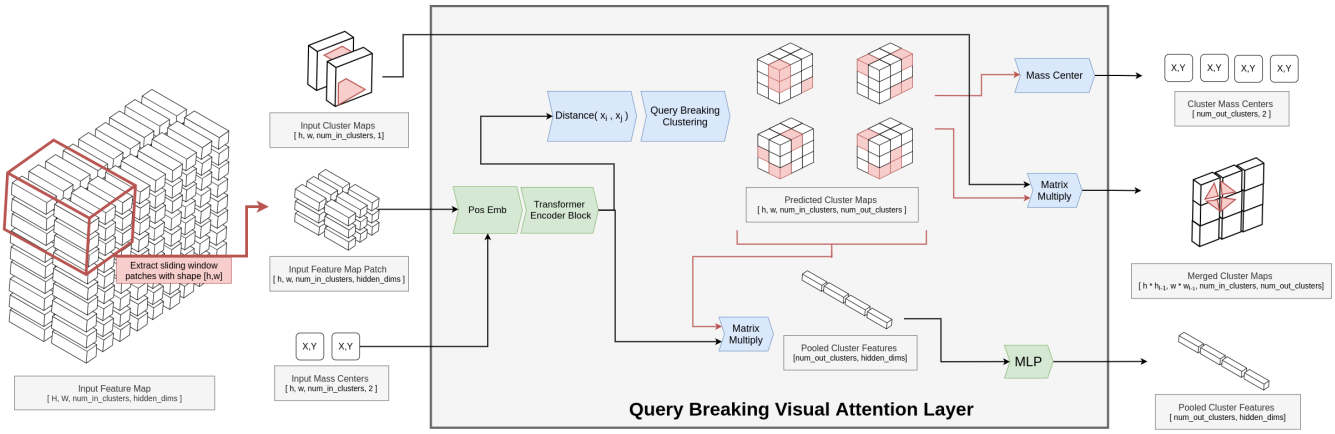

Figure 3.2: Processing pipeline of the Query Breaking Visual Attention Layer

unrelated hence should be put into different clusters. To do so, QBVA first leverages *dynamic message passing* between input elements using a popular version of Self-Attention [Vaswani et al., 2017], hence refining and relating representations of similar input elements and contrasting the ones of dissimilar ones within the local neighborhood. After two Self-Attentional layers, QBVA projects each input element onto a shared embedding space to differentiate between distinct clusterings. Using this shared embedding space, a novel clustering algorithm that is inspired by the Stick Breaking Clustering, [Burgess et al., 2019, Engelcke et al., 2019, Engelcke et al., 2021], generates cluster assignment matrices in a *non-parametric and differentiable* way. Finally, QBVA pools input features just like [Locatello et al., 2020, Engelcke et al., 2019] using freshly generated cluster assignment matrices and aggregates both the pooled features and their corresponding cluster assignments. By doing so, QBVA separates the representation of differing clusters in a sequential manner, substantially reduces the number of elements ($N \gg K$) from input to output and reserve its rights to dynamically alter the number of objects that it can represent if the need emerges.

3.1.1 Partitioning into Sliding Windows and Positional Encoding

the input image is first run through a convolutional backbone that resembles a ResNet [He et al., 2015] and the position information of individual features is in-

fused by a positional encoding scheme borrowed from [Locatello et al., 2020]. Thus, the first QBVA layer accepts a position encoded convolutional feature map X with dimensions $X \in \mathbb{R}^{H \times W \times D}$ where H and W denotes spatial dimensions height and width whereas D stands for the input feature dimension. Upon entry, QBVA separates the input into spatial patches in a sliding window fashion, resulting in the unfolded input, $X_{unf} \in \mathbb{R}^{L^2 \times N \times D}$ where L is the number of windows along a spatial axis, $N = W_P \cdot H_P$ and W_P, H_P denote the specific patch dimensions. QBVA processes each such patch in parallel, hence alleviate the scalability issues that most of the contemporary slot based approaches face. To infuse position information, just like Slot Attention, we produce a 4-dimensional vector for each possible coordinate inside the input. These 4 dimensions embed the distance of the specific coordinate to 4 different outer edges of the image, scaled between $[-1, 1]$. Next, we feed each of these 4-d vectors through a full-connected layer to increase their dimensionality from 4 to the hidden dimension of the model and finally we sum up these freshly upsampled position encoding vectors with the original feature map.

3.1.2 Refining the Input Nodes

Next, in order to relate and contrast appropriate input nodes in a specific patch, QBVA leverages two cascaded Transformer Encoder blocks, as it is originally proposed in the seminal paper by [Vaswani et al., 2017]. Our work assumes that the node features obtained from the image via early convolutional layers, in fact admit a good initialization to obtain the initial cluster assignments due to the homophily property of modern neural networks [Mcpherson et al., 2001, Bianchi et al., 2019]. In addition, the features of nodes in closely related groupings tend to become more similar due to the smoothing effect of message passing operations such as Self-Attention [Bianchi et al., 2019]. In more detail, a single Transformer Encoder Block (TEB) in QBVA adopts a Multi-Head Self-Attention (MHSA) layer followed by a residual Multi Layer Perceptron (MLP) in a Post-Norm (PN) configuration [Liu et al., 2021]. Mathematically, this corresponds to generating auxiliary embeddings for all of the $N = H_P \times W_P$ unfolded input nodes as:

$$X_{attn} = \textbf{Softmax}\left(\frac{(\mathbf{W_Q} \cdot X_{unf}) \times (\mathbf{W_K} \cdot X_{unf})^T}{\sqrt{D}}\right)(\mathbf{W_V} \cdot X_{unf}) \tag{3.1}$$

$$X_{attn} = \textbf{LayerNorm}\,(X_{unf} + X_{attn}) \tag{3.2}$$

$$X_{attn} = \textbf{LayerNorm}\,(X_{attn} + \textbf{MLP}(X_{attn})) \tag{3.3}$$

where W_Q, W_K and W_V stand for the query, key and value projections respectively. To ease the notation used, the attention formulation shared above involves only a single attention head, for the details of the multi-headed version, the reader can refer to the original work [Vaswani et al., 2017]. With applying two consecutive TEBs to the input as decribed above, our assumption is that, message passing is carried out between elements that should be clustered together and nearly no information is exchanged between elements that ought to be in separate clusters. After refining input nodes with a several MHSA operations, QBVA is now ready to apply its novel clustering procedure called the Query-Breaking Clustering (QBC).

3.1.3 Query-Breaking Clustering

The QBC is a differentiable and non-parametric algorithm inspired by "scope" based sequential masking introduced by [Burgess et al., 2019] and thereafter successfully adapted by [Engelcke et al., 2019]. The QBC is designed to separate the input nodes into a variable number of clusters according to their representational similarity. In order to establish a common ground to measure similarity between varying nodes in a local neighborhood, QBC first projects each node's features onto a shared embedding space by using a dedicated MLP, then forces this feature vector to select its own dominant feature by applying a LayerNorm [Ba et al., 2016] operation:

$$X_{embed} = \textbf{LayerNorm}\,(\mathbf{MLP_{proj}}\,(X_{attn})) \tag{3.4}$$

Following the shared projection, QBVA constructs a soft Affinity Matrix, $\mathbf{A} \in [0,1]^{M^2 \times N \times N}$, where $N = H_P \times W_P$. The affinity matrix $\mathbf{A}$ encodes whether or not each pair of nodes are connected to each other, in terms of soft binary values (i.e. $\in [0,1]$). Thus, in contrast to the SBC procedure found in MoNET, GENESIS

and GENESIS-v2 where a single attention map is produced at each iteration, QBC starts off with constructing an all-to-all node affinity matrix and sequentially finds distinct clusters amongst them. Dealing with full scale affinity matrices are generally known to be prohibitive in terms of time and memory efficiency (i.e. $\mathcal{O}(n^2)$) however, since QBC operates on a large number of small patches in parallel, (i.e. $H \gg H_P$), the computational overhead can be mitigated. In order to construct the aforementioned affinity matrix, we first compute the Euclidean Distance between the representations of each node pair. Then, we normalize the distances from each node to all other nodes within the patch using the *Softmin* operator. Finally, we re-scale the output of the Softmin operation between 0 and 1 using the min-max normalization scheme. To make the above discussion a bit more concrete, related equations are given below:

$$\mathcal{D}_{ij}^{N} = \underset{j}{\mathbf{Softmin}}\left(\|X_{embed}^{i} - X_{embed}^{j}\|^2 \right) \quad \text{and} \quad \mathbf{A_{ij}} = \frac{\mathcal{D}_{ij}^{N}}{\max\left(\mathcal{D}_{i}^{N}\right)} \tag{3.5}$$

where $\mathcal{D}_{ij}^{N}$ denotes the Softmin normalized Euclidean distances between each node pair i, j and $\mathbf{A_{ij}}$ stands for the constructed affinity matrix. At this point in the processing pipeline, QBVA layer obtains a soft binary value (i.e. $\in [0, 1]$) for each possible input node pair i and j, indicating how strongly they are connected to each other.

After the construction of the initial affinity matrix, we treat each row of it as a singular attention map of a specific input node, $\mathbf{A_i} \in [0, 1]^{M^2 \times 1 \times N}$. Under such a setting, the QBC procedure aims at finding the attention maps with the *highest energy* and *lowest entropy* scores at each step. Energy of an attention map can be easily defined as the Frobenious Norm of the attention matrix. On the other hand, in the realm of information theory, entropy $H(X)$ of a random variable X that takes on values in space $\mathcal{X}$ and distributed according to $p : \mathcal{X} \to [0, 1]$, can be described as the amount of *surprise* or *uncertainty* that is inherent to the possible outcomes of X. This intuition, that is measuring the amount of *uncertainty* in a probabilistic

event is embedded to mathematical terms as:

$$\mathrm{H}(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{3.6}$$

where $\sum$ denotes summation over values that X's might possibly take on. [Pathria and Beale, 2011]Thus, QBC aims at discovering attention maps that yield high magnitudes and low uncertainty.

Following the scoring of each attention map according to their energy and entropy properties, QBC is required to select the top performing one. This selection of highest scoring attention map indeed admits a discrete categorical distribution, a subject that most of the contemporary neural networks fail at dealing with. This is due to the fact that discreteness introduced into the computational pipeline prevents naive usage of the back-propagation algorithm to obtain gradients during the optimization stage. This is primarily because mathematical operations that involve discrete jumps are not differentiable. To overcome this challenge of obtaining useful gradients while being able to select the top performing attention map in a categorical way, we refer to the Gumbel-Softmax operator [Jang et al., 2016]. Gumbel-Softmax is an efficient gradient estimator that approximates the non-differentiable sample from a discrete categorical distribution with a continuous relaxation of it, hence is capable of calculating useful gradients using the prominent "reparametrization trick".[Jang et al., 2016]

Now that the top performing attention map is discovered in a differentiable way, QBC assigns the active nodes of this map as its next output cluster. After the first cluster assignment is complete, QBC updates all of the remaining attention maps to *mask out* the recently clustered nodes and this overall process is repeated until there are no nodes left to be explained. The latter update of the attention maps, where we mask out the recently clustered nodes, is implemented as the scope update mechanism that can be found in the SBC procedure of models [Burgess et al., 2019, Engelcke et al., 2019, Engelcke et al., 2021]. The pseudo-algorithm for the QBC procedure is shared with Algorithm 1.

Notice that the search for distinct clusters carried out by the QBC algorithm is

Algorithm 1 Query-Breaking Clustering Process. The input to the process is the constructed affinity matrix $\mathbf{A}$ and embedded feature tensor $\mathbf{X_{embed}} \in \mathbb{R}^{N \times D}$, containing each vector in a spatial window. The process generates a set of K output clusters in a differentiable and non-parametric way.

Inputs: Affinity Matrix: $\mathbf{A} \in [0,1]^{N \times N}$, Embedded Features: $\mathbf{X_{embed}} \in \mathbb{R}^{N \times D}$, Number of output clusters: $\mathbf{K}$

Initialize: Scope List: $\mathbf{S} \in \left[\mathbf{1}^{1 \times N}\right]$, Input Cluster List: $\mathbf{C}_i \in \varnothing$

Outputs: Output Cluster List: $\mathbf{C} \in [0,1]^{K \times N}$

for `i in range(`**K**`):`
    `masked_maps =` **A** $*$ **S**`[i]`
    `energy_score = Norm(masked_maps, axis=-1)`
    `entropy_score =` $-1*$`Entropy(masked_maps, axis=-1)`
    `energy_score = exp(LayerNorm(energy_score, axis=-2))`
    `entropy_score = exp(LayerNorm(entropy_score, axis=-2))`
    `score = energy_score` $*$ `entropy_score`
    `cluster_node = Gumbel_Softmax(score, hard = True, axis=-2)`
    `cluster_map =` **A**`[cluster_node, :]`
    C`.append(`**S**`[i]` $*$ `cluster_map)`
    S`.append(`**S**`[i]` $*$ `(1`$-$`cluster_map))`
end for

designed to obtain optimum clustering performance when presented with an affinity matrix. The model prefers to select attention maps that explain a large portion of the input, hence favors maps with higher total energy. At the same time, QBC tries to penalize wrong and/or uncertain groupings by endorsing a lower entropy score.

3.1.4 Pool and Aggregate

Once the cluster assignment matrices are finalized by the Query-Breaking Clustering Process, QBVA pools its input features according to the freshly generated cluster assignment matrices and aggregate these pooled features. With the pooling opera-

tion, QBVA generates an output cluster feature map of size $X_{clus} \in \mathbb{R}^{L^2 \times K \times D}$ where L is the number of windows that are processed in parallel, K denotes the number of clusters and D stands for the hidden dimensions. Furthermore, as soon as the second cluster assignment matrix is generated by QBVA-Net, we merge these consecutive assignments in a dendrogram like fashion, hence increasing the spatial resolution of cluster assignments at each step. Finally, in order to facilitate positional encoding at each QBVA Layer, we first obtain the centre of mass of each cluster assignment map. This can be simply achieved by calculating a weighted average of the cluster assignment map along both the horizontal and vertical axes. In the end, QBVA aggregates 3 items; pooled cluster features, merged cluster assignment maps and corresponding centre of mass coordinates.

3.1.5 Timing and Memory Considerations

Compared to Slot Attention [Locatello et al., 2020], QBVA-Net is almost as fast, in terms of both the memory and timing complexity. On Tetrominoes dataset, when the batch size is set to 64 and a single T4 GPU is used, QBVA-Net takes 8 hours to train whereas Slot Attention takes 7. In addition, a single layer of QBVA has a $\mathcal{O}(M^2)$ time and $\mathcal{O}(M)$ space complexity, where $M = H_P \times W_P$ is the number of all nodes in a spatial patch, similar to vanilla Self-Attention operation and most of the hierarchical clustering approaches [Sibson, 1973]. This $\mathcal{O}(M^2)$ complexity is essentially generated by pairwise distance calculation between all nodes in a patch. After the distance calculation is done, QBVA selects clusters in linear time, hence, $\mathcal{O}(M^2)$. Despite having $\mathcal{O}(M^2)$ time and $\mathcal{O}(M)$ space complexity per layer, QBVA balances this computational load by operating on a large batch of small spatial patches in parallel, hence keeping the value of M small. Furthermore, at each layer, QBVA substantially reduces the number of input elements to output elements, i.e. $(M \gg K)$, hence introducing significant advantages in terms of timing and memory complexity as the model runs deeper. For instance, at first layer, QBVA decomposes the input feature map with $N \times N$ pixels into $\left(\frac{N}{N_{P_0}}\right)^2$ windows where each such window has $N_{P_0}^2 = M_0$ pixels. QBVA processes each window in parallel, thus,

first layer introduces a time complexity of $\mathcal{O}(M_0^2)$ due to pairwise distance matrix calculations. Then, this first layer outputs C_0 singular vectors for each window, where C_0 is the desired number of output clusters for the first layer. Therefore, second layer of QBVA-Net obtains a feature map containing a total of $\left(\frac{N}{N_{P_0}}\right)^2 \times C_0$ input nodes, decomposes it into $\left(\frac{N}{N_{P_0} \times N_{P_1}}\right)^2$ windows where each window contains $N_{P_1}^2 = M_1$ nodes. Accordingly, the second layer of QBVA-Net introduces a time complexity of $\mathcal{O}(M_1^2)$ where M_1 is generally selected as $M_1 \leq M_0$.

Chapter 4

EXPERIMENTS

The aim of this chapter is to assess the capabilities of the proposed Query Breaking Visual Attention layer on an object-centric learning paradigm, namely, unsupervised scene segmentation.

Baseline We compare our method to one of the state-of-the-art work for this task, the Slot Attention module [Ronneberger et al., 2015]. Recall the discussion about Instance Slots back in Chapter 2, Slot Attention is a member of Instance Slots class where a single multidimensional Gaussian distribution is learned as a meta feature vector. Upon retrieving position encoded convolutional features, Slot-Attention first samples a fixed size of points from the latter Gaussian distribution and iteratively refines this samples using a recursive attention scheme that enforces each input pixel to select a specific slot. However, the number of elements that Slot-Attention can model should be determined apriori to its usage and since it operates on full scale convolutional output, it wouldn't scale nicely to images of higher resolutions. We now present detailed experimental setup and implementation specifics for both the baseline and the proposed model. To establish a fair comparison of QBVAs segmentation performance as an image encoder to the baseline, we make use of the exact same decoder architecture as Slot Attention, the Spatial Broadcast Decoder. Furthermore, we train our model using only the image reconstruction loss, just like Slot Attention. Finally, we provide the necessary quantitative and qualitative results showcasing the advantages of our model.

Dataset For the unsupervised scene segmentation task, we use the Tetrominoes [Greff et al., 2019] multi object dataset, which is a popular synthetic dataset that has

been used in many recent works addressing the problem at hand. Similar to [Greff et al., 2019, Locatello et al., 2020] we use the first 60K samples of Tetrominoes for training and evaluate on 320 test samples.

4.1 Unsupervised Scene Segmentation on Tetrominoes

Training As mentioned previously, the training takes place in a fully unsupervised fashion. Only information bearing signal for the model is the mean squared error calculated for image reconstruction. Similar to the baseline, our model is trained using the Adam optimizer [Kingma and Ba, 2014] with a batch size of 64 images. Our method makes use of the linear warm-up, exponential decay paradigm for the learning rate and the maximum learning rate reached after linear warm up is set to be 1×10^{-4}. For the Tetrominoes dataset, we use a fixed number of object slots, $K = 4$ for easier parallelism on GPU however, it should be noted that QBVA enjoys dynamic selection of number of objects that it can represent, thanks to its sequential slot generation scheme. Loss plots describing the training behavior of both the baseline Slot Attention and the proposed QBVA-Net is shared with Figure 4.1. Inspecting the plots, one can concur that both models behave similarly during their corresponding training processes.

Metrics In accordance with the baseline and related work, we assess the segmentation maps that are obtained for each separate object at the end of the decoder pipeline against the ground truth segmentation maps. In contrast to earlier work, we propose to compare the performance of our model against the baseline using the segmentation maps that are generated at the end of the encoder network. To quantitatively assess all these maps, we refer to the widely adopted clustering similarity metric called the Adjusted Rand Index (ARI) [Rand, 1971].

Results Qualitative results are displayed in Figure 4.2 whereas Table 4.1 summarizes the quantitative results. Looking at both qualitative and quantitative results, it can be observed that our proposed methodology compares on par with the state-of-the-art baseline Slot Attention model at unsupervised segmentation task using

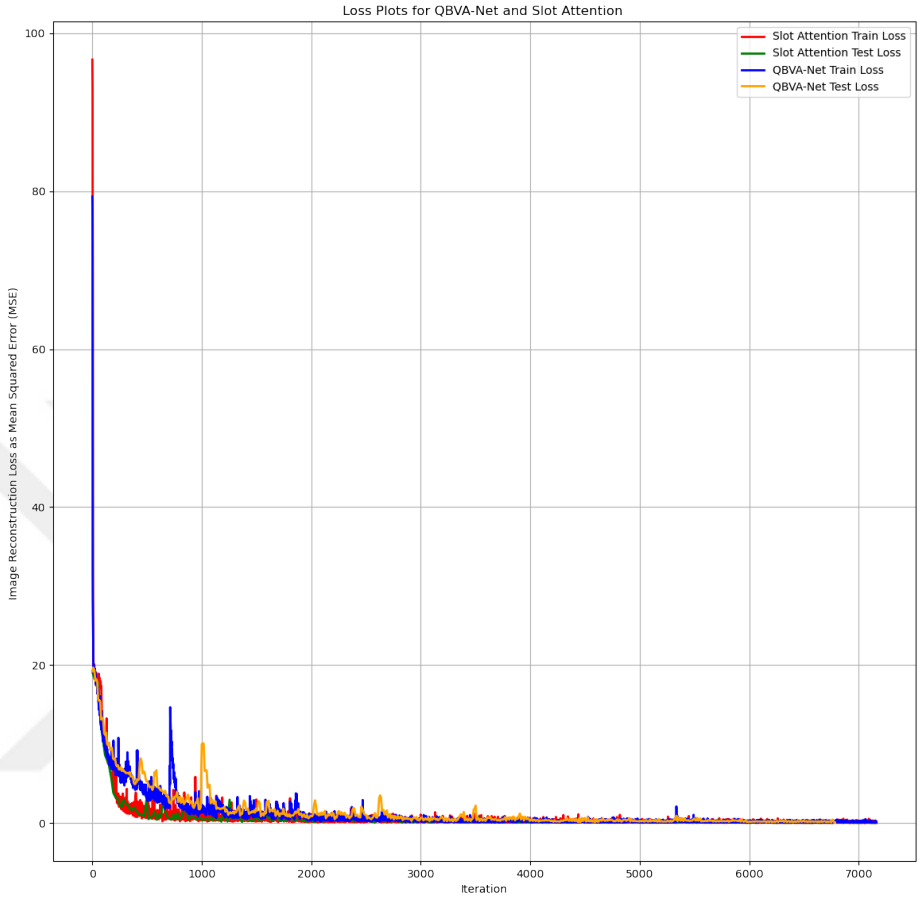

Figure 4.1: Loss Plots obtained for the baseline Slot Attention and Proposed QBVA-Net models.

predictions generated at the end of the decoder sub-network. Moreover, it is evident from the shared results that the proposed QBVA layer and it's hierarchical architecture QBVA-Net proves to be favorable compared to Slot Attention at the same task using predictions obtained at the end of the Encoder pipeline. In addition, it should be noted that QBVA layer does not explicitly construct encoder segmentation maps on the original pixel space but rather builds up a combination of its own segmentation maps that were first created as the output of the QBVA stem layer.

Table 4.1: Encoder and Decoder Quantitative Segmentation Performance

Model	Encoder	Decoder
Slot Attention	95.9	99.5
QBVA-Net (ours)	97.2	99.2

Adjusted Rand Index (ARI) scores (in %, mean ± stddev for 3 seeds) for unsupervised object discovery in multi-object dataset Tetrominoes.

In other words, despite being more accurate when it comes to encoder generated segmentation maps, QBVA achieves this performance as a byproduct of its hierarchical learning scheme. Furthermore, it is interesting to observe that QBVA-Net manages to capture the background with a dominant and separate slot whereas Slot Attention tends to equally distribute background elements to each slot.

Looking at the qualitative results of encoder generated segmentation maps that are shared with Figure 4.2, one can clearly distinguish QBVA's performance over Slot Attention. On the other hand, it is apparent that both QBVA and Slot Attention perform akin to each other when the decoder generated segmentation maps are evaluated. Further, we examine that the Query Breaking Visual Attention layer is as efficient as its competitor Slot Attention, when both approaches are compared as singular modules. Nonetheless, Slot Attention is a stand-alone module that can only operate on full image resolution, hence the efficiency advantages of QBVA Layer begins to dominate over its counterpart when the input image size is scaled up to higher resolutions.

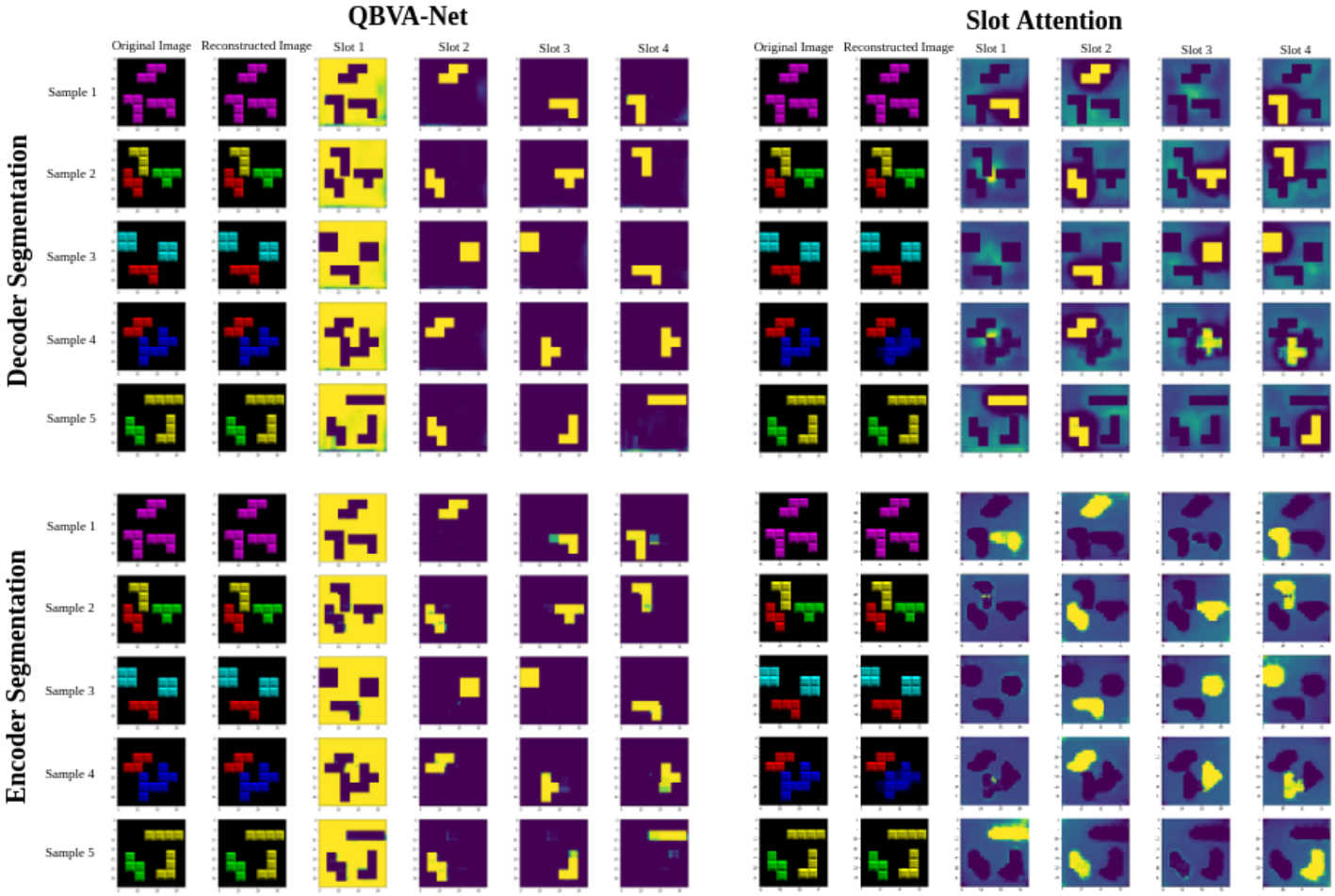

Figure 4.2: Qualitative segmentation results obtained for QBVA-Net and Slot Attention from both the encoder and decoder subnetworks.

Chapter 5

CONCLUSION

We have proposed, defined and analyzed the Query Breaking Visual Attention module, an efficient and effective building block that introduces a "divide and conquer" strategy to object-centric representation learning while solving the unsupervised scene segmentation task. Being a hierarchical image segmentation architecture that is composed of a series of the same module, QBVA holds the promise to scale up to images of higher resolutions without any drastic increase in computational resources, unlike any other baseline. In addition, output cluster assignments that are generated by the QBVA architecture is not restricted by a fixed representational capacity, which proves advantageous over most related competitors. Experiments conducted on the Tetrominoes multi-object dataset has demonstrated highly competitive segmentation performance of QBVA both at the end of the encoder and the decoder subnetworks.

As an intuitive next step, we plan on increasing the diversity of our experiments by testing the unsupervised scene segmentation performance of QBVA-Net on new multi-object datasets. In addition, we are eager to explore additional decoder designs that may be more suited for a hierarchical encoder architecture such as the proposed QBVA-Net. A decoder design that is closer in spirit to the widely adopted U-Net [Ronneberger et al., 2015] architecture would be a reasonable and favorable canditate. Further, we intend to assess the quality of object-centric representations that QBVA learns by employing QBVA-Net for differing downstream tasks such as object detection, set prediction or visual reasoning.[Locatello et al., 2020] Finally, we are determined to evaluate QBVA-Nets scaling abilities by testing it on a dataset that contains higher resolution images compared to datasets that any other related

work uses.

BIBLIOGRAPHY

[Ba et al., 2016] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization.

[Bianchi et al., 2019] Bianchi, F. M., Grattarola, D., and Alippi, C. (2019). Spectral clustering with graph neural networks for graph pooling.

[Burgess et al., 2019] Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. (2019). Monet: Unsupervised scene decomposition and representation.

[Carion et al., 2020] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers.

[Crawford and Pineau, 2019] Crawford, E. and Pineau, J. (2019). Spatially invariant unsupervised object detection with convolutional neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3412–3420.

[Dosovitskiy et al., 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

[Engelcke et al., 2021] Engelcke, M., Jones, O. P., and Posner, I. (2021). Genesis-v2: Inferring unordered object representations without iterative refinement.

[Engelcke et al., 2019] Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. (2019). Genesis: Generative scene inference and sampling with object-centric latent representations.

[Gao and Ji, 2019] Gao, H. and Ji, S. (2019). Graph u-nets.

[Goyal et al., 2019] Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. (2019). Recurrent independent mechanisms.

[Greff et al., 2019] Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. (2019). Multi-object representation learning with iterative variational inference.

[Greff et al., 2017] Greff, K., van Steenkiste, S., and Schmidhuber, J. (2017). Neural expectation maximization.

[Greff et al., 2020] Greff, K., van Steenkiste, S., and Schmidhuber, J. (2020). On the binding problem in artificial neural networks.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

[Jang et al., 2016] Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax.

[Jiang et al., 2019] Jiang, J., Janghorbani, S., de Melo, G., and Ahn, S. (2019). Scalor: Generative world models with scalable object representations.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

[Lee et al., 2019] Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. (2019). Set transformer: A framework for attention-based permutation-invariant neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR.

[Lin et al., 2020] Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. (2020). Space: Unsupervised object-oriented scene representation via spatial attention and decomposition.

[Liu et al., 2021] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao,

Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. (2021). Swin transformer v2: Scaling up capacity and resolution.

[Locatello et al., 2020] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. (2020). Object-centric learning with slot attention.

[Mcpherson et al., 2001] Mcpherson, M., Smith-Lovin, L., and Cook, J. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–.

[Nalisnick and Smyth, 2016] Nalisnick, E. and Smyth, P. (2016). Stick-breaking variational autoencoders.

[Pathria and Beale, 2011] Pathria, R. and Beale, P. D. (2011). 3 - the canonical ensemble. In Pathria, R. and Beale, P. D., editors, *Statistical Mechanics (Third Edition)*, pages 39–90. Academic Press, Boston, third edition edition.

[Rand, 1971] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

[Sibson, 1973] Sibson, R. (1973). Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16:30–34.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

[Watters et al., 2019] Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. (2019). Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes.

[Ying et al., 2018] Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. (2018). Hierarchical graph representation learning with differentiable pooling.

[Yuan et al., 2019] Yuan, J., Li, B., and Xue, X. (2019). Generative modeling of infinite occluded objects for compositional scene representation. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7222–7231. PMLR.

[Zaheer et al., 2017] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., and Smola, A. (2017). Deep sets.

[Zhang et al., 2019a] Zhang, Y., Hare, J., and Prugel-Bennett, A. (2019a). Deep set prediction networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Zhang et al., 2019b] Zhang, Y., Hare, J., and Prügel-Bennett, A. (2019b). Fspool: Learning set representations with featurewise sort pooling.