HEAV: HIERARCHICAL ENSEMBLING OF AUGMENTED VIEWS FOR IMAGE CAPTIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

A great deal of progress has been made in image captioning, driven by research into how to encode the image using pre-trained models. This includes visual encodings (e.g. image grid features or detected object) and more recently textual encodings (e.g. image tags or text descriptions of image regions). As more advanced encodings are available and incorporated, it is natural to ask: *how to **efficiently** and **effectively** leverage and ensemble the heterogeneous set of encodings?* In this paper, we propose to regard the encodings as augmented views of the input image. The model encodes each view independently with a shared encoder efficiently, and a contrastive loss is incorporated across the encoded views to improve the representation quality, as well as to enable semi-supervised training of image captioning. Our proposed hierarchical decoder then adaptively ensembles the encoded views according to their usefulness by first ensembling within each view at the token level, and then across views at the view level. We demonstrate significant performance improvements of +5.6% CIDEr on MS-COCO compared to state of the art under the same trained-from-scratch setting and +16.8% CIDEr on Flickr30K with semi-supervised training, and conduct rigorous analyses to demonstrate the importance of each part of our design.

1 INTRODUCTION

A large amount of progress has been made in vision-and-language (VL) tasks such as image captioning (Chen et al., 2015; Agrawal et al., 2019), visual question answering (Goyal et al., 2017; Hudson & Manning, 2019), and image-text retrieval. For these tasks, recent methods (Zhang et al., 2021a; Li et al., 2020; Shen et al., 2021; Kuo & Kira, 2022) observe that encoding the input image by an object detector (Ren et al., 2015) pre-trained on Visual Genome (Krishna et al., 2017) is not sufficient. To provide information complementary to detected objects, recent works proposed to encode an input image by different pre-trained models and into different modalities, and achieve substantial performance improvement. For example, some works encode from the visual perspective (*e.g.* stronger object detector pre-trained on a larger vocabulary and datasets (Zhang et al., 2021a) or global image features (Ji et al., 2021)), while other works encode from the textual perspective (*e.g.* image tags (Li et al., 2020) and text descriptions of image regions (Kuo & Kira, 2022)).

Given the great success of incorporating various heterogeneous encodings or "*views*", one research question emerges naturally: *how to **efficiently** and **effectively** leverage and ensemble these heterogeneous views?* For **efficiency**, three factors are particularly important: computation, parameter, and data efficiency. State-of-art VL models are typically a transformer-based model (Vaswani et al., 2017), which has undesirable quadratic computational complexity with respect to the input sequence size. Therefore, as more views are incorporated, each represented by a sequence of tokens, we should carefully manage the computation and model size. Moreover, on the medium-scale MS-COCO image captioning benchmark (Chen et al., 2015) (∼0.6M training samples), we should take data efficiency into consideration when training the data-hungry (Dosovitskiy et al., 2021) transformer model to avoid negative effects such as overfitting. For **effectiveness**, different views encode some shared and some complementary information of the input image. Therefore, it is important to leverage the views as much as possible, and at the same time adaptively weigh each view according to their usefulness. Take image captioning in Figure 1 as an example, when generating the word "*sofa*", if the view of detected objects fails to detect the sofa in the input image, the model should down-weigh the view of detected objects and rely more on other views that properly encode the information of sofa.

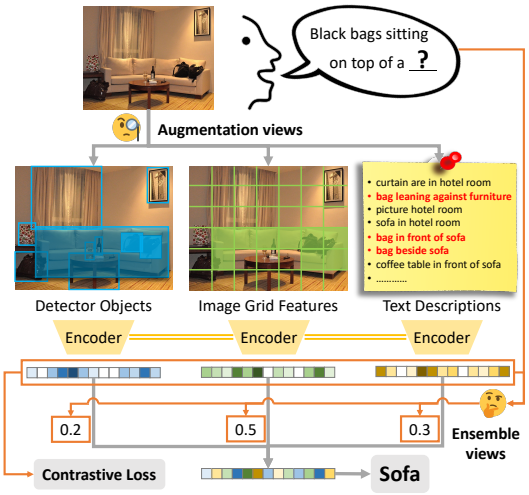

Figure 1: HEAV, **H**ierarchical **E**nsembling of **A**ugmented **V**iews, for image captioning at the step of generating the word "*sofa*". First, heterogeneous views such as detected objects (Anderson et al., 2018), image grid features (Shen et al., 2021), and text descriptions (Kuo & Kira, 2022) are generated from the input image. These views are regarded as the augmentations of the input image and are separately encoded by a shared transformer encoder. A contrastive loss is incorporated to help representation learning of heterogeneous views. Our proposed hierarchical decoder then ensembles the encoded views by adaptively weighing the views according to their usefulness. Finally, the next word "*sofa*" is predicted from the ensembled views.

With these considerations in mind, we propose HEAV, **H**ierarchical **E**nsembling of **A**ugmented **V**iews. In HEAV, we (1) regard heterogeneous views as augmentations of the input image, and (2) devise a hierarchical decoder layer to ensemble the heterogeneous views. For (1), by regarding views as augmentations, we naturally choose to use a *shared* transformer encoder to encode each view *independently*. Compared to methods that concatenate all views into a long sequence as input (Li et al., 2020; Zhang et al., 2021a; Kuo & Kira, 2022), where the computational complexity scales up quadratically with respect to the number of views, our method scales up linearly. Compared to methods that encode each view with unshared encoders (Li et al., 2021; Akbari et al., 2021; Hu & Singh, 2021; Alayrac et al., 2020), our method is more parameter efficient. Furthermore, data augmentation increases data diversity and thus improves data efficiency, which is particularly important for training data-hungry transformer models. Last but not least, by regarding views as augmentations of the input image, we can naturally incorporate a contrastive loss to help representation learning of encoded views and increase data efficiency (Grill et al., 2020; Chen et al., 2020b; Goyal et al., 2021). Different from how other VL methods (Radford et al., 2021; Jia et al., 2021; Yu et al., 2022; Li et al., 2021; Yang et al., 2022) incorporate a contrastive loss, our formulation does **not** require annotated pairs (*e.g.* human annotated image-caption pairs in MS-COCO) and can work with unlabeled image-only data to achieve better performance. Also crucially, for how to effectively ensemble the views, in (2) we devise a hierarchical decoder layer, which modifies the standard transformer decoder layer by introducing two-tiered cross-attention modules. The hierarchical decoder first ensembles within each view at the token level and then ensembles across views at the view level. By introducing this hierarchical structure for ensembling, we can better model the importance of each view and adaptively weigh each view according to their usefulness.

To sum up, we make the following contributions in this paper: (1) regard heterogeneous views as augmentations of the input image to improve computation, parameter, and data efficiency; (2) devise a hierarchical decoder layer to ensemble the views by adaptively weighing the views according to their usefulness; (3) achieve significant improvement of +5.6% CIDEr on MS-COCO over state of the art in the trained-from-scratch setting, and achieve comparable or often better performance compared with methods using large-scale transformer pre-training even though we do not do so; (4) demonstrate semi-supervised training of image captioning and achieve substantial improvement of +16.8% CIDEr on Flickr30K by leveraging additional image-only data from MS-COCO; and (5) provide thorough ablations and analyses to validate our proposed method for efficiency and effectiveness.

2 Related Works

Image captioning. The goal of image captioning is to generate text descriptions for an image. It can be roughly divided into two settings: (1) trained-from-scratch and (2) with pre-training, depending on whether the model is pre-trained on a large image-text corpus or not. For the trained-from-scratch setting (Anderson et al., 2018; Huang et al., 2019; Cornia et al., 2020; Rennie et al., 2017; Jiang et al., 2018; Pan et al., 2020; Dou et al., 2022; Li et al., 2022; Kim et al., 2021; Yang et al., 2022),

researchers train the model only on the image captioning dataset such as MS-COCO (Lin et al., 2014; Chen et al., 2015). They mainly focus on the improvements of model architecture and/or training losses, and tackle the image captioning task alone. On the other hand, for the pre-training setting (Li et al., 2020; Zhang et al., 2021a; Chen et al., 2020d; Tan & Bansal, 2019; Lu et al., 2019; Su et al., 2020; Li et al., 2019; 2021; Hu et al., 2021; Wang et al., 2022b; Zeng et al., 2021), researchers first pre-train the model on a large image-text corpus and then fine-tune to various downstream VL tasks. They mainly focus on how to effectively pre-train the model so that it transfers well to a broad set of downstream VL tasks. In this paper, we also use a transformer model and work on the trained-from-scratch setting.

Image encodings. One important aspect of VL approaches is to properly encode relevant information from the input image, on which the captioning model is conditioned on for captions generation. Anderson et al. (2018) proposed to encode finer-grained information of the input image into a sequence of objects detected by an object detector pre-trained on Visual Genome (Krishna et al., 2017). This method achieved great success and soon became the dominant approach in many VL tasks (Li et al., 2020; Chen et al., 2020d; Tan & Bansal, 2019; Lu et al., 2019; Su et al., 2020; Li et al., 2019). In most recent works, thanks to advances in foundation models (Kolesnikov et al., 2020; Jia et al., 2021; Radford et al., 2021) trained on large-scale datasets, some works (Li et al., 2021; Shen et al., 2021; Wang et al., 2022c; Dou et al., 2022) encoded the input image into image grid features and achieved good performance. Nevertheless, these works still encode an input image from a single point of view by a single pre-trained model. To encode complementary information, recent works proposed to include other heterogeneous "*views*" including object tags (Li et al., 2020; Zhang et al., 2021a), global image features (Su et al., 2020; Yu et al., 2021; Zhang et al., 2021b; Ji et al., 2021), or text descriptions (Kuo & Kira, 2022) of the input image. In this paper, we use multiple views including detected objects, image grid features, and text descriptions, and focus on how to leverage and ensemble these views efficiently and effectively.

Ensemble of views. To ensemble multiple heterogeneous views, common approaches include (1) ensemble of models (Huang et al., 2019; Cornia et al., 2020; Rennie et al., 2017; Jiang et al., 2018), which trains $|V|$ (number of views) models, one for each view, and average the word predicted by each model, and (2) concatenated views (Li et al., 2020; Zhang et al., 2021a; Kuo & Kira, 2022), which concatenates the views, each represented by a sequence of tokens, into a long single view, and let the transformer encoder-decoder learn to ensemble the views internally. However, an ensemble of models is parameter inefficient as the number of model grows linearly with $|V|$. Concatenated views are computationally inefficient as the computational complexity from the transformer model is quadratic with respect to $|V|$. In this paper, in pursuit of efficiency and effectiveness, we encode the views separately with a shared encoder, and devise a novel hierarchical decoder to first ensemble within each view at the token level and then ensemble across views at the view level.

Representation learning. To properly encode heterogeneous views, existing methods (Li et al., 2021; Akbari et al., 2021; Hu & Singh, 2021; Alayrac et al., 2020) typically encode each view with a dedicated encoder. This is parameter inefficient as the number of encoders grows linearly with respect to $|V|$. This may also not be as data efficient and be prone to overfitting on the relative smaller-scale MS-COCO image captioning datasets. To learn a better representation of heterogeneous views, we propose to incorporate a contrastive loss, which facilitates superior self-/un-supervised representation learning (Chen et al., 2020c; Grill et al., 2020; Tian et al., 2020; Chen et al., 2020a), and is beneficial in low-label settings (Grill et al., 2020; Chen et al., 2020b; Goyal et al., 2021). Different from existing multi-modal works (Radford et al., 2021; Jia et al., 2021; Yu et al., 2022; Li et al., 2021; Yang et al., 2022) that incorporate contrastive loss only in the pre-training stage, or require annotated pairs (*e.g.* human annotated image-caption pairs in MS-COCO), our method incorporate contrastive loss together with the target image captioning task, and requires **no** annotated pairs.

3 METHOD

Given heterogeneous views of the input image (see Figure 1) such as objects detected by an object detector (Anderson et al., 2018), image grid features from a pre-trained image encoder (Shen et al., 2021), and text descriptions that describe image regions by cross-modal retrieval (Kuo & Kira, 2022), our goal is to **efficiently** and **effectively** leverage these views for the image captioning task. By revisiting the probabilistic model of image captioning in Section 3.1, we found two major design

choices: (1) how to encode heterogeneous views, and (2) how to ensemble these views. In Section 3.2, we propose to regard views as augmentations of the input image, and thus naturally choose to encode each view *independently* with a *shared* encoder. This formulation is more efficient in terms of computation, parameter, and data. It also allows us to add a contrastive loss to help representation learning of encoded views and increase data efficiency. In Section 3.3, we propose a *hierarchical decoder* layer to first ensemble within each view at the token level and then ensemble across views at the view level. By introducing this hierarchical structure for ensembling, the decoder better models the importance of each view and adaptively weighs each view according to their usefulness. The overall model architecture is illustrated in Figure 2.

3.1 Heterogeneous Views for Image Captioning

We start by reviewing the probabilistic model of an auto-regressive image captioning model that generates captions $\boldsymbol{y}$ from an input image $\boldsymbol{x}$:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_t p(y_t|\boldsymbol{x}, y_{1:t-1}), \tag{1}$$

where the next token (word) y_t is generated conditioned on previously generated tokens $y_{1:t-1}$ and the input image $\boldsymbol{x}$. Following modern approaches (Anderson et al., 2018; Li et al., 2020; Kuo & Kira, 2022; Shen et al., 2021), the input image $\boldsymbol{x}$ is first encoded by some pre-trained models into different views $\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_j}\}$ (*e.g.* detected objects, image grid features, or text descriptions shown in Figure 1), and then the captions are generated from these views. We can model this process in a probabilistic way by introducing a new variable $\boldsymbol{V}$ into Equation 1:

$$\prod_t p(y_t|\boldsymbol{x}, y_{1:t-1}) = \prod_t \sum_j p(\boldsymbol{v}_j|\boldsymbol{x}, y_{1:t-1}) p(y_t|\boldsymbol{v}_j, \boldsymbol{x}, y_{1:t-1}) \tag{2}$$

$$\simeq \prod_t \sum_j p(\boldsymbol{v}_j|y_{1:t-1}) p(y_t|\boldsymbol{v}_j, y_{1:t-1}) \tag{3}$$

By the law of total probability, we introduce a new variable $\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_j}\}$ and arrive at the right-hand side of Equation 2. Since $\boldsymbol{x}$ is encoded by *frozen* pre-trained models into heterogeneous views $\boldsymbol{v}_j$ and the caption is predicted from $\boldsymbol{v}_j$, we omit $\boldsymbol{x}$ and arrive at Equation 3. Equation 3 can be regarded as ensembling in the output space of the model. Suppose the captioning model is an over-parameterized deep network with parameter θ, we can also ensemble in the feature space with the following approximation:

$$\prod_t \sum_j \underbrace{p(\boldsymbol{v}_j|y_{1:t-1})}_{\equiv \beta_j} \underbrace{p(y_t|\boldsymbol{v}_j, y_{1:t-1})}_{\equiv f_\theta(\boldsymbol{v}_j)} = \prod_t \sum_j \beta_j f_\theta(\boldsymbol{v}_j) \simeq \prod_t f_{\theta'}\left(\sum\nolimits_j \beta_j \boldsymbol{v}_j\right), \tag{4}$$

where β is the ensembling weight and $f_\theta(\boldsymbol{v}_j)$ is the captioning model. Substituting $f_\theta(\cdot)$ back to its original form, we have the final formulation that ensembles in the feature space:

$$\prod_t f_{\theta'}\left(\sum\nolimits_j \beta_j \boldsymbol{v}_j\right) = \prod_t p\Big(y_t \;\Big|\; y_{1:t-1}, \underbrace{\sum\nolimits_j \beta_j \boldsymbol{v}_j}_{\equiv \boldsymbol{\mathcal{V}}}\Big) = \prod_t p\left(y_t \mid y_{1:t-1}, \boldsymbol{\mathcal{V}}\right) \tag{5}$$

Comparing our final formulation in Equation 5 with the original single-view formulation in Equation 1, the key is to compute the ensemble of views $\boldsymbol{\mathcal{V}} = \sum_j \beta_j \boldsymbol{v}_j$, which involves (1) encoding the views (Section 3.2) and (2) ensembling the views (Section 3.3).

3.2 Heterogeneous View Augmentation

To model the ensemble of views $\boldsymbol{\mathcal{V}}$, a common approach (Li et al., 2020; Kuo & Kira, 2022) is to concatenate all the views along the sequence dimension into a long single view, and let the transformer encoder and decoder ensemble them internally at the token level by the attention module. However, one major drawback of this approach is that the computational complexity scales up quadratically with respect to the number of views ($\mathcal{O}(|V|^2)$). To overcome this issue, we propose to regard the heterogeneous views as augmentations of the input image and model the encoding of views $\boldsymbol{v}_j$ and ensembling of views β_j in $\boldsymbol{\mathcal{V}} = \sum_j \beta_j \boldsymbol{v}_j$ separately.

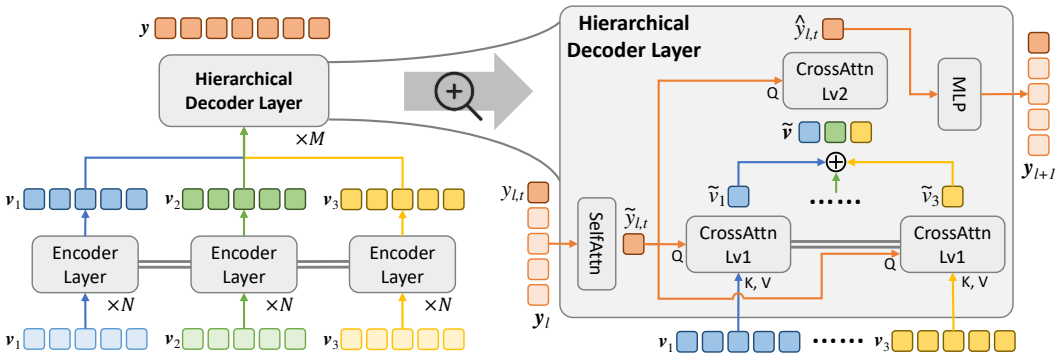

Figure 2: Model architecture. Views are represented by a sequence of d-dimensional tokens ☐. ***(Left)*** Heterogeneous views are encoded independently with a shared transformer encoder. The encoded views are then ensembled within the proposed hierarchical decoder. ***(Right)*** The hierarchical decoder layer first ensembles within each view at the token level with a shared $\text{CrossAttn}_{\text{Lv1}}$ module, and then ensembles across views at the view level with a $\text{CrossAttn}_{\text{Lv2}}$ module. For clarity of illustration, we only show operations for the $\tilde{y}_{l,t}$ token and not the rest of $\tilde{y}_{l,1:t-1}$ tokens.

Just like how we use data augmentation in computer vision, we would not concatenate all the augmented images into one single image before sending it into the model or use different encoders to encode different augmented images. Naturally, the views are encoded independently using a shared encoder. By encoding each view independently, the computational complexity scales up linearly ($\mathcal{O}(|V|)$), which is computationally efficient. By encoding each view with a shared transformer encoder, the model size stays constant ($\mathcal{O}(1)$), which is parameter efficient. Furthermore, data augmentation increases data diversity and thus improves data efficiency.

Since we regard the views as augmentations of the input image, we can add a contrastive loss (He et al., 2019; Chen et al., 2020c;a; Grill et al., 2020) along with model training to improve the representation quality of encoded heterogeneous views. In contrastive learning, augmented views from the same input image are positive pairs and those from different images are negative pairs. The representation of positive pairs is pulled together in an embedding space while the representation of negative pairs is pushed apart. The diversity of views is critical to contrastive learning (Chen et al., 2020a; Tian et al., 2020; Sohn et al., 2020; Kuo et al., 2020) and thus we add channel-wise and sequence-wise dropout during training. For channel-wise dropout, each channel across views and sequences are randomly zero-ed out with probability p_c. For sequence-wise dropout, each token, which is a d-dimensional feature vector, across views are randomly zero-ed out with probability p_s. Note that this usage of contrastive loss is different from most other VL pre-training methods (Radford et al., 2021; Jia et al., 2021; Yu et al., 2022; Li et al., 2021; Yang et al., 2022), where paired annotations such as image and text pairs annotated by human or scraped from the internet are required. On the other hand, our method only requires input image *without* paired annotations to construct positive and negative pairs for the contrastive loss. Our novel formulation enables training an image captioning model in a semi-supervised way to achieve better performance with extra image-only data.

3.3 Heterogeneous View Ensembling

To ensemble the encoded heterogeneous views from the last section, we specifically focus on how to learn the ensembling weights β in the ensemble of views $\boldsymbol{\mathcal{V}} = \sum_j \beta_j \boldsymbol{v}_j$. Since β plays the same role as $p(\boldsymbol{v}_j|y_{1:t-1})$ in Equation 4, it should also be conditioned on previously generated words $y_{1:t-1}$. It should also adaptively weigh the views according to their usefulness. For example, when a view fails to properly encode important information from the input image, β for that view should be lower and the model should weigh more on views that encode important information. Lastly, we spent great efforts in the last section to encode the views efficiently. Therefore, the computation of β should be at least as efficient as the view encoding strategy described in the last section.

We propose a hierarchical decoder layer shown in Figure 2, which modifies the standard transformer decoder layer by introducing two-tiered cross-attention structure. The hierarchical decoder first

ensemble within each view at the token level by a shared $\text{CrossAttn}_{\text{Lv1}}$ module. The outputs from $\text{CrossAttn}_{\text{Lv1}}$ of all views are collected and concatenated into a sequence along the sequence dimension, and then ensembled across views at the view level by the $\text{CrossAttn}_{\text{Lv2}}$ module. By introducing this hierarchical structure for ensembling, we can better model the usefulness of each view by $\text{CrossAttn}_{\text{Lv1}}$ and adaptively adjust the ensembling weights of each view according to their usefulness by $\text{CrossAttn}_{\text{Lv2}}$. To encourage the decoder to better leverage all views instead of focusing on a few certain views, we add a view-wise dropout to randomly drop an encoded view with probability p_v during training. Please see Supplementary D for step-by-step operations.

We now discuss the properties of our proposed hierarchical decoder layer. One important aspect is efficiency. The hierarchical decoder layer uses a shared $\text{CrossAttn}_{\text{Lv1}}$ module to first ensemble within each view. The computational complexity scales up linearly, and the model size is constant with respect to $|V|$. It then ensembles across views using a $\text{CrossAttn}_{\text{Lv2}}$ module with linear computational complexity, and constant model size with respect to $|V|$. Overall, the computational complexity and model size do not exceed our proposed view encoding strategy in Section 3.2. Another desired property is that β should be conditioned on previously generated tokens $y_{1:t-1}$, and should adaptively adjust for each view according to their usefulness. Since β is modeled as the attention weights within $\text{CrossAttn}_{\text{Lv2}}$ with $\tilde{y}_{l,t}$ as query, which self-attends to $y_{1:t-1}$, it is indeed conditioned on $y_{1:t-1}$. Furthermore, since the attention weights of $\text{CrossAttn}_{\text{Lv2}}$ are computed by the similarity between $\tilde{y}_{l,t}$ and $\tilde{v}$, the usefulness of each view can be taken into account when computing β.

4 Experiment

4.1 Implementation Details

Our proposed HEAV can be easily incorporated into existing encoder-decoder transformer models. In this paper, we choose Xmodal-Ctx (Kuo & Kira, 2022) as our base image captioning model. Following the conventional training procedure (Cornia et al., 2020; Li et al., 2020; Vinyals et al., 2015), the model is first trained with cross-entropy loss and then fine-tuned with SCST (Rennie et al., 2017) loss, which optimizes CIDEr score by reinforcement learning. To generate heterogeneous views of the input image, we use detected objects from Anderson et al. (2018), CLIP ViT-B/32 (Radford et al., 2021) image grid features from Shen et al. (2021), and text descriptions retrieved by CLIP ViT-L/14 from (Kuo & Kira, 2022). All pre-trained models for generating the views are frozen and thus the views can all be pre-generated offline. For the contrastive loss, we prepend a learnable [CLS] token for each view before sending it into the transformer encoder and use the encoded [CLS] token as a view-level representation for computing the contrastive loss. We adapt MoCo-v3 (Ji et al., 2021) with an exponential moving average (EMA) transformer encoder and a memory buffer to compute the contrastive loss. The loss is included in both the cross-entropy and SCST training stages, with loss weights set to 0.05 and 0.2, respectively. More implementation details and hyper-parameters can be found in Supplementary E.

4.2 Main Results

In Table 1, we show the results of our HEAV on the test set of MS-COCO Karpathy split (Karpathy & Fei-Fei, 2015). It is worth noting that HEAV is **not** pre-trained on external image-and-text corpus

Table 1: Image captioning results on the test set of MS-COCO Karpathy split (Karpathy & Fei-Fei, 2015). Since our HEAV is trained from scratch, we separately compare methods with large-scale transformer pre-training on the left, and those trained from scratch on the right.

Method	Pre-train	B-4	M	C	S
VLP (Zhou et al., 2020)	3M	39.5	29.3	129.3	23.2
X-VLM (Zeng et al., 2021)	16M	40.4	-	139.3	-
Oscar (Li et al., 2020)	6.5M	40.5	29.7	137.6	22.8
VinVL (Zhang et al., 2021a)	8.8M	40.9	**30.9**	140.4	**25.1**
GIT (Wang et al., 2022a)	4M	**41.3**	30.4	139.1	24.3
ViTCap (Fang et al., 2022)	4M	41.2	30.1	138.1	24.1
HEAV (ours)	None	41.0	30.2	**141.5**	23.9

Method	B-4	M	C	S
SCST (Rennie et al., 2017)	34.2	26.7	114.0	-
Up-Down (Anderson et al., 2018)	36.3	27.7	120.1	21.4
AoANet (Huang et al., 2019)	38.9	29.2	129.8	22.4
$\mathcal{M}^2$ (Cornia et al., 2020)	39.1	29.1	131.2	22.6
CLIP-ViL (Shen et al., 2021)	40.2	29.7	134.2	23.8
Xmodal-Ctx (Kuo & Kira, 2022)	39.7	30.0	135.9	23.7
HEAV (ours)	**41.0**	**30.2**	**141.5**	**23.9**

and instead is trained from scratch only on the MS-COCO image captioning dataset (Chen et al., 2015). Therefore, for fair comparison, we separate these two different settings in Table 1, where methods on the left use pre-training while those on the right are trained from scratch. In the same trained-from-scratch setting, our HEAV outperforms previous state-of-art Xmodal-Ctx (Kuo & Kira, 2022) by 5.6% in CIDEr and 1.3% in BLEU-4. On the other hand, when compared with methods with transformer pre-training, our HEAV, despite being only trained on MS-COCO, achieves comparable or often better performance.

To further demonstrate the data efficiency of HEAV, we train it on the Karpathy split (Karpathy & Fei-Fei, 2015) of Flickr30K, which only has ∼0.15M training data (4x smaller than MS-COCO), and show the results in Table 2. The results of other methods are taken from Stefanini et al. (2022). We can see that our HEAV outperforms previous state-of-art ORT substantially by 12.9% in CIDEr and 4.3% in BLEU-4. The significant improvement may come from the data efficiency of our method, which is particularly important when trained on the smaller-scale dataset of Flickr30K.

Table 2: Image captioning results on Flickr30K Karpathy split (Karpathy & Fei-Fei, 2015). We also demonstrate semi-supervised learning (SSL) for image captioning with labeled data from Fickr30K and unlabeled data from MS-COCO.

Method	B-4	M	C	S
Show & Tell (Vinyals et al., 2015)	21.5	18.3	41.7	12.2
Show, Attend & Tell (Xu et al., 2015)	23.6	19.2	49.1	13.3
Up-Down (Anderson et al., 2018)	28.3	21.6	63.3	15.9
$\mathcal{M}^2$ (Cornia et al., 2020)	29.8	22.4	68.4	16.2
ORT (Herdade et al., 2019)	30.1	22.8	68.8	16.9
HEAV (ours)	**34.4**	24.6	81.7	18.0
HEAV + SSL (ours)	34.3	**25.1**	**85.6**	**19.0**

4.3 Ablations and Analyses

The goal of this paper is to propose an *efficient* and *effective* way for leveraging and ensembling heterogeneous views. Therefore, in this section we closely examine whether our proposed HEAV achieves these goals in Section 4.3.1 for efficiency and Section 4.3.2 for effectiveness. Following the convention in Huang et al. (2019); Kuo & Kira (2022); Herdade et al. (2019), we only train the model with cross-entropy loss for all ablations and analyses.

4.3.1 Is HEAV computation, parameter, and data efficient?

In Section 3.2, we propose to regard views as augmentations of the input image and encode the views independently with a shared transformer encoder. To demonstrate the efficiency of our method, in Table 3, we show the theoretical computation and parameter complexity as well as the actual training speed and trainable parameters. Since we do not want to trade performance in pursuit of efficiency, we also show in Table 3 that our method achieves better performance despite being more efficient.

Comparison with common ensembling approaches. Other common approaches include (1) ensemble of models (Rennie et al., 2017; Cornia et al., 2020; Huang et al., 2019; Anderson et al., 2018), which train $|V|$ image captioning models, one for each view, and ensembles their predicted words, and (2) concatenated views (Li et al., 2020; Kuo & Kira, 2022), which concatenates all views along the sequence dimension into a long single view, and let the transformer model ensemble the views internally at the token level by the attention module. In Table 3, ensemble of models has to train $|V|$ models, one for each view, and thus is parameters inefficient ($\mathcal{O}(|V|)$). Concatenated views concatenates all views into one long single view and is computationally expensive ($\mathcal{O}(|V|^2)$). Compared to these two approaches, our HEAV has linear computation ($\mathcal{O}(|V|)$) and constant parameter ($\mathcal{O}(1)$) complexity, and performs consistently better across all metrics by large margins.

Shared *v.s.* unshared encoder. Another design choice is whether or not to use a shared encoder for each view. In Table 3, in the case of *w/o* $\mathcal{L}_{con}$, using unshared encoders (Li et al., 2021; Akbari et al., 2021; Hu & Singh, 2021; Alayrac et al., 2020) for each view does not bring any performance benefits compared to using a shared encoder, but with the cost of higher parameter complexity ($\mathcal{O}(|V|)$). Furthermore, we ablate the proposed contrastive loss $\mathcal{L}_{con}$, and found that the unshared encoder does not benefit from $\mathcal{L}_{con}$ as much as the shared encoder. Although $\mathcal{L}_{con}$ introduces additional overhead such as extra parameters from the projection heads and extra computation from pairwise similarity during the training time (not for inference), the increase in complexity is moderate but the improvement in performance is significant when incorporated into our shared-encoder strategy.

Table 3: Comparison with other encoding and ensembling methods in terms of complexity and performance. All models are trained with cross-entropy loss only. Please see Section 4.3.1 for more details of different conditions.

Conditions	$\mathcal{L}_{\text{con}}$	Computation		Parameter		Performance			
		complexity	iter/sec ↑	complexity	#params ↓	B-4	M	C	S
Ensemble of models		$\mathcal{O}(\|V\|)$	2.23	$\mathcal{O}(\|V\|)$	52.5M	38.5	28.6	121.1	21.3
Concatenated views		$\mathcal{O}(\|V\|^2)$	4.20	$\mathcal{O}(1)$	13.1M	38.5	28.7	122.8	21.7
Unshared encoders		$\mathcal{O}(\|V\|)$	5.33	$\mathcal{O}(\|V\|)$	20.8M	39.7	29.1	125.4	22.1
Unshared encoders	✓	$\mathcal{O}(\|V\|)$	3.96	$\mathcal{O}(\|V\|)$	22.7M	39.7	29.3	125.8	22.2
Shared encoder		$\mathcal{O}(\|V\|)$	5.97	$\mathcal{O}(1)$	13.5M	39.7	29.1	125.6	22.1
Ours (Shared encoder)	✓	$\mathcal{O}(\|V\|)$	4.83	$\mathcal{O}(1)$	15.4M	**40.5**	**29.4**	**127.6**	**22.3**

Data efficiency with fewer training data. To test the data efficiency of HEAV, in Figure 3 we train the model with $\{10, 20, ..., 90\}\%$ of data, and compare the CIDEr score with other common ensembling approaches, ensemble of models (red) and concatenated views(gold), trained on 100% of data. We can see that HEAV only needs about 40-50% training data to achieve comparable performance. With the same input views and similar model architectures, this indicates that our method is more data efficient, likely due to our novel use of heterogenous views as augmentations and the contrastive loss $\mathcal{L}_{con}$. In Supplementary A, we also show that HEAV suffers less from overfitting on MS-COCO, indicating that our model is more data efficient.

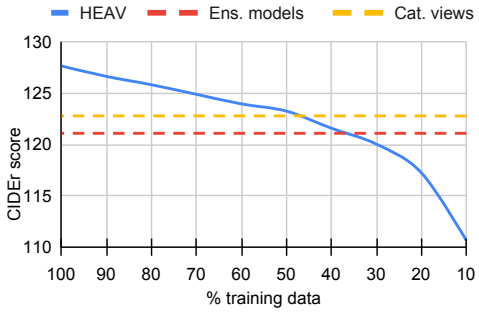

Figure 3: Data efficiency of HEAV. HEAV (blue) only needs about 40-50% training data to achieve similar performance as other common ensembling methods trained on 100% of data such as ensemble of models (red) and concatenated views (gold). All models are trained with cross-entropy loss only.

Semi-supervised training. To demonstrate data efficiency brought by the contrastive loss $\mathcal{L}_{con}$, we train a semi-supervised image captioning model. Specifically, the model is trained on the labeled data (image-caption pairs) from Flickr30K, and unlabeled data (image only) from MS-COCO. Since our contrastive loss is applied differently than other VL methods and does not require annotated pairs (*e.g.* human-annotated image-caption pairs in MS-COCO), it can be applied on the unlabeled image-only data to aid representation learning of encoded views. In Table 2, with semi-supervised training, the already strong HEAV achieves even better performance of +3.9% CIDEr and +1% SPICE.

4.3.2 Is hierarchical decoder effective?

In Section 3.3, we propose to model ensembling weights β in the ensemble of views $\boldsymbol{\mathcal{V}} = \sum_j \beta_j \boldsymbol{v}_j$ by our proposed hierarchical decoder layer, which first ensembles within each view at the token level and then ensembles across views at the view level. To understand how the hierarchical decoder benefits an image captioning model, we closely examining the ensembling weights β in two control studies. To verify the effectiveness of our design, we compare with other common designs in Table 4.

Adaptive ensembling weights $\boldsymbol{\beta}$. To better understand how the hierarchical decoder benefits the image captioning model, we design two control studies in Figure 4 to show how the ensembling weights β vary adaptively according to the usefulness of a view at the view level and at the word level. In the first experiment, we add noise to a view by randomly zeroing out tokens in a view to make a view *less useful*, and expect a drop of β toward that noised view. To measure β, we take the multi-head attention weights of $\text{CrossAttn}_{\text{Lv2}}$ at the last decoder layer and average the attention weights across heads. In Figure 4a, β for the noised view drops consistently across all caption generation steps compared to the same view *without* added noise. This means that our hierarchical decoder indeed leans to ensemble views according to their usefulness at the view level. In the second experiment, we randomly mask out a prominent region of the input image in a view. For example, we mask out dog in the input image (Figure 4b) with caption "*a* ***dog*** *laying down beside a little couch*"

(a) Ensembling weights ageraged across heads for a noised view at each caption generation step.

(b) Input image with caption: *a dog laying down beside a little couch*

(c) Ensembling weight of each attention head at the step of generating "*dog*", which is masked out in the input image.

Figure 4: The ensembling weights β vary adaptively according to the usefulness of a view at the view level and at the word level when generating a caption for the center image. ***(a)*** We add random noise to a view and show that β averaged across different attention heads drop consistently at each step of caption generation. ***(c)*** We mask out dog in the input image and show that β of different attention heads drop consistently at the step of generating the word "*dog*".

to make a view *less useful* at the step of generating the word "*dog*". We expect a drop of β toward the masked view at the step of generating the word "*dog*". To measure β, we take the multi-head attention weights of $\text{CrossAttn}_{\text{Lv2}}$ at the last decoder layer and measure the attention weights of each head at the step of generating the word "*dog*". In Figure 4c, β for the masked view drops consistently across all attention heads compared to the same view *without* masking. This means that our hierarchical decoder indeed learns to ensemble views according to their usefulness at the word level. The example in Figure 4 is randomly chosen and more examples can be found in Supplementary D.

Design choices of hierarchical decoder. In Table 4, we ablate different design choices of the hierarchical decoder. One simple alternative is to first concatenate the encoded views along the sequence dimension into a long single view, and use a single cross-attention module to jointly ensemble the views at the token level. One can also mean-/max-pool to ensemble the views in place of the $\text{CrossAttn}_{\text{Lv2}}$ module. Other works also propose to use a sigmoid/tanh gating mechanism (Huang et al., 2019) to ensemble across layers (Cornia et al., 2020), which can be generalized to ensemble across views. In Table 4, we can see that our proposed hierarchical decoder layer with a two-tiered cross-attention structure achieves the best performance compared to all other designs.

Table 4: Ablations of design choices for the proposed hierarchical decoder layer. The models are trained with cross-entropy only.

Method	B-4	M	C	S
Concatenate	39.2	29.2	125.1	22.1
Mean pool	39.4	29.2	125.2	22.2
Max pool	38.7	28.8	122.9	21.5
Sigmoid	39.4	29.2	125.5	22.2
Tanh	39.4	29.3	124.8	22.0
Ours	**40.5**	**29.4**	**127.6**	**22.3**

5 Conclusion

In this paper, we focus on the problem of how to *efficiently* and *effectively* leverage and ensemble heterogeneous views. To tackle this problem, we propose HEAV to (1) regard heterogeneous views as augmentations of the input image, and naturally encode each view independently with a shared encoder, (2) incorporate a contrastive loss across encoded views to improve representation quality and enable semi-supervised training to leverage image-only unlabeled data, and (3) ensemble the encoded views adaptively by our carefully designed hierarchical decoder layer that first ensembles within each view at the token level and then across views at the view level. Through rigorous analysis, HEAV is computation, parameter, and data efficient, and outperforms other less efficient designs and existing approaches for views ensembling. We also demonstrate that our hierarchical decoder successfully models the usefulness of views and weigh the views adaptively according to their usefulness. We demonstrate significant performance improvements of +5.6% CIDEr compared to state-of-art *w/o* transformer pre-training on MS-COCO and +16.8% CIDEr with SSL on Flickr30K.

REFERENCES

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8948–8957, 2019.

Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021.

Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *Advances in Neural Information Processing Systems*, 33:25–37, 2020.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020b.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pp. 104–120. Springer, 2020d.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10578–10587, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Nanyun (Violet) Peng, Zicheng Liu, and Michael Zeng. An empirical study of training end-to-end vision-and-language transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR), 2022.*, June 2022.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lin Liang, Zhe Gan, Lijuan Wang, Yezhou Yang, and Zicheng Liu. Injecting semantic concepts into end-to-end image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18009–18019, 2022.

Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *Advances in Neural Information Processing Systems*, 32, 2019.

Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1439–1449, 2021.

Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. *arXiv preprint arXiv:2111.12233*, 2021.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4634–4643, 2019.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Jiayi Ji, Yunpeng Luo, Xiaoshuai Sun, Fuhai Chen, Gen Luo, Yongjian Wu, Yue Gao, and Rongrong Ji. Improving image captioning by leveraging intra-and inter-layer global representation in transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 1655–1663, 2021.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pp. 4904–4916. PMLR, 2021.

Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 499–515, 2018.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pp. 5583–5594. PMLR, 2021.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 491–507. Springer, 2020.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Chia-Wen Kuo and Zsolt Kira. Beyond a pre-trained object detector: Cross-modal textual and visual context for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *European Conference on Computer Vision*, pp. 479–495. Springer, 2020.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pp. 121–137. Springer, 2020.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10971–10980, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28: 91–99, 2015.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7008–7024, 2017.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: a survey on deep learning-based image captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SygXPaEYvH.

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022a.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*, 2022b.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*, 2022c. URL https://openreview.net/forum?id=GUrhfTuf_3.

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

Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15671–15680, 2022.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. In *AAAI*, 2021.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5579–5588, 2021a.

Xuying Zhang, Xiaoshuai Sun, Yunpeng Luo, Jiayi Ji, Yiyi Zhou, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Rstnet: Captioning with adaptive attention on visual and non-visual words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15465–15474, 2021b.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13041–13049, 2020.

APPENDIX

A OVERFITTING

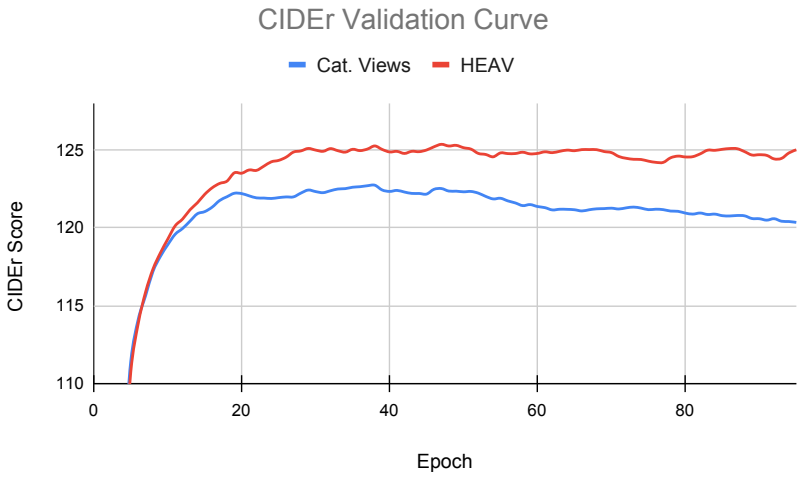

Figure 5: CIDEr validation curve of HEAV *v.s.* concatenated views.

Training a data-hungry transformer model on a medium-scale dataset of MS-COCO (around 0.6M training samples) is prone to overfitting. In HEAV, we propose to regard heterogeneous views as augmentations of the input image and encode the views independently with a shared encoder. We claim that this formulation increases data diversity and is more parameter and data efficient. Furthermore, we add a contrastive loss to improve representation quality of encoded views, which is also beneficial for data efficiency. In Figure 5, compared to concatenated views, our HEAV indeed suffers less from overfitting. Due to overfitting, the CIDEr score of concatenated views drops by 3.6 from the highest to the end of training.

B GENERATED CAPTIONS

In Figure 6, we show some random examples of different captions generated by our HEAV and another trained-from-scratch SoTA method Xmodal-Ctx Kuo & Kira (2022). Qualitatively, HEAV is capable of generating captions in more details and more closely related to the input image rather than generating a generic sentence. For example, in Figure 6a, HEAV generates "*a man standing in a living room holding a nintendo wii game controller*", while Xmodal-Ctx generates a more generic description of "*a group of people sitting on a couch playing a video game*". Another example in Figure 6e shows that HEAV describes the train in more details as "*a yellow and purple train*" rather than just "*a train*" by Xmodal-Ctx.

(a)
HEAV: a man standing in a living room holding a nintendo wii game controller
Xmodal-Ctx: a group of people sitting on a couch playing a video game

(b)
HEAV: a batter catcher and umpire during a baseball game
Xmodal-Ctx: a baseball player holding a bat on a field

(c)
HEAV: a bunch of umbrellas hanging from a ceiling
Xmodal-Ctx: a bunch of flowers hanging from a ceiling

(d)
HEAV: two people playing a video game in a room
Xmodal-Ctx: a person standing in front of a tv

(e)
HEAV: a yellow and purple train parked at a train station
Xmodal-Ctx: a train that is sitting on the tracks

(f)
HEAV: a piece of cake on a plate with a flower
Xmodal-Ctx: a slice of cake on a plate with a fork

Figure 6: Captions generated by HEAV and another trained-from-scratch SoTA method Xmodal-Ctx (Kuo & Kira, 2022)

C STEP-BY-STEP OPERATIONS OF HIERARCHICAL DECODER

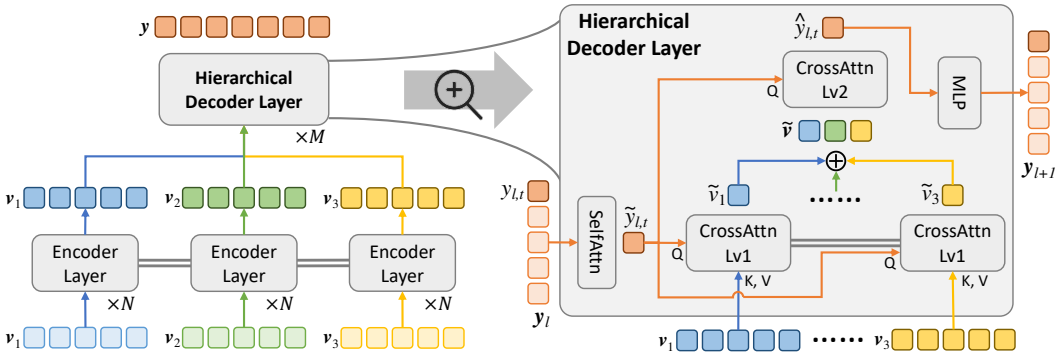

Figure 7: Model architecture. Each □ represents a d-dimensional token. ***(Left)*** Heterogeneous views (different views represented by different colors) are encoded independently with a shared transformer encoder. The encoded views are then ensembled within the proposed hierarchical decoder. ***(Right)*** The hierarchical decoder layer first ensembles within each view at the token level with a shared $\text{CrossAttn}_{\text{Lv1}}$ module, and then ensembles across views at the view level with a $\text{CrossAttn}_{\text{Lv2}}$ module. For clarity of illustration, we only show operations with respect to the $\tilde{y}_{l,t}$ token and not the rest of $\tilde{y}_{l,1:t-1}$ tokens.

In this section, we detail the step-by-step operations of our proposed hierarchical decoder layer in Figure 7. At the l-th decoder layer, given previously generated $1{:}t$ words at the l-th layer $\boldsymbol{y}_l = [y_{l,1}, y_{l,2}, ..., y_{l,t}]$ and encoded views $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_j]$, we first pass $\boldsymbol{y}_l$ through a causal self-attention module to get $\tilde{\boldsymbol{y}}_l = [\tilde{y}_{l,1}, \tilde{y}_{l,2}, ..., \tilde{y}_{l,t}] = \text{SelfAttn}(\boldsymbol{y}_l)$ such that y_{l,t_1} does not attend to y_{l,t_2} for $t_1 < t_2$. For the clarity of illustration, we will only show the operations with respect to $\tilde{y}_{l,t}$ and not the rest of $\tilde{y}_{l,1:t-1}$. We first perform Lv1 cross-attention to ensemble tokens *within* each view at the token level as $\tilde{v}_i = \text{CrossAttn}_{\text{Lv1}}(\tilde{y}_{l,t}, \boldsymbol{v}_i, \boldsymbol{v}_i)$, where $\text{CrossAttn}(Q, K, V)$ takes query, key, and value as input and output a sequence same shape as the query. After the operation of Lv1 cross-attention, each view $\boldsymbol{v}_i$, a sequence of tokens, are ensembled into $\tilde{v}_i$, a single d-dimensional token. All $\tilde{v}_i$ are collected and concatenated along the sequence dimension into $\tilde{\boldsymbol{v}} = [\tilde{v}_1, \tilde{v}_2, ..., \tilde{v}_N]$. We then perform Lv2 cross-attention to ensemble *across* views at the view level as $\hat{y}_{l,t} = \text{CrossAttn}_{\text{Lv2}}(\tilde{y}_{l,t}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{v}})$. The final output of the l-th layer decoder is $y_{l+1,t} = \text{MLP}(\hat{y}_{l,t})$.

D ADAPTIVE ENSEMBLING WEIGHTS β

We show more examples of how the hierarchical decoder adaptively weigh each view according to their usefulness at the view level and at the word level in Figure 8-14. At the view level (figures on the left), we add noise to a view by randomly zeroing out tokens in a view to make a view *less useful*. β is measured as the the multi-head attention weights of $\text{CrossAttn}_{\text{Lv2}}$ at the last decoder layer average across heads. Overall, we can see that β for the noised view drops across caption generation steps compared to the same view *without* added noise. This means that our hierarchical decoder indeed leans to ensemble views according to their usefulness at the view level. At the word level (figures on the right), we randomly mask out a prominent object of the input image in a view. β is measured as the multi-head attention weights of $\text{CrossAttn}_{\text{Lv2}}$ at the last decoder layer at the step of generating the word corresponding to the masked object. Overall, we can see that β for the masked view drops across attention heads compared to the same view *without* masking. This means that our hierarchical decoder indeed leans to ensemble views according to their usefulness at the word level.

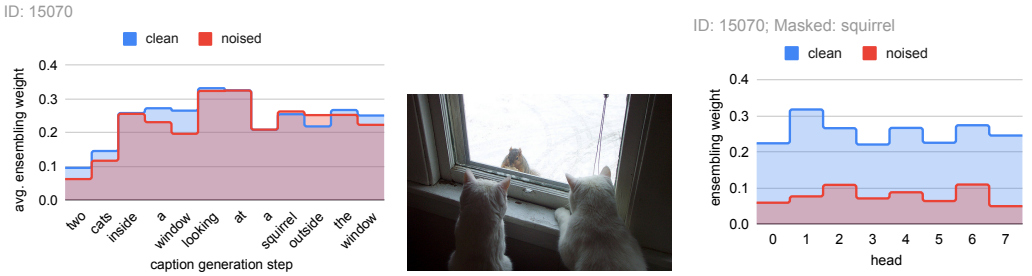

Figure 8: ***(left)*** Ensembling weights ageraged across heads for a noised view at each caption generation step. ***(center)*** Input image with caption: "*two cats inside a window looking at a squirrel outside the window*". ***(right)*** Ensembling weight of each attention head at the step of generating "*squirrel*", which is masked out in the input image.

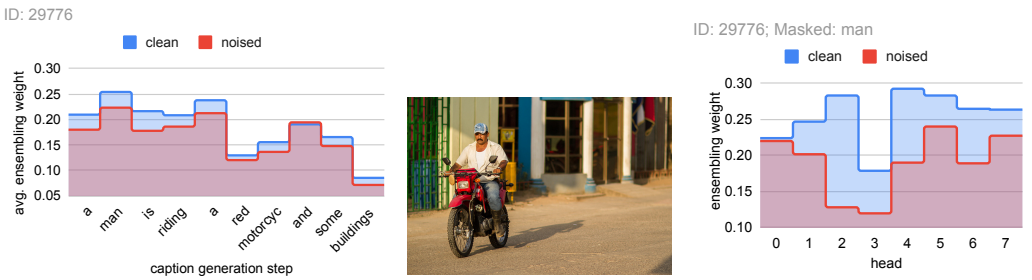

Figure 9: ***(left)*** Ensembling weights ageraged across heads for a noised view at each caption generation step. ***(center)*** Input image with caption: "*a man is riding a red motorcycle and some buildings*". ***(right)*** Ensembling weight of each attention head at the step of generating "*man*", which is masked out in the input image.

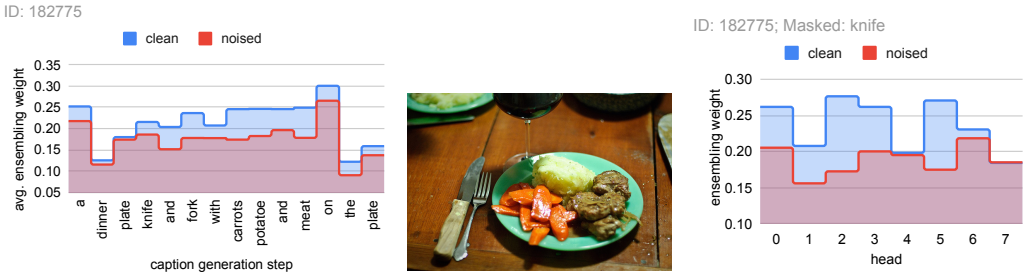

Figure 10: ***(left)*** Ensembling weights ageraged across heads for a noised view at each caption generation step. ***(center)*** Input image with caption: "*a dinner plate knife and fork with carrots potatoes and meat on the plate*". ***(right)*** Ensembling weight of each attention head at the step of generating "*knife*", which is masked out in the input image.

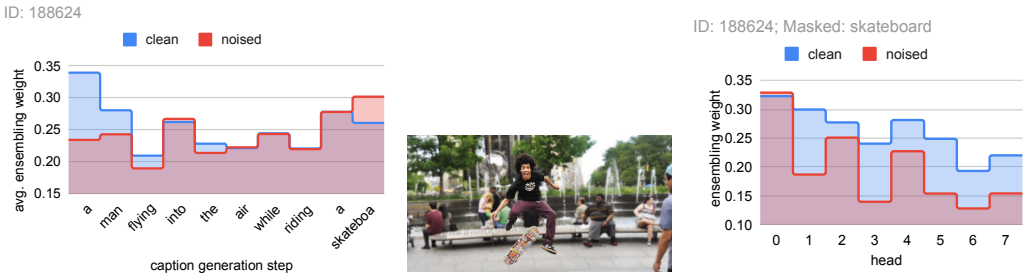

Figure 11: ***(left)*** Ensembling weights ageraged across heads for a noised view at each caption generation step. ***(center)*** Input image with caption: "*a man flying into the air while riding a skateboard*". ***(right)*** Ensembling weight of each attention head at the step of generating "*skateboard*", which is masked out in the input image.

Figure 12: ***(left)*** Ensembling weights ageraged across heads for a noised view at each caption generation step. ***(center)*** Input image with caption: "*cat standing in toilet next to a tile floor*". ***(right)*** Ensembling weight of each attention head at the step of generating "*cat*", which is masked out in the input image.

Figure 13: ***(left)*** Ensembling weights ageraged across heads for a noised view at each caption generation step. ***(center)*** Input image with caption: "*a train rolls down the tracks at the train station*". ***(right)*** Ensembling weight of each attention head at the step of generating "*train*", which is masked out in the input image.

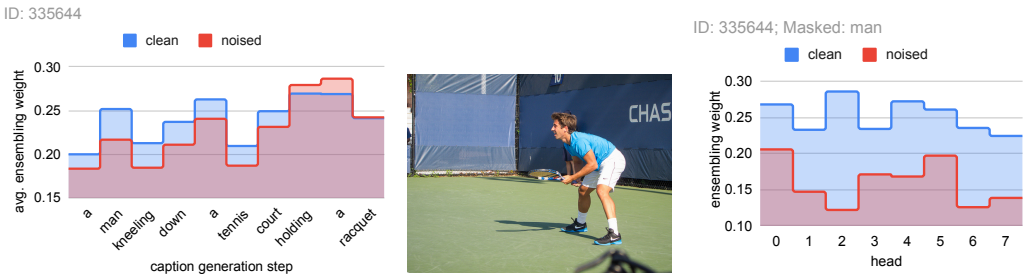

Figure 14: ***(left)*** Ensembling weights ageraged across heads for a noised view at each caption generation step. ***(center)*** Input image with caption: "*a man kneeling down a tennis court holding a racquet*". ***(right)*** Ensembling weight of each attention head at the step of generating "*man*", which is masked out in the input image.

E IMPLEMENTATION DETAILS

We provide a detailed list of hyperparameters including their values and whether they are tuned in Table 5 (cross-entropy training) and Table 6 (SCST training). For cross-entropy training, the model can be trained with a single Nvidia 2080 Ti GPU in 2 days. For SCST training, the model can be trained with a single Nvidia A40 GPUs in 4 days.

Table 5: Hyperparameters for cross-entropy training. The values for untuned parameters are inherent from the base image captioning model Xmodal-Ctx (Kuo & Kira, 2022).

Hyperparameter	Value	Tuned	Note
N	3		Number of encoder layers
M	3		Number of decoder layers
lr	2e-5	✓	learning rate
bs	50		batch size
wd	0.05		weight decay
λ	0.05	✓	loss weight for $\mathcal{L}_{con}$
p_c	0.1		drop rate for channel-wise dropout
p_s	0.1		drop rate for sequence-wise dropout
p_v	0.1		drop rate for view-wise dropout
optimizer	AdamW		Adam with decoupled weight decay (Loshchilov & Hutter, 2019)
lr scheduler	constant with warmup		linearly warm up lr from 0.0 and then stay constant
warmup steps	10k		
K	8k	✓	size of memory buffer for MoCo contrastive learning
τ	0.06	✓	temperature for MoCo contrastive learning
ema	0.999		exponential moving avergae for MoCo contrastive learning

Table 6: Hyperparameters for SCST (Rennie et al., 2017) training. The values for untuned parameters are inherent from the base image captioning model Xmodal-Ctx (Kuo & Kira, 2022), or from the tuned value in cross-entropy training.

Hyperparameter	Value	Tuned	Note
N	3		Number of encoder layers
M	3		Number of decoder layers
lr	5e-6		learning rate
bs	40		batch size
wd	0.0		weight decay
λ	0.2	✓	loss weight for $\mathcal{L}_{con}$
p_c	0.1		drop rate for channel-wise dropout
p_s	0.1		drop rate for sequence-wise dropout
p_v	0.1		drop rate for view-wise dropout
optimizer	AdamW		Adam with decoupled weight decay (Loshchilov & Hutter, 2019)
lr scheduler	None		do not use any lr scheduler
K	8k		size of memory buffer for MoCo contrastive learning
τ	0.06		temperature for MoCo contrastive learning
ema	0.999		exponential moving avergae for MoCo contrastive learning