Multimodal Token Fusion for Vision Transformers

Yikai Wang[1] Xinghao Chen[2] Lele Cao[1] Wenbing Huang[3] Fuchun Sun[1]✉ Yunhe Wang[2]

[1]Beijing National Research Center for Information Science and Technology (BNRist),
State Key Lab on Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University
[2]Huawei Noah's Ark Lab [3]Institute for AI Industry Research (AIR), Tsinghua University

wangyk17@mails.tsinghua.edu.cn, xinghao.chen@huawei.com, caolele@gmail.com,
hwenbing@126.com, fuchuns@tsinghua.edu.cn, yunhe.wang@huawei.com

Abstract

Many adaptations of transformers have emerged to address the single-modal vision tasks, where self-attention modules are stacked to handle input sources like images. Intuitively, feeding multiple modalities of data to vision transformers could improve the performance, yet the inner-modal attentive weights may be diluted, which could thus greatly undermine the final performance. In this paper, we propose a multimodal token fusion method (TokenFusion), tailored for transformer-based vision tasks. To effectively fuse multiple modalities, TokenFusion dynamically detects uninformative tokens and substitute these tokens with projected and aggregated inter-modal features. Residual positional alignment is also adopted to enable explicit utilization of the inter-modal alignments after fusion. The design of TokenFusion allows the transformer to learn correlations among multimodal features, while the single-modal transformer architecture remains largely intact. Extensive experiments are conducted on a variety of homogeneous and heterogeneous modalities and demonstrate that TokenFusion surpasses state-of-the-art methods in three typical vision tasks: multimodal image-to-image translation, RGB-depth semantic segmentation, and 3D object detection with point cloud and images. Code will be released [1] [2].

1. Introduction

Transformer is initially widely studied in the natural language community as a non-recurrent sequence model [37] and it is soon extended to benefit vision-language tasks. Recently, numerous studies have further adopted transformers for computer vision tasks with well-adapted architectures and optimization schedules. As a result, vision transformer variants have shown great potential in many single-modal vision tasks, such as classification [5, 19], segmentation [41, 44], detection [2, 7, 20, 45], image generation [14].

Yet up until the date of this work, the attempt of extending vision transformers to handle multimodal data remains scarce. When multimodal data with complicated alignment relations are introduced, it poses great challenges in designing the fusion scheme for model architectures. The key question to answer is how and where the interaction of features from different modalities should take place. There have been a few methods for transformer-based vision-language fusion, *e.g.*, VL-BERT [35] and ViLT [15]. In these methods, vision and language tokens are directly concatenated before each transformer layer, making the overall architecture very similar to the original transformer. Such fusion is actually alignment-agnostic, which means the inter-modal alignments are not explicitly utilized. We also try to apply such intuitive fusion methods on multimodal vision tasks (Sec. 4). Unfortunately, this intuitive transformer fusion cannot bring promising gains or may even result in worse performance than the single-modal counterpart, which is mainly due to the fact that the inter-modal interaction is not fully exploited. There are also several attempts for fusing multiple vision modalities. For example, TransFuser [24] leverages transformer modules to connect CNN backbones of images and LiDAR points. However, these methods still neglect to find an effective and general method to insert inter-modal alignments into transformers.

We aim to benefit the learning process by multimodal data while also leveraging inter-modal alignments, which are naturally available in many vision tasks, *e.g.*, with camera intrinsics/extrinsics, a world-space point can be projected and correspond to a pixel on the camera plane. Unlike the alignment-agnostic fusion (will be described in Sec. 3.1), the alignment-aware fusion explicitly involves

✉ Corresponding author: Fuchun Sun.

[1]https://github.com/huawei-noah/noah-research

[2]https://gitee.com/mindspore/models/tree/master/research/cv/TokenFusion

the alignment relations of different modalities. Yet, since inter-modal projections are introduced to the transformer, alignment-aware fusion may greatly alter the original model structure and data flow, which potentially undermines the success of single-modal architecture designs or learned attention during pretraining. Thus, one may have to determine the "correct" layers/tokens/channels for multimodal projection and fusion, and also re-design the architecture or re-tune optimization settings for the new model. To avoid dealing with these challenging matters and inherit the majority of the original single-modal design, we propose multimodal token fusion, termed **TokenFusion**, which adaptively and effectively fuses multiple single-modal transformers.

The basic idea of our TokenFusion is to prune multiple single-modal transformers and then re-utilize pruned units for multimodal fusion. We apply individual pruning to each single-modal transformer and each pruned unit is substituted by projected alignment features from other modalities. This fusion scheme is assumed to have a limited impact on the original single-modal transformers, as it maintains the relative attention relations of the important units. TokenFusion also turns out to be superior in allowing multimodal transformers to inherit the parameters from single-modal pretraining, *e.g.*, on ImageNet.

To demonstrate the advantage of the proposed method, we consider extensive tasks including multimodal image translation, RGB-depth semantic segmentation, and 3D object detection based on images and point clouds, covering up to four public datasets and seven different modalities. TokenFusion obtains state-of-the-art performance on these extensive tasks, demonstrating its great effectiveness and generality. Specifically, TokenFusion achieves 64.9% and 70.8% mAP@0.25 for 3D object detection on the challenging SUN RGB-D and ScanNetV2 benchmarks, respectively.

2. Related Work

Transformers in computer vision. Transformer is originally designed for NLP research fields [37], which stacking multi-head self-attention and feed-forward MLP layers to capture the long-term correlation between words. Recently, vision transformer (ViT) [5] reveals the great potential of transformer-based models in large-scale image classification. As a result, transformer has soon achieved profound impacts in many other computer vision tasks such as segmentation [41, 44], detection [2, 7, 20, 45], image generation [14], video processing [18], etc.

Fusion for vision transformers. Deep fusion with multimodal data has been an essential topic which potentially boosts the performance by leveraging multiple sources of inputs, and it may also unleash the power of transformers further. Yet it is challenging to combine multiple off-the-rack single transformers while guaranteeing that such combination will not impact their elaborate singe-modal designs. [1] and [18] process consecutive video frames with transformers for spatial-temporal alignments and capturing fine-grained patterns by correlating multiple frames. Regarding multimodal data, [24, 38] utilize the dynamic property of transformer modules to combine CNN backbones for fusing infrared/visible images or LiDAR points. [8] extends the coarse-to-fine experience from CNN fusion methods to transformers for image processing tasks. [12] adopts transformers to combine hyperspectral images by the simple feature concatenation. [22] inserts intermediate tokens between image patches and audio spectrogram patches as bottlenecks to implicitly learn inter-modal alignments. These works, however, differ from ours since we would like to build a general fusion pipeline for combing off-the-rack vision transformers without the need of re-designing their structures or re-tuning their optimization settings, while explicitly leveraging inter-modal alignment relations.

3. Methodology

This part intends to provide a full landscape of the proposed methodology. We first introduce two naïve multimodal fusion methods for vision transformers in Sec. 3.1. Given the limitations of both intuitive methods, we then propose multimodal token fusion in Sec. 3.2. We elaborate the fusion designs for both homogeneous and heterogeneous modalities to evaluate the effectiveness and generality of our method in Sec. 3.4 and Sec. 3.5, respectively.

3.1. Basic Fusion for Vision Transformers

Suppose we have the i-th input data $\boldsymbol{x}^{(i)}$ that contains M modalities: $\boldsymbol{x}^{(i)} = \{\boldsymbol{x}_m^{(i)} \in \mathbb{R}^{N\times C}\}_{m=1}^{M}$, where N and C denote the number of tokens and input channels respectively. For simplicity, we will omit the subscript $^{(i)}$ in the upcoming sections. The goal of deep multimodal fusion is to determine a multi-layer model $f(\boldsymbol{x})$, and its output is expected to close to the target $\boldsymbol{y}$ as much as possible. Specifically in this work, $f(\boldsymbol{x})$ is approximated by a transformer-based network architecture. Suppose the model contains L layers in total, we represent the input token feature of the l-th layer ($l = 1, \ldots, L$) as $\boldsymbol{e}^l = \{\boldsymbol{e}_m^l \in \mathbb{R}^{N\times C'}\}_{m=1}^{M}$, where C' denotes the number of feature channels of the layer in scope. Initially, $\boldsymbol{e}_m^1$ is obtained using a linear projection of $\boldsymbol{x}_m$, which is a widely adopted approach to vectorize the input tokens (*e.g.* image patches), so that the first transformer layer can accept tokens as input.

We use different transformers for input modalities and denote $f_m(\boldsymbol{x}) = \boldsymbol{e}_m^{L+1}$ as the final prediction of the m-th transformer. Given the token feature $\boldsymbol{e}_m^l$ of the m-th modality, the l-th layer computes

$$\hat{\boldsymbol{e}}_m^l = \mathrm{MSA}\big(\mathrm{LN}(\boldsymbol{e}_m^l)\big),\ \boldsymbol{e}_m^{l+1} = \mathrm{MLP}\big(\mathrm{LN}(\hat{\boldsymbol{e}}_m^l)\big), \quad (1)$$

where MSA, MLP, and LN denote the multi-head self-attention, multi-layer perception, and layer normalization,

receptively. $\hat{\boldsymbol{e}}_m^l$ represents the output of MSA.

During multimodal fusion for vision tasks, the alignment relations of different modalities may be explicitly available. For example, pixel positions are often used to determine the image-depth correlation; and camera intrinsics/extrinsics are important in projecting 3D points to images. Based on the involvement of alignment information, we consider two kinds of transformer fusion methods as below.

Alignment-agnostic fusion does not explicitly use the alignment relations among modalities. It expects the alignment may be implicitly learned from large amount of data. A common method of the alignment-agnostic fusion is to directly concatenate multimodal input tokens, which is widely applied in vision-language models. Similarly, the input feature $\boldsymbol{e}_l$ for the l-th layer is also the token-wise concatenation of different modalities. Although the alignment-agnostic fusion is simple and may have minimal modification to the original transformer model, it is hard to directly benefit from the known multimodal alignment relations.

Alignment-aware fusion explicitly utilizes inter-modal alignments. For instance, this can be achieved by selecting tokens that correspond to the same pixel or 3D coordinate. Suppose $\boldsymbol{x}_m[n]$ is the n-th token of the m-th modality input $\boldsymbol{x}_m$, where $n = 1, \cdots, N_m$. We define the "token projection" from the m-th modality to the m'-th modality as

$$\mathrm{Proj}_{m'}^{\mathrm{T}}(\boldsymbol{x}_m[n_m]) = h(\boldsymbol{x}_{m'}[n_{m'}]), \tag{2}$$

where h could simply be an identity function (for homogeneous modalities) or a shallow multi-layer perception (for heterogeneous modalities). And when considering the entire N tokens, we can conveniently define the "modality projection" as the concatenation of token projections:

$$\mathrm{Proj}_{m'}^{\mathrm{M}}(\boldsymbol{x}_m) = \big[\mathrm{Proj}_{m'}^{\mathrm{T}}(\boldsymbol{x}_m[1]); \cdots; \mathrm{Proj}_{m'}^{\mathrm{T}}(\boldsymbol{x}_m[N])\big]. \tag{3}$$

Eq. (3) only depicts the fusion strategy on the input side. We can also perform middle-layer or multi-layer fusion across different modality-specific models, by projecting and aggregating feature embeddings $\boldsymbol{e}_m$ which possibly enables more diversified and accurate feature interactions. However, with the growing complexity of transformer-based models, searching for optimal fusion strategies (*e.g.* layers and tokens to apply projection and aggregation) for merely two modalities (*e.g.* 2D and 3D detection transformers) can grow into an extremely hard problem to solve. To tackle this issue, we propose multimodal token fusion in Sec. 3.2.

3.2. Multimodal Token Fusion

As described in Sec. 1, multimodal token fusion (TokenFusion) first prunes single-modal transformers and further re-utilizes the pruned units for fusion. In this way, the informative units of original single-modal transformers are assumed to be preserved to a large extent, while multimodal interactions could be involved for boosting performance.

As previously shown in [30], tokens of vision transformers could be pruned in a hierarchical manner while maintaining the performance. Similarly, we can select less informative tokens by adopting a scoring function $s^l(\boldsymbol{e}^l) = \mathrm{MLP}(\boldsymbol{e}^l) \in [0, 1]^N$, which dynamically predicts the importance of tokens for the l-th layer and the m-th modality. To enable the back propagation on $s^l(\boldsymbol{e}^l)$, we re-formulate the MSA output $\hat{\boldsymbol{e}}_m^l$ in Eq. (1) as

$$\hat{\boldsymbol{e}}_m^l = \mathrm{MSA}\big(\mathrm{LN}(\boldsymbol{e}_m^l) \cdot s^l(\boldsymbol{e}_m^l)\big). \tag{4}$$

We use $\mathcal{L}_m$ to denote the task-specific loss for the m-th modality. To prune uninformative tokens, we further add a token-wise pruning loss (an l_1-norm) on $s^l(\boldsymbol{e}_m^l)$. Thus the overall loss function for optimization is derived as

$$\mathcal{L} = \sum_{m=1}^{M} \Big(\mathcal{L}_m + \lambda \sum_{l=1}^{L} \big|s^l(\boldsymbol{e}_m^l)\big|\Big), \tag{5}$$

where λ is a hyper-parameter for balancing different losses.

For the feature $\boldsymbol{e}_m^l \in \mathbb{R}^{N \times C'}$, token-wise pruning dynamically detects unimportant tokens from all N tokens. Mutating unimportant tokens or substituting them with other embeddings are expected to have limited impacts on other informative tokens. We thus propose a token fusion process for multimodal transformers, which substitute unimportant tokens with their token projections (defined in Sec. 3.1) from other modalities. Since the pruning process is dynamic, *i.e.*, conditioned on the input features, the fusion process is also dynamic. This process performs token substitution before each transformer layer, thus the input feature of the l-th layer, *i.e.*, $\boldsymbol{e}_m^l$, is re-formulated as

$$\boldsymbol{e}_m^l = \boldsymbol{e}_m^l \odot \mathbb{I}_{s^l(\boldsymbol{e}_m^l) \geq \theta} + \mathrm{Proj}_{m'}^{\mathrm{M}}(\boldsymbol{e}_m^l) \odot \mathbb{I}_{s^l(\boldsymbol{e}_m^l) < \theta}, \tag{6}$$

where $\mathbb{I}$ is an indicator asserting the subscript condition, therefore it outputs a mask tensor $\in \{0, 1\}^N$; the parameter θ is a small threshold (we adopt 10^{-2} in our experiments); and the operator $\odot$ resents the element-wise multiplication.

In Eq. (6), if there are only two modalities as input, m' will simply be the other modality other than m. With more than two modalities, we pre-allocate the tokens into $M - 1$ parts, each of which is bound with one of the other modalities than itself. More details of this pre-allocation will be described in Sec. 3.4.

3.3. Residual Positional Alignment

Directly substituting tokens will risk completely undermining their original positional information. Hence, the model can still be ignorant of the alignment of the projected features from another modality. To mitigate this problem, we propose the method of Residual Positional Alignment (RPA) that leverages Positional Embeddings (PEs) for multimodal alignment.

As illustrated in Fig. 1 and Fig. 2 which will be detailed later, the key idea of RPA lies in injecting equivalent PEs to

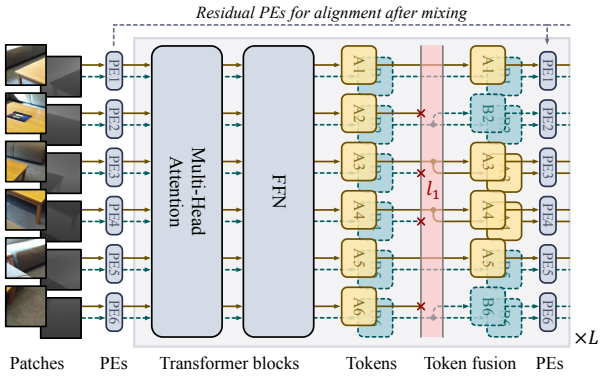

Figure 1. Framework of TokenFusion for homogeneous modalities with RGB and depth as an example. Both modalities are sent to a shared transformer with also shared positional embeddings.

subsequent layers. Moreover, the back propagation of PEs stops after the first layer, which means only the gradients of PEs at the first layer are retained while for the rest of the layers are frozen throughout the training. In this way, PEs serve a purpose of aligning multimodal tokens despite the substitution status of the original token. In summary, even if a token is substituted, we still reserve its original PEs that are added to the projected feature from another modality.

3.4. Homogeneous Modalities

In the common setup of either a generation task (multimodal image-to-image translation) or a regression task (RGB-depth semantic segmentation), the homogeneous vision modalities $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_M$ are typically aligned with pixels, such that the pixels located at the same position in RGB or depth input should share the same label. We also expect that such property allows the transformer-based models to benefit from joint learning. Hence, we adopt shared parameters in both MSA and MLP layers for different modalities; yet rely on modality-specific layer normalizations to uncouple the normalization process, since different modalities may vary drastically in their statistical means and variances by nature. In this scenario, we simply set function h in Eq. (6) as an identity function, and we also let $n_{m'} = n_m$, which means we always substitute each pruned token with the token sharing the same position.

An overall illustration of TokenFusion for fusing homogeneous modalities is depicted in Fig. 1. Regarding two input modalities, we adopt bi-directional projection and apply token-wise pruning on both modalities respectively. Then the token substitution process is performed according to Eq. (6). When there are $M > 2$ modalities, we also apply the token-wise pruning on all modalities with an additional pre-allocation strategy that selects m' in based on m according to Eq. (6). To be specific, for the m-th modality, we randomly pre-allocate N tokens into $M-1$ groups with equal group sizes. This pre-allocation is carried out prior to the commence of training procedure, and the obtained groups will be fixed throughout the training. We denote the group allocation as $\boldsymbol{a}_{m'}(m) \in \{0,1\}^N$, where $\boldsymbol{a}_{m'}(m)[n] = 1$ indicates that if the n-th token of the m-th modaltity is pruned, it will be substituted by the corresponding token of the m'-th modality, otherwise $\boldsymbol{a}_{m'}(m)[n] = 0$. Having obtained the pre-allocation strategy for $M > 2$ modalties, Eq. (6) can be further developed into a more specific form:

$$\boldsymbol{e}_m^l = \boldsymbol{e}_m^l \odot \mathbb{I}_{s^l(\boldsymbol{e}_m^l) \geq \theta} + \sum_{\substack{m'=1 \\ m' \neq m}}^{M} \boldsymbol{a}_{m'}(m) \odot \mathrm{Proj}_{m'}^{\mathrm{M}}(\boldsymbol{e}_m^l) \odot \mathbb{I}_{s^l(\boldsymbol{e}_m^l) < \theta}. \quad (7)$$

3.5. Heterogeneous Modalities

In this section, we further explore how TokenFusion handles heterogeneous modalities, in which input modalities exhibit quite different data formats and large structural discrepancies, *e.g.*, different number of layers or embedding dimensions for the transformer architectures. A concrete example would be to learn 3D object detection (based on point cloud) and 2D object detection (based on images) simultaneously with different transformers. Although there are already specific transformer-based models designed for 3D or 2D object detection respectively, there still lacks a fast and effective method to combine these models and tasks.

An overall structure of TokenFusion for fusing heterogeneous modalities is depicted in Fig. 2. Different from the homogeneous case, we approximate the token projection function h in Eq. (2) with a shallow multi-layer perception (MLP), since transformers for these heterogeneous modalities may have different hidden embedding dimensions. For the case of 3D object detection with 3D point cloud and 2D image, we project each point to the corresponding image based on camera intrinsics and extrinsics. Likewise, we also project 3D object labels to the images for obtaining the corresponding 2D object labels. We train two standalone transformers with unshared parameters in an end-to-end manner. Regarding the 3D object detection with point cloud as input, we follow the architecture used in Group-Free [20], where N_{point} sampled seed points and K_{point} learned proposal points are considered as input tokens, which are sent to the transformer for predicting K_{point} 3D bounding boxes and object categories. For the 2D object detection with images as input, we follow the framework in YOLOS [7] which sends N_{img} image patches and K_{img} object queries to the transformer to predict K_{img} 2D bounding boxes together with their associated object categories.

The inter-modal projection maps seed points to image patches, *i.e.*, an N_{point}-to-N_{img} mapping. Specifically, the token-wise pruning is applied on the N_{point} seed point tokens. Once a certain token obtains a low importance score,

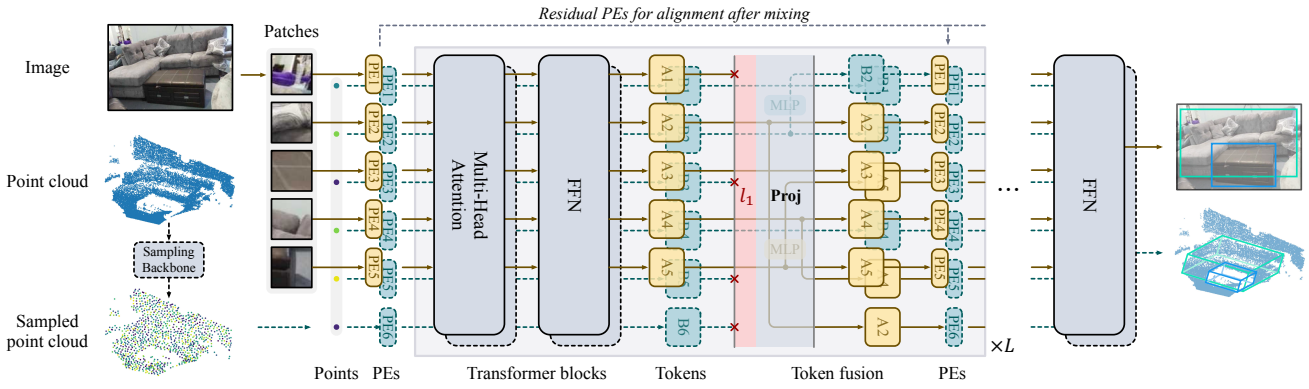

Figure 2. Framework of TokenFusion for heterogeneous modalities with point clouds and images. Both modalities are sent to individual transformer modules with also individual positional embeddings. Additional inter-modal projections (Proj) are needed which is different from the fusion for homogeneous modalities.

we project the 3D coordinate of this token to a 2D pixel on the corresponding image input. It is now viable to locate the specific image patch based on the 2D pixel. Suppose this projection obtains the n_{img}-th image patch based on the n_{point}-th seed point which is pruned. We substitute m and m' in Eq. (2) with the subscripts "point" and "img" respectively, *i.e.*, $\text{Proj}^{\text{T}}_{\text{img}}(\boldsymbol{x}_{\text{point}}[n_{\text{point}}]) = h(\boldsymbol{x}_{\text{img}}[n_{\text{img}}])$. Thus the relation between n_{point} and n_{img} captured by the token projection satisfies

$$\left[u,\, v,\, z\right]^{\top} = \mathbf{K} \cdot \mathbf{R_t} \cdot \left[x_{n_{\text{point}}},\, y_{n_{\text{point}}},\, z_{n_{\text{point}}},\, 1\right]^{\top}, \quad (8)$$

$$n_{\text{img}} = \left\lfloor \frac{\lfloor v/z \rfloor}{P} \right\rfloor \times \left\lfloor \frac{W}{P} \right\rfloor + \left\lfloor \frac{\lfloor u/z \rfloor}{P} \right\rfloor, \quad (9)$$

where $\mathbf{K} \in \mathbb{R}^{4\times4}$ and $\mathbf{R_t} \in \mathbb{R}^{4\times4}$ are camera intrinsic and extrinsic matrices, respectively; $[x_{n_{\text{point}}},\, y_{n_{\text{point}}},\, z_{n_{\text{point}}}]$ denotes the 3D coordinate of the n_{point}-th point; u, v, z are temporary variables with $\left[\lfloor u/z \rfloor,\, \lfloor v/z \rfloor\right]$ actually being the projected pixel coordinate of the image; P is the patch size of the vision transformer and W denotes the image width.

4. Experiments

To evaluate the effectiveness of the proposed TokenFusion, we conduct comprehensive experiments towards both homogeneous and heterogeneous modalities with state-of-the-art (SOTA) methods. Experiments are conducted on totally seven different modalities and four application scenarios, implemented with PyTorch [23] and MindSpore [13].

4.1. Multimodal Image-to-Image Translation

The task of multimodal image-to-image translation aims at generating a target image modality based on different image modalities as input (*e.g.* Normal+Depth→RGB). We evaluate TokenFusion in this task using the Taskonomy [42] dataset, which is a large-scale indoor scene dataset containing about 4 million indoor images captured from 600 buildings. Taskonomy provides over 10 multimodal representations in addition to each RGB image, such as depth (euclidean or z-buffering), normal, shade, texture, edge, principal curvature, etc. The resolution of each representation is 512×512. To facilitate comparison with the existing fusion methods, we adopt the same sampling strategy as [39], resulting in 1,000 high-quality multimodal images for training, and 500 for validation.

Our implementation contains two transformers as the generator and discriminator respectively. We provide configuration details in our supplementary materials. The resolution of the generator/discriminator input or the generator prediction is 256×256. We adopt two kinds of architecture settings, the tiny (Ti) version with 10 layers and the small (S) version with 20 layers, and both settings are only different in layer numbers. The learning rates of both transformers are set to 2×10^{-4}. We adopt overlapped patches in both transformers inspired by [41].

In our experiments for this task, we adopt shared transformers for all input modalities with individual layer normalizations (LNs) that individually compute the means and variances of different modalities. Specifically, parameters in the linear projection on patches, all linear projections (*e.g.* for key, queries, etc) in MSA, and MLP are shared for different modalities. Such a mechanism largely reduces the total model size which as discussed in the supplementary materials, even achieves better performance than using individual transformers. In addition, we also adopt shared positional embeddings for different modalities. We let the sparsity weight $\lambda = 10^{-4}$ in Eq. (5) and the threshold $\theta = 2 \times 10^{-2}$ in Eq. (7) for all these experiments.

Our evaluation metrics include FID/KID for RGB predictions and MAE/MSE for other predictions. These metrics are introduced in the supplementary materials.

Results. In Table 1, we provide comparisons with extensive baseline methods and a SOTA method [39] with the

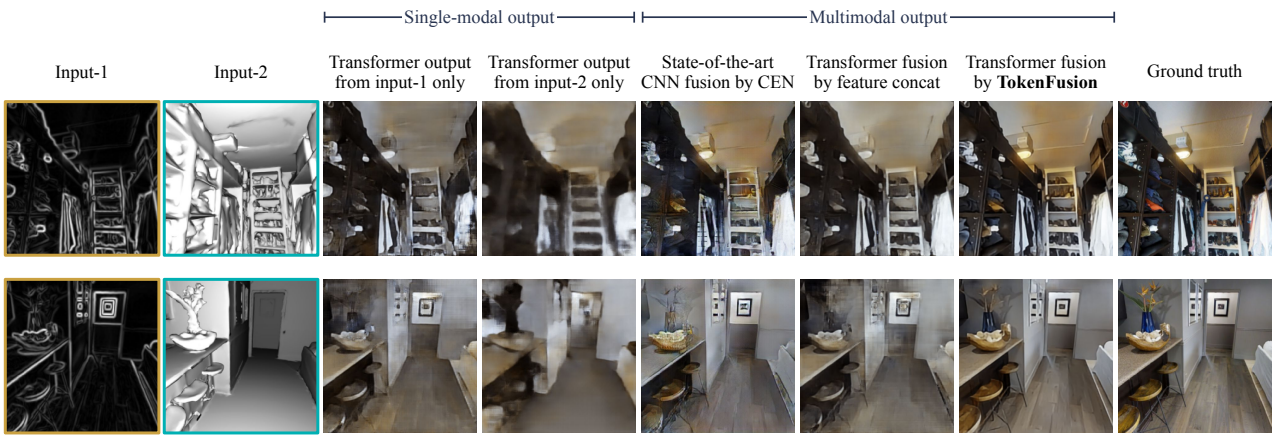

Figure 3. Comparison on the *validation* data split for image-to-image translation (Texture+Shade→RGB). The resolution of all input/output images is 256×256. The third/forth column is predicted by the single modality, and the following three columns are predicted by CEN [39], the intuitive transformer fusion by feature concatenation, and our TokenFusion, respectively. Best view in color and zoom in.

same data settings. All methods adopt the learned ensemble over the two predictions which are corresponded to the two modality branches. In addition, all predictions have the same resolution 256×256 for a fair comparison. Since most existing methods are based on CNNs, we further provide two baselines for transformer-based models including the baseline without feature fusion (only uses ensemble for the late fusion) and the feature fusion method. By comparison, our TokenFusion surpasses all the other methods with large margins. For example, in the Shade+Texture→RGB task, our TokenFusion (S) achieves $43.92/0.94$ FID/KID scores, remarkably better than the current SOTA method CEN [39] with 29.8% relative FID metric decrease.

In supplementary materials, we consider more modality inputs up to 4 which evaluates our group allocation strategy.

Visualization and analysis. We provide qualitative results in Fig. 3, where we choose tough samples for comparison. The predictions with our TokenFusion obtain better natural patterns and are also richer in colors and details. In Fig. 4, we further visualize the process of TokenFusion of which tokens are learned to be fused under our l_1 sparsity constraints. We observe that the tokens for fusion follow specific regularities. For example, the texture modality tends to preserve its advantage of detailed boundaries, and meanwhile seek facial tokens from the shade modality. In this sense, TokenFusion combines complementary properties of different modalities.

4.2. RGB-Depth Semantic Segmentation

We then evaluate TokenFusion on another homogeneous scenario, semantic segmentation with RGB and depth as input, which is a very common multimodal task and numerous methods have been proposed towards better performance. We choose the typical indoor datasets, NYUDv2 [31] and SUN RGB-D [32]. For NYUDv2, we follow the standard

Method	Shade+Texture →RGB	Depth+Normal →RGB	RGB+Shade →Normal	RGB+Normal →Shade	RGB+Edge →Depth
CNN-based models					
Concat [39]	78.82/3.13	99.08/4.28	1.34/2.85	1.28/2.02	0.33/0.75
Self-Att. [36, 39]	73.87/2.46	96.73/3.95	1.26/2.76	1.18/1.76	0.30/0.70
Align. [34, 39]	92.30/4.20	105.03/4.91	1.52/3.25	1.41/2.21	0.45/0.90
CEN [39]	62.63/1.65	84.33/2.70	1.12/2.51	1.10/1.72	0.28/0.66
Transformer-based models					
Concat (Ti)	76.13/2.85	102.70/4.74	1.52/3.15	1.33/2.20	0.40/0.83
Ours (Ti)	50.40/1.03	76.35/2.19	0.73/1.83	0.95/1.54	0.21/0.57
Concat (S)	72.55/2.39	96.04/4.09	1.18/2.73	1.30/2.07	0.35/0.68
Ours (S)	**43.92/0.94**	**70.13/1.92**	**0.58/1.51**	**0.79/1.33**	**0.16/0.47**

Table 1. Results on Taskonomy for multimodal image-to-image translation. Evaluation metrics are FID/KID ($\times10^{-2}$) for RGB predictions and MAE ($\times10^{-1}$)/MSE ($\times10^{-1}$) for other predictions. Lower values indicate better performance for all the metrics.

795/654 images for train/test splits to predict the standard 40 classes [9]. SUN RGB-D is one of the most challenging large-scale indoor datasets, and we adopt the standard 5,285/5,050 images for train/test of 37 semantic classes.

Our models include TokenFusion (tiny) and TokenFusion (small), of which the single-modal backbones follow the B1 and B2 settings of SegFormer [41]. Both tiny and small versions adopt the pretrained parameters on ImageNet-1k for initialization following [41]. Similar to our implementation in Sec. 4.1, we also adopt shared transformers and positional embeddings for RGB and depth inputs with individual LNs. We let the sparsity weight $\lambda = 10^{-3}$ in Eq. (5) and the threshold $\theta = 2\times10^{-2}$ in Eq. (7) for all these experiments.

Results. Results provided in Table 2 conclude that current transformer-based models equipped with our TokenFusion surpass SOTA models using CNNs. Note that we choose relatively light backbone settings (B1 and B2 as mentioned in Sec. 4.2). We expect that using larger backbones (*e.g.*, B5) would yield better performance.

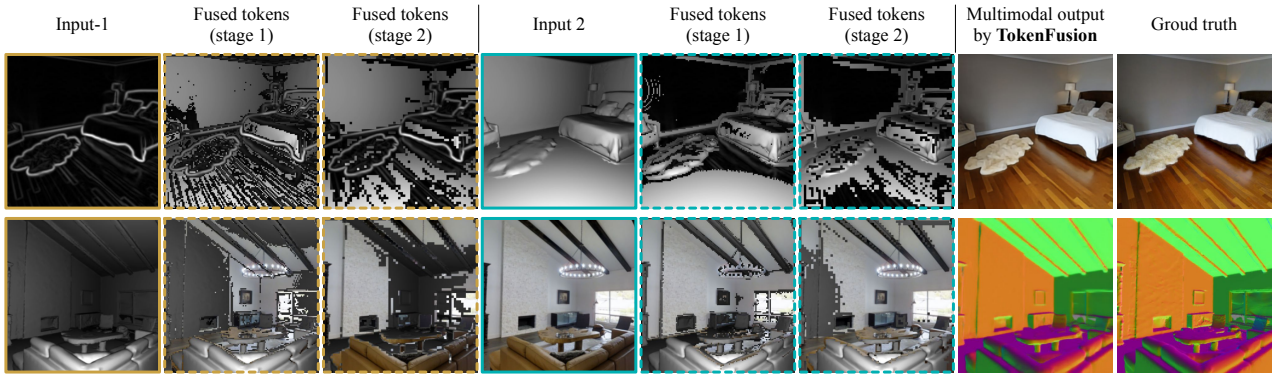

Figure 4. Illustrations of which tokens are fused in our TokenFusion, performed on the *validation* data split. We provide two cases including Texture+Shade→RGB (first row) and Shade+RGB→Normal (second row). The resolution of all images is 256×256. We choose the last layers in the first and second transformer stages respectively. Best view in color and zoom in.

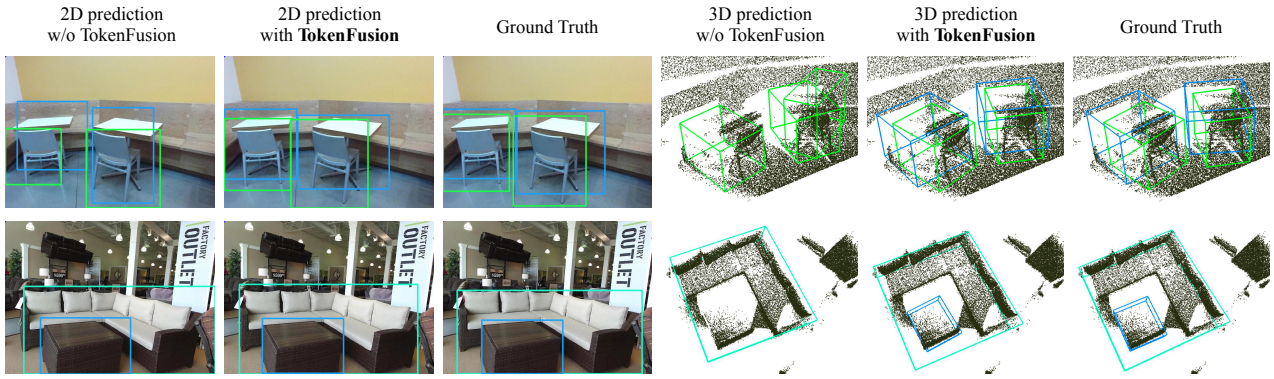

Figure 5. Results visualization on the *validation* data split for heterogeneous modalities including point clouds and images, where 3D object detection and 2D object detection are learned simultaneously. We compare the performance without (w/o) or with our TokenFusion. Our TokenFusion mainly benefits 3D object detection results.

Method	Inputs	NYUDv2			SUN RGB-D		
		Pixel Acc.	mAcc.	mIoU	Pixel Acc.	mAcc.	mIoU
CNN-based models							
FCN-32s [21]	RGB	60.0	42.2	29.2	68.4	41.1	29.0
RefineNet [17]	RGB	74.4	59.6	47.6	81.1	57.7	47.0
FuseNet [11]	RGB+D	68.1	50.4	37.9	76.3	48.3	37.3
SSMA [36]	RGB+D	75.2	60.5	48.7	81.0	58.1	45.7
RDFNet [16]	RGB+D	76.0	62.8	50.1	81.5	60.1	47.7
AsymFusion [40]	RGB+D	77.0	64.0	51.2	-	-	-
CEN [39]	RGB+D	77.7	65.0	52.5	83.5	63.2	51.1
Transformer-based models							
w/o fusion (Ti)	RGB	75.2	62.5	49.7	82.3	60.6	47.0
Concat (Ti)	RGB+D	76.5	63.4	50.8	82.8	61.4	47.9
Ours (Ti)	RGB+D	78.6	66.2	53.3	84.0	63.3	51.4
w/o fusion (S)	RGB	76.0	63.0	50.6	82.9	61.3	48.1
Concat (S)	RGB+D	77.1	63.8	51.4	83.5	62.0	49.0
Ours (S)	RGB+D	**79.0**	**66.9**	**54.2**	**84.7**	**64.1**	**53.0**

Table 2. Comparison results on the NYUDv2 and SUN RGB-D datasets with SOTAs for RGB and depth (D) semantic segmentation. Evaluation metrics include pixel accuracy (Pixel Acc.) (%), mean accuracy (mAcc.) (%), and mean IoU (mIoU) (%).

4.3. Vision and Point Cloud 3D Object Detection

We further apply TokenFusion for fusing heterogeneous modalities, specifically, the 3D object detection task which has received great attention. We leverage 3D point clouds and 2D images to learn 3D and 2D detections, respectively, and both processes are learned simultaneously. We expect the involvement of 2D learning boosts the 3D counterpart.

We adopt SUN RGB-D [33] and ScanNetV2 [4] datasets. For SUN RGB-D, we follow the same train/test splits as in Sec. 4.2 and detect the 10 most common classes. For ScanNetV2, we adopt the 1,201/312 scans as train/test splits to detect the 18 object classes. All these settings (splits and detected target classes) follow current works [20, 26] for a fair comparison. Note that different from SUN RGB-D, ScanNetV2 provides multi-view images for each scene alongside the point cloud. We randomly sample 10 frames per scene from the scannet-frames-25k samples provided in [4].

Our architectures for 3D detection and 2D detection follow GF [20] and YOLOS [7], respectively. We adopt the "L6, O256" or "L12, O512" versions of GF for the 3D detection branch. We combine GF with the tiny (Ti) and small (S) versions of YOLOS, respectively, and adopt mAP@0.25 and mAP@0.5 as evaluation metrics following [20, 26].

Results. We provide results comparison in Table 3 and

Method	Backbone	Inputs	mAP@0.25	mAP@0.5
CNN-based models				
VoteNet [27]	PointNet++	Points	59.1	35.8
VoteNet [27]*	PointNet++	Points+RGB	58.0	34.3
MLCVNet [29]	PointNet++	Points	59.8	-
HGNet [3]	GU-net	Points	60.1	39.0
H3DNet [43]	4×PointNet++	Points	61.6	-
imVoteNet [25]	PointNet++	Points+RGB	63.4	-
Transformer-based models				
GF [20] (L6, O256)	PointNet++	Points	63.0 (62.6)	45.2 (44.4)
GF [20] (L6, O256)*	PointNet++	Points+RGB	62.1 (61.0)	42.7 (41.9)
Ours (L6, O256; Ti)	PointNet++	Points+RGB	64.5 (64.2)	47.8 (47.3)
Ours (L6, O256; S)	PointNet++	Points+RGB	**64.9** (64.4)	**48.3** (47.7)

Table 3. Comparison on SUN RGB-D with SOTAs for 3D object detection, including best results and average results in brackets. * indicates appending RGB to the points as described in Sec. 4.3.

Method	Backbone	Inputs	mAP@0.25	mAP@0.5
CNN-based models				
GSDN [10]	MinkNet	Points	62.8	34.8
3D-MPA [6]	MinkNet	Points	64.2	49.2
VoteNet [27]	PointNet++	Points	62.9	39.9
MLCVNet [29]	PointNet++	Points	64.5	41.4
H3DNet [43]	PointNet++	Points	64.4	43.4
H3DNet [43]	4×PointNet++	Points	67.2	48.1
Transformer-based models				
GF [20] (L6, O256)	PointNet++	Points	67.3 (66.3)	48.9 (48.5)
GF [20] (L6, O256)*	PointNet++	Points+RGB	66.3 (65.7)	47.5 (47.0)
GF [20] (L12, O512)	PointNet++w2×	Points	69.1 (68.6)	52.8 (51.8)
GF [20] (L12, O512)*	PointNet++w2×	Points+RGB	68.2 (67.6)	50.3 (49.4)
Ours (L6, O256; Ti)	PointNet++	Points+RGB	68.8 (68.0)	51.9 (51.2)
Ours (L12, O512; S)	PointNet++w2×	Points+RGB	**70.8** (69.8)	**54.2** (53.6)

Table 4. Comparison on ScanNetV2 with SOTAs for 3D object detection, including best results and average results in brackets.

Table 4. The main comparison is based on the best results of five experiments between different methods, and numbers within the brackets are the average results. Besides, we perform intuitive multimodal experiments by appending the 3-channel RGB vectors to the sampled points after PointNet++ [28]. Such intuitive experiments are marked by the subscript * in both tables. We observe, however, that simply appending RGB information even leads to the performance drop, indicating the difficulty of such a heterogeneous fusion task. By comparison, our TokenFusion achieves new records on both datasets, which are remarkably superior to previous CNN/transformer models in terms of both metrics. For example, with TokenFusion, YOLOS-Ti can be utilized to boost the performance of GF by further 2.4 mAP@0.25 improvements, and using YOLOS-S brings further gains.

Visualizations. Fig. 5 illustrates the comparison of detection results when using TokenFusion for multimodal interactions against individual learning. We observe that TokenFusion benefits the 3D detection part. For example, with the help of images, models with TokenFusion can locate 3D objects even with sparse or missing point data (second row). In addition, using images also benefits when the points of two objects are largely overlapped (first row). These observations demonstrate the advantages of our TokenFusion.

l_1-norm	Fusion strategy	Seg. (NYUDv2) Pixel Acc.	mAcc.	mIoU	3D det. (SUN RGB-D) mAP@0.25	mAP@0.5
×	×	75.2	62.5	49.7	62.8	45.1
×	Random (10%)	75.6	63.0	50.1	62.3	44.5
×	Random (30%)	74.2	61.0	48.2	59.5	42.4
✓	×	75.0	62.5	49.5	62.6	44.9
✓	✓(with RPA)	78.6	66.2	53.3	64.9	48.3

Table 5. Effectiveness of l_1-norm and token fusion. Experiments include RGB-depth segmentation (seg.) on NYUDv2 and 3D detection (det.) with images and points on SUN RGB-D.

Token fusion (with l_1-norm)	RPA	Seg. (NYUDv2) Pixel Acc.	mAcc.	mIoU	3D det. (SUN RGB-D) mAP@0.25	mAP@0.5
×	×	75.2	62.5	49.7	62.8	45.1
×	✓	75.7	62.9	50.3	63.0	45.3
✓	×	78.3	65.8	52.9	63.6	46.2
✓	✓	78.6	66.2	53.3	64.9	48.3

Table 6. Effectiveness of RPA proposed in Sec. 3.4. Experimental tasks and datasets follow Table 5.

5. Ablation Study

l_1-norm and token fusion. In Table 5, we demonstrate the advantages of l_1-norm and token fusion. We additionally conduct experiments with random token fusion. We observe that applying l_1-norm itself has little effect on the performance yet it is essential to reveal tokens for fusion. Our token fusion together with l_1-norm achieves much better performance than the random fusion baselines.

Evaluation of RPA. Table 6 evaluates RPA proposed in Sec. 3.3. Results indicate that only using RPA without token fusion does not noticeably affect the performance, but is important when combined with the token fusion process for alignments, especially for the 3D detection task.

6. Conclusion

We propose TokenFusion, an adaptive method generally applicable for fusing vision transformers with homogeneous or heterogeneous modalities. TokenFusion exploits uninformative tokens and re-utilizes them to strengthen the interaction of other informative multimodal tokens. Alignment relations of different modalities are explicitly utilized due to the residual positional alignment and inter-modal projection. TokenFusion surpasses state-of-the-art methods on a variety of tasks, demonstrating its superiority and generality for multimodal fusion.

Acknowledgement

This work is funded by Major Project of the New Generation of Artificial Intelligence (No. 2018AAA0102900), the Sino-German Collaborative Research Project Crossmodal Learning (NSFC 62061136001/DFG TRR169), and National Natural Science Foundation of China (No. 62006137). We acknowledge the support of MindSpore, CANN and Ascend AI Processor.

References

[1] Aljaz Bozic, Pablo R. Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular RGB scene reconstruction using transformers. In *NeurIPS*, 2021. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2

[3] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *CVPR*, 2020. 8

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 7

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 2

[6] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *CVPR*, 2020. 8

[7] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *arXiv preprint arXiv:2106.00666*, 2021. 1, 2, 4, 7

[8] Yu Fu, TianYang Xu, XiaoJun Wu, and Josef Kittler. Ppt fusion: Pyramid patch transformerfor a case study in image fusion. *arXiv preprint arXiv:2107.13967*, 2021. 2

[9] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *CVPR*, 2013. 6

[10] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3d single-shot object detection. *arXiv preprint arXiv:2006.12356*, 2020. 8

[11] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In *ACCV*, 2016. 7

[12] Jin-Fan Hu, Ting-Zhu Huang, and Liang-Jian Deng. Fusformer: A transformer-based fusion approach for hyperspectral image super-resolution. *arXiv preprint arXiv:2109.02079*, 2021. 2

[13] Huawei. Mindspore. https://www.mindspore.cn/, 2020. 5

[14] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. In *NeurIPS*, 2021. 1, 2

[15] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*, 2021. 1

[16] Seungyong Lee, Seong-Jin Park, and Ki-Sang Hong. Rdfnet: RGB-D multi-level residual feature fusion for indoor semantic segmentation. In *ICCV*, 2017. 7

[17] Guosheng Lin, Fayao Liu, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for dense prediction. In *IEEE Trans. PAMI*, 2019. 7

[18] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *ICCV*, 2021. 2

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[20] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *ICCV*, 2021. 1, 2, 4, 7, 8

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 7

[22] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *arXiv preprint arXiv:2107.00135*, 2021. 2

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[24] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021. 1, 2

[25] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *CVPR*, 2020. 8

[26] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 7

[27] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 8

[28] Charles R Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 8

[29] Xie Qian, Lai Yu-kun, Wu Jing, Wang Zhoutao, Zhang Yiming, Xu Kai, and Wang Jun. Mlcvnet: Multi-level context votenet for 3d object detection. In *CVPR*, 2020. 8

[30] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 3

[31] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 6

[32] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 6

[33] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 7

[34] Sijie Song, Jiaying Liu, Yanghao Li, and Zongming Guo. Modality compensation network: Cross-modal adaptation for action recognition. In *IEEE Trans. Image Process.*, 2020. 6

[35] 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 *ICLR*, 2019. 1

[36] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. In *IJCV*, 2020. 6, 7

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 1, 2

[38] Vibashan VS, Jeya Maria Jose Valanarasu, Poojan Oza, and Vishal M Patel. Image fusion transformer. *arXiv preprint arXiv:2107.09011*, 2021. 2

[39] Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. Deep multimodal fusion by channel exchanging. In *NeurIPS*, 2020. 5, 6, 7

[40] Yikai Wang, Fuchun Sun, Ming Lu, and Anbang Yao. Learning deep multimodal feature representation with asymmetric multi-layer fusion. In *ACM MM*, 2020. 7

[41] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 1, 2, 5, 6

[42] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 5

[43] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. *arXiv preprint arXiv:2006.05682*, 2020. 8

[44] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 1, 2

[45] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 2