# A Receptor Skeleton for Capsule Neural Networks

Jintai Chen [1] Hongyun Yu [1] Chengde Qian [2] Danny Z. Chen [3] Jian Wu [4]

## Abstract

In previous Capsule Neural Networks (CapsNets), routing algorithms often performed clustering processes to assemble the child capsules' representations into parent capsules. Such routing algorithms were typically implemented with iterative processes and incurred high computing complexity. This paper presents a new capsule structure, which contains a set of optimizable receptors and a transmitter is devised on the capsule's representation. Specifically, child capsules' representations are sent to the parent capsules whose receptors match well the transmitters of the child capsules' representations, avoiding applying computationally complex routing algorithms. To ensure the receptors in a CapsNet work cooperatively, we build a skeleton to organize the receptors in different capsule layers in a CapsNet. The receptor skeleton assigns a share-out objective for each receptor, making the CapsNet perform as a hierarchical agglomerative clustering process. Comprehensive experiments verify that our approach facilitates efficient clustering processes, and CapsNets with our approach significantly outperform CapsNets with previous routing algorithms on image classification, affine transformation generalization, overlapped object recognition, and representation semantic decoupling.

## 1. Introduction

Capsule Neural Networks (CapsNets) (Sabour et al., 2017; Hinton et al., 2018; Hahn et al., 2019; Rajasegaran et al., 2019; Tsai et al., 2020; Ribeiro et al., 2020b) offer effective capabilities of assembling object part representations to synthesize object whole representations for better generalization and robustness (see Fig. 1(a)), thus performing well in image classification (Tsai et al., 2020), affine transformation generalization (Sabour et al., 2017), and occluded/ overlapped object (e.g., organs in an X-ray) recognition (Sabour et al., 2017; LaLonde & Bagci, 2018; Bonheur et al., 2019). In a typical CapsNet, several neurons are grouped as units, called capsules, whose representations are convex combinations of (possibly some of) the child capsule representations under the guidance of various routing algorithms.

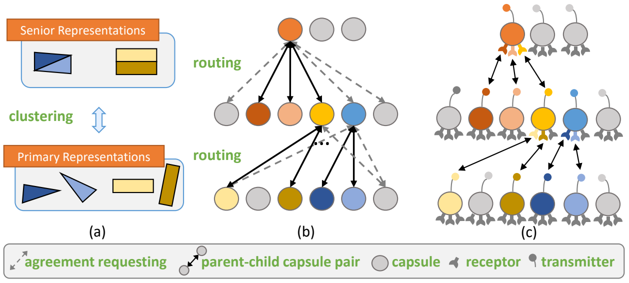


*Figure 1.* (a) The idea of CapsNets, which clusters low-level representations to synthesize higher-level representations. (b) The typical routing algorithms (Sabour et al., 2017; Hinton et al., 2018) use iterative processes to implement clustering, requesting agreements between the parent and child capsules repeatedly. (c) In our approach, we define receptors and transmitters for capsules, and cluster the child capsules' representations for the parent capsules based on the matching between the parent capsules' receptors and the transmitters of the child capsules' representations.

It was pointed out (Malmgren, 2019) that typical routing algorithms are in essence clustering algorithms, organizing the capsules in different layers to hierarchically assemble the representations of object parts into object wholes. For example, Dynamic Routing (Sabour et al., 2017) was a soft $k$-Means, and EM Routing (Hinton et al., 2018) was based on the Gaussian mixture model (Reynolds, 2009). Clustering in such a routing algorithms is typically implemented by iterative processes (see Fig. 1(b)), which repeatedly request agreements between the representations of the parent and child capsules before finalizing the clusters. Unlike a neural layer (e.g., a fully connected layer) in which low-level representations are combined into high-level representations with the parameter coefficients, those routing algorithms assemble low-level representations to synthesize high-level

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China; [2]School of Statistics and Data Science, Nankai University, China; [3]Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA; [4]The First Affiliated Hospital, and Department of Public Health, Zhejiang University School of Medicine, Hangzhou, China;. Correspondence to: Jian Wu <wujian2000@zju.edu.cn>.

*Proceedings of the 38th International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

representations according to the coefficients determined by the similarities with the estimated cluster centroids.

However, the iterative processes in routing algorithms incur high computing complexity. Then, some straight-through routing algorithms were given to avoid iterative processes, such as single-layer perceptron (Hahn et al., 2019), attention module (Choi et al., 2019), and Gumbel-Softmax (Ahmed & Torresani, 2019). These straight-through routing algorithms passed representations forward according to the learnable parameters, and did not explicitly cluster low-level representations. An inverted dot-product based attention routing (Tsai et al., 2020) used a space-time trade-off strategy but did not fundamentally reduce the computing complexity. In summary, clustering in a routing algorithm picks major representations and learns the part-whole relationship (Sabour et al., 2017). However, the previous routing algorithms did not provide an efficient way to handle clustering; thus, the low computing complexity of these routing algorithms did not seem to directly benefit representation clustering.

In this work, we present a new capsule structure by adding optimizable receptors to capsules, in which the receptors are on behalf of cluster centroids in the representation clustering. Also, we devise a transmitter on the capsule's representation. Using our approach, the representations of the parent capsules are convex combinations of the child capsules' representations whose transmitters match well with the parent capsule's receptors (as illustrated in Fig. 1 (c)). Previous routing algorithms obtained the combination coefficients by iteratively computing the capsule representation similarities, while our approach directly determines the coefficients based on the similarities of receptors and transmitters, thus avoiding the iterative process of common routing algorithms. Since our approach performs a process similar to information transmission among neurons in which neurotransmitters released by a neuron are received by some specific receptors of the target neurons, we call the proposed compositions *transmitters* and *receptors*, respectively.

As the receptors are learnable and determine which child capsule representations to pick, it is clear that the capsule's representations can be highly affected by its receptors. Previous CapsNets naturally learn the part-whole relationship hierarchically via layer-by-layer clustering. But, in this design, if each receptor in a CapsNet operates independently, then there is no guarantee that all the clustering centroids represented by the receptors serve for the part-whole relationship capture. Hence, we embed a receptor skeleton into the CapsNet to organize the receptors, constructing associated relations among receptors. In (Wan et al., 2020), the average of representations was used to obtain their higher-level semantics, providing good interpretations. Inspired by this, we require that the parent capsules' receptors be the averages of some receptors in the child capsules using the skeleton, embedding a hierarchical relationship of receptors in adjacent layers. Thus, the parent capsules' receptors can be regarded as "outlines" of the child capsules' receptors, while the child capsules' receptors can be viewed as various "embodiments" of the parent capsules' receptors. In data processing, as the receptors represent the cluster centroids, the representations can be grouped and combined hierarchically by "attaching" to the receptors of the skeleton. To avoid introducing biases to this skeleton, we propose three reasonable conditions for the skeleton topology, in order to ensure that the receptors are treated uniformly and specifically, and provide a solution for finding a skeleton topology inspired by the Latin square design (Colbourn & Dinitz, 2006).

In summary, the contributions of our work are as follows:

- We develop a new capsule structure and a new approach to assemble the child capsule representations into the parent capsules using a process similar to information transmission among neurons. Our approach performs a representation clustering process without using the time-consuming iterations of common routing algorithms.
- We propose a receptor skeleton to specify a hierarchical relationship of receptors in adjacent layers which provides a share-out objective to each receptor, making the receptors in a CapsNet work collectively to recognize objects by hierarchically organizing objects via a composition of part representations.
- Extensive experiments conducted on several datasets demonstrate the superiority of our approach, which outperforms the previous routing algorithms with lower computing complexity on various applications, especially occluded/overlapped object recognition and affinity transformation generalization.

## 2. Preliminaries

### 2.1. CapsNets

In (Sutskever et al., 2011), neurons in a layer were divided into groups called capsules, and a capsule neural network (called CapsNet) was proposed (Sabour et al., 2017) in which capsule representations were presented as vectors. A matrix capsule was introduced in (Hinton et al., 2018) which is beneficial for dealing with image data. Later, many CapsNets (e.g., (Tsai et al., 2020; Choi et al., 2019; Ribeiro et al., 2020a;b)) followed these capsule structure, and various routing algorithms were explored. When voting which child capsule representations to be used by the parent capsules, given the representations $\mathbf{u}_i$ of the $i$-th child capsule, it is first transformed into representation votes $\hat{\mathbf{u}}_{j|i}$ for the

$j$-th parent capsule, by:

$$\hat{\mathbf{u}}_{j|i} = W_{ij}(\mathbf{u}_i) \tag{1}$$

where $W_{ij}(\cdot)$ is a learnable transformation function (e.g., an affine transformation). Then, a routing algorithm combines $\hat{\mathbf{u}}_{j|i}$ ($i \in \{1, 2, \ldots, A\}$) to synthesize the representations of the $j$-th parent capsule, $\mathbf{u}_j$, by $\mathbf{u}_j = \sum_{i=1}^{A} c_{j|i}\hat{\mathbf{u}}_{j|i}$, where the coefficients $c_{j|i}$ of $\hat{\mathbf{u}}_{j|i}$ are computed based on their similarities to the estimated cluster centroids and $A$ is the number of child capsules. Note that some of the coefficients $c_{j|i}$ are close to zero (since the clustering processes in routing algorithms are soft versions (Malmgren, 2019)). Related work on new capsule structures was limited, including probabilistic models (Ribeiro et al., 2020b), stacked capsule autoencoder (Kosiorek et al., 2019), and 3D versions for point clouds (Zhao et al., 2019; Srivastava et al., 2019). Our paper presents a new capsule structure, which provides individual parameters to select the desired representations.

### 2.2. Routing algorithms

Some routing algorithms seek similar representations for parent capsules by clustering, implemented with iterative processes (Sabour et al., 2017; Hinton et al., 2018; Rajasegaran et al., 2019). The Dynamic Routing (Sabour et al., 2017) could be regarded as a soft $k$-Means algorithm (Malmgren, 2019). In (Hinton et al., 2018), a routing algorithm was presented, implemented by the Expectation-Maximum (EM) algorithm and the Gaussian mixture model (Reynolds, 2009), which are commonly used in clustering. However, the iterative processes incurred high computing complexity. To address this, a single-layer perceptron (Hahn et al., 2019), attention scores computed by the child capsules (Choi et al., 2019; Xinyi & Chen, 2018), global variational inference (Ribeiro et al., 2020b), and Gumbel-Softmax (Ahmed & Torresani, 2019) were proposed to substitute the routing algorithms with iterative processes. DeepCaps (Rajasegaran et al., 2019) performed only the iterative processes in the last capsule layer, thus reducing the computing complexity. An inverted dot-product based attention routing (Tsai et al., 2020) substituted the iterative processes with a new parallel operation, which accelerated the inference. But, the parallel operation actually provided a trade-off between the memory space and inference time, and the high computing complexity still remained. In general, these approaches did not take into account both the clustering mechanism and reduction of computing complexity of the routing algorithms.

## 3. Receptors in a Capsule

We propose a new approach for clustering representations with low computing complexity. In previous work, the representations of a capsule are determined by routing algorithms, and the capsule is just a representation container. Given the representations of $A$ child capsules for the $j$-th parent capsule $\hat{\mathbf{u}}_{j|i}$ (obtained by Eq. (1), $i \in \{1, 2, \ldots, A\}$), routing algorithms progressively estimate the representation centroids iteratively. These routing algorithms ignore some irrelevant representations and focus on some fixation patterns (Sabour et al., 2017). However, the process of human vision does not work in this way. In a scene, people make some preliminary judgments according to certain cognitive prioris (Spelke, 1990), so that human can respond to a scene subconsciously and quickly. Such prioris, learned from some related experiences, are activated by a few representative details in the scene.

Following this view, we design a new capsule structure with a set of optimizable receptors. Also, we define a transmitter on a representation of the capsule, which is on behalf of the representation. In data processing, a receptor picks the relevant representations for the parent capsule by matching with the transmitters of the child capsules' representations, just like a priori (receptor) being activated by some representative information (transmitter).

### 3.1. Receptors and transmitters

The receptors of a capsule are defined as $\{v_m \in \mathbb{R}^C\}$ ($m \in \{1, 2, \ldots, M\}$), where $C$ is the capsule size and $M$ is the number of receptors per capsule. The receptors can be directly optimized by back-propagation without any extra losses. Notably, each receptor belongs to a capsule, which is not a shared element like routing algorithms.

A transmitter is defined on a capsule representation. For a representation of the $i$-th matrix child capsule for $j$-th parent capsule, $\hat{\mathbf{u}}_{j|i} \in \mathbb{R}^{C\times W\times H}$ (where $W$ and $H$ are the width and height of the feature maps, respectively), we define a lower-dimensional transmitter $\mathbf{s}_{j|i} \in \mathbb{R}^C$ of $\hat{\mathbf{u}}_{j|i}$ using a squeezing function S, as:

$$\mathbf{s}_{j|i} = \mathrm{S}(\hat{\mathbf{u}}_{j|i}) \tag{2}$$

such that the squeezing function S reduces the dimensions (e.g., the $W$ dimension and $H$ dimension) of $\hat{\mathbf{u}}_{j|i}$. In practice, the squeezing function is implemented by the global average pooling, adopting the idea from (Hu et al., 2018). The transmitter $s_{j|i}$ can be viewed as a simplified version of $\hat{\mathbf{u}}_{j|i}$, providing core information of $\hat{\mathbf{u}}_{j|i}$. Note that the transmitters should have the identical size as the receptors, for ease of similarity computing. We do not use high-dimensional tensors (like $\mathbb{R}^{C\times W\times H}$) to define the receptors and transmitters, since this would incur high computation complexity.

### 3.2. How does a receptor match to transmitters?

To obtain the representation of the $j$-th parent capsule, we compute the similarities between the parent capsule's receptor (here we assume $M = 1$) and all the child capsules' transmitters $\mathbf{s}_{j|i}$ ($i \in \{1, 2, \ldots, A\}$). Formally, the simi-

**Algorithm 1** A procedure for synthesizing the $j$-th parent capsule's representation by our proposed approach.

```
1: Input: The representations of $A$ child capsules, $\mathbf{u}_i$ ($i \in \{1, 2, \ldots, A\}$); the receptors of the $j$-th parent capsule, $\{v_m\}$ ($m \in \{1, 2, \ldots, M\}$); the number $K$ of the nearest neighbors (see Sec. 3.2); the coefficient vector $\mathbf{r}$ (see Eq. (5)).
2: Output: The $j$-th parent capsule's representation $\mathbf{u}_j$.
3: Transform $\mathbf{u}_i$ to $\hat{\mathbf{u}}_{j|i}$ by function $W_{ij}$.   ▷ by Eq. (1)
4: Compute the transmitters $\mathbf{s}_{j|i}$ of $\hat{\mathbf{u}}_{j|i}$.   ▷ by Eq. (2)
5: for $m = 1$ to $M$ do
6:     Compute the similarities between the transmitters $\mathbf{s}_{j|i}$ and the receptors $v_m$.   ▷ by Eq. (3)
7:     Rank the similarities and combine the $K$ nearest neighboring representations to be $\hat{\mathbf{u}}_{j;m}$.   ▷ by Eq. (4)
8: end for
9: Combine $\hat{\mathbf{u}}_{j;m}$ ($m \in \{1, 2, \ldots, M\}$) with $\mathbf{r}$ to obtain the $j$-th parent capsule's representation $\hat{\mathbf{u}}_j$.   ▷ by Eq. (5)
```

larity $d_{j|i;m}$ between the receptor $v_m$ and a representation transmitter $s_{j|i}$ is defined as:

$$d_{j|i;m} = \text{sigmoid}(s_{j|i}^T \cdot \frac{v_m}{||v_m||_2}) \in (0, 1) \tag{3}$$

where $d_{j|i;m}$ is a scalar. We use a sigmoid function to limit the bounds of $d_{j|i;m}$ to avoid extreme values. The receptor $v_m$ is normalized by its $l_2$ norm to avoid the influence of the vector length.

After obtaining the similarities $d_{j|i;m}$ ($i \in \{1, 2, \ldots, A\}$), we pick the $K$ nearest neighboring representations from the $A$ child capsules, $\{\hat{\mathbf{u}}_{j|k;m}\}$ ($k \in \{1, 2, \ldots, K\}$), using the $K$ nearest neighbor grouping ($K$-NN grouping). Then, the $K$ picked representations are combined to form a higher-level representation, with similarity-based coefficients, by:

$$\hat{\mathbf{u}}_{j;m} = \sum_{k=1}^{K} w_{j|k;m} \hat{\mathbf{u}}_{j|k;m} \tag{4}$$

where the coefficients are computed as $\mathbf{w} = \text{softmax}(\mathbf{d})$, in which $\mathbf{w} = [w_{j|1;m}, w_{j|2;m}, \ldots, w_{j|K;m}]$, and $\mathbf{d} = [d_{j|1;m}, d_{j|2;m}, \ldots, d_{j|K;m}]$. Thus, those $K$ representations whose transmitters are similar to the receptor $v_m$ are combined into a higher-level representation $\hat{\mathbf{u}}_{j;m}$.

### 3.3. How do receptors of a capsule collaborate?

A capsule can represent complex semantics (e.g., compound semantics) with multiple neurons (Sutskever et al., 2011; Sabour et al., 2017). Typically, a representation $\hat{\mathbf{u}}_{j;m}$ provided by a receptor (see Eq. (4)) can represent only one single semantic. We define a set of receptors in a capsule, making it available to learn compound semantics. Given $\hat{\mathbf{u}}_{j;m}$ ($m \in \{1, 2, \ldots, M\}$) obtained by $M$ receptors in the $j$-th capsule, we define the $j$-th parent capsule's representation as a convex combination of $\{\hat{\mathbf{u}}_{j;m}\}$. Besides, we add the averages of representation votes (by a shortcut connection) to smooth the output representation to stabilize the training process. Formally, the representation of the $j$-th parent capsule is defined by:

$$\mathbf{u}_j = \sum_{m=1}^{M} \gamma_{j;m} \hat{\mathbf{u}}_{j;m} + \frac{1}{A} \sum_{i=1}^{A} \hat{\mathbf{u}}_{j|i} \tag{5}$$

where $\gamma_{j;m}$ is an element in a vector $\mathbf{r} = [\gamma_{j;1}, \gamma_{j;2}, \ldots, \gamma_{j;M}]$, $\mathbf{r} = \text{softmax}(\mathbf{z})$, and $\mathbf{z}$ is an optimizable vector of the size $|\mathbf{r}|$. Then, the representation $\mathbf{u}_j$ of the $j$-th parent capsule is further passed forward, and is possibly combined into its parent capsule's representation in the next layer.

The whole procedure for synthesizing a capsule's representation by our approach is given in Alg. 1. For ease of understanding, we show the operations of the receptors using the loop in Lines 5–8 of Alg. 1. In practice, the receptors can perform in parallel. In our approach, the capsules pick the low-level representations via the receptors, avoiding using routing algorithms. Actually, our approach plays the role of the routing algorithms.

## 4. Receptor Skeleton

Previous CapsNets worked in a hierarchical agglomerative clustering process to learn the part-whole relationship in recognizing objects. In our approach, we add a constraint on the receptors to ensure that a receptor is associated with those receptors in the adjacent capsule layers, presenting the hierarchical cluster centroids for the global part-to-whole assembling. In (Wan et al., 2020), the averages of the low-level representations were used as the higher-level representations, attaining good hierarchical interpretations for the objects. Inspired by this, we require the receptors (representing cluster centroids) $v_m^{(p)}$ of a parent capsule be the averages of some receptors $\{v_i^{(c)}\}$ of the child capsules, by:

$$v_m^{(p)} = \frac{\sum_{i=1}^{N_m} v_i^{(c)}}{N_m} \tag{6}$$

where $v_i^{(c)}$ ($i \in \{1, 2, \ldots, N_m\}$) denote the child capsules' receptors connecting to the parent capsule's receptor $v_m^{(p)}$ (we call the relationship between two receptors a "connection"), and $N_m$ is the number of $\{v_i^{(c)}\}$. An example of receptor relationships is shown in Fig. 2. In this way, the semantics learned by $v_i^{(c)}$ are possibly part of the compound semantics learned by $v_m^{(p)}$. In implementation, we can initialize some meta-receptors and synthesize receptors for all capsules, sharing the underlying data storage. Below we will introduce our skeleton topology design.

### 4.1. Skeleton topology

For building the skeleton to specify the receptor relations in adjacent capsule layers, we assume that the following three reasonable conditions hold for the skeleton of receptors, making the receptors be treated uniformly and individually and avoiding introducing biases. The three conditions respectively ensure that all the receptors (i) are considered in the skeleton, (ii) are equally treated in the view of the child and parent capsules, and (iii) represent specifically different semantics. These three conditions are as follows.

**Condition 1.** *Each parent-child capsule pair should have at least one connection (by receptors).*

*Remark.* In two adjacent capsule layers, each high-level capsule possibly takes in the representations from any low-level capsules. This means that every high-level capsule is potentially the parent capsule of the low-level capsules. Thus, there should be at least one receptor in each high-level capsule connecting to a receptor of each low-level capsule.

**Condition 2.** *All the parent-child capsule pairs should have identical connection amounts, and all the receptors in the same capsule layer are connected with identical amounts.*

*Remark.* To treat all the capsules and receptors equally, we require that the connections be distributed uniformly. This means that we do not give any priori to a CapsNet, and all the properties of the CapsNet (e.g., the importance of certain capsules) are adaptive to the dataset.

**Condition 3.** *No two receptors should have identical connections.*

*Remark.* It is more efficient to make every receptor unique, facilitating the CapsNet to learn more varied semantics.

Suppose there are $B$ parent capsules and each parent capsule contains $M$ receptors. A naive solution is: each receptor in every child capsule connects to a different receptor of the parent capsules; thus the receptor number inside a child capsule is equal to the total receptor number of the parent capsules ($B \times M$). Furthermore, if there are $A$ child capsules, then there should be $A \times B \times M$ receptors inside each grandchild capsule. This means that the number of receptors in a CapsNet increases exponentially with its layers, which will cause high computing complexity. A better solution is to keep the receptor number per capsule a constant, and let each receptor of the child capsules connect to more than one receptor in the parent capsules (if possible).

To satisfy the three conditions above and keep the receptor number $M$ per capsule a constant, we propose an intuitive solution inspired by the Latin square design (Colbourn & Dinitz, 2006; Hedayat et al., 2012). Let the number $A$ of the low-level capsules, the number $M$ of receptors per capsule, and the number $B$ of the high-level capsules satisfy:

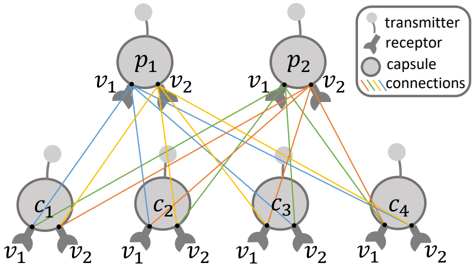


*Figure 2.* An example of connection topology (see Eq. (8)) in a two-capsule-layer module. The first receptor $v_1$ of the first parent capsule $p_1$, as specified by Eq. (6), is the average of some child capsule receptors, by $v_1^{(p_1)} = (v_1^{(c_1)}+v_1^{(c_2)}+v_2^{(c_3)}+v_2^{(c_4)})/4$.

$$B \times M = A \tag{7}$$

Note that the constraint in Eq. (7) is not a strong one, since the number of the higher-level capsules should be less than the lower-level capsules, in a clustering view.

We define the skeleton connection topology using an $A \times A$ square matrix $\mathbf{\Gamma}$. In the matrix $\mathbf{\Gamma}$, let the rows index the child capsules in a fixed permutation, and the columns index the receptors in the parent capsules (since $B \times M = A$) by a sequence $[v_1^{(p_1)}, v_1^{(p_2)}, \ldots, v_1^{(p_B)}, v_2^{(p_1)}, v_2^{(p_2)}, \ldots, v_2^{(p_B)}, \ldots, v_M^{(p_1)}, v_M^{(p_2)}, \ldots, v_M^{(p_B)}]$, where $v_m^{(p_j)}$ denotes the $m$-th receptor in the $j$-th parent capsule. An entry $\mathbf{\Gamma}[r, c]$ indicates which receptor of the $r$-th child capsule connects to the $c$-th receptor in the receptor sequence of the parent capsule layer ($r \in \{1, 2, \ldots, A\}$, $c \in \{1, 2, \ldots, B \times M\}$). Formally, we connect the parent capsule receptor $v_m^{(p_j)}$ with the ($\mathbf{\Gamma}[i, Bm - B + j] + 1$)-th receptor of the $i$-th child capsule.

To obtain $\mathbf{\Gamma}$, we adopt the idea of the Latin square design. A Latin square of order $A$ is an $A \times A$ matrix containing numbers in $\{0, 1, \ldots, A-1\}$ such that each number occurs exactly once in each row and each column (Hedayat et al., 2012). The Latin square design uses a Latin square to uniformly arrange the levels of the factor variables exactly once in various experimental settings (indicated by rows and columns). The uniformity and uniqueness properties of the Latin square meet our three conditions. Thus, we construct the skeleton connection topology based on a Latin square.

In practice, we construct a standard $A \times A$ Latin square $\mathbf{L}$, by taking the entry $\mathbf{L}[r, c] = (r + c - 2) \ \mathbf{mod} \ A$ (Hedayat et al., 2012). Given a Latin square $\mathbf{L}$ of size $A \times A$, we obtain $\mathbf{\Gamma}$ as $\mathbf{\Gamma}[r, c] = \lfloor \mathbf{L}[r, c]/B \rfloor \in \{0, 1, \ldots, M-1\}$, where $\lfloor \cdot \rfloor$ is the floor rounding operator. Here we give an example with $A = 4$ and $M = 2$ to illustrate the procedure of building a skeleton topology, in Eq. (8) and Fig. 2.

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{pmatrix}, \quad \mathbf{\Gamma} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

**Lemma 1**. $\Gamma$ thus obtained satisfies the three conditions aforementioned.

*Proof.* In a Latin square $\mathbf{L}$, every element in $\{0, 1, \ldots, A-1\}$ appears exactly once in each row and each column of $\mathbf{L}$. Thus, every element in $\{0, 1, \ldots, M-1\}$ appears exactly $B$ times in each row and each column of $\mathbf{\Gamma}$ due to Eq. (7). Hence condition 1 and condition 2 are satisfied.

For condition 3, by the above construction procedure of $\mathbf{L}$ and $\mathbf{\Gamma}$, one can easily find that $\mathbf{\Gamma}$ is a Hankel matrix (a row-reversed Toeplitz matrix) in which each ascending skew-diagonal from left to right is constant. Based on the above property of the Hankel matrix, without loss of generality, we only need to show that the first column of $\mathbf{\Gamma}$ is different from the other columns.

In the first row of $\mathbf{L}$, $\mathbf{L}[1,t] = t-1$ for $1 \leq t \leq A$. In its last row, $\mathbf{L}[A,1] = A-1$ and $\mathbf{L}[A,t] = t-2$ for $2 \leq t \leq A$. By the relationship between $\mathbf{L}$ and $\mathbf{\Gamma}$, we have $\mathbf{\Gamma}[1,1] = 0 \neq t-1 = \mathbf{\Gamma}[1,t]$ for $A-B+1 \leq t \leq A$. Also, we have $\mathbf{\Gamma}[A,1] = M-1 > \mathbf{\Gamma}[A,t]$ for $2 \leq t \leq A-B+1$, since $\mathbf{\Gamma}[A,t] = M-1$ if and only if $A-B+2 \leq t \leq A$ or $t = 1$. Hence, the first column of $\mathbf{\Gamma}$ is different from the other columns and condition 3 is satisfied. □

## 5. Architecture

We build four capsule layers on the top of the backbones (the ResNet-18 backbone (He et al., 2016), the simple backbone (Tsai et al., 2020)), with our receptor skeleton. We use a non-linear function for the layer normalization, as in (Tsai et al., 2020). We take the Cross-Entropy loss as in (Tsai et al., 2020) as our loss function.

## 6. Experiments and Results

### 6.1. Basic setups

We use the PyTorch library to implement our approach. We let $K = 6$ for the $K$ nearest neighbor grouping (see Sec. 3.2), and the number of receptors per capsule $M = 2$. The numbers of capsules in the four capsule layers are 40, 20, 10, and the class number of the corresponding dataset, respectively. In the training process, we use the SGD optimizer with weight decay $5^{-4}$ and momentum $0.9$ (Sutskever et al., 2013). We run 300 epochs, and the initial learning rate is 0.1, which is reduced by 10× at the 60-th, 120-th, and 160-th epochs. The models are trained and tested on GeForce RTX 3090 Ti GPUs. For fair comparison, we unify the backbones, number of capsule layers, capsule number, capsule sizes, etc. The performances of the known approaches are obtained by running their open source codes.

### 6.2. Performance comparison

**(1) Classification performances**

We evaluate the classification performances of CapsNet with our approach on several non-performance-saturated datasets, including AffNIST, Fashion-MNIST, SmallNorb, SVHN, Multi-MNIST (following (Sabour et al., 2017)), CIFAR-10, and CIFAR-100. We generate the Multi-MNIST dataset similarly as in (Sabour et al., 2017), and the only difference is that each digit is shifted up to 2 pixels in each direction, which results in an image of size $32 \times 32$. With larger occluded areas than those in (Sabour et al., 2017), the classification task becomes more difficult. The Multi-MNIST dataset was used to evaluate the ability in processing overlapped objects. The images in the AffNIST dataset are MNIST images transformed by 32 random affine operators. To verify the affine transformation robustness, we train all the CapsNets only on the MNIST training set (Notably, the training set of AffNIST is not used), and evaluate their generalization abilities on the AffNIST test set. Similar to (Sabour et al., 2017; Ribeiro et al., 2020b;a), the MNIST images for training are randomly placed in a $40 \times 40$ black background. For the SmallNorb dataset, we follow the experimental design as in (Ribeiro et al., 2020b) to evaluate the viewpoint-invariance. For the CIFAR-10 and CIFAR-100 datasets, we use the following data augmentations as in (Tsai et al., 2020): (1) pad four zero-value pixels to the input images, (2) randomly crop the images to size $32 \times 32$, and (3) horizontally flip images with a probability of $0.5$. For the other datasets, the official train-test splits are adopted, and no data augmentation is applied. To keep fairness, we use the same configurations in training and inference for all the models (CapsNets) on the same datasets.

We unify the simple backbone using a two-layer convolutional neural network as in (Tsai et al., 2020). As for the ResNet backbones, we use the first two blocks of ResNet-18 (He et al., 2016). The capsule size $C$ is 36 for CIFAR-100, and 16 for the other datasets. Since our approach is an alternative of routing algorithms, we compare our approach with the state-of-the-art routing algorithms, including Dynamic Routing (Sabour et al., 2017), EM Routing (Hinton et al., 2018), 3D Routing (Rajasegaran et al., 2019), and inverted dot-product attention routing (IDPA-Routing) (Tsai et al., 2020).

As shown in Table 1, our approach considerably outperforms the routing algorithms with simple backbones on the datasets (except for the SmallNorb dataset) **by clear margins (1.1%–3.9%)**. On the SmallNorb, our method obtain a slightly lower performance than 3D-Routing and EM Routing, but still comparable. Our approach attains good perfor-

*Table 1.* Classification performance comparison among various routing algorithms and our approach **with simple backbones**. The best performances are marked in **bold**, and the second-best performances are underlined.

| Methods | AffNIST | Fashion-MNIST | SmallNorb | SVHN | Multi-MNIST | CIFAR-10 | CIFAR-100 |
|---|---|---|---|---|---|---|---|
| 3D-Routing | 87.8% | 91.4% | 97.5% | 90.5% | 94.8% | 82.9% | 24.2% |
| Dynamic Routing | 81.2% | 92.3% | 97.1% | 90.9% | 95.3% | 84.5% | 57.4% |
| EM Routing | 93.8% | 90.3% | **98.0%** | 90.2% | 94.8% | 82.2% | 34.3% |
| IDPA-Routing | 86.1% | 92.0% | 96.3% | 90.6% | 95.6% | 85.0% | 57.1% |
| Our approach | **96.7%** | **93.4%** | 97.2% | **94.8%** | **96.9%** | **87.4%** | **60.5%** |

*Table 2.* Classification performance comparison among ResNet-18, the other routing algorithms, and our approach **on the ResNet-18 backbones.** The best performances are marked in **bold**.

| Methods | CIFAR-10 | CIFAR-100 |
|---|---|---|
| ResNet-18 | 95.1% | 77.9% |
| 3D-Routing | 91.2% | - |
| Dynamic Routing | 90.4% | - |
| EM Routing | 86.2% | - |
| IDPA-Routing | 93.5% | 72.3% |
| Our approach | **93.7%** | **72.8%** |

*Table 3.* Overlapped object segmentation results (Dice scores, %) on the JSRT dataset. "LL" and "RL" denote the left and right lungs, "LC" and "RC" denote the left and right clavicles, and "H" denotes the heart, respectively. The prefix "V" indicates using vector capsules, and "M" indicates using matrix capsules.

| Methods | Dice (%) | | | | | |
|---|---|---|---|---|---|---|
| | LL | RL | LC | RC | H | Mean |
| M-Dual | 96.8 | 97.3 | 88.0 | 87.2 | 94.1 | 92.68 |
| M-(Ours) | 97.4 | 98.0 | 89.8 | 88.3 | 95.1 | **93.72** |
| V-Dual | 96.5 | 97.1 | 86.3 | 86.2 | 93.3 | 91.88 |
| V-Dynamic | 95.4 | 96.3 | 82.7 | 82.3 | 92.3 | 89.80 |
| V-(Ours) | 97.1 | 97.7 | 88.2 | 87.6 | 94.0 | **92.92** |

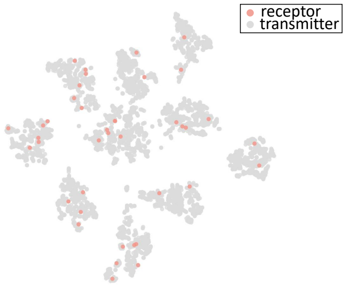


*Figure 3.* T-SNE embeddings of the receptors (orange) and transmitters (grey) in the penultimate layer. One can see that the distributions of the transmitters and receptors are overlapped.

mances on Multi-MNIST, which indicates that our approach can separate overlapped objects, possibly due to the part-to-whole learning scheme. Besides, one can observe our performance gains compared with the previous CapsNets on the AffNIST test set. Recall that we train the CapsNets only using the original MNIST images without affine transformation, this experiment verifies that our proposed approach can better handle affine transformation than the known approaches. Also, with the ResNet-18 backbone, our approach outperforms the known previous work, especially the current best-performed routing algorithm IDPA-Routing, as shown in Table 2. These results on the complex datasets show that our approach has good potential for image classification.

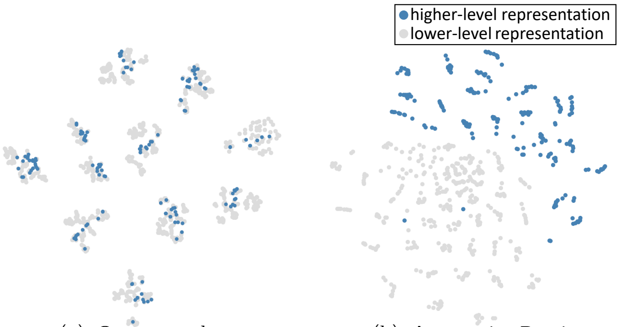
**(2) Segmentation performances**

CapsNets with routing algorithms can pick similar representations and thus are effective in segmenting highly overlapped objects (Bonheur et al., 2019; Sabour et al., 2017). Radiograph is a type of medical images in which tissues and organs can largely overlap. Hence, we follow the model in (Bonheur et al., 2019), replacing Dynamic Routing and Dual Routing (Bonheur et al., 2019) by our approach and embedding two receptor skeletons in the U-shape model (one in the encoder and the other in the decoder). We compare our approach with those with the matrix capsule version (Hinton et al., 2018) and vector capsule version (Sabour et al., 2017) on the Japanese Society of Radiological Technology (JSRT) dataset (Shiraishi et al., 2000). The JSRT dataset contains 247 chest radiographs, and segmentation annotations of the left and right lungs, the left and right clavicles, and the hearts are provided. The Dice scores of Dual Routing and Dynamic Routing are from (Bonheur et al., 2019). As shown in Table 3, **our approach outperforms the two routing algorithms by 1.04%–3.12%**. Since our approach performs well on Multi-MNIST classification (as shown in Table 1) and overlapped object segmentation, it suggests that our approach excels at overlapped object processing, which is possibly due to our design (the receptors, transmitters, and receptor skeleton) for clustering.

### 6.3. How well does our approach work?

**(1) How do the receptors and transmitters perform?**

Here we inspect the performances and operation mechanism with the MNIST dataset. Transmitters are on behalf of representations; receptors are on behalf of representation centroids. The receptors pick representations for the parent



(a) Our approach (b) An attention Routing

*Figure 4.* T-SNE embedding illustration of the representations in two adjacent layers. The blue points indicate the parent capsules' representations and the grey points indicate the child capsules' representations. (a) The T-SNE embedding of the representations learned with our approach; (b) the T-SNE embedding of the representations learned with an attention routing without the skeleton.

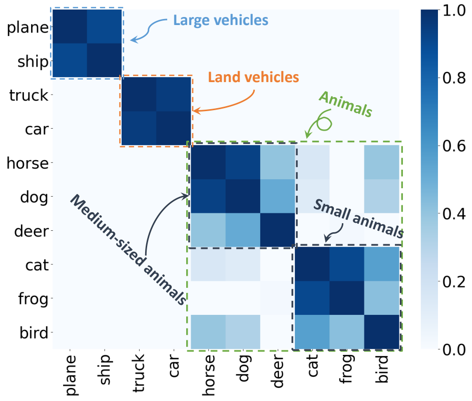


*Figure 5.* A heatmap illustrating the correlation of 10 classes on the CIFAR-10 dataset based on the capsule activation frequencies.

capsules by matching with the transmitters. As shown in Fig. 3, the transmitters are twinkled around the receptors, verifying that the receptors act as representation centroids. Also, the distributions of the receptors and transmitters are highly overlapped, suggesting that the receptors work collectively, seeking varied semantics for the parent capsules.

**(2) Is the receptor skeleton helpful?**
In Fig. 4 (a), the t-SNE embedding shows that the representations are clustered hierarchically by our approach. With 10 digits (on the MNIST dataset), the representations in adjacent layers are coarsely partitioned into 10 clusters, and the parent capsules' representations "generalize" the major patterns of the child capsules' representations. This is because the model objective (classification) transmits to each capsule via the receptor skeleton. Without such a skeleton, an attention routing does not obtain a similar embedding (see Fig. 4(b)). Thus, the receptor skeleton can make a CapsNet work as an agglomerative clustering process.

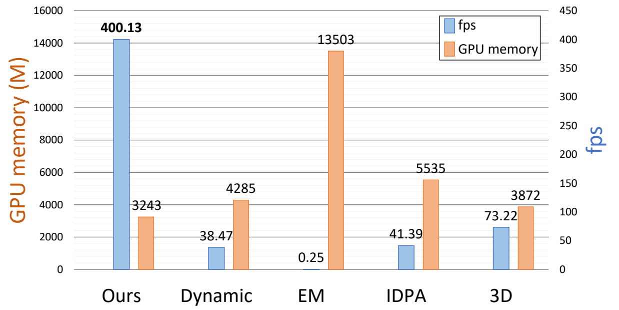


*Figure 6.* Bar graphs for illustrating computing complexities. One can see that our approach saves 5.5× — 1600× inference time and uses only 22% — 83% GPU memory.

**(3) How does our approach affect capsules?**
To further explore how our approach guides the capsules to learn semantics, we compute the capsules' activation frequencies (in the penultimate layer) for each class on the CIFAR-10 dataset, and then compute the correlations among object classes based on the capsule activation frequencies. As shown in Fig. 5, the semantics are captured by the capsules with our approach; further, the more semantically similar objects (e.g., cars and trunks) activate the same capsules more largely, while semantically irrelevant objects hardly activate the same capsules.

## 6.4. Computing complexity

To fairly evaluate the computing complexity of the routing approaches, we compare the inference time (frames per second (fps)) and the GPU memory usage (M) instead of the parameter count on GeForce RTX 3090 Ti GPUs, as some routing algorithms are "parameterless" but compute many intermediate variables during iterations. We focus on the computing costs of the procedure for processing the representation $\hat{\mathbf{u}}_{j|i}$ to obtain $\mathbf{u}_j$ (which is what the routing algorithms and our approach are for). The results are shown in Fig. 6. Clearly, our approach takes less GPU memory and much shorter inference time in dealing with input and output tensors (in computing GPU memory, batch size = 64, capsule number = 40, capsule size = 16, width = 8, and height = 8; in computing fps, batch size = 1). Comparing with the routing algorithms, our approach outperforms them **by 5.5× (comparing with 3D routing) to 1600× (comparing with EM routing) on inference time, with only 22% (comparing with EM routing) to 83% (comparing with 3D routing) GPU memory.** The reason for our reduction on computing complexity is that we avoid using iterative processes, and thus accelerate the procedure and avoid storing many intermediate variables.

## 6.5. Reconstruction and dimension perturbation

To inspect the semantics of the individual dimensions represented by the class capsules, we employ the decoder network

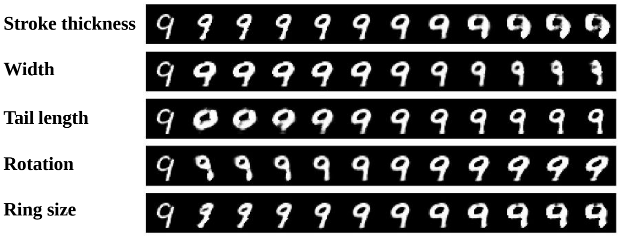


*Figure 7.* Illustrating dimension perturbations. Each row presents our reconstruction results when one of the class capsule dimensions is tweaked (by intervals of 0.04 in the range [-0.2, 0.2]). The first digit in each row is the original digit.

in (Sabour et al., 2017) and tweak the values in the class capsule dimensions by intervals of 0.04 in the range from -0.2 to 0.2. Some reconstruction results with dimension perturbations are illustrated in Fig. 7, which shows that our approach preserves the reconstruction information of digits, and every dimension in the class capsules represents steady and explainable semantics, including the stroke thickness, width, tail length, rotation, ring size, as summarized in Fig. 7. It is also evident that our approach performs well in reconstruction.

### 6.6. Impacts of various combinations of $K$ and $M$

We evaluate the classification performance of our approach with the simple backbone on the CIFAR-10 dataset, with various combinations of $K = 2, 4, 6, 8, 10$ for the $K$-NN grouping (see Sec. 3.2) and the receptor number per capsule $M = 1, 2, 3, 6$. Since the receptor number $M$ is under the constraint of Eq. (7), we build a CapsNet with $M = 6$ as baseline, and obtain another CapsNet version with $M = 3$ by combining every two receptors inside each capsule into one, by computing the average in the combination. Similarly, we obtain CapsNets with $M = 1, 2$. For $M = 1$, we discard some connections randomly to make the connections in different capsules vary.

As shown in Table 4, as the $K$ value increases, the classification performance of our approach on the CIFAR-10 dataset first increases and then decreases. For example, with $M = 2$, the best classification performance is attained at $K = 6$. This may be because clusters of medium sizes work better in representation clustering. For the receptor number $M$ per capsule, $M = 2$ yields better performance than $M = 1$, with various $K$ values. When $M > 2$ increasing, the classification performances fluctuate slightly. The reason for this may be that a capsule does not combine many semantics well (e.g., 6 types of semantics when $M = 6$) in one step (in a capsule layer), and a better solution may be to combine few representations (e.g., with $M = 2$) in a step and capture the complex compound semantics progressively. In our experiments, we use $M = 2$ and $K = 6$ to keep a balance between the performance and the model size.

*Table 4.* Classification performances of our approach with the simple backbone on the CIFAR-10 dataset with various combinations of $K = 2, 4, 6, 8, 10$ for the $K$-NN grouping and the receptor number per capsule $M = 1, 2, 3, 6$. **Bold** entries denote the best performances of the columns (the best $M$ with a certain $K$); underlined entries denote the best performances of the rows (the best $K$ with a certain $M$).

| M | K | | | | |
|---|---|---|---|---|---|
| | 2 | 4 | 6 | 8 | 10 |
| 1 | 85.6% | 86.1% | 86.8% | 86.5% | 85.1% |
| 2 | 86.6% | 87.0% | 87.4% | 86.8% | 85.2% |
| 3 | **86.9%** | 87.2% | **87.5%** | **86.9%** | 85.2% |
| 6 | 86.8% | **87.5%** | 87.2% | 86.3% | **85.3%** |

## 7. Conclusions

In this paper, we proposed a new approach to replace routing algorithms in CapsNets. We introduced a new capsule structure with a set of receptors and devised a transmitter on a representation loaded in each capsule. By matching the receptors and transmitters, the child capsules' representations are clustered without iterative processes. To hierarchically cluster the representations, we designed a receptor skeleton to organize the receptors. Under three conditions in view of uniformity and uniqueness, we developed an intuitive solution by adopting the idea of Latin square design. Experiments on multiple datasets provided strong evidences for the superiority of our new approach for image classification, overlapped object recognition, affine transformation robustness, and representation clustering, with lower model complexity.

## Acknowledgements

This research was partially supported by the National Research and Development Program of China under grant No. 2019YFC0118802, No. 2019YFB1404802, and No. 2018AAA0102102, the National Natural Science Foundation of China under grant No. 61672453, the Zhejiang University Education Foundation under grants No. K18-511120-004, No. K17-511120-017, and No. K17-518051-02, the Zhejiang public welfare technology research project under grant No. LGF20F020013, the Medical and Health Research Project of Zhejiang Province of China (No. 2019KY667), the Wenzhou Bureau of Science and Technology of China (No. Y2020082), and the Key Laboratory of Medical Neurobiology of Zhejiang Province. D. Z. Chen's research was supported in part by NSF Grant CCF-1617735.

## References

Ahmed, K. and Torresani, L. STAR-Caps: Capsule networks with straight-through attentive routing. In *NeurIPS*, 2019.

Bonheur, S., Štern, D., Payer, C., Pienn, M., Olschewski, H., and Urschler, M. Matwo-CapsNet: A multi-label semantic segmentation capsules network. In *MICCAI*, 2019.

Choi, J., Seo, H., Im, S., and Kang, M. Attention routing between capsules. In *CVPR Workshops*, 2019.

Colbourn, C. J. and Dinitz, J. H. *Handbook of Combinatorial Designs*. CRC press, 2006.

Hahn, T., Pyeon, M., and Kim, G. Self-routing capsule networks. In *NeurIPS*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hedayat, A. S., Sloane, N. J. A., and Stufken, J. *Orthogonal Arrays: Theory and Applications*. Springer Science & Business Media, 2012.

Hinton, G. E., Sabour, S., and Frosst, N. Matrix capsules with EM routing. In *ICLR*, 2018.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *CVPR*, 2018.

Kosiorek, A. R., Sabour, S., Teh, Y. W., and Hinton, G. E. Stacked capsule autoencoders. In *NeurIPS*, 2019.

LaLonde, R. and Bagci, U. Capsules for object segmentation. In *MIDL*, 2018.

Malmgren, C. A comparative study of routing methods in capsule networks. Master's thesis, Linkping University, 2019.

Rajasegaran, J., Jayasundara, V., Jayasekara, S., Jayasekara, H., Seneviratne, S., and Rodrigo, R. DeepCaps: Going deeper with capsule networks. In *CVPR*, 2019.

Reynolds, D. A. Gaussian mixture models. *Encyclopedia of Biometrics*, 2009.

Ribeiro, F. D. S., Leontidis, G., and Kollias, S. Capsule routing via variational bayes. In *AAAI*, 2020a.

Ribeiro, F. D. S., Leontidis, G., and Kollias, S. Introducing routing uncertainty in capsule networks. In *NeurIPS*, 2020b.

Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. In *NeurIPS*, 2017.

Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, et al. Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American Journal of Roentgenology*, 2000.

Spelke, E. S. Principles of object perception. *Cognitive Science*, 1990.

Srivastava, N., Goh, H., and Salakhutdinov, R. Geometric capsule autoencoders for 3D point clouds. *arXiv preprint arXiv:1912.03310*, 2019.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, 2011.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

Tsai, Y.-H. H., Srivastava, N., Goh, H., and Salakhutdinov, R. Capsules with inverted dot-product attention routing. In *ICLR*, 2020.

Wan, A., Dunlap, L., Ho, D., Yin, J., Lee, S., Jin, H., Petryk, S., Bargal, S. A., and Gonzalez, J. E. NBDT: Neural-backed decision trees. In *ICLR*, 2020.

Xinyi, Z. and Chen, L. Capsule graph neural network. In *ICLR*, 2018.

Zhao, Y., Birdal, T., Deng, H., and Tombari, F. 3D point capsule networks. In *CVPR*, 2019.