A Additional Details for Object Detection

A.1 Object Detection Datasets
When pretraining our model on the four datasets (i.e., Visual Genome (VG), COCO, OpenImages, and Objects365), we follow [10] to build a unified training corpus with the statistics shown in Table 1 except that we do not use the annotations from COCO stuff [4]. The resultant corpus has 2.49M unique images with 1848 categories.

| Source | VG  | COCO | Objects365 | OpenImages |
|--------|-----|------|------------|------------|
| Images | 97k | 111k | 609k       | 1.67M      |
| Categories | 1594 | 80   | 365        | 500        |
| Sampling | ×8  | ×8   | ×2         | ×1         |

A.2 Implementation Details
For the object detector, we set the number of queries \( N = 150 \), the number of sampling points equal to 4, and the hidden dimension \( d = 512 \). The backbone network weights are intialized by the weights of Swin-Base \((384 \times 384)\) pretrained on ImageNet21K [9]. Following [11], the loss for normalized bounding box regression for object \( i \), \( L_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) \) is computed as the weighted summation of a box distance \( L_{t_i} \) and a GIoU loss \( L_{\text{iou}} \):

\[
L_{t_i}(b_i, \hat{b}_{\sigma(i)}) = ||b_i - \hat{b}_{\sigma(i)}||_1, \tag{1}
\]

\[
L_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) = 1 - \frac{|b_i \cap \hat{b}_{\sigma(i)}|}{|b_i \cup \hat{b}_{\sigma(i)}|} - \frac{|B(b_i, \hat{b}_{\sigma(i)}) \setminus b_i \cup \hat{b}_{\sigma(i)}|}{|B(b_i, \hat{b}_{\sigma(i)})|}, \tag{2}
\]

\[
L_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) = \alpha_{t_i} L_{t_i}(b_i, \hat{b}_{\sigma(i)}) + \alpha_{\text{iou}} L_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}), \tag{3}
\]
where $\alpha_{l_1} = 5$, $\alpha_{iou} = 2$, and $B$ outputs the largest box covering $b_i$ and $\hat{b}_{\sigma(i)}$. We also employ two training strategies, i.e., iterative bounding box refinement and auxiliary losses; see [11] and our configuration files for details.

Table 2: Performance of object detection on the COCO and Visual Genome datasets. ‘4DS’ denotes the four object detection datasets.

| Model  | Training Data | mAP (COCO) | mAP^{50} (VG) |
|--------|----------------|-------------|---------------|
| BUTD [3] | VG             | -           | 10.2          |
| VinVL [10] | 4DS          | 50.5        | 13.8          |
| GRIT   | VG             | 33.6        | 14.2          |
| GRIT†  | 4DS            | 50.8        | 15.1          |

A.3 Object Detection Results

Table 2 shows the performance on the COCO validation split and the Visual Genome test split of our object detector compared with VinVL and BUTD [3]. It is seen that the object detector of GRIT attains comparable or higher performance on the two datasets as compared with BUTD and VinVL when pretrained on the similar datasets.

B Additional Details for Image Captioning

Class Token We prepend a class token embedding $g_{\langle\text{cls}\rangle} \in \mathbb{R}^d$ to $G_0$ before forwarding them to the grid feature network. We use this class token embedding to predict the emotion category of the input image when training an emotion-grounded model on the ArtEmis dataset; see Sec. B.2.

Boundary Tokens Following previous studies, we prepend a special token $\langle\text{sos}\rangle$ to the beginning of captions, and append another special token $\langle\text{eos}\rangle$ to the end of captions during training. During inference, we start the generation by setting the first token to $\langle\text{sos}\rangle$.

B.1 Image Captioning on the COCO dataset

SPICE Sub-category and CLIPscore Metrics Table 3 reports a breakdown of SPICE F-scores over various sub-categories on the “Karpathy” test split, in comparison with the region-based methods: Up-Down [3], vanilla Transformer [5], and $M^2$ Transformer [5]. These scores give a quantitative assessment of performance on different aspects when describing the content of images. As seen in Table 3, our method attains better scores over all sub-categories, showing
Table 3: Breakdown of SPICE F-scores over various sub-categories and the CLIP scores.

| Method           | SPICE | Object | Attr. | Relation | Color | Count | Size | CLIP |
|------------------|-------|--------|-------|----------|-------|-------|------|------|
| Up-Down [3]      | 21.4  | 39.1   | 10.0  | 6.5      | 11.4  | 18.4  | 3.2  | -    |
| Transformer [5]  | 21.1  | 38.6   | 9.6   | 6.3      | 9.2   | 17.5  | 2.0  | -    |
| \(\mathcal{M}^2\) Trans. [5] | 22.6  | 40.0   | 11.6  | 6.9      | 12.9  | 20.4  | 3.5  | 73.4 |
| GRIT†            | 24.3  | 42.7   | 13.5  | 7.7      | 14.7  | 29.3  | 4.5  | 77.2 |

significant improvement on identifying and counting objects, attributes, and relationships between objects. The table also reports the CLIP scores [6] of the two methods, showing consistent improvement of our method over the compared method.

B.2 Image Captioning on the ArtEmis dataset

ArtEmis Dataset This dataset consists of 80,031 unique images divided into the training, validation, and test splits with the ratios of 85%, 5%, and 10%, respectively. Each caption of a given image is annotated with an emotion label. In total, there are 454,684 captions along with 8 unique emotion categories; see [1] for details.

Emotion Grounded Model Following [1], we also trained an emotion grounded model, which predicts the emotion associated with the caption. Specifically, we mapped the updated class embedding \(g_{(cls)}\) into an 8-dimensional vector using a linear projection. During training, we minimized the summation of the two losses, i.e., emotion prediction and caption generation.

Full Results Table 4 shows the full results of different models on the test split of the Artemis dataset including the emotion grounded models. It is noted that the ground truth emotion labels are not provided during inference.

B.3 Image Captioning on the nocaps Dataset

Full results We report the full results on the validation split of the nocaps dataset for different domains, i.e., in-domain, near-domain, and out-of-domain, in Table 5.

B.4 Computational Efficiency

We measured the inference time of GRIT and two representative region-based methods, VinVL [10] and \(\mathcal{M}^2\) Transformer [5], on the same machine having a
Table 4: Performance on the ArtEmis test split.

| Method       | Emotion Type | Performance Metrics |
|--------------|--------------|---------------------|
|              |              | B@1 B@2 B@3 B@4 M  R |
| NN [1]       | No           | h 36.4 13.9 5.4 2.2 10.2 21.0 |
| ANP [1]      | No           | G 39.6 13.4 4.2 1.4 8.8 20.2 |
| M²Trans. [1] | Yes          | R 51.1 28.2 15.4 9.0 13.7 28.6 |
| M²Trans. [1] | No           | R 50.7 28.2 15.9 9.5 14.0 28.0 |
| SAT [1]      | Yes          | G 52.0 28.0 14.6 7.9 13.4 29.4 |
| SAT [1]      | No           | G 53.6 29.0 15.5 8.7 14.2 29.7 |
| GRIT†        | Yes          | R+G 69.3 39.4 19.2 11.1 16.5 33.0 |
| GRIT†        | No           | R+G 70.1 40.1 20.9 11.3 16.8 33.3 |

Table 5: Performance on the nocaps validation split.

| Method     | V.E in-domain | near-domain | out-domain | Overall |
|------------|---------------|-------------|------------|---------|
|            | Type          | C S         | C S        | C S     | C S     |
| NBT [2]    | R             | 62.7 10.1   | 51.9 9.2   | 54.0 8.6 | 53.9 9.2 |
| Up-down [2]| R             | 78.1 11.6   | 57.7 10.3  | 31.3 8.3 | 55.3 10.1 |
| Trans. [5] | R             | 78.0 11.0   | - -        | 29.7 7.8 | 54.7 9.8 |
| M²Trans. [5]| R         | 85.7 12.1  | - -        | 38.9 8.9 | 64.5 11.1 |
| GRIT†      | R+G           | 105.9 13.6  | 92.16 13.05 | 72.6 11.1 | 90.2 12.8 |

Tesla V100-SXM2 of 16GB memory with CUDA version 10.0 and Driver version 410.104. It has Intel(R) Xeon(R) Gold 6148 CPU. The comparison was conducted following [7,8]. Specifically, we excluded the time of preprocessing the image and loading it to the GPU device. Also, the images are rescaled to the resolutions such that all the compared methods achieve its highest performance for image captioning. For the compared methods, we used the official implementations of M² Transformer³ and VinVL⁴.

Regarding feature extraction, we extracted the region features from Faster R-CNN using the original implementation⁵ used by M² Transformer and another implementation⁶ used by VinVL. It is seen that VinVL and M² Transformer spend considerable time on feature extraction due to the forward pass through the CNN backbone with high resolution inputs and the computationally expensive regional operations. It is also noted that VinVL introduced class-agnostic NMS operations, which reduce a great amount of time consumed by class-aware NMS operations in the standard Faster R-CNN. On the other hand, we employ

³ https://github.com/aimagelab/meshed-memory-transformer
⁴ https://github.com/pzzhang/VinVL
⁵ https://github.com/peteanderson80/bottom-up-attention
⁶ https://github.com/microsoft/scene_graph_benchmark
a Deformable DETR-based detector to extract region features without using all such operations. Table 6 shows the comparison on feature extraction.

Table 6: The inference time on feature extraction of different methods.

| Method     | Backbone     | Detector     | Regional Operations | Inference Time |
|------------|--------------|--------------|---------------------|----------------|
| VinVL\textsubscript{large}[10] | ResNeXt-152  | Faster R-CNN | Class-Agnostic NMS  | 304 ms         |
|            |              |              | Rol Align, etc      |                |
| M\textsuperscript{2} Trans. [5] | ResNet-101   | Faster R-CNN | Class-Aware NMS     | 736 ms         |
|            |              |              | Rol Align, etc      |                |
| GRIT       | Swin-Base    | DETR-based   | -                   | 31 ms          |

Regarding caption generation, all the methods use beam search as the decoding strategy, with beam size of 5 and the maximum caption length of 20. Both M\textsuperscript{2} Transformer and GRIT employ a lightweight caption generator (caption decoder) having only 3 transformer layers with hidden dimension of 512 while VinVL\textsubscript{large} has 24 transformer layers with hidden dimension of 1024; see Table 7. Thus, with the visual features as inputs, M\textsuperscript{2} Transformer and GRIT spend less inference time generating words than VinVL\textsubscript{large} in the autoregressive manner.

Table 7: The inference time on caption generation of different methods.

| Method       | No. of Layers | Hidden Dim | Inference Time |
|--------------|---------------|------------|----------------|
| VinVL\textsubscript{large}[10] | 24            | 1024       | 542 ms         |
| M\textsuperscript{2} Transformer [5] | 3             | 512        | 174 ms         |
| GRIT         | 3             | 512        | 138 ms         |

B.5 Qualitative Examples

Figure 1, 2, 3, and 4 show some examples of the captions generated by our proposed method (GRIT) and another region-based method (M\textsuperscript{2} Transformer) given the same input images from the COCO test split. It is observed that the generated captions from GRIT are qualitatively better than those generated by the baseline method in terms of detecting and counting objects as well as describing their relationships in the given images. The inaccuracy of the captions generated by the baseline method might be due to the drawbacks of the region features extracted by a frozen pretrained object detector which produces wrong detection and lacks of contextual information.
GT-1: a child is brushing her hair in the mirror
GT-2: a little girl is brushing her hair in a bathroom
M2: a young girl holding a baseball bat in a
GRIT: a little girl brushing her hair with a brush

GT-1: 2 female tennis players standing with their rackets
GT-2: a pair of young women hold tennis balls and rackets
M2: a woman hitting a tennis ball with a tennis racket
GRIT: 2 people hold tennis rackets and balls on a court

GT-1: an elephant walking not too far from a rhino in a forest
GT-2: an elephant and a rhino share a field with a pond
M2: a group of elephants grazing in a field
GRIT: an elephant and a rhino standing in a field

GT-1: a bike is parked alongside the lake shore
GT-2: a bike is parked on the grass in front of the lake
M2: a bicycle leaning against a bridge over the water
GRIT: a bike parked next to a bridge on the water

GT-1: the boy is playing video games in his bedroom
GT-2: a young man is sitting in a chair playing a video game
M2: a young man sitting in a chair holding a wii remote
GRIT: a man sitting in a chair playing a video game

GT-1: a woman is taking a turkey out of the oven
GT-2: a woman is taking the cooked turkey out of the oven.
M2: a woman taking a pizza out of an oven with a
GRIT: a woman taking a turkey out of an oven with

GT-1: a giraffe standing outside of a building next to a tree.
GT-2: a giraffe standing in a small piece of shade.
M2: two giraffes are standing in a zoo enclosure
GRIT: a giraffe standing in the dirt next to a building

GT-1: bowls on a table with meat and vegetables.
GT-2: four plates of different kind of food sitting on a table
M2: three plates of food on a wooden table with a
GRIT: four bowls of food and a spoon on a table

Fig. 1: Qualitative examples from our method (GRIT) and a region-based method (M2 Transformer) on the COCO test images. Zoom in for better view.
Fig. 2: Qualitative examples from our method (GRIT) and a region-based method ($M^2$ Transformer) on the COCO test images. Zoom in for better view.
Fig. 3: Qualitative examples from our method (GRIT) and a region-based method ($M^2$ Transformer) on the COCO test images. Zoom in for better view.
Fig. 4: Qualitative examples from our method (GRIT) and a region-based method ($M^2$ Transformer) on the COCO test images. Zoom in for better view.
References

1. Achlioptas, P., Ovsjanikov, M., Haydarov, K., Elhoseiny, M., Guibas, L.J.: Artemis: Affective language for visual art. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11569–11579 (2021)
2. Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., Anderson, P.: nocaps: novel object captioning at scale. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8948–8957 (2019)
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: 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)
4. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: Computer vision and pattern recognition (CVPR), 2018 IEEE conference on. IEEE (2018)
5. Cornia, M., Stefanini, M., Baraldi, L., Cucchiara, R.: Meshed-memory transformer for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10578–10587 (2020)
6. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. arXiv preprint arXiv:2104.08718 (2021)
7. Jiang, H., Misra, I., Rohrbach, M., Learned-Miller, E., Chen, X.: In defense of grid features for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10267–10276 (2020)
8. Kim, W., Bokyung, S., Ildoo, K., Kim, W.: Vilt: Vision-and-language transformer without convolution or region supervision. In: International Conference on Machine Learning (2021)
9. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10012–10022 (2021)
10. Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., Gao, J.: Vinvl: Revisiting visual representations in vision-language models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5579–5588 (2021)
11. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: Proceedings of International Conference of Learning Representations (2021)