SoftSkip: Empowering Multi-Modal Dynamic Pruning for Single-Stage Referring Comprehension

Dulanga Weerakoon
Singapore Management University
mweerakoon.2019@phdcs.smu.edu.sg

Vigneshwaran Subbaraju
IHPC, A*STAR, Singapore
vigneshwaran_subbaraju@ihpc.a-star.edu.sg

Tuan Tran
Singapore Management University
tuantran@smu.edu.sg

Archan Misra
Singapore Management University
archanm@smu.edu.sg

ABSTRACT

Supporting real-time referring expression comprehension (REC) on pervasive devices is an important capability for human-AI collaborative tasks. Model pruning techniques, applied to DNN models, can enable real-time execution even on resource-constrained devices. However, existing pruning strategies are designed principally for uni-modal applications, and suffer a significant loss of accuracy when applied to REC tasks that require fusion of textual and visual inputs. We thus present a multi-modal pruning model, LGMDP, which uses language as a pivot to dynamically and judiciously select the relevant computational blocks that need to be executed. LGMDP also introduces a new SoftSkip mechanism, whereby 'skipped' visual scales are not completely eliminated but approximated with minimal additional computation. Experimental evaluation, using 3 benchmark REC datasets and an embedded device implementation, shows that LGMDP can achieve *33%* latency savings, with an accuracy loss *0.5% - 2%*.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**; • **Computing methodologies → Computer vision**; **Natural language processing**; • **Human-centered computing** → *Human computer interaction (HCI).*

KEYWORDS

Human-Robot Interaction; Referring Expression Comprehension; Pruning; Computer Vision; Natural Language Processing

ACM Reference Format:
Dulanga Weerakoon, Vigneshwaran Subbaraju, Tuan Tran, and Archan Misra. 2022. SoftSkip: Empowering Multi-Modal Dynamic Pruning for Single-Stage Referring Comprehension. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548432

1 INTRODUCTION

Referring Expression Comprehension (REC) is a multi-modal task where the goal is to correctly identify objects or segments within an image that is being referred to by an accompanying verbal instruction. In this task, the input consists of (a) an image and (b) an accompanying natural language text that refers an object or objects within the art of the image, and the desired output is a set of co-ordinates or a bounding box that encompasses the referred part(s) of the image. REC is a foundational construct that underpins AI-driven capabilities such as Question-Answering [35] and Human Robot Interaction [24]. A variety of deep learning based approaches for REC, trained using large-scale datasets of crowd-sourced "image—description" pairs [13, 14, 20, 21, 33], have been proposed. Initial approaches adopt a three-stage process, viz., (Step 1) generate region proposals on the image, (Step 2) extract visual features from the proposed regions and textual features from the natural language text, and finally (Step 3) rank candidate proposed regions using a metric that reflects the match between each region's visual features and the instruction's textual features. Experimental studies showed that REC performance was crucially dependent on the effective modeling of visual context, at scales corresponding to either the entire image [13], individual objects [33] or at multiple levels [32]. To reduce the high computational complexity and latency associated with the execution of three distinct stages, more recent REC approaches [29, 30, 34] have adopted a single stage approach with multi-modal fusion. These approaches employ multi-modal attention mechanisms that fuse verbal and visual cues, and often provide better modeling of contextual information at both global and local scales.

Despite the reduced complexity, such single-stage neural models for REC are still resource intensive and ill-suited to support real-time, interactive applications on resource-constrained embedded platforms, such as wearables and IoT devices. To support such real-time pervasive REC execution, we thus explore the use of neural optimization techniques that bypass redundant/inconsequential computation blocks in such single-stage models. Broadly speaking, such neural optimization can broadly involve either (a) static pruning, which reduces the model size by eliminating neural nodes or layers during the offline training phase, or (b) dynamic pruning techniques, such as convolutional layer sparsification [2] or dynamic routing [27], which selectively eliminate some of the computations in the complex backbone network during the *inference phase.* While dynamic pruning approaches are more nimble as they customize the computation to each input sample, all extant methods have been

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548432

designed for uni-modal tasks (e.g., solely based on visual input). Our main contribution is to develop a novel, generalized dynamic pruning strategy for REC tasks, which are inherently *multi-modal* (consisting of both verbal and visual inputs).

Our proposed approach adopts the same basic principle associated with all run-time pruning methods: skip computational blocks on a per-input basis, thereby reducing latency and computational energy overheads. However, our runtime pruning strategy explicitly uses the REC-specific property that the textual input contains critical information in identifying important/relevant regions in the image, and consequently develops mechanisms that optimize the visual processing pipeline (and the subsequent stages, such as attentional modules, that fuse visual and verbal cues) based on features embedded in the textual input. To the best of our knowledge, our work is the first to propose a multi-modal pruning approach for REC tasks. More specifically, we hypothesize that computational blocks at certain visual scale can be safely skipped depending on the sizes of both the target object and the objects referred to in the verbal input. Accordingly, in our approach, we use the textual features as a pivot to determine the necessity or relative importance of computing features at certain image scales. This approach contrasts with traditional unimodal runtime skipping mechanisms, where computational blocks are skipped primarily based on background vs. foreground differentiation and without regard to the size or saliency of individual objects.

Determining these scales are, however, a challenging problem for REC tasks due to the possibility of multiple relevant visual scales and saliency, such as when the textual reference is made with respect to another anchor object–e.g.,"small clock on the table", where the table and clock require different scales. Accordingly, adopting the prior *binarized* approaches (where a specific computational block is either executed in its entirety or completely skipped) runs the risk of missing crucial contextual information. Based on empirical observations that corroborate this anticipated pitfall of binarized skipping, we thus introduce a novel *"soft-skipping"* strategy, where certain computational blocks determined to be suitable for skipping are approximated using an alternate (convolutional) pathway that consumes dramatically lower computational resources. We believe that this approach, called **SoftSkip**, represents a general and powerful design paradigm for optimizing neural computation for tasks, such as REC, that involve correlated multi-modal inputs and require processing at different visual scales. We design a modified single-stage REC model, called *LGMDP* (Language-Guided Multi-modal Dynamic Pruning), that incorporates the SoftSkip mechanism into multiple stages, such as visual feature extraction, adaptive feature selection and global attention computation, associated with the execution of REC tasks. Via extensive studies, we demonstrate how LGMDP provides significantly superior performance compared to standard static and dynamic pruning approaches, achieving lower latency while offering far higher comprehension accuracy (almost comparable to a non-optimized heavyweight baseline model).

Key Contributions: Our key contributions are as follows:

- We introduce a novel run-time DNN optimization approach called LGMDP that is useful for supporting multi-modal tasks such as REC. To the best of our knowledge, LGMDP is the first model that uses textual features as a pivot to skip computations in both the visual processing and the subsequent multi-modal fusion stages. In addition, LGMDP employs the novel concept of SoftSkip, where computational blocks are not completely eliminated but rapidly approximated, thereby ensuring that features at different visual scales are at least partially preserved.
- We implement LGMDP, as well as a variety of competitive state-of-the-art (SOTA) alternatives. Using three different benchmark datasets (ReferIt[14], RefCOCO [33] and Cops-Ref [7]), we show that LGMDP offers a far superior accuracy-vs.-latency tradeoff and is able to offer a significant reduction in computational latency with negligible loss in accuracy. In particular, LGMDP suffers only an ∼0.5% loss (65.3%→64.6%) in comprehension accuracy compared to the non-optimized RealGIN baseline, but achieves more than 33% reduction in processing latency when executed on a NVIDIA Jetson TX2 device. In addition, LGMDP's accuracy of 65.37% @ 220 ms of latency far outperforms the best performing SOTA uni-modal pruning alternative (only 58.31% accuracy at similar latency). In addition, in Section 4.2, we provide more granular insights into how LGMDP's multi-scale SoftSkip mechanism is able to leverage on appropriate textual cues.
- We also demonstrate how LGMDP's SoftSkip-based approach can be combined with standard static pruning approaches to support ultra-lightweight, real-time REC execution on the Jetson TX2, a representative embedded platform. While the combined model suffers an 18% loss in accuracy compared to the RealGIN baseline, it achieves a significant 2.75x reduction in latency and 7x reduction in memory overhead, which is superior to that achieved by static pruning alone (17% accuracy loss with 2.2x and 7x reduction in latency and memory, respectively).

Overall, we believe that LGMDP's paradigm of language-driven SoftSkip-based dynamic pruning represents a significant, foundational advance towards the goal of supporting accurate, real-time REC on embedded and pervasive devices.

2 RELATED

Although the referring expression comprehension problem has been studied for a while, deep learning based approaches gained ground after the release of the large-scale ReferIt dataset [14] generated using a crowd-sourced two-player game. To generate ReferIt samples, one of the players views an image and writes an expression that refers to the target object, while the other player using this expression to click on the relevant region in the image. This approach was later extended to include images in the MSCOCO [18] dataset, with the instructions then as RefCOCO and RefCOCO+ datasets [33]. Using the ReferIt dataset, the authors also studied the visuo-linguistic characteristics of referring expressions and showed how visual attributes in the image input correlated with words used in the verbal instruction (e.g.,'Big' is most commonly associated with larger target regions whereas 'small', tiny', little' etc are associated with smaller regions).

Initially, deep neural models for REC were based on the CNN-LSTM [20, 33] framework, where the LSTM takes a word vector at each time step and attempts to match it with CNN-based visual features extracted from a candidate region within the image. These models adopt a max-margin based training method for the LSTM such that the probability of referring expression is higher for the referred image region. Many of these early works showed that accurate modeling of the contextual information was critical for achieving effective target inference. For example, Yu et.al. [33] used visual differences between objects to represent the visual context, achieving higher comprehension accuracy than [20]. Subsequently, [13] used whole-image CNN features to represent the context, while [21] proposed the use of use multiple-instance learning for effective context modeling. With the advent of attention mechanisms, MAttNet [32] introduced a modular framework that further enhanced REC accuracy. In Mattnet, separate modules, each with their individual attention mechanisms, utilize features of object locations, context or relationships.

Such multi-stage pipelines, however, are computationally prohibitive due to the need for a separate region proposal network. To address this, recent *single-stage* neural approaches [23, 29, 30, 34] have replaced the region ranking with a multi-modal bounding box regression stage. Yang et al [30] proposed an approach based on YOLOv3 [22], where they obtained a visual feature pyramid via the Darknet-53 [22] backbone network, and the language features for textual referring expression via BERT [8]. The visual feature pyramid is then concatenated with the verbal features at each level, and subsequently combined with a normalized spatial feature, to execute the bounding box regression. The authors recently extended their work further [29] by including a sub-query learning and modulation framework that decomposes long textual descriptions into shorter sub-queries; they demonstrated that recursive use of such visual features improved the ability to resolve ambiguity associated with longer verbal instructions. The Zero-Shot Grounding (ZSG) method [23] extracted image features by combining a ResNet (instead of Darknet) backbone with a feature pyramid network (FPN), while using a Bi-LSTM to extract the language representation. More recently, Zhou et al. [34] further extended ZSG to develop the RealGIN model. RealGIN includes a separate Adaptive Feature Selection (AFS) method that uses textual information to identify REC-relevant visual features and a multi-modal global attention mechanism named GARAN, and achieved 30fps processing throughput (a 10-fold increase over MattNet) for REC tasks.

A broader body of research has tackled the problem of pruning DNN models, either statically or during run-time [2, 3, 16, 26, 31], to support efficient, low-latency DNN execution on pervasive devices. Efficient networks such as MobileNet & ShuffleNet embody the principle of static model compression, where redundancy is identified and eliminated in the channels, weights and filters in a complex visual backbone such as the VGG16, ResNet etc. Approaches such as [16], SkipNet [27] and Dynamic Convolutions [26] are examples of run-time pruning, where unwanted computations are eliminated during inference time with minimal impact on overall accuracy. Reinforcement learning is typically used to train the modules that decide on such run-time, input-dependent skipping of specific computational blocks. All extant approaches, however, tackle only unimodal neural models; refactoring them to our multi-modal REC task is non-trivial due to the need to preserve relevant contextual and relational information across the different modes.

2.1 Baseline Model: RealGIN

We use the RealGIN model [34] as the representative, state-of-the-art single-stage model for real-time, multi-modal REC tasks. As shown in Figure 1, RealGIN employs a ResNet [12] based backbone for extracting features from the image and a bi-directional LSTM [10] for extracting language features from verbal instruction. RealGIN then uses a novel Adaptive Feature Selection (AFS) module that identifies image features that are relevant to the text instruction. This is followed by a new multi-modal attention mechanism (GARAN) to facilitate language-guided visual attention. As shown in [34], RealGIN achieves a 10x improvement in throughput, while achieving accuracy very close to that of multi-stage approaches such as MAttNet [32].

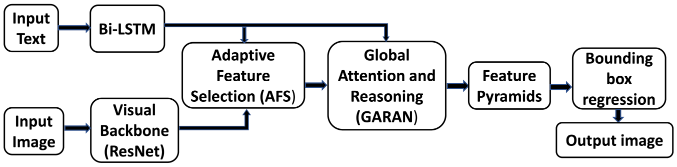

Figure 1: RealGIN single-stage REC model

3 LGMDP

Figure 2 presents the architecture of our proposed LGMDP model for single-stage, dynamically optimized REC. The model includes (a) a novel skippable visual backbone which enables input-specific, alternate efficient pathways for visual feature extraction, (b) a bi-directional GRU for textual feature extraction, (c) a module to compute multiple scale-specific skip factor, (d) a modified and skippable multi-scale adaptive feature selection (borrowed from RealGIN), (e) a global attention module (borrowed from RealGIN), (f) a skippable feature pyramid network and finally (g) bounding box regression layers (similar to YoLo3). The parameters of all these modules are trained using an end-to-end supervised learning process.

Our run-time SoftSkip approach (shown in Fig 3) is based on the following set of principles:

(a) The information embedded in the textual input should be used to determine the relevant scales in the visual backbone. This choice of verbal→visual dependence is modeled on known models of human comprehension [4, 15], which show that humans utilize verbal cues to adjust visual attention (and not vice versa) and driven by the high complexity of the visual feature extraction.

(b) Since it is difficult to precisely determine whether the skipped layers of the visual backbone contain pertinent contextual information, we do not completely eliminate a computational block but instead use an alternative computationally-lightweight computational pipeline to approximate the features derived by the block. The approximated features continue to serve as useful input for the subsequent processing blocks.

(c) Processing blocks in the subsequent AFS and FPN stages, which fuse verbal and visual features, utilize the same set of scale-specific skipping parameters used by the visual backbone. In other words, a single set of skipping parameters are used across multiple processing modules, implying that if a particular NxN scale is skipped (or approximated) in the visual backbone, the corresponding NxN scale is also approximated in the AFS and FPN modules. Using a common set of 3 universally-applied skipping parameters also avoids unnecessary additional computation, as compared to an alternative approach where different stages of the model are associated with different sets of skipping parameters.

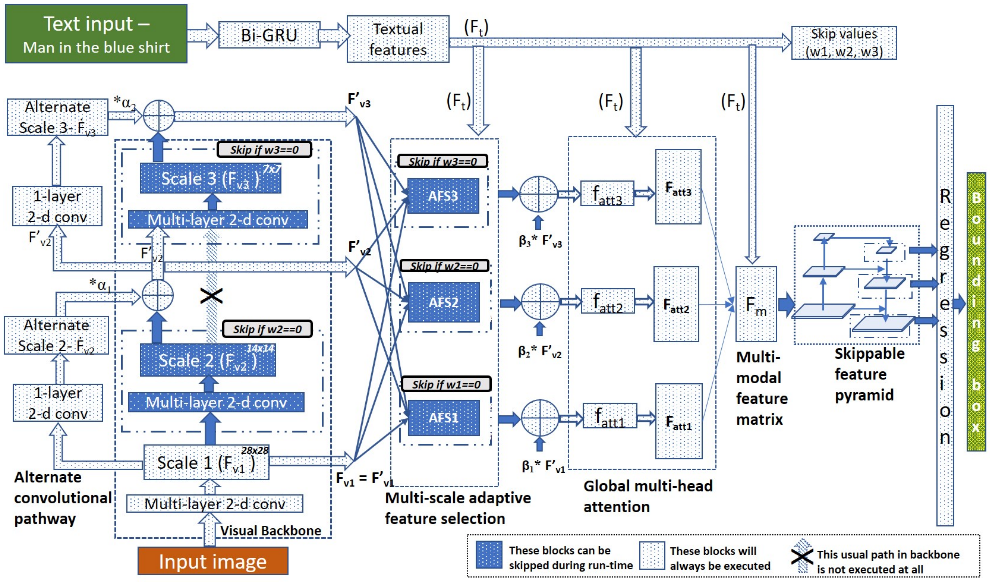

Figure 2: LGMDP - Model architecture

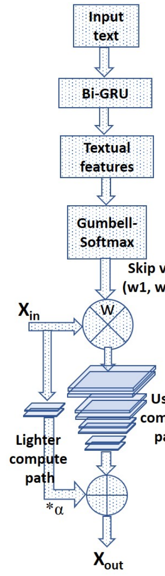

Figure 3: Soft-Skipping

In this paper, we compute 3 distinct skipping parameters, which are applied to conditionally skip the corresponding last three stages of the visual pipeline. The choice of 3 parameters, corresponding to scales of 28x28, 14x14 and 7x7, respectively, are chosen as they intuitively correspond to small, medium and large-sized objects.

In the training mode, all the blocks are still computed and multiplied by the skipping parameter. This way we make sure the model weights are differentiable for backward propagation. Formally, let C be the computational block to be skipped, x be the input to the computational block, w be the skipping parameter, $C^{'}$ be the low complexity approximation to the computational block, α be a trainable scaling parameter and x_{out} be the output. Then, forward computation in training mode is defined as follows.

$$x_{out} = w * C(x) + \alpha * C^{'}(x);\ where\ \ w \in 0,1,\ \ \alpha \in [0,1] \qquad (1)$$

During run-time, in order to achieve true latency savings we propose a different forward propagation where the decision to execute a computational block or not is based on the value of the relevant skipping parameter. When the skipping parameter is 0, relevant computational block will not be executed returning latency savings. In general, let C be the computational block to be skipped, x be the input to the computational block, w be the pruning weight, $C^{'}$ be the low complexity approximation to the computational block, α be a trainable scaling parameter and x_{out} be the output. Then, forward computation in inference mode is defined as follows.

$$x_{out} = \begin{cases} C(x) + C^{'}(x) \ \ if w = 1 \\ \alpha * C^{'}(x) \ \ if w = 0 \end{cases} \qquad (2)$$

3.1 Language-based scale-specific skipping parameters

We use the Gumbell-Softmax on the language embedding generated by the GRU network to compute 3 discrete values that serve as common skipping parameters across different stages of the overall neural model. The language features (f_t) are captured by a 256-dimensional vector. This embedding vector serves as an input to a fully-connected layer with 3 output neurons. We then use the Gumbell-Softmax activation mechanism to obtain the 3 corresponding discrete binary values as follows.

$$w_1, w_2, w_3 = G(F(f_t));\ where\ \ w_1, w_2, w_3 \in \{0,1\} \qquad (3)$$

In the above equation, G is the Gumbell-Softmax activation function and F is the Fully-connected layer function. A value of '0' for a skipping parameter implies that the relevant module will not be executed (or, more precisely, will only be approximately executed), while a value of '1' implies normal execution of the module. In our current design, for a given input referring expression, only one of these parameters are assigned a value of '1', while the others are set to '0'. This discrete value of either '1' or '0' is taken through the output of Gumbell-Softmax activation function which directly returns a one-hot encoded 3-dim vector. Each dimension of this vector refers to a skipping parameter.

3.2 Skippable visual backbone

We evaluated two visual backbones for LGMDP, viz. (i) ResNet [34] and (ii) ShuffleNet [19]. Our baseline RealGIN model also uses the same ResNet visual backbone. A Resnet-based backbone is, however, not amenable to execution on low-resource embedded devices, such as the Jetson Nano. Therefore, we also consider lower complexity ShuffleNet as the backbone, to support real-time, on-device execution despite a loss in accuracy. ResNet has a residual structure and several earlier studies have proposed ways of skipping convolutions, using strategies such as early exit [25] and gated skipping [27], to reduce latency. However, such hard skipping strategies simply terminate computation of visual features at lower scales and thus cannot extract the multi-scale contextual information required for effective REC. In contrast, as shown in Figure 3, we adopted the SoftSkip approach: in LGMDP, whenever a certain block of convolutions at a certain scale is identified as suitable for 'skipping' in the visual backbone, a single-layer convolution is always used as an alternate pathway to provide approximate information at this scale. Accordingly, if F_{v1}, F_{v2}, F_{v3} denote the 3 visual feature scales in the backbone, we compute $\dot{F}_{v1}$, $\dot{F}_{v2}$ and $\dot{F}_{v3}$ which are ideally low complexity alternatives for F_{v1}, F_{v2} and F_{v3}. Then, we use w_1, w_2 and w_3 for skipping F_{v1}, F_{v2} and F_{v3} as follows.

$$F'_{vi} = w_(i) * L_b(F_{vi}) + \alpha_i * Conv2D(F_{v(i-1)}),\ for\ i \in 1, 2, 3 \quad (4)$$

In the above equation, L_b represents the usual convolutional blocks in the backbone which computes the respective feature scales. $Conv2D$ is a single convolutional layer with a stride of 2 and kernel size of 1. In the event that the w_i is 0, corresponding feature scale will be approximated with this single convolutional layer.

3.3 Skippable Adaptive Feature Selection

Outputs from different scales of the visual backbone are used as input to AFS modules (described in [34]) operating at the corresponding scales. The function of the AFS module is to take the visual feature maps from different scales and project them to the same resolution and depth using multiple convolutional layers. The visual feature maps from different scales represent visual information at different semantic levels. Separately, the language priors for these semantic levels are learnt from the textual feature vector (f_t) resulting in three separate fusion weights (v_1, v_2 and v_3) corresponding to these feature scales. These language priors are found to be useful to represent the wide variations in the content of textual expressions. For example, when an expression contains semantic information such as color/texture, AFS can increase the fusion weights for low/mid-level features. After that, a final visual feature map is computed as the weighted sum of these individual feature maps.

In LGMDP, we use multi-scale AFS wherein three AFS modules are involved each producing output at three different scales. We then use the skipping parameters w_1, w_2 and w_3 calculated earlier to determine whether the AFS modules at a certain scale should be subjected to soft-skipping or not. By sticking to our principle of soft-skipping, whenever an AFS module is subjected to soft-skipping, its output is always approximated by a proportion of the corresponding visual feature map obtained from the skippable visual backbone (as described earlier). Formally,

$$x_i = w_i * AFS_i(F'_{v1}, F'_{v2}, F'_{v3}, F_t) + \beta_i * F'_{vi} \quad (5)$$

When w_i is 0, respective AFS stage output will be 0 and relevant AFS activated features will be approximated with $\beta_i * F'_{vi}$.

3.4 Global Attention Module

In this module, the AFS-derived feature maps are used as input to an attention mechanism (mimicking that used in [34]). This module, called GARAN, uses the textual features to collect expression-related information over the whole image and then selectively diffuse this information to all anchors (predefined (location, size) templates for objects). As described in [34], the differential attention maps can be obtained by using the ground-truth bounding box of the target object, during training, as a supervision signal to calculate the attention loss.

Even though scale-specific skipping behavior can conceptually also be incorporated in the GARAN, we empirically found that such skipping resulted in a significant drop in accuracy with only minimal latency benefits. As the GARAN module is relatively computationally lightweight, there is not much benefit in implementing the skipping behavior at this stage. The output of the GARAN module is the multi-modal feature matrix F_m, which is then utilized to perform a YoLo-style bounding box regression.

3.5 Skippable Feature Pyramids and bounding box regression

The multi-modal feature matrix obtained from the GARAN module is used by a feature pyramid network [17] to perform bounding box regression. The feature pyramid network (FPN) is generally used to overcome the problem of limited receptive field exhibited by convolutional layers. FPN usually has two pathways: (1) *Bottom-up pathway*, which ideally is the feed-forward computation of convolutional layers with a scaling factor of 2, and (2) *Top-down pathway*, which up-samples feature vectors that are spatially coarser but semantically stronger, using a scaling step of 2. We apply our soft-skipping approach only to the top-down pathway in a similar fashion as described in the visual backbone. The FPN outputs feature maps at multiple scales that would be used for regression. Our intuition is that the language may have a hint on the size of the object of interest which may in turn help us to skip the irrelevant feature scales.

4 RESULTS

We evaluate LGMDP’s performance for REC tasks using three benchmark datasets.

1. **ReferIt or RefCLEF** [14] - This dataset contains about 19997 images selected from the ImageCLEF competition [11], with the objects in them referred through a two-player game, where the players alternate between generating and comprehending referring expressions. Apart from collecting this pioneering set of data of <*image,expression*> pairs, the authors also provided an analysis of the visuo-linguistic characteristics observed, such as the relationship between target bounding-box area versus the key-words used in the expressions etc. Thus, we used this dataset first to evaluate multiple SoftSkip strategies and empirically identify the preferred LGMDP model.
2. **RefCOCO** [33] - This dataset was also collected using the same game paradigm as ReferIt dataset, but the stimulus images are selected from the MSCOCO [18] dataset which is often used for training deep learning based object detectors. In this, there are 142,209 expressions referring at 50,000 objects in 19,994 images.
3. **COPS-Ref** [7] - This is a recently released dataset containing 148,712 expressions, that collectively refer to 1,307,885 regions on 75,299 images, making it the current largest real-world image dataset for referring expressions. The <*image,expression*> pairs in this dataset are considered challenging due to the presence of “distractor" objects which are similar to the target objects. Thus, this dataset serves to analyze the deeper reasoning abilities, such as logic and relational inference, of various REC techniques.

To evaluate the inference latency (time taken to process a single <*image,expression*> pair) of LGMDP and other competing models, we implement and deploy these models on an *NVIDIA Jetson TX2* [1], a representative embedded device. TX2 comprises of a 256-core GPU,a dual-core NVIDIA processor and 8GB system memory.

4.1 Different SoftSkip & HardSkip Strategies

	Val	Test	Lat(ms)
LGMDP-skip1	68.21	65.17	290
LGMDP-skip2	67.90	64.98	255
LGMDP-skip3	63.16	60.18	200
LGMDP	67.19	64.56	220
LGMDP-Hardskip	57.54	53.11	218

Table 1: Comparison of LGMDP variants on referit dataset

We first start by evaluating, on the ReferIt dataset, four distinct variants of our proposed SoftSkip strategy vs. a candidate Hardskip alternative, to help establish the preferred LGMDP alternative and its absolute performance. We evaluated a few variants of SoftSkip behavior, resulting in 5 different LGMDP variants, as follows:

(a) *LGMDP-skip1:* In this variant, only the largest scale (scale 3=7x7) is amenable to dynamic runtime soft-skipping, with SoftSkip enabled across all of the backbone, AFS and FPN stages; the other scales are always computed in their entirety.
(b) *LGMDP-skip2:* In this variant, SoftSkip is enabled for two scales (7x7 & 14x14), across all of the backbone, AFS and FPN stages.
(c) *LGMDP-skip3:* Here, SoftSkip is enabled across all three scales (7x7, 14x14 and 28x28), and throughout the entire DNN including the backbone, AFS and FPN stages.
(d) *LGMDP:* In this preferred model (which diverges only minutely from LGMDP-skip3), SoftSkip is enabled for all 3 scales in the AFS and FPN stages, but it is applied only to the two larger scales (7x7 and 14x14) on the visual backbone.
(e) *LGMDP-HardSkip:* This variant is identical to LGMDP, except that it replaces SoftSkip with a hard skipping strategy, where the computation for DNN states identified for skipping is eliminated entirely (instead of computing an approximate set of features).

Table 1 compares their relative performance. As expected, the average accuracy values degrade slightly as more scales are progressively possible candidates for soft-skipping; conversely, the overall execution latency decreases as well. We observe that option (d) (where skipping scale 1 in the visual backbone is prohibited) performs significantly better (suffering an accuracy drop of $< 1\%$) than LGMDP-skip3 (which suffers an accuracy drop of $\sim 5\%$), while exhibiting comparable latency. Accordingly, we pick and use (d) as our preferred LGMDP embodiment for all subsequent experiments.

We also note that LGMDP-Hard suffers a significant ($> 12\%$) drop in accuracy, validating our belief that completely eliminating feature extraction at certain scales is inadvisable for REC tasks where language-guided visual reasoning often occurs at multiple scales. We hypothesize that this performance loss could be due to two different factors: (a) *Vanishing Gradient Problem:* During the learning phase, if the skipping weight equals 0, the relevant feature scale results in a null matrix, which results in a zero gradient that in turn affects the efficacy of backpropagation; (b) *Over-reliance on language features:* Hard-skipping implicitly assumes that the verbal instruction provides sufficient cues for determining the appropriate visual scales needed. This is, of course, not universally true; in cases where the verbal cues are imprecise, hard skipping effectively obliterates features at certain scales, making the eventual recognition of target objects extremely difficult. SoftSkip, in contrast, is more permissive of situations where the language pruner makes mistakes.

4.2 Qualitative Insights on Multi-Scale SoftSkip

Before returning to using macroscopic metrics, such as comprehension accuracy, to compare LGMDP against non-dynamic baselines, we analyze the corpus of instructions in the ReferIt dataset to reveal deeper insights into the functioning of our SoftSkip strategy.

Figure 4 provides some typical examples of the referring expressions and the associated images in the ReferIt corpus. The top row contains examples of images where the referred object is generally large in size. The first two images in the middle row contain references to small target objects, while third image contains a reference to a prominent target. The last row of images are examples of where the verbal expression employs color attributes. We can observe that the textual references in these examples often provide indicative hints about the most salient visual scale. Small objects may need a scale of 28x28 (scale 1), big objects may need a scale of 7x7 (scale 3), references to low-level image properties like color/texture may benefit from scale 1, while the presence of relative positional references (left/right/foreground/background) may suggest the need for multiple scales. A fairly detailed analysis of the visuo-linguistic characteristics of the ReferIt dataset was provided by the authors in [14]. We used this to examine the semantic relevance of the SoftSkip behavior exhibited by LGMDP. The individual bars in Figure 4 provide the percentage of expressions where LGMDP chose a certain scale for full execution without skipping (with the corresponding skipping parameter $w_i == 1$), for instructions that contained specific keywords such as 'big', 'little', 'foreground', 'background' etc. In CNN-based object detection, spatial resolution diminishes rapidly as computation proceeds to the deeper layers; accordingly, deeper layers are likely to be less useful for capturing features of smaller objects. Therefore, one would expect scale 1 to be active and scale 3 skipped ($w_1 = 1, w_3 = 0,$) more often for scenarios involving detection of small objects. Similarly, one would expect scale 3 to be active ($w_3 = 1$) mainly for target images with larger bounding box area and expressions needing visual context corresponding to larger object sizes.

Some key points we observed are given below,

- As reported in [14], in the ReferIt dataset, the bounding box area of the target objects was found to be larger for expressions involving the keyword 'big', whereas the keywords 'tiny, little and short' were often associated with target objects with smaller bounding boxes. Correspondingly, in our skipping mechanism, the scale 1 (28x28) is used without skipping ($w_1 = 1$) for 43% of the instructions containing 'tiny, little and short', whereas the scale 3 (7x7) is used without skipping ($w_3 = 1$) for 26% of such instructions. When we looked at instructions containing the word 'big', the proportion of the instructions for $w1 = 1$ dropped to 38% while the proportion for scale 3 ($w_3 = 1$) increased to 32% of the instructions. The results broadly confirm our hypothesis that larger target objects would preferentially activate coarse scales (scale 3) more often. However, note that additional contextual factors, beyond just the target's bounding box size, can affect the choice of scale, and in fact, require *multiple* scales. For example, consider the *ReferIt* expression: "the door of the big building".

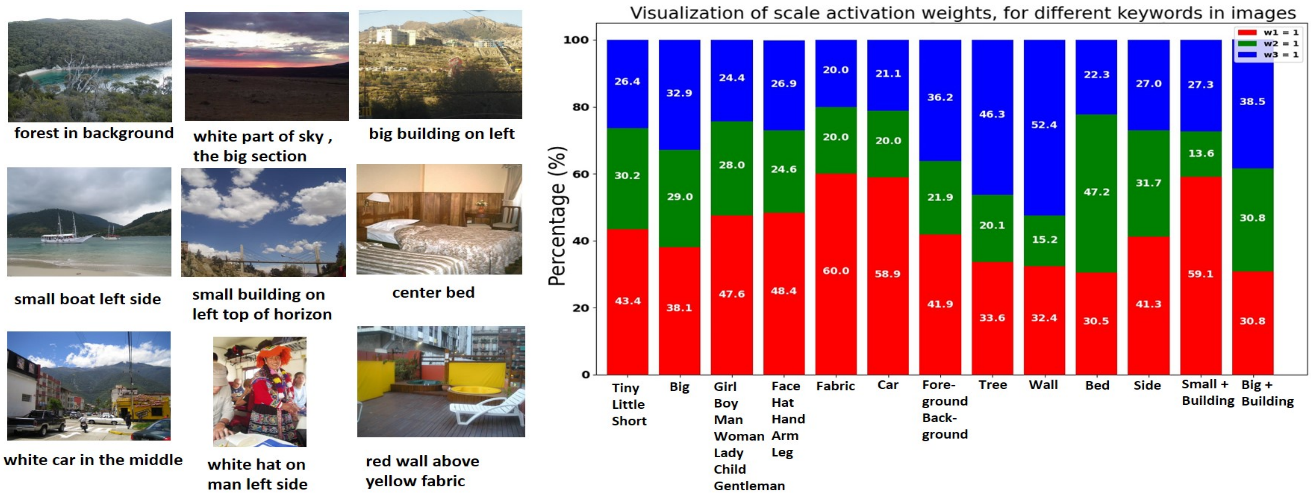

Figure 4: Examples of size, position and color based references in the ReferIt dataset and a visualization of scale activation weights, for different keywords in images

Here, while the target object is a 'small' door, comprehension requires identification of the 'big building' first.

- We then explored to filter out cases where the references to the same object are made but with different size adjectives. We chose 'Building' as our target object and selected the expressions containing the word 'Building'. Within these expressions, we looked at references to 'big' building and 'small' building separately. We found that in cases involving 'small' buildings, scale 1 was used for 59% of the expressions. In contrast, for expressions that involved 'big' building, scale 1 was used only about 30% of the time, while the use of scales 2 and 3 increased dramatically.
- In this dataset, color adjectives were used very often to refer to relatively smaller target objects, such as 'car' and 'fabric'. Accordingly, we observed that scale 1 was activated in LGMDP ($w_1 = 1$) for a vast majority of instructions involving these objects.
- Relative references in referring expressions may involve a mix of words such as positional keywords or objects. In ReferIt, the words 'foreground', 'background', 'side (left side, right side)', 'guy', 'man', 'woman', 'tree' and 'wall' were among the most frequently used to describe relative location of the target object. We observed a mix of scales being activated for these keywords. First, we looked at expressions that involved explicit references to foreground/background and side keywords. From a visual analysis in Figure 4, we could not determine an obvious 'preferred' scale associated with these words. Next, we looked at the expressions involving relative position with respect to the objects ('tree' and 'wall'). We observed that Scale 3 ($w3 = 1$) was fully activated for a large proportion of such expressions. However, for references to human objects (e.g., girl, boy, woman, man, child, lady, gentleman) or constituent body parts (e.g., face, hand, arm and leg), Scale 1 was activated more often. We note that human objects and/or their body parts are often easily described via the use of low-level adjectives (e.g., color/texture for skin or clothing).
- The authors in [14] had reported that the target object 'bed' was very often referred by 'absolute location' ("center bed", "bed on the left" etc.). In LGMDP, we find that medium scale 2 is invoked for a large proportion of these expressions.

These observations intuitively justify the scale-specific skipping strategy adopted in LGMDP, where a necessary scale is activated according to the semantic level of the visual context needed. Given that a single visual scale may not be directly inferred from verbal instructions or may not be appropriate for many images, LGMDP's SoftSkip approach helps cushion the effect of erroneous computation of skip weights.

4.3 Comparison of LGMDP with RealGIN

We now compare the performance of LGMDP with our chosen baseline - RealGIN. From Table 2, we observe that LGMDP's performance is very close to that of the baseline RealGIN approach. The drop in accuracy is a very meager <1% for ReferIt and RefCOCO (testB) datasets. The overall accuracy of RealGIN as well as LGMDP is lower for the more challenging COPS-Ref dataset, with LGMDP suffering a more discernible (~1.3%) relative loss in comprehension accuracy. Such modest loss in accuracy is, however, balanced by the significant (~33%) reduction in latency achieved by our dynamic pruning approach. The latency of LGMDP (220 msec, 225msec and 218msec for the ReferIt, COPS-Ref and RefCOCO dataset respectively) is about 105-110msec lower than the baseline.

4.3.1 Pervasive versions: Static Pruning vs. Dynamic SoftSkip. The original RealGIN uses ResNet-152 as its visual backbone. However, ResNet-152 is computationally intensive and not suitable for pervasive applications–e.g., the original RealGIN model cannot be executed on the Jetson Nano, a more resource-constrained pervasive device. As a form of static pruning, RealGin's ResNet-152 visual backbone can be replaced Shufflenet [19], a lightweight DNN model specially curated for pervasive applications with low latency and memory requirements. Thus, we experimented both with (a) a static pruning model, *RealGIN-staticPr* (where Shufflenet is used as part of the RealGIN backbone), and as well as (b) *LGMDP-staticPr* (an exemplar of dynamic soft skipping, where our SoftSkip paradigm is applied on top of RealGIN-staticPr). Applying static pruning does result, as expected, in a significant (18%) loss of accuracy, while enabling a 2x faster execution on the Jetson Nano. In comparison,

Method	Backbone	ReferIt			COPS-Ref		RefCOCO			
		Val (%)	Test (%)	Lat(ms)	Test (%)	Lat(ms)	Val (%)	Test A (%)	Test B (%)	Lat(ms)
RealGIN	ResNet-152	68.29	65.37	330	48.17	330	75.10	76.71	68.22	330
LGMDP	ResNet-152	67.19	64.56	220	46.81	225	73.27	75.19	67.23	218
RealGIN-SkipNet	ResNet-152	61.49	58.31	250	38.66	250	69.12	70.23	72.10	250
RealGIN-staticPr	Shufflenet	51.21	48.29	150	45.98	150	60.45	62.29	64.97	150
LGMDP-staticPr	Shufflenet	49.90	47.11	120	44.12	124	58.10	61.11	63.27	120

Table 2: Performance comparison of LGMDP against RealGIN, Static Pruning and Dynamic Pruning

LGMDP suffers only a marginal 0.8% degradation in performance while enabling a 33% further reduction of latency (from 330 msec to 220 msec). Similar results also hold for the execution on our representative Jetson TX2 (see Table 2), demonstrating the general applicability of our proposed SoftSkip approach.

4.4 Dynamic Pruning: SkipNets vs LGMDP

In contrast to existing dynamic pruning approaches designed for uni-modal neural models, LGMDP is designed for a multi-modal REC task. To evaluate the benefit of our multi-modal dynamic pruning strategy against the existing uni-modal pruning approaches, we take Skipnet [27] as an alternative baseline. Skipnet uses a gated network architecture where individual layers are dynamically skipped (or not), based on gating parameters computing using the Gumbell Softmax activation function applied on existing residual blocks in the convolutional backbone. For comparative assessment, we implement *RealGIN-SkipNet*, a model where we replace RealGIN's visual backbone with the Skipnet architecture. As shown in Table 2, RealGIN-Skipnet suffers an accuracy loss of ∼7%, in contrast to a mere 0.8% accuracy loss for LGMDP, while also incurring a ∼10% higher latency than LGMDP. This result conclusively establishes the superiority of our proposed SoftSkip-based dynamic pruning for multi-modal REC tasks.

5 DISCUSSION

While we have demonstrated that the soft-skipping strategy in LGMDP has several benefits in saving computational resources in REC task alone, we believe that the same paradigm could be applicable in several related tasks such as referring expression generation, simultaneous REC + segmentation, natural language image editing and video based REC as well. Video-based REC would also bring in new challenges in the form of varying levels of visual context in different frames, even though the textual referring expression may remain unchanged. Therefore, there may be a need to constantly iterate between visual and textual information to achieve an accurate grounding. However, even in this case, we believe that our principle of "scale-specific soft-skipping" would still prove relevant in helping ascertain the visual scales most pertinent for accurate instruction comprehension.

An important recent trend in this field is to use additional, naturally-occurring non-verbal cues (e.g. pointing) as part of the referring expression [6, 28]. This opens up more opportunities and challenges–e.g., pointing gestures can provide crucial hints regarding the location of the target object on the image/video, even though pointing accuracy degrades with distance. Furthermore, verbal expressions could be very different when users are allowed to point at the target object (an aspect not explicitly considered in the game setup of [14]). However, we note that LGMDP can be easily extended to accommodate pointing gestures by including a pointing affinity field in the GARAN mechanism. We also believe that calculating pointing affinity maps would also involve deep neural networks operating at multiple scales, while factoring in the aforementioned distance dependence of pointing fidelity. Thus, we believe that scale-specific soft-skipping could still be relevant in such situations. Another important capability pursued is *zero-shot grounding*, where the textual expression contains words that have not been encountered during training. Limited recent work [23] has shown that it is still possible to achieve accurate comprehension. One would imagine that non-verbal cues like pointing and gaze would facilitate zero-shot grounding of referring expressions.

Further, recently transformer-based methods such as GPT3[9], DETR [5] have demonstrated superior performance in both traditional natural language processing and computer vision tasks. Hence, transformers may also be useful in REC tasks. However, these models are still computationally very complex and resource-intensive, and face the same challenges for pervasive deployment discussed earlier. While developing scale-specific skipping strategies for such transformer-based models is an interesting proposition, it is likely that this will require significant changes to the methods and design proposed in this paper.

6 CONCLUSION

We have demonstrated that it is possible to save computational resources for real-time execution of REC tasks using a novel *dynamic, multi-scale, soft-skipping mechanism.* Proposed LGMDP model utilizes the textual information to determine which scale of visual features is more crucial for comprehension, and then reduces (but does not eliminate) computational effort, across multiple DNN stages, at other visual scales. Using this strategy, we show, across 3 benchmark datasets, that LGMDP is able to (a) save about 33% of latency, with just an ∼0.5% loss in accuracy, and (b) substantially outperform other static and uni-modal dynamic pruning approaches.

7 ACKNOWLEDGMENTS

This work was supported partially by 1) National Research Foundation, Singapore under its NRF Investigatorship grant (NRF-NRFI05-2019-0007), 2) Ministry of Education, Singapore under its Academic Research Fund Tier-1 grant (19-C220-SMU-008) and 3) Agency for Science, Technology and Research (A*STAR), Singapore under its AME Programmatic Funding Scheme (Project #A18A2b0046). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore or the Ministry of Education, Singapore or the A*STAR, Singapore.

REFERENCES

[1] . Jetson TX2 Module. https://developer.nvidia.com/embedded/jetson-tx2. Accessed: 2022-04-11.

[2] Sourav Bhattacharya and Nicholas D Lane. 2016. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. 176–189.

[3] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. 2017. Adaptive neural networks for efficient inference. *arXiv preprint arXiv:1702.07811* (2017).

[4] Bastien Boutonnet and Gary Lupyan. 2015. Words jump-start vision: A label advantage in object recognition. *Journal of Neuroscience* 35, 25 (2015), 9329–9335.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 213–229.

[6] Yixin Chen, Qing Li, Deqian Kong, Yik Lun Kei, Song-Chun Zhu, Tao Gao, Yixin Zhu, and Siyuan Huang. 2021. Yourefit: Embodied reference understanding with language and gesture. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1385–1395.

[7] Zhenfang Chen, Peng Wang, Lin Ma, Kwan-Yee K Wong, and Qi Wu. 2020. Cops-ref: A new dataset and task on compositional referring expression comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10086–10095.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30, 4 (2020), 681–694.

[10] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional LSTM networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*. Springer, 799–804.

[11] Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. 2006. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International workshop ontoImage*, Vol. 2.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[13] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4555–4564.

[14] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 787–798.

[15] Peter Kok, Michel F Failing, and Floris P de Lange. 2014. Prior expectations evoke stimulus templates in the primary visual cortex. *Journal of cognitive neuroscience* 26, 7 (2014), 1546–1554.

[16] Ji Lin, Yongming Rao, Jiwen Lu, and Jie Zhou. 2017. Runtime neural pruning. In *Advances in neural information processing systems*. 2181–2191.

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[19] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*. 116–131.

[20] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 11–20.

[21] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. 2016. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*. Springer, 792–807.

[22] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[23] Arka Sadhu, Kan Chen, and Ram Nevatia. 2019. Zero-shot grounding of objects from natural language queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4694–4703.

[24] Mohit Shridhar and David Hsu. 2018. Interactive visual grounding of referring expressions for human-robot interaction. *arXiv preprint arXiv:1806.03831* (2018).

[25] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2464–2469.

[26] Thomas Verelst and Tinne Tuytelaars. 2020. Dynamic Convolutions: Exploiting Spatial Sparsity for Faster Inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2320–2329.

[27] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. 2018. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 409–424.

[28] Dulanga Weerakoon, Vigneshwaran Subbaraju, Nipuni Karumpulli, Tuan Tran, Qianli Xu, U-Xuan Tan, Joo Hwee Lim, and Archan Misra. 2020. Gesture enhanced comprehension of ambiguous human-to-robot instructions. In *Proceedings of the 2020 International Conference on Multimodal Interaction*. 251–259.

[29] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. 2020. Improving One-stage Visual Grounding by Recursive Sub-query Construction. In *ECCV*.

[30] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. 2019. A Fast and Accurate One-Stage Approach to Visual Grounding. In *ICCV*.

[31] Shuochao Yao, Yiran Zhao, Huajie Shao, ShengZhong Liu, Dongxin Liu, Lu Su, and Tarek Abdelzaher. 2018. Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. 278–291.

[32] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1307–1315.

[33] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *European Conference on Computer Vision*. Springer, 69–85.

[34] Yiyi Zhou, Rongrong Ji, Gen Luo, Xiaoshuai Sun, Jinsong Su, Xinghao Ding, Chia-Wen Lin, and Qi Tian. 2021. A real-time global inference network for one-stage referring expression comprehension. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[35] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4995–5004.