Channel Exchanging Networks for Multimodal and Multitask Dense Image Prediction

Yikai Wang, Fuchun Sun, Wenbing Huang, Fengxiang He, Dacheng Tao

Abstract—Multimodal fusion and multitask learning are two vital topics in machine learning. Despite the fruitful progress, existing methods for both problems are still brittle to the same challenge—it remains dilemmatic to integrate the common information across modalities (resp. tasks) meanwhile preserving the specific patterns of each modality (resp. task). Besides, while they are actually closely related to each other, multimodal fusion and multitask learning are rarely explored within the same methodological framework before. In this paper, we propose Channel-Exchanging-Network (CEN) which is self-adaptive, parameter-free, and more importantly, applicable for multimodal and multitask dense image prediction. At its core, CEN adaptively exchanges channels between subnetworks of different modalities. Specifically, the channel exchanging process is self-guided by individual channel importance that is measured by the magnitude of Batch-Normalization (BN) scaling factor during training. For the application of dense image prediction, the validity of CEN is tested by four different scenarios: multimodal fusion, cycle multimodal fusion, multitask learning, and multimodal multitask learning. Extensive experiments on semantic segmentation via RGB-D data and image translation through multi-domain input verify the effectiveness of CEN compared to state-of-the-art methods. Detailed ablation studies have also been carried out, which demonstrate the advantage of each component we propose. Our code is available at https://github.com/yikaiw/CEN.

Index Terms—Multimodal Fusion, Multitask Learning, Channel Exchanging, Semantic Segmentation, Image-to-Image Translation.

◆

1 INTRODUCTION

ENCOURAGED by the growing availability of low-cost sensors, *multimodal fusion* that takes advantage of multiple data sources for classification or regression becomes one of the central problems in machine learning [1]. Joining the success of deep learning, multimodal fusion is recently specified as *deep multimodal fusion* by introducing end-to-end neural integration of multiple modalities [2], and it has exhibited remarkable benefits against the unimodal paradigm in semantic segmentation [3], [4], action recognition [5], [6], [7], visual question answering [8], [9], and many others [10], [11], [12]. *Multitask learning* [13] is another crucial topic in machine learning. It aims to seek models to solve multiple tasks simultaneously, which enjoys the benefit of model generation and data efficiency against the methods that learn each task independently. Similar to multimodal fusion, multitask learning has also been developed from previously shallow methods [14] to deep variants [15], [16], [17], [18], [19] by taking advantage of deep learning. The successful applications of multitask learning include navigation [20], robot manipulation [21], etc.

In general, dense image prediction could be a collection of computer vision tasks that aim at classifying (*e.g.*, segmentation [3], [22], [23], [24]) or regressing (*e.g.*, image-to-image translation [25], [26], [27], [28]) every pixel in an image, namely, producing pixel-wise output based on the given input pixels. The learning pipeline for dense prediction is usually expected to capture rich spatial details or

strong semantics, which also benefits greatly from multimodal data sources or the multitask joint training. A variety of works tailored for dense image prediction have been done towards multimodal fusion and multitask learning. For multimodal fusion, regarding the type of how they fuse, existing methods are generally categorized into *aggregation-based* fusion [4], [29], [30], *alignment-based* fusion [7], [31], and the mixture of them [1]. As for multitask learning, in the context of deep learning, two types of contemporary techniques are identified: *hard parameter-sharing* [32], [33] and *soft parameter-sharing* [15], [34]. Despite the fruitful progress, existing methods for both problems are still brittle to the same challenge—it remains dilemmatic to integrate the common information across modalities (resp. tasks) meanwhile preserving the specific patterns of each modality (resp. task) for multimodal fusion (resp. multitask learning). To be more specific, for multimodal fusion, the aggregation-based fusion is prone to underestimating the intra-modal propagation, whereas the alignment-based fusion mostly delivers ineffective inter-modal fusion owing to the weak message exchanging by solely training alignment losses [30], [35], [36]. A similar issue exists in multitask learning. Current hard/soft parameter sharing schemes could be vulnerable to the negative transfer issue across different tasks owing to the insufficient balance between inter-task knowledge sharing and intra-task information processing [37]. When focusing on dense image prediction, multimodal fusion and multitask learning can also be regarded as the dual problem of each other. As will be described in § 3, multimodal fusion corresponds to the multiple-input-single-output problem while multitask learning, inversely, is of the single-input-multiple-output formulation. Yet, most previous works study these two problems separately without revealing their common property.

———————————————————

Y. Wang and F. Sun are with 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. W. Huang is with Gaoling School of Artificial Intelligence, Renmin University of China and Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China. F. He and D. Tao are with JD Explore Academy, JD.com Inc.

In this paper, we propose Channel-Exchanging-Network (CEN) which is self-adaptive, parameter-free, and applicable for multimodal and multitask dense image prediction. For unification, we refer to both the modality-specific network in multimodal fusion and the task-specific network in multitask learning as a subnetwork. To enable message passing among different modalities/tasks, CEN adaptively exchanges the channels between subnetworks. The core of CEN lies in its smaller-norm-less-informative assumption inspired by network pruning [38], [39]. To be specific, we utilize the scaling factor (*i.e.*, γ) of Batch-Normalization (BN) [40] or Instance-Normalization (IN) [41] as the importance measurement for each corresponding channel, and replace the channels associated with close-to-zero factors of each subnetwork with the mean of other subnetworks. Such message exchanging is self-adaptive in determining when to exchange, and hence it is capable of accomplishing better trade-off between inter-subnetwork knowledge sharing and intra-subnetwork information processing, in contrast to conventional multimodal and multitask learning methods. Further, the channel exchanging operation itself is parameter-free, making CEN less prone to overfitting, while, for example, the attention-based fusion [4] needs extra parameters to adjust the importance of each subnetwork. Another hallmark of CEN is that the encoder parameters except for BN layers of all subnetworks are shared with each other (§ 3.2). Apart from compacting the model size, we apply the idea here to serve specific purposes in CEN: by using private BNs, we can determine the channel importance for each individual modality; by sharing convolutional filters, the corresponding channels among different modalities are embedded with the same mapping, thus more capable of modelling the modality-common statistic.

CEN is generally powerful, capable of addressing four different problems in image dense prediction: multimodal fusion, cycle multimodal fusion, multitask learning, and multitask multitask learning. For multimodal fusion, we conduct channel exchanging on the encoder side to allow information integration between different input modalities. We also design cycle multimodal fusion to reuse the knowledge among different generation flows, which can promote performance for each flow. As natural extensions, channel exchanging could be applied to the decoder side or both the decoder and encoder to exchange task-specific information for multitask learning or for multimodal multitask learning. These details will be provided in § 3.

To sum up, our contributions are as follows:

- We propose CEN for message fusion, which is self-adaptive and parameter-free. The core of CEN is to replace the channels associated with close-to-zero BN or IN scaling factors of each subnetwork with the mean of others.
- CEN is generally powerful and is applied to multimodal fusion, cycle multimodal fusion, multitask learning, and multimodal multitask learning. To the best of our knowledge, it is the first time that one single technique is explicitly employed to address multimodal fusion, multitask learning, or both, particularly on dense image prediction.
- Experimental evaluations are conducted on semantic

segmentation via RGB-D data [42], [43] and image translation through multi-domain input [44]. It demonstrates that CEN yields remarkably superior performance to various kinds of multimodal fusion methods and multitask learning methods under a fair condition of comparison.

2 RELATED WORK

We introduce the methods of deep multimodal fusion and deep multitask learning, especially using dense image prediction as examples. We also discuss other related concepts.

Deep multimodal fusion. Regarding dense image prediction, deep multimodal fusion uses multiple data sources to enhance pixel-level semantics and fine-grained details against the single-modality counterpart. To this end, related methods toward dense image prediction are basically categorized into aggregation-based fusion and alignment-based fusion. Aggregation-based fusion methods apply a certain operation (*e.g.*, averaging [29], concatenation [30], [45], and attention-based modules [4], [46]) to fuse high-resolution feature maps and combine multimodal subnetworks into a single network. For example, U2Fusion [47] concatenates source images and puts forward the information measurement for unsupervised learning. RDFNet [36] adopts multi-layer fusion and iteratively refines fused features with additional convolutional blocks for aggregation. Due to the weakness in intra-modal processing, recent aggregation-based works perform feature fusion while still maintaining the subnetworks of all modalities [35], [48]. Alignment-based fusion methods [7], [31], instead, adopt regulation losses to align the embedding of subnetworks while keeping full propagation for each of them. These methods align multimodal features by applying the similarity regulation, where Maximum-Mean-Discrepancy (MMD) [49] is usually adopted for the measurement. However, simply focusing on unifying the whole distribution may overlook the specific patterns in each domain/modality [7], [50]. Hence, [31] provides a way that might alleviate this issue, which correlates modality-common features while simultaneously maintaining modality-specific information. Another categorization of multimodal fusion towards dense prediction could be generally specified as early, middle, and late fusion, depending on when to fuse, which have been discussed in earlier works [51], [52], [53], [54] and also in the current deep learning literature [1], [55], [56], [57]. Besides, evaluations in [36] indicate that the single-layer fusion can not effectively exploit multimodal features, especially for addressing high-resolution predictions torward dense image prediction. [29] points out that the performance of dense feature fusion is highly affected by the choice of which layer to fuse. Beyond dense image prediction, there are other portions of the multimodal learning literature, *e.g.*, based on modulation [57], [58], [59]. Different from these categories of fusion methods, we propose a new fusion method by channel exchanging, which potentially enjoys the guarantee of both sufficient inter-model interactions and intra-modal learning.

Deep multitask learning. In general, multitask visual perception predicts multiple output domains based on one same vision domain. Typical approaches could include designing hard parameter-sharing and soft parameter-sharing.

Specifically, hard parameter-sharing imposes a fixed subset of hidden layers to be shared across tasks and other layers to be task-specific, for example, UberNet [32], U2Fusion [47], and others [33], [60], [61]. Differently, for soft (or partial) parameter-sharing, there could be a separate set (or a significant fraction) of parameters per task, and models are correlated either by adaptive feature sharing or by aligning parameters to be similar, for example, Cross-stitch [15], Sluice [34], and NDDR [62]. Yet, compared with the learning upon single modalities, multitask learning is not always beneficial, since the performance might be harmed by the negative transfer (negative knowledge transfer across tasks), which is discussed in [63], [64], [65]. In addition, many multitask learning methods are specifically designed for dense image prediction, which is also the main focus of this paper. For example, MTI-Net [66] distills dense features across different tasks with multimodal feature aggregation. [65], [67] explicitly enforce cycle-based consistency between domains to improve performance and generalization. U2Fusion [47] develops joint training and sequential training that leverages a shared model to handle multitask learning for image-to-image translation. In this paper, we integrate the benefits of both hard parameter-sharing and soft parameter-sharing. Specifically, for multitask learning, we share the parameters of encoders for all tasks (hard parameter-sharing) and then conduct CEN on decoders (soft parameter-sharing).

Other related concepts. The idea of using the BN scaling factor to evaluate the importance of CNN channels has been studied in network pruning [38], [39] and representation learning [68]. Moreover, [38] enforces ℓ_1 norm penalty on the scaling factors and explicitly prunes out filters meeting sparsity criteria. Here, we apply this idea as an adaptive tool to determine where to exchange and fuse. CBN [57] performs cross-modal message passing by modulating BN of one modality conditional on the other, which is different from our method that directly exchanges channels across modalities for fusion. ShuffleNet [69] proposes to shuffle a portion of channels among multiple groups for efficient propagation in light-weight networks, which is similar to our idea of exchanging channels for message fusion. Yet, while the motivation of our paper is highly different, the exchanging process is self-determined by the BN scaling factors, instead of the random exchanging in ShuffleNet.

3 CHANNEL EXCHANGING NETWORKS

We first introduce the general formulation of CEN, and then follow it up by specifying the design of four different settings: multimodal fusion, cycle multimodal fusion, multitask learning, and multimodal multitask learning.

3.1 The general mechanism

For either multimodal or multitask learning, we are interested in studying the relationship between subnetworks on different streams of input-output pairs. Suppose we have the data of M streams $\{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^M$, where $\boldsymbol{x}_m$ and $\boldsymbol{y}_m$ represent the input data point and output label, respectively. The subnetwork of the m-th stream is dubbed as f_m. The notion of "stream" can be flexibly specified: for multimodal fusion, a different stream corresponds to a different modality where $\boldsymbol{x}_m$ varies but $\boldsymbol{y}_m$ keeps unchanged in terms

of different m; for multitask learning, on the contrary, a different stream implies a different task, where $\boldsymbol{x}_m$ usually keeps the same and $\boldsymbol{y}_m$ represents the label for task m.

A trivial training paradigm is minimizing the loss of each subnetwork f_m independently, which leads to the loss between the prediction $\hat{\boldsymbol{y}}_m := f_m(\boldsymbol{x}_m)$ and the label $\boldsymbol{y}_m$[1],

$$\min_{f_{1:M}} \sum_{m=1}^M \mathcal{L}\big(f_m(\boldsymbol{x}_m), \boldsymbol{y}_m\big). \tag{1}$$

However, the independent training strategy fails to characterize the affinity between different streams, limiting the expressivity of multimodal information fusion or multitask knowledge transfer.

In this work, we propose CEN that adaptively exchanges the knowledge between different subnetworks in an end-to-end manner. In form, the training objective in Eq. 1 can be rewritten as

$$\min_{f_{1:M}} \sum_{m=1}^M \mathcal{L}\big(f_m(\boldsymbol{x}_{1:M}), \boldsymbol{y}_m\big) + \lambda\|\hat{\boldsymbol{\gamma}}_m\|_1 \tag{2}$$

where,

- The subnetwork $f_m(\boldsymbol{x}_{1:M})$ (instead of $f_m(\boldsymbol{x}_m)$ in Eq. 1) fuses multimodal information by channel exchanging from other subnetworks to the m-th subnetwork, as we will detail later;
- Each subnetwork is equipped with BN layers containing the scaling factors $\boldsymbol{\gamma}_m$, and we will penalize the ℓ_1 norm of their certain portion $\hat{\boldsymbol{\gamma}}_m$ for sparsity. The ℓ_1 norm is uniformly applied to all BN layers. Here, we omit the layer index for simplicity.

Prior to introducing the mechanism of channel exchanging, we first review the Batch-Normalization (BN) layer [40], which is used widely in deep learning to eliminate covariate shift and improve generalization. For a certain BN layer, we denote by $\boldsymbol{x}_m$ the feature map of the m-th subnetwork, and by $\boldsymbol{x}_{m,c}$ the c-th channel. The BN layer performs a normalization of $\boldsymbol{x}_m$ followed by an affine transformation, namely,

$$\boldsymbol{x}'_{m,c} = \gamma_{m,c} \frac{\boldsymbol{x}_{m,c} - \mu_{m,c}}{\sqrt{\sigma_{m,c}^2 + \epsilon}} + \beta_{m,c}, \tag{3}$$

where, $\mu_{m,c}$ and $\sigma_{m,c}$ compute the mean and the standard deviation, respectively, of all activations over all pixel locations (H and W) for the current mini-batch data; $\gamma_{m,c}$ and $\beta_{m,c}$ are the trainable scaling factor and offset, respectively; ϵ is a small constant to avoid divisions by zero. The following layer takes $\{\boldsymbol{x}'_{m,c}\}_c$ as input after a non-linear function.

The factor $\gamma_{m,c}$ in Eq. 3 evaluates the correlation between the input $\boldsymbol{x}_{m,c}$ and the output $\boldsymbol{x}'_{m,c}$ during training. The gradient of the loss *w.r.t.* $\boldsymbol{x}_{m,c}$ will approach 0 if $\gamma_{m,c} \to 0$ at one training step, implying that $\boldsymbol{x}_{m,c}$ will almost lose its influence to the final prediction and become redundant thereby at this traing step.

In addition, as will be shown in Fig. 8 (a), if the scaling factor of one channel (with sparsity constraints) is lower than the small threshold at one training step, this channel

---

1. Note that this loss should be summed over all data points in real implementation. Here we consider a single data point throughout the paper for simplicity.
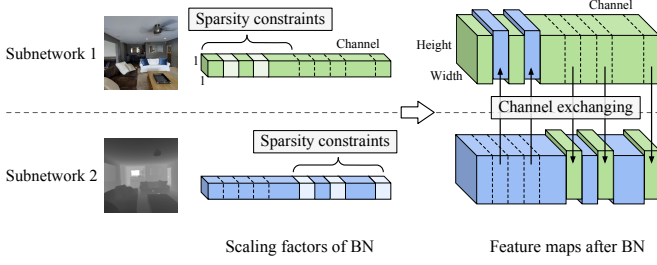
Fig. 1: An illustration of CEN. The sparsity constraints on scaling factors are applied to disjoint channel regions of different modalities. A feature map will be replaced by that of other modalities at the same position, if its scaling factor is lower than a threshold.

will hardly recover and almost become redundant during the later training process.

It motivates us to replace the channels of small scaling factors with the ones of other subnetworks, since those channels potentially are redundant. To do so, we have

$$
\boldsymbol{x}'_{m,c} = \begin{cases} \gamma_{m,c} \dfrac{\boldsymbol{x}_{m,c} - \mu_{m,c}}{\sqrt{\sigma^2_{m,c} + \epsilon}} + \beta_{m,c}, & \text{if } \gamma_{m,c} > \theta; \\ \dfrac{1}{M-1} \sum\limits_{m' \neq m}^{M} \gamma_{m',c} \dfrac{\boldsymbol{x}_{m',c} - \mu_{m',c}}{\sqrt{\sigma^2_{m',c} + \epsilon}} + \beta_{m',c}, & \text{else}; \end{cases} \tag{4}
$$

where the current channel is replaced with the mean of other channels if its scaling factor is smaller than a certain threshold $\theta \approx 0^+$. In a nutshell, if one channel of one modality has little impact on the final prediction, then we replace it with the mean of other modalities. We apply Eq. 4 for each modality before feeding them into the nonlinear activation followed by the convolutions in the next layer. Gradients are detached from the replaced channel and back-propagated through the new ones.

Fig. 11 illustrates our channel exchanging process in each of the layers. In order to In our implementation, we equally divide the whole channels into M sub-parts and only per-form the channel exchanging in each corresponding sub-part for each modality. This is mainly to avoid a portion of channels being redundant *w.r.t.* all modalities. More detailed reasons are described in § 4.5. We denote the scaling factors that are allowed to be replaced as $\hat{\gamma}_m$. We further impose the sparsity constraint on $\hat{\gamma}_m$ in Eq. 2 to discover unnecessary channels. As the exchanging in Eq. 4 is a directed process within only one sub-part of channels, it hopefully can not only retain modal-specific propagation in the other $M - 1$ sub-parts but also avoid unavailing exchanging since $\gamma_{m',c}$, different from $\hat{\gamma}_{m,c}$, is out of the sparsity constraint.

Regarding specific tasks where Instance-Normalizations (INs) are used for normalization instead of BNs, the sparsity constraints are similarly applied to scaling factors of INs, and the channel exchanging design (Eq. 4) is still applicable.

We summarize the advantages of our CEN below:

- **Prameter-free**. As specified in Eq. 4, CEN involves no additional parameter and applies BN scaling factors to control the exchanging process.
- **Self-adaptive**. The channel exchanging could take place at every layer throughout the encoder or/and decoder. BN scaling factors are learned from the data, which adaptively balances the inter-subnetwork processing and inter-subnetwork fusion.

3.2 Multimodal fusion via CEN on encoders

In this part, we focus particularly on multimodal fusion $\{\boldsymbol{x}_m\}_{m=1}^{M} \to \boldsymbol{y}$, where $\boldsymbol{x}_m$ denotes the m-th input modality, and all subnetworks generate the same output $\boldsymbol{y}$, *i.e.*, $\boldsymbol{y}_m = \boldsymbol{y}, \forall m = 1, \cdots, M$. Given that this paper mainly copes with dense prediction problems (such as depth estimation or semantic segmentation), the subnetwork f_m is of the encoder-decoder style. The goal of multimodal fusion is to effectively fuse the information of all modalities to improve the prediction accuracy for the target output. It is thus natural to fix the same decoder for all subnetworks and conduct CEN between their encoders. The architecture of multimodal fusion is depicted in Fig. 2 (a).

We first carry out sparsity penalty on BN scaling factors for the m-th encoder following Eq. 2, and then perform channel exchanging. Besides, the final output of the decoder is an ensemble of all modalities associated with the decision scores $\{\alpha_m\}_{m=1}^{M}$[2]; in our implementation, these decision scores are learned by an additional softmax output to meet the simplex constraint $\sum_{m=1}^{M} \alpha_m = 1$.

It is known in [70] that leveraging individual BN layers characterizes the traits of different domains or modalities. In our method, specifically, different scaling factors (Eq. 3) evaluate the importance of the channels of different modalities, and they should be decoupled. With the exception of BN (or IN) layers, all subnetworks share all parameters (*e.g.* convolutional filters[3]) in the encoder with each other. The hope is that we can further reduce the network complexity and therefore improve the predictive generalization. Rather, considering the specific design of our framework, sharing convolutional filters is able to capture the common patterns in different modalities, which is a crucial purpose of multimodal fusion. This design further compacts the multimodal architecture to be almost as small as the unimodal one, as will be evaluated in Table 2. In our experiments, we conduct multimodal fusion on RGB-D images or on other domains of images corresponding to the same image content. In this scenario, all modalities are homogeneous in the sense that they are just different views of the same input. Thus, sharing parameters between different subnetworks still yields promisingly expressive power. Nevertheless, when we are dealing with heterogeneous modalities (*e.g.*, images and text sequences), it would impede the expressive power of the subnetworks if keeping sharing their parameters, hence a more dexterous mechanism is suggested, and the discussion of which is left for future exploration.

3.3 Cycle multimodal fusion via CEN on encoders

In the previous section (§ 3.2), we have introduced how to apply CEN on multimodal fusion. Here, we discuss a more complicated setting: cycle multimodal fusion. Assuming we have $\{\boldsymbol{x}_m\}_{m=1}^{M} \to \boldsymbol{x}_{M+1}$, where the output is specified as the $(M + 1)$-th modality for consistent denotation. Note that such learning task is related to a different task

---

2. The decision scores are learnable scalars, optimized by comparing ensembled outputs with labels while temporally freezing (detaching) the subnetworks. The decision scores are fixed during inference.

3. If the input channels of different modalities are different (*e.g.*, RGB and depth), we will broaden their sizes to be the same as their Least Common Multiple (LCM).

(a) Multimodal learning   **(b)** Cycle multimodal learning   **(c)** Multitask learning   **(d)** Multimodal multitask learning
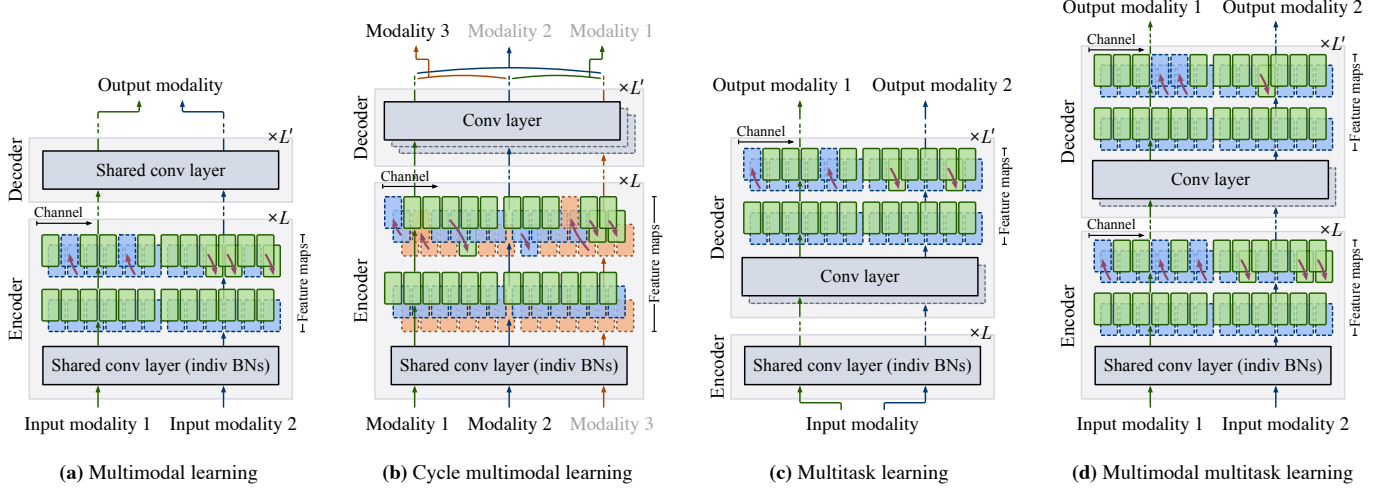
Fig. 2: Structures of CENs for multimodal fusion, cycle multimodal fusion, multitask learning, and multimodal multitask learning. For cycle multimodal learning, given the case with three modalities, only two of the three forward passes are performed at each time. Here, "conv" and "indiv" are abbreviations for "convolutional" and "individual", respectively. L and L' denote layer numbers of the encoder and the decoder, respectively.

$\{\boldsymbol{x}_m\}_{m=1,m\neq j}^{M+1} \to \boldsymbol{x}_j$, which, inversely, uses modality $M+1$ along with the remaining modalities to generate modality j. Actually, we can go through all the $M+1$ cases by cycling different output modality, which leads to a set of cycle multimodal fusion tasks $\{\mathcal{T}_j := \{\boldsymbol{x}_m\}_{m=1,m\neq j}^{M+1} \to \boldsymbol{x}_j\}_{j=1}^{M+1}$.

By § 3.2, a straightforward way is applying CEN independently to each multimodal fusion task $\mathcal{T}_j$ for fusing the input modalities. Nevertheless, such an independent learning fashion is unable to reveal the relationships between $\mathcal{T}_j$s. Although different tasks conduct different generation directions, these tasks are tackling overlapping modalities, hence potentially, their learning knowledge might be reused and the learning processes could be coupled. Towards this purpose, we enforce all $\mathcal{T}_j$s to share the same encoder except the BN parameters. Specifically, for each task $\mathcal{T}_j$, we utilize distinct sets of BN parameters for different input modalities, giving rise to the total number of BN parameter sets for all tasks as $M(M+1)$. With the separated BNs, we then carry out CEN on the encoder for multimodal fusion for each task $\mathcal{T}_j$. The sketched pipeline is illustrated in Fig. 2 (b). Note that for the case with three modalities, channels are still divided into two parts, since for cycle multimodal fusion, only two of the three modalities are sent to the encoder at each time.

Obviously, cycle multimodal fusion is a multitask generalization of the multimodal fusion in § 3.2. The key benefit is that it simultaneously addresses all combinations of the cycling generation tasks with only one single pair of the encoder and decoder, which dramatically decreases the model complexity. More interestingly, as we will demonstrate in our experiments, the cycle multimodal fusion can improve each of the single-task multimodal fusion, probably thanks to the knowledge transfer by parameter sharing and joint training. We will provide more details and evaluations for cycle multimodal fusion in the experiment section.

3.4 Multitask learning via CEN on decoders

Different from multimodal fusion, multitask learning requires to predict different labels for different subnetworks:

$\boldsymbol{x} \to \{\boldsymbol{y}_m\}_{m=1}^M$, where we assume all tasks have the same input, *i.e.*, $\boldsymbol{x}_m = \boldsymbol{x}, \forall m = 1, \cdots, M$ and the output label is $\boldsymbol{y}_m$ for the task m. The advantage of multitask learning is to improve model generalization and data efficiency, by sharing task-common knowledge while retaining task-specific information. One of the widely-used methods is employing the hard parameter-sharing mechanism [71] that shares the encoder and uses task-specific decoders. Despite its popularity in previous applications, modelling the multitask relationship by solely sharing the encoder is insufficient in characterizing high-level patterns, particularly the related features across decoders.

To address the aforementioned issues, we propose to perform channel exchanging on the decoders. Our goal of employing CEN on decoders lies in adaptively discovering the redundant channels in decoders and compensating for the information from the channels of other tasks. The methodology is illustrated is in Fig. 2 (c). Specifically, the sparsity penalty of BN (or IN) scaling factors is added to the decoder part. Accordingly, for the m-th subnetwork, channel exchanging is conducted from other decoders to the m-th decoder.

3.5 Multimodal multitask learning via CEN on both encoders and decoders

It could be straightforward to combine the designs in § 3.2 and § 3.4 to handle multimodal multitask learning tasks, with multiple input and output modalities, as illustrated in Fig. 2 (d). It requires to address $\{\{\boldsymbol{x}_{m_1}\}_{m_1=1}^{M_1} \to \boldsymbol{y}_{m_2}\}_{m_2=1}^{M_2}$, where M_1 and M_2 are the numbers of input and output modalities, respectively. To enable simultaneous multimodal fusion and multitask learning, we perform CEN on both encoders and decoders. The input for each decoder is given by CEN on all encoders. In this case, we share the convolutional layers at the encoder part and privatize $M_1 M_2$ groups of BN (or IN) parameters. Similarly, for the m_2-th task/decoder (where $m_2 = 1, \cdots, M_2$), we adopt $\{\alpha_{m_1}^{m_2}\}_{m_1=1}^{M_1}$ as decision scores for ensemble that meet $\sum_{m_1=1}^{M_1} \alpha_{m_1}^{m_2} = 1$.

4 EXPERIMENTS

We contrast the performance of CEN against existing methods on the four problems in Fig. 2. For multimodal fusion, we conduct experiments on the two tasks: semantic segmentation and image-to-image translation. For the other three problems, we evaluate the performance mainly on image-to-image translation, since this task contains a rich number of image modalities and is suitable for evaluations under various settings. The datasets and implementation details for semantic segmentation and image-to-image translation are provided below.

Semantic segmentation. We evaluate our method on two public datasets NYUDv2 [42] and SUN RGB-D [43], which consider RGB and depth as input. Regarding NYUDv2, we follow the standard settings and adopt the split of 795 images for training and 654 for testing, predicting standard 40 classes [72]. SUN RGB-D is one of the most challenging large-scale benchmarks for indoor semantic segmentation, containing 10,335 RGB-D images of 37 semantic classes. We use the public train-test split (5,285 vs 5,050). We consider RefineNet [3]/PSPNet [73] as our segmentation framework whose backbone is implemented by ResNet [74] pretrained from ImageNet dataset [75]. The initial learning rates are set to 5×10^{-4} for the encoder and 3×10^{-3} for the decoder, respectively, both of which are reduced to their halves every 100/150 epochs (of total epochs 300/450) on NYUDv2 with ResNet101/ResNet152 and every 20 epochs (of total epochs 60) on SUN RGB-D. The mini-batch size, momentum, and weight decay are selected as 6, 0.9, and 10^{-5}, respectively, on both datasets. We set $\lambda = 5 \times 10^{-3}$ in Eq. 2 and the threshold to $\theta = 2 \times 10^{-2}$ in Eq. 4. Unless otherwise specified, we adopt the multi-scale strategy [3], [36] during the test time. We employ common evaluation metrics including Mean IoU, Pixel Accuracy, and Mean Accuracy [3]. Full implementation details are provided in the appendix.

Image-to-image translation. We adopt Taskonomy [44], a dataset with 4 million images of indoor scenes gathered from about 600 buildings. Each image in Taskonomy has more than 10 multimodal representations, including depth (euclidean/zbuffer), shade, normal, texture, edge, principal curvature, etc. For efficiency, we sample 1,000 high-quality multimodal images for training, and 500 for validation. We also provide experiments with 15,000 sampled images for training in the appendix. Following Pix2pix [25], we adopt the U-Net-256 structure for image translation with the consistent setups with [25]. The BN computations are replaced with Instance Normalization layers (INs), and our method (Eq. 4) is still applicable. We adopt individual INs in the encoder, and share all other parameters including INs in the decoder. We set λ to 10^{-3} for sparsity constraints and the threshold θ to 10^{-2}. FID [76] and KID [77] are adopted as evaluation metrics, as will be introduced in the appendix.

4.1 Evaluations on multimodal fusion

We first assess the importance of each component in CEN solely on the semantic segmentation dataset NYUDv2, and then compare the performance with other multimodal fusion baselines and SOTA methods on semantic segmentation and image-to-image translation.

TABLE 1: Detailed results for different versions of our CEN on NYUDv2. All results are obtained with the backbone RefineNet (ResNet101) of single-scale evaluation for test. "Ens." is the abbreviation for "Ensemble".

Convs	BNs	ℓ_1 Regulation	Exchange	Mean IoU (%)		
				RGB	Depth	Ens.
Unshared	Unshared	×	×	45.5	35.8	47.6
Shared	Shared	×	×	43.7	35.5	45.2
Shared	Unshared	×	×	46.2	38.4	48.0
Shared	Unshared	×	✓ (fixed 30%)	44.9	40.3	47.2
Shared	Unshared	×	✓ (random)	44.2	40.5	46.8
Unshared	Unshared	All-channel	×	44.6	35.3	46.6
Unshared	Unshared	All-channel	✓	46.8	41.7	49.1
Shared	Unshared	All-channel	×	46.1	37.9	47.5
Shared	Unshared	All-channel	✓	48.6	39.0	49.8
Unshared	Unshared	Half-channel	×	45.1	35.5	47.3
Unshared	Unshared	Half-channel	✓	46.5	41.6	48.5
Shared	Unshared	Half-channel	×	46.0	38.1	47.7
Shared	Unshared	Half-channel	✓	**49.7**	**45.1**	**51.1**

4.1.1 Semantic Segmentation

The validity of each proposed component. Table 1 summarizes the results of different variants of CEN on NYUDv2. We have the following observations:

- Compared to the unshared baseline, sharing convolutional parameters greatly boosts the performance, particularly on the Depth modality (35.8 vs 38.4). Yet, the performance will encounter a clear drop if we additionally share the BN layers. This observation is consistent with our analyses in § 3.2 due to the different roles of convolutional filters and BN parameters.
- As ℓ_1 enables the discovery of unnecessary channels, naively exchanging channels with a fixed portion (without using ℓ_1) could not reach good performance. For example, exchanging a fixed portion of 30% channels (close to the averaged number of exchanged channels in CEN) only gets IoU 47.2. Besides, we try to exchange channels randomly like ShuffleNet or directly discard unimportant channels without channel exchanging, the IoUs of which are 46.8 and 47.5, respectively.
- After carrying out directed channel exchanging under the ℓ_1 regulation, our model gains a huge improvement on both modalities, *i.e.* from 46.0 to 49.7 on RGB, and from 38.1 to 45.1 on Depth, and finally increases the ensemble Mean IoU from 47.6 to 51.1. It thus verifies the effectiveness of our proposed mechanism on this task.
- Note that the channel exchanging is only available on a certain portion of each layer, *i.e.*, exchanging only half of the channels in the two-modal case. When we remove this constraint and allow all channels to be exchanged by Eq. 4, the accuracy decreases, which we conjecture is owing to the detriment by impeding modal-specific propagation, if all channels are engaged in cross-modal fusion.

After training CEN (with sparsity constraints on disjoint channel regions, as illustrated in Fig. 11), each certain channel belongs to one of the three categories: ($\gamma_{rgb} \approx 0, \gamma_{depth} > 0$), ($\gamma_{rgb} > 0, \gamma_{depth} \approx 0$), and ($\gamma_{rgb} > 0, \gamma_{depth} > 0$). There
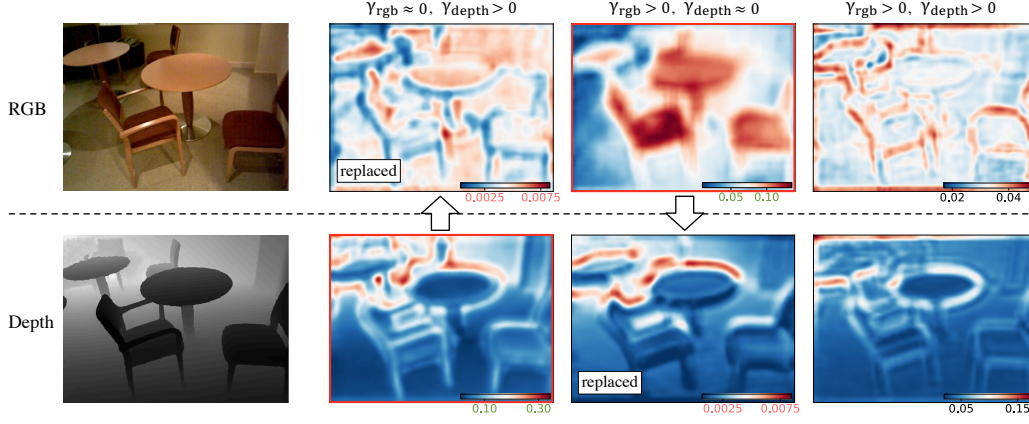
Fig. 3: Visualization of the *averaged* feature maps for RGB and Depth. From left to right: the input images, the channels of $(\gamma_{rgb} \approx 0, \gamma_{depth} > 0)$, $(\gamma_{rgb} > 0, \gamma_{depth} \approx 0)$, and $(\gamma_{rgb} > 0, \gamma_{depth} > 0)$. The feature maps are collected in a single layer, specifically, the 9th layer of ResNet, *i.e.* the 2nd layer of the 3rd stage (with 256 channels) of ResNet. Values under color bars correspond to the actual values of averaged feature maps.

TABLE 2: Comparison with three typical fusion methods including concatenation (concat), fusion by alignment (align), and self-attention (self-att.) on NYUDv2. All results are obtained with RefineNet (ResNet101) of single-scale evaluation for test. "Ens." is the abbreviation for "Ensemble".

| Modality | Approach | Commonly-used setting | | Same with our setting | | Params. used for fusion (M) |
		Mean IoU (%)	Params. in total (M)	Mean IoU (%) RGB / Depth / Ens.	Params. in total (M)	
RGB	Uni-modal	45.5	118.1	45.5 / - / -	118.1	-
Depth	Uni-modal	35.8	118.1	- / 35.8 / -	118.1	-
RGB-D	Concat (early)	47.2	120.1	47.0 / 37.5 / 47.6	118.8	0.6
	Concat (middle)	46.7	147.7	46.6 / 37.0 / 47.4	120.3	2.1
	Concat (late)	46.3	169.0	46.3 / 37.2 / 46.9	126.6	8.4
	Concat (all-stage)	47.5	171.7	47.8 / 36.9 / 48.3	129.4	11.2
	Align (early)	46.4	238.8	46.3 / 35.8 / 46.7	120.8	2.6
	Align (middle)	47.9	246.7	47.7 / 36.0 / 48.1	128.7	10.5
	Align (late)	47.6	278.1	47.3 / 35.4 / 47.6	160.1	41.9
	Align (all-stage)	46.8	291.9	46.6 / 35.5 / 47.0	173.9	55.7
	Self-att. (early)	47.8	124.9	47.7 / 38.3 / 48.2	123.6	5.4
	Self-att. (middle)	48.3	166.9	48.0 / 38.1 / 48.7	139.4	21.2
	Self-att. (late)	47.5	245.5	47.6 / 38.1 / 48.3	203.2	84.9
	Self-att. (all-stage)	48.7	272.3	48.5 / 37.7 / 49.1	231.0	112.8
	Our CEN	-	-	**49.7 / 45.1 / 51.1**	**118.2**	**0.0**

will not be $(\gamma_{rgb} \approx 0, \gamma_{depth} \approx 0)$ since we apply sparsity constraints on disjoint channels. To further explain why channel exchanging works, Fig. 3 displays the averaged feature maps of RGB and Depth. Here, "averaged" means: Firstly, extracting feature maps at all specific channels (in a layer) that belong to the same (aforementioned) category; Secondly, averaging these feature maps along the channels. We observe from Fig. 3 that RGB channels with non-zero scaling factors mainly characterize the texture, while Depth channels with non-zero factors focus more on the boundary; in this sense, performing channel exchanging can better combine the complementary properties of both modalities.

Comparison with fusion baselines. In Table 2, we report comparison results of our CEN with two aggregation-based methods: concatenation [30] and self-attention [4], and one alignment-based approach [31], using the same backbone. All baselines are implemented with the early, middle, late, and all stage fusion. For a more fair comparison, all base-

lines are further conducted under the same setting (except channel exchanging) with ours, namely, sharing convolutions with individual BNs, and preserving the propagation of all subnetworks (with also the ensemble). Full details are provided in the appendix. It demonstrates that, in both settings, our method always outperforms others by an average improvement of larger than 2%. We also report the parameters used for fusion, *e.g.* the aggregation weights of two modalities in concatenation. While self-attention (all-stage) attains the closest performance to ours (49.1 vs 51.1), its parameters used for fusion are considerable, whereas our fusion is parameter-free.

Visualizations are provided in Fig. 4. We choose the hard cases including the images containing tables and chairs, as well as those with low/high light intensity. We observe that the concatenation method is more sensitive to noises in the depth input. Both concatenation and self-attention methods are weak in predicting thin objects, *e.g.*, table legs and chair
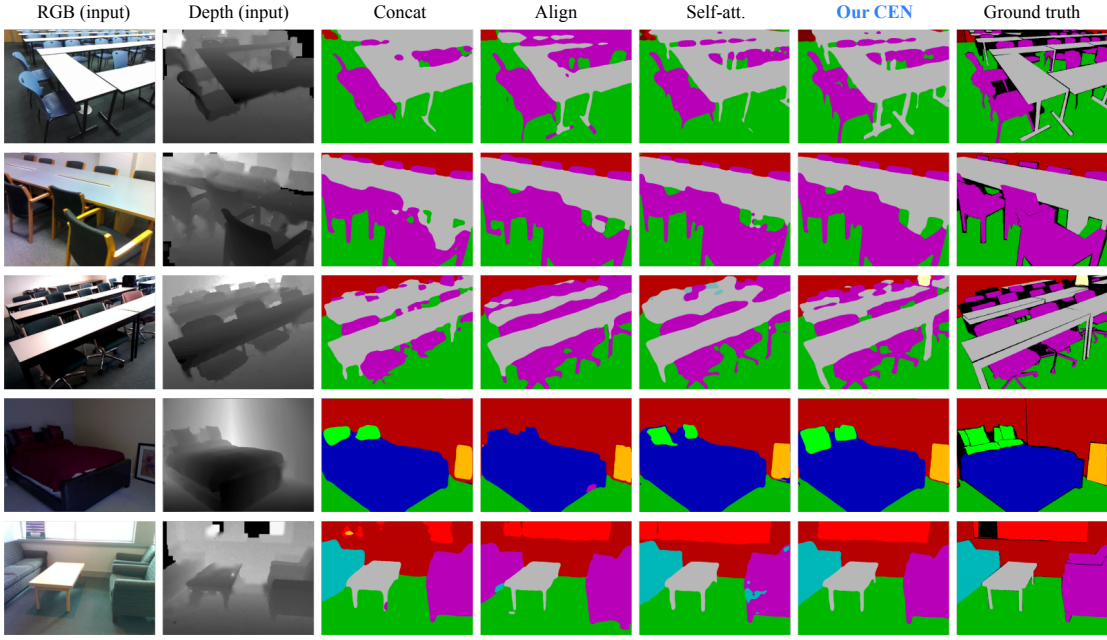
Fig. 4: Visualization results of semantic segmentation. Images are collected from NYUDv2 and SUN RGB-D datasets. All results are obtained with the backbone RefineNet (ResNet101) of single-scale evaluation for test.

TABLE 3: Comparison with SOTA semantic segmentation methods on NYUDv2 and SUN RGB-D datasets. † indicates our implemented results. Evaluation metrics include Pixel Accuracy, Mean Accuracy and Mean IoU.

Modality	Approach	Backbone Network	NYUDv2			SUN RGB-D		
			Pixel Acc. (%)	Mean Acc. (%)	Mean IoU (%)	Pixel Acc. (%)	Mean Acc. (%)	Mean IoU (%)
RGB	FCN-32s [22]	VGG16	60.0	42.2	29.2	68.4	41.1	29.0
	RefineNet [3]	ResNet101	73.8	58.8	46.4	80.8	57.3	46.3
	RefineNet [3]	ResNet152	74.4	59.6	47.6	81.1	57.7	47.0
RGB-D	FuseNet [29]	VGG16	68.1	50.4	37.9	76.3	48.3	37.3
	ACNet [78]	ResNet50	-	-	48.3	-	-	48.1
	SSMA [4]	ResNet50	75.2	60.5	48.7	81.0	58.1	45.7
	SSMA [4] †	ResNet101	75.8	62.3	49.6	81.6	60.4	47.9
	CBN [57] †	ResNet101	75.5	61.2	48.9	81.5	59.8	47.4
	3DGNN [79]	ResNet101	-	-	-	-	57.0	45.9
	SCN [80]	ResNet152	-	-	49.6	-	-	50.7
	CFN [48]	ResNet152	-	-	47.7	-	-	48.1
	RDFNet [36]	ResNet101	75.6	62.2	49.1	80.9	59.6	47.2
	RDFNet [36]	ResNet152	76.0	62.8	50.1	81.5	60.1	47.7
	Ours-RefineNet (single-scale)	ResNet101	76.2	62.8	51.1	82.0	60.9	49.6
	Ours-RefineNet	ResNet101	77.2	63.7	51.7	82.8	61.9	50.2
	Ours-RefineNet (single-scale)	ResNet152	77.0	64.4	51.6	82.3	61.7	50.0
	Ours-RefineNet	ResNet152	77.4	64.8	52.2	83.2	62.5	50.8
	Ours-PSPNet	ResNet152	**77.7**	**65.0**	**52.5**	**83.5**	**63.2**	**51.1**

legs. These objects are usually missed in the depth input, which may hinder the prediction results after fusion. On the contrary, the prediction results of our method preserve more details and are more robust to the light intensity.

Comparison with SOTAs. In Table 3, we contrast our method against a wide range of state-of-the-art methods. Their results are directly copied from previous papers if provided or re-implemented by us otherwise, as marked with annotations. Results conclude that our method equipped with PSPNet (ResNet152) achieves new records remarkably superior to previous methods in terms of all metrics on both datasets. In particular, given the same backbone, our method is still much better than RDFNet [36]. To isolate the contribution of RefineNet in our method, Table 3 also

provides the uni-modal results, where we observe a clear advantage of multimodal fusion.

4.1.2 Image-to-Image Translation

Comparison with baseline fusion methods. We then evaluate the performance given five specific translation cases, including Shade+Texture→RGB, Depth+Normal→RGB, RGB+Shade→Normal, RGB+Normal→Shade and RGB+ Edge →Depth. In addition to the three baselines used in semantic segmentation (Concat, Self-attention, Align), we conduct an extra aggregation-based method by using the average operation. All baselines perform fusion under four different kinds of strategies: early (at the 1st Conv-layer), middle (the 4th Conv-layer), late (the 8th Conv-layer), and
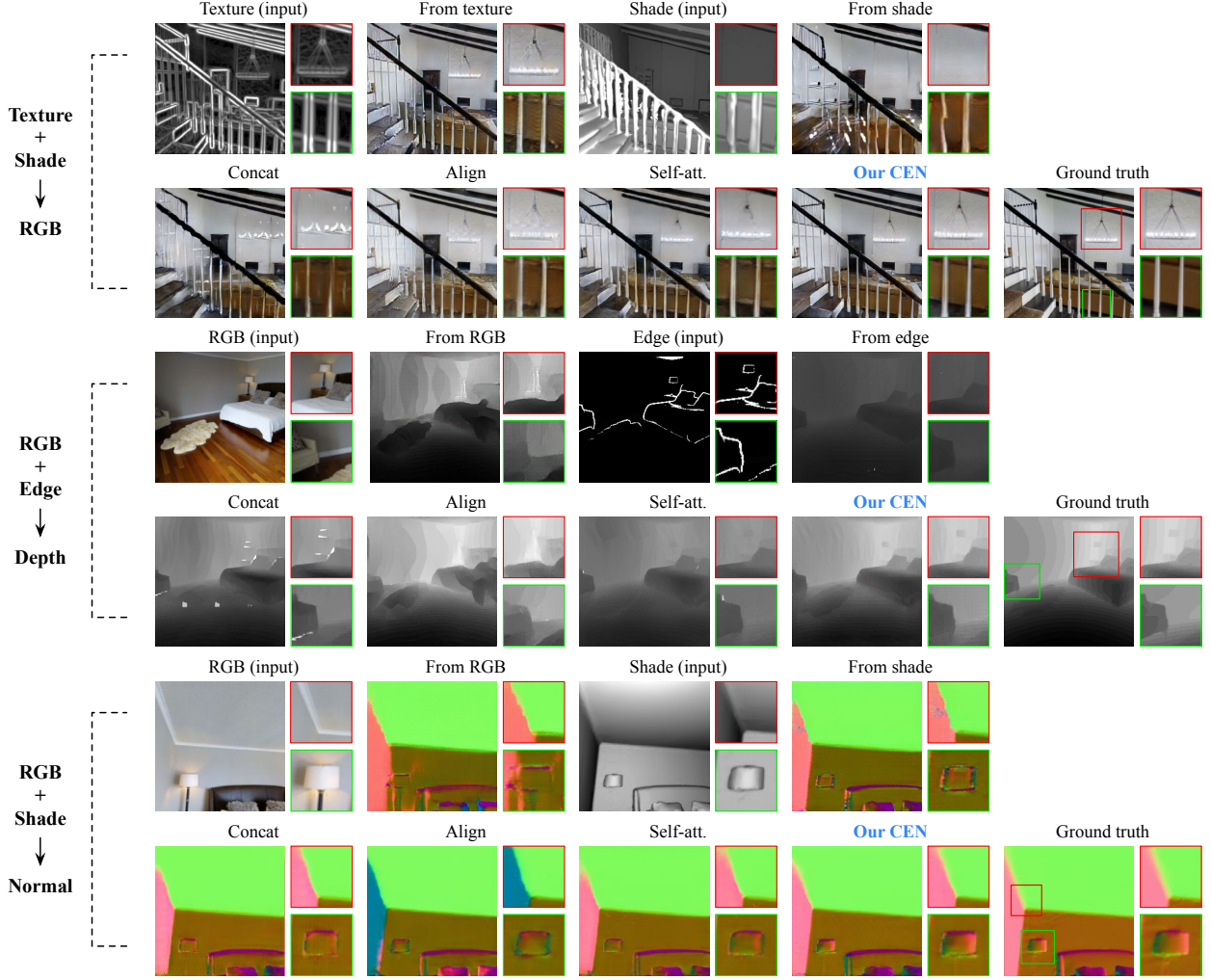
Fig. 5: Visualization results of multimodal image-to-image translation including Texture+Shade→RGB (top group), RGB+Edge→Depth (middle group), and RGB+Shade→Normal (bottom group), respectively. The resolution of each predicted image is 256×256. More visualizations are provided in the appendix.

all-layer fusion. Our method yields much lower FID/KID or MAE/MSE than others, especially when predicting the RGB modality, as detailed in Table 15. These results support the benefit of our proposed idea once again.

Main visualizations are provided in Fig. 5. We observe that when predicting RGB given texture and shade, the prediction solely predicted from the texture is vague at boundary lines, while the prediction solely from the shade misses some opponents, *e.g.* the pendant lamp, and is weak in predicting handrails. When fusing both input modalities, the concatenation method is uncertain in the regions where both modalities have disagreements. Alignment and self-attention are still weak in combining both modalities at details. Our results are clear at boundaries and fine-grained details. When predicting depth given RGB and edge, it is straightforward to find the benefits of multimodal fusion in this figure. The depth predicted by RGB is good at predicting numerical values, but is weak in capturing boundaries, which results in vague and curving boundaries. Oppositely, the depth predicted by the edge well captures boundaries, but is relatively weak in determining numerical values. The alignment fusion method is still weak in capturing

boundaries. Both concatenation and self-attention methods are able to combine the advantages of both modalities, but numerical values are still obviously lower than the ground truth. All illustrations verify that our CEN achieves better performance compared to baseline methods. More visualizations and baseline settings are provided in the appendix.

Considering more input modalities. We test whether our method is applicable to the case with more than two modalities. For this purpose, Table 5 presents the results of image translation to RGB by inputting from one to four modalities of Depth, Normal, Texture, and Shade. It is observed that increasing the number of modalities improves the performance consistently, suggesting much potential of applying our method towards various cases.

4.2 Evaluations on cycle multimodal fusion

In this subsection, we evaluate CEN-cycle, a cycle multimodal fusion mode of CEN to simultaneously tackle three generation flows with a compact structure. As described in § 3.3, in cycle multimodal fusion, we go through all 6 flows where each flow contains two input modalities and one output modality. The subnetwork is trained with all the

TABLE 4: Comparison on multimodal image-to-image translation task. Evaluation metrics are FID/KID ($\times 10^{-2}$) for RGB predictions and MAE ($\times 10^{-2}$)/MSE ($\times 10^{-2}$) for other predictions. Lower values indicate better performance for all metrics.

Modality	Our CEN	Baseline	Early	Middle	Late	All-layer
Shade+Texture →RGB	**62.63 / 1.65**	Concat Average Align Self-att.	87.46 / 3.64 93.72 / 4.22 99.68 / 4.93 83.60 / 3.38	95.16 / 4.67 93.91 / 4.27 95.52 / 4.75 90.79 / 3.92	122.47 / 6.56 126.74 / 7.10 98.33 / 4.70 105.62 / 5.42	78.82 / 3.13 80.64 / 3.24 92.30 / 4.20 73.87 / 2.46
Depth+Normal →RGB	**84.33 / 2.70**	Concat Average Align Self-att.	105.17 / 5.15 109.25 / 5.50 111.65 / 5.53 100.70 / 4.47	100.29 / 3.37 104.95 / 4.98 108.92 / 5.26 98.63 / 4.35	116.51 / 5.74 122.42 / 6.76 105.85 / 4.98 108.02 / 5.09	99.08 / 4.28 99.63 / 4.41 105.03 / 4.91 96.73 / 3.95
RGB+Shade →Normal	**11.23 / 25.09**	Concat Average Align Self-att.	13.34 / 28.27 14.24 / 30.47 14.50 / 31.07 12.99 / 28.21	12.15 / 26.54 12.62 / 27.02 13.92 / 29.34 11.75 / 25.86	13.93 / 28.80 14.01 / 28.95 12.81 / 27.55 14.22 / 29.07	13.36 / 28.51 12.82 / 28.28 15.18 / 32.50 12.63 / 27.61
RGB+Normal →Shade	**11.03 / 17.16**	Concat Average Align Self-att.	15.62 / 24.49 14.63 / 22.88 13.88 / 22.62 12.14 / 18.26	13.81 / 21.24 12.83 / 20.42 13.16 / 21.55 11.52 / 17.33	12.62 / 19.17 15.11 / 23.92 12.73 / 20.41 14.47 / 22.82	12.83 / 20.18 12.28 / 18.64 14.09 / 22.05 11.79 / 17.62
RGB+Edge →Depth	**2.75 / 6.60**	Concat Average Align Self-att.	3.43 / 7.53 3.62 / 7.78 4.38 / 8.93 3.03 / 7.05	3.17 / 7.39 3.41 / 7.64 3.86 / 8.16 3.32 / 7.29	3.82 / 7.87 3.56 / 7.73 4.19 / 8.61 3.40 / 7.47	3.25 / 7.46 3.30 / 7.44 4.38 / 9.03 3.01 / 6.98

TABLE 5: Multimodal fusion on image translation (to RGB) with $1 \sim 4$ input modalities.

Modality	Depth	Normal	Texture	Shade	Depth+Normal	Depth+Normal +Texture	Depth+Normal +Texture+Shade
FID	113.91	108.20	97.51	100.96	84.33	60.90	57.19
KID ($\times 10^{-2}$)	5.68	5.42	4.82	5.17	2.70	1.56	1.33

TABLE 6: Experimental results of cycle multimodal fusion. Evaluation metrics are FID/KID ($\times 10^{-2}$) for RGB predictions and MAE ($\times 10^{-2}$)/MSE ($\times 10^{-2}$) for other predictions. Lower values indicate better performance for all these metrics. "Curve" and "SemSeg" are abbreviations for the principle curve and semantic segmentation, respectively.

Modality	CEN (IN×6, enc×3, dec×3)	CEN-random (IN×6, enc×1, dec×3)	CEN-cycle (IN×6, enc×1, dec×3)	CEN-cycle (IN×6, enc×1, dec×1)
RGB+Shade → Texture	1.74 / 3.05	2.17 / 4.53	**1.54 / 2.56**	1.62 / 2.81
RGB+Texture → Shade	16.53 / 25.07	18.26 / 28.60	**15.53 / 23.77**	16.10 / 24.36
Shade+Texture → RGB	62.63 / 1.65	73.27 / 2.33	**61.03 / 1.50**	61.25 / 1.60
RGB+Depth → SemSeg	21.52 / 36.24	22.80 / 37.09	**18.57 / 33.29**	18.71 / 33.56
RGB+SemSeg → Depth	4.63 / 8.59	5.03 / 8.81	**4.02 / 7.90**	4.27 / 8.26
Depth+SemSeg → RGB	99.60 / 4.18	102.97 / 4.31	**96.13 / 3.66**	97.01 / 3.94
RGB+Depth → Normal	13.03 / 28.75	15.72 / 31.15	12.26 / 27.12	**11.94 / 26.79**
RGB+Normal → Depth	3.34 / 5.22	4.67 / 6.73	2.63 / 4.70	**2.57 / 4.45**
Depth+Normal → RGB	84.33 / 2.70	90.49 / 3.73	**82.81 / 2.64**	83.73 / 2.66
RGB+Depth → Curve	5.42 / 15.09	5.73 / 16.08	**4.83 / 13.71**	5.03 / 14.15
RGB+Curve → Depth	2.62 / 3.87	2.82 / 4.23	**2.14 / 3.47**	2.25 / 3.67
Depth+Curve → RGB	85.13 / 2.82	88.69 / 3.39	**83.85 / 2.42**	84.52 / 2.64
Depth+Normal → Shade	7.10 / 11.22	7.47 / 11.45	7.03 / 10.65	**6.60 / 10.31**
Shade+Depth → Normal	13.11 / 31.57	13.74 / 32.20	13.12 / 31.65	**12.92 / 31.30**
Shade+Normal → Depth	1.62 / 2.91	1.92 / 3.18	1.56 / 2.94	**1.50 / 2.87**
Total params. (M)	Gen: 163.3; Dis: 8.3	Gen: 124.2; Dis: 8.3	Gen: 124.2; Dis: 8.3	Gen: **54.5**; Dis: 8.3

three flows at each step. For each flow, our default setting is employing the encoders and decoders with shared convolution parameters but unshared INs. To demonstrate the benefit of CEN-cycle, we also implement these baselines in Table 6: independent CEN that trains each flow separately, CEN-random that randomly samples one of the three flows per training step, and CEN-cycle with unshared decoders.

We observe that compared with independent CEN, CEN-cycle with unshared decoders not only compacts the overall model but also achieves provably better prediction performance. By further sharing the decoders, CEN-cycle further reduces the model size (needing about $1/3$ parameter) and

still yields better results than independent CENs. In addition, CEN-random is inferior to independent CEN, probably because it is ineffective to balance the training between different flows if only one flow is trainable per step. In summary, the results here support that performing CEN-cycle is valuable, and it is able to reuse the information in different generation flows that involve overlapping input/output modalities by parameter sharing and joint training.

4.3 Evaluations on multitask learning

This subsection evaluates multitask learning which adopts a single modality as input and simultaneously predicts two
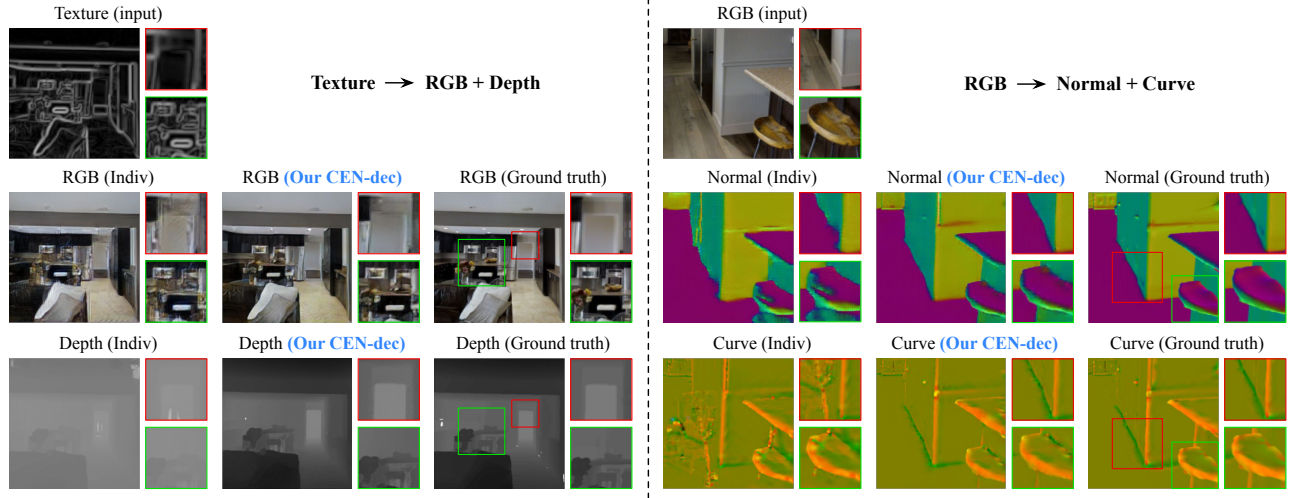
Fig. 6: Visualization results of multitask image-to-image translation including Texture→RGB+Depth (left group) and RGB→Normal+Curve (right group), respectively. "Curve" is the abbreviation for the principle curve modality. We compare the individual (Indiv) baseline with unshared encoders and our CEN-dec. The resolution of each predicted image is 256×256.

TABLE 7: Experimental results of multitask learning. Evaluation metrics are FID/KID ($\times 10^{-2}$) for RGB predictions and MAE ($\times 10^{-2}$)/MSE ($\times 10^{-2}$) for other predictions. Lower values indicate better performance. Individual (Indiv) learning and Cross-Task Consistency (X-TC) [65] are served as baselines. We provide numbers of groups for instance normalization (IN), encoder (enc) and decoder (dec), and the total parameters (params.) in generator (Gen) and discriminator (Dis), respectively. "Curve" and "SemSeg" are abbreviations for the principle curve and semantic segmentation, respectively.

Modality		Indiv (IN×2, enc×2, dec×2)	Indiv (IN×2, enc×1, dec×2)	X-TC [65] (IN×2, enc×1, dec×2)	CEN-dec (IN×2, enc×1, dec×2)	CEN-dec + X-TC [65] (IN×2, enc×1, dec×2)
RGB →	SemSeg	26.71 / 40.15	27.14 / 41.90	23.83 / 38.10	23.02 / 37.54	**21.78 / 37.32**
	Depth	5.35 / 9.13	5.51 / 9.42	5.22 / 8.98	4.82 / 8.50	**4.76 / 8.43**
RGB →	Normal	18.74 / 37.24	18.15 / 36.82	18.18 / 36.49	16.74 / 32.26	**14.85 / 29.33**
	Curve	6.24 / 16.97	6.02 / 16.70	5.33 / 14.76	4.92 / 14.08	**4.50 / 13.81**
RGB →	Shade	24.04 / 33.85	23.63 / 32.92	19.04 / 29.87	18.77 / 27.94	**17.07 / 27.10**
	Texture	2.40 / 4.93	2.19 / 4.66	2.33 / 4.85	1.83 / 3.67	**1.64 / 2.99**
Texture →	RGB	97.51 / 4.82	96.81 / 4.57	95.81 / 3.94	92.92 / 3.25	**90.85 / 2.81**
	Depth	4.20 / 8.16	4.05 / 7.94	3.54 / 6.07	3.19 / 5.05	**2.90 / 4.87**
Normal →	Depth	2.59 / 3.92	2.72 / 4.16	2.20 / 3.54	1.97 / 3.30	**1.85 / 3.04**
	Shade	8.08 / 12.40	7.90 / 12.03	7.26 / 11.52	7.09 / 11.14	**6.94 / 10.88**
Total params. (M)		Gen: 108.7; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3

or three different modalities. As introduced in § 3.4, CEN is conducted on the decoder side, abbreviated as CEN-dec.

Table 7 reports the case of predicting two modalities. Besides individual training with shared/unshared encoders, we consider a stronger baseline named Cross-Task Consistency (X-TC) [65] under the triangle loss setting. X-TC basically enforces an addition supervision to let one predicted modality generate the other one. As observed, CEN-dec outperforms individual learning and X-TC in all tasks, and its performance is further promoted if used along with X-TC, showing the compatibility between CEN-dec and X-TC.

In Fig. 6, we further provide visualizations of multitask learning. We observe that by simultaneously predicting RGB and depth from texture, our CEN-dec predicts noticeably better results. By simultaneously normal and the principle curve from RGB, predicted normal boundaries of the table and wall are more accurate with CEN-dec.

We also consider the case of predicting three modalities. To this end, we implement two recent popular methods

including AdaShare (AS) [81] and Task-Grouping (TG) [64] which consider multitask learning by parameter sharing. Table 8 summarizes the experimental results. We find that both AS and TG usually achieve better accuracy than individual learning on some tasks (for example RGB→Depth) but at the sacrifice of other tasks (RGB→SemSeg), probably owing to the negative transfer. Yet, our CEN-dec, which simply shares the encoders with individual INs and performs channel exchanging in the decoder, outperforms all methods by noticeable margins in all tasks, supporting the superiority of channel exchanging for message fusion between different tasks. Interestingly, when combined with TG, the performance of CEN-dec is boosted remarkably, implying the flexibility of integrating our method with other techniques.

4.4 Evaluations on multimodal multitask learning

We evaluate our multimodal multitask CEN as a combination of multimodal fusion and multitask learning, as shown

TABLE 8: Experimental results of simultaneously predicting three tasks. Evaluation metrics and abbreviations follow Table 7. AdaShare (AS) [81] and Taskgrouping (TG) [64] are additionally served as baselines.

Modality		Indiv (IN×3, enc×3, dec×3)	AS [81] (IN×1, enc×2, dec×3)	TG [64] (IN×2~3, enc×2~3, dec×3)	CEN-dec (IN×3, enc×1, dec×3)	CEN-dec + TG [64] (IN×3, enc×2~3, dec×3)
RGB →	SemSeg	26.68 / 40.11	29.50 / 43.71	26.72 / 40.15	25.30 / 39.64	**22.97 / 37.50**
	Depth	5.35 / 9.15	5.02 / 8.71	5.15 / 8.80	4.81 / 8.51	**4.71 / 8.40**
	Normal	18.70 / 37.18	17.57 / 33.80	17.92 / 34.39	17.05 / 33.19	**16.63 / 32.02**
Texture →	RGB	97.45 / 4.80	99.23 / 5.11	97.40 / 4.78	94.04 / 3.76	**92.71 / 3.20**
	Depth	4.24 / 8.19	4.16 / 8.05	4.27 / 8.25	3.19 / 5.05	**3.08 / 4.96**
	Edge	0.97 / 1.73	1.16 / 2.24	0.95 / 1.70	0.90 / 1.66	**0.86 / 1.61**
Normal →	RGB	108.28 / 5.42	114.74 / 5.89	108.13 / 5.40	102.55 / 5.20	**99.18 / 4.86**
	Depth	2.60 / 3.92	2.77 / 4.30	2.41 / 3.80	1.93 / 3.25	**1.83 / 3.01**
	Shade	8.11 / 12.38	7.86 / 11.95	7.75 / 11.77	6.86 / 10.82	**6.79 / 10.73**
Total params. (M)		Gen: 163.1; Dis: 12.5	Gen: 143.7; Dis: 12.5	Gen: 143.7~163.1; Dis: 12.5	Gen: 124.3; Dis: 12.5	Gen: 143.7~163.1; Dis: 12.5

TABLE 9: Experimental results of multimodal multitask learning. Evaluation metrics are FID/KID ($\times 10^{-2}$) for RGB predictions and MAE ($\times 10^{-2}$)/MSE ($\times 10^{-2}$) for other predictions. Lower values indicate better performance. Individual (Indiv) learning is served as the baseline. We provide numbers of groups for instance normalization (IN), encoder (enc) and decoder (dec), and the total parameters (params.) in generator (Gen) and discriminator (Dis). "Curve" and "SemSeg" are abbreviations for the principle curve and semantic segmentation, respectively.

Modality			Indiv (IN×4, enc×2, dec×2)	CEN-enc (IN×4, enc×1, dec×2)	CEN-dec (IN×4, enc×1, dec×2)	CEN-enc & dec (IN×4, enc×1, dec×2)
RGB Depth } →	{	SemSeg	26.86 / 40.24	21.17 / 36.05	25.22 / 39.36	**20.25 / 35.17**
		Curve	5.97 / 16.51	5.49 / 15.30	5.76 / 16.04	**5.27 / 14.93**
RGB Depth } →	{	Nomal	18.68 / 37.11	13.54 / 29.03	16.81 / 32.75	**12.23 / 27.39**
		Shade	8.62 / 12.76	7.37 / 11.09	8.20 / 12.14	**7.08 / 10.91**
RGB Edge } →	{	Depth	4.49 / 9.80	2.81 / 6.77	4.02 / 8.53	**2.47 / 6.33**
		Normal	16.56 / 33.40	13.28 / 29.32	15.14 / 32.72	**12.62 / 28.71**
Texture Shade } →	{	RGB	97.31 / 4.76	62.47 / 1.63	87.50 / 3.72	**60.26 / 1.57**
		Depth	2.66 / 4.20	1.64 / 3.03	2.18 / 3.77	**1.58 / 2.94**
Total params. (M)			Gen: 108.7; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3

in Table 9. We compare four different settings including individual training (Indiv), CEN on the encoder (CEN-enc), CEN on the decoder (CEN-dec), and CEN on both the encoder and decoder (CEN-enc & dec). All the settings maintain four individual INs that correspond to the four different input-output combinations, respectively.

Results indicate that performing CEN either on the encoder or decoder is beneficial compared with the individual training baseline. Generally speaking, CEN-enc obtains more benefits compared with CEN-dec. This is natural as each input modality contains complementary information for predicting each output modality, hence CEN-enc is particularly advantageous. But different output modalities might not be necessarily related, and as a result, CEN-dec gains smaller improvement. As expected, combining CEN-enc and CEN-dec can further improve each of them and delivers the best performance in all considered cases.

4.5 Discussions

Why dividing channels into M sub-parts. We describe in § 3.1 and Fig. 11 that we evenly divide the whole channels into M sub-parts (where M is the number of input modalities), and apply sparsity constraints only to one sub-part for each modality. Otherwise, if we do not divide channels and apply sparsity constraints to all scaling factors for each
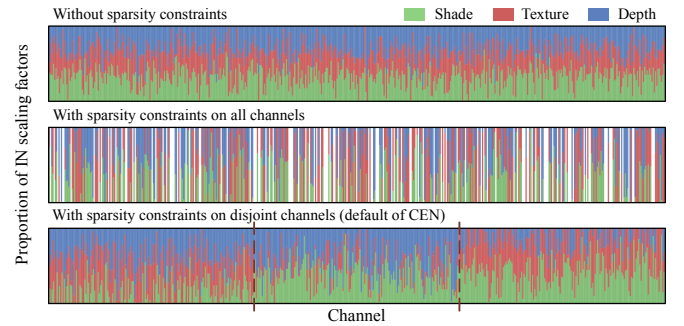

Fig. 7: We adopt shared Convs and unshared INs, and plot the proportion of scaling factors for each modality at the 7th Conv-layer, i.e. $\gamma_{m,c}/(\gamma_{1,c} + \gamma_{2,c} + \gamma_{3,c})$, where $m = 1, 2, 3$ being Shade, Texture and Depth, respectively. Note that we use the white space to represent a channel c if all of the three scaling factors ($\gamma_{1,c}, \gamma_{2,c}, \gamma_{3,c}$) are less than the threshold.

modality, there is likely to be a portion of channels with close-to-zero scaling factors *w.r.t.* all modalities. We provide the illustration in Fig. 12. We observe that with sparsity constraints on all channels, Fig. 12 (middle) has a number of channels with small scaling factors, which are thus considered to be redundant *w.r.t.* all modalities, which might lead to the decline of model capacity. Besides, it is hard to decide

the exchanging direction on these redundant channels based on Eq. 4. We provide corresponding experimental results in Table 1 and the appendix (Table 13).

Typical values of scaling factors. Fig. 8 demonstrates typical values of BN scaling factors vs training steps, consisting of four combinations: within/beyond sparsity constraints, and with/without channel exchanging. Experimental details are provided in the caption of Fig. 8. From the first two subfigures, we observe that whether applying channel exchanging or not, scaling factors that are close to zero can hardly recover (in the later training process). In addition, according to the last two subfigures, it seems channel exchanging increases the learning speed of a portion of scaling factors without sparsity constraints, probably due to the accumulated gradient on both the RGB branch and the depth branch by channel exchanging (Eq. 4).

Effect of zeroing out channels and channel exchanging. This part provides the sensitivity analysis for two essential hyper-parameters of CEN, including the weight λ (Eq. 2) of sparsity constraint, and the threshold θ (Eq. 4) that identifies close-to-zero scaling factors. Experimental details are provided in the caption of Fig. 9. To isolate the advantage of channel exchanging, Fig. 9 (a) indicates that by zeroing out channels with small scaling factors (instead of channel exchanging), the performance slightly drops with the increase of λ or θ since the percentage of zeroing out channels increases accordingly. Nevertheless, such a drop is moderate, given that under the sparsity constraints, the zeroed-out channels are less influential (as analyzed in § 3.1). Fig. 9 (b) provides the sensitivity analysis of our channel exchanging. We observe that both hyper-parameters λ and θ are not sensitive around their default settings. It is also noticeable that without channel exchanging, simply zeroing out channels reaches much inferior performance.

Importance of the exchanging process. We provide additional experiments in the appendix (Table 12), to evaluate the importance of the exchanging process. We try other approaches to replace zeroed-out channels with: concatenated multimodal features (followed by a Conv-layer) instead of the average, evenly spaced channels from the same modality or other modalities, channels with the largest scaling factors, etc. Results indicate the superiority of our current design. In summary, albeit the simplicity of using the average of other modalities in CEN, it is also effective and competitive.

Evaluation for the unsupervised learning. In a part of multimodal fusion tasks, there is no ground truth during training [27], [47], [82]. As a general multimodal/multitask method, CEN is also potentially applicable to unsupervised learning tasks. For example, we apply CEN to the Saliency Network in [82] for RGB-D unsupervised saliency detection, a dense image prediction task aiming to effectively find and segment the most distinctive objects in a scene. Quantitive results and visualizations are provided in the appendix (Fig. 15 and Table 14), where improvements are also achieved, indicating the effectiveness of CEN in this case.

5 CONCLUSION

We propose Channel-Exchanging-Network (CEN), a novel framework for multimodal fusion and multitask learning, which is parameter-free and self-adaptive. The motivation

(a) Scaling factors of the **first** 128 channels (**within** sparsity constraints) when channel exchanging is **NOT** applied

(b) Scaling factors of the **first** 128 channels (**within** sparsity constraints) when channel exchanging is applied

(c) Scaling factors of the **last** 128 channels (**beyond** sparsity constraints) when channel exchanging is **NOT** applied

(d) Scaling factors of the **last** 128 channels (**beyond** sparsity constraints) when channel exchanging is applied
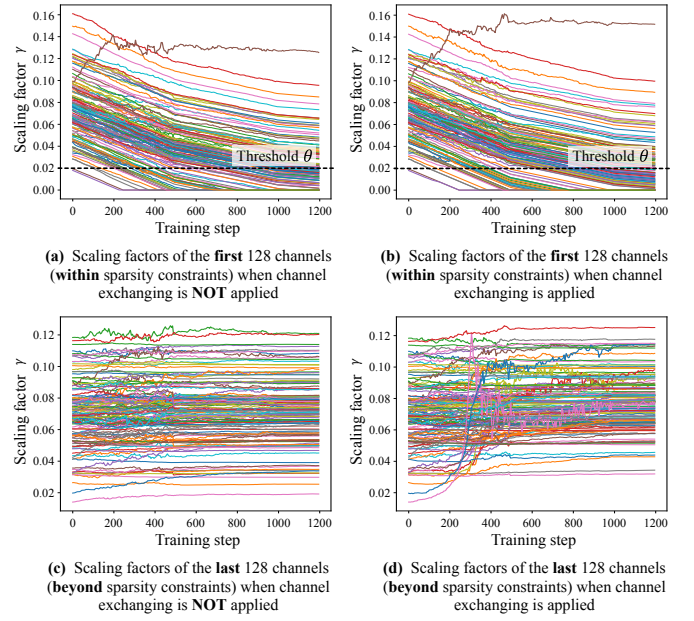
Fig. 8: Typical values of BN scaling factors (*w.r.t.* the RGB modality) within/beyond sparsity constraints vs training steps. We compare circumstances when channel exchanging is and is not applied. Experiments are conducted on NYUDv2 with RefineNet (ResNet101). We choose the 8th layer of convolutional layers that have 3×3 kernels, and there are 256 channels. Regarding RGB, sparsity constraints to scaling factors are applied on the first 128 channels.
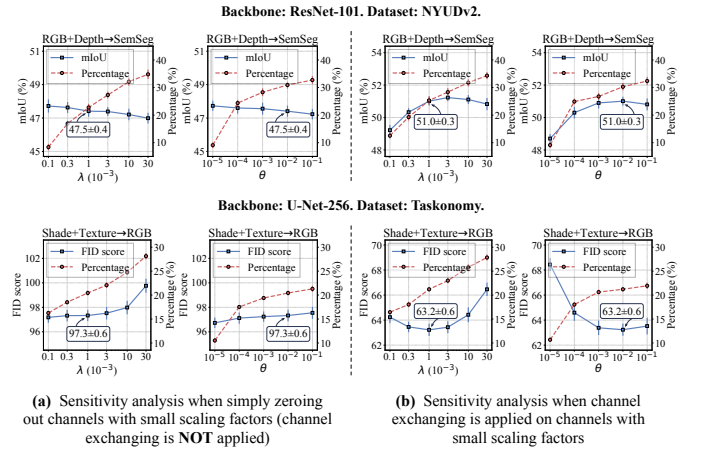

(a) Sensitivity analysis when simply zeroing out channels with small scaling factors (channel exchanging is **NOT** applied)

(b) Sensitivity analysis when channel exchanging is applied on channels with small scaling factors

Fig. 9: Effect when zeroing out channels (without channel exchanging) as well as the sensitivity analysis for λ and θ. Experiments include RGB-D semantic segmentation (SemSeg) on NYUDv2 (top group) and Texture+Shade→RGB on Taskonomy (bottom group). We conduct five experiments for each parameter setting. Default settings are $\lambda = 10^{-3}$ and $\theta = 10^{-2}$. The left y-axis indicates the metric (mIoU ↑ or FID score ↓). The right y-axis indicates the percentage of channels that are lower than θ and these channels will be replaced by zeros (left group) or by cross-modal channels (right group). Metric results at default settings are marked.

behind this is to boost inter-subnetwork fusion while simultaneously keeping sufficient intra-subnetwork processing. The channel exchanging is self-guided by channel impor-

tance measured by individual BNs, making our framework self-adaptive and compact. Extensive evaluations in four cases (multimodal fusion, cycle multimodal fusion, multi-task learning, and multimodal multitask learning) verify the effectiveness of our method.

APPENDIX A
IMPLEMENTATION DETAILS

In our experiments, we adopt ResNet101 and ResNet152 for semantic segmentation, and U-Net-256 for image-to-image translation. We use an NVIDIA Tesla V100 with 32GB for each experiment. Regarding both ResNet structures, we apply sparsity constraints on Batch-Normalization (BN) scaling factors *w.r.t.* each Convolutional-layer (Conv-layer) with 3×3 kernels. These scaling factors further guide the channel exchanging process that exchanges a portion of feature maps after BN. For the Conv-layer with 7×7 kernels at the beginning of ResNet, and all other Conv-layers with 1×1 kernels, we do not apply sparsity constraints or channel exchanging. For U-Net, we apply sparsity constraints on Instance-Normalization (IN) scaling factors *w.r.t.* all Conv-layers (eight layers in total) in the encoder of the generator, and each is followed by channel exchanging.

We mainly adopt three multimodal fusion baselines in our paper, including concatenation, alignment, and self-attention. Regarding the concatenation method, we stack multimodal feature maps along the channel, and then add a 1×1 convolutional layer to reduce the number of channels back to the original number. The alignment fusion method is a re-implementation of [31], and we follow its default settings for hyper-parameter, *e.g.* using 11 kernel functions for the multiple kernel Maximum Mean Discrepancy. The self-attention method is a re-implementation of the SSMA block proposed in [4], where we also follow the default settings, *e.g.* setting the channel reduction ratio η to 16.

In Table 2 of our main paper, we adopt early, middle, late and all-stage fusion for each baseline method. In ResNet101, there are four stages with 3, 4, 23, and 3 blocks, respectively. The early fusion, middle fusion, and late fusion refer to fusing after the 2nd stage, 3rd stage, and 4th stage respectively. All-stage fusion refers to fusing after the four stages.

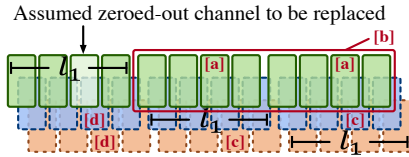Assumed zeroed-out channel to be replaced

Fig. 10: Illustrations of channels as a complement to Table 12.

We now introduce the metrics (including FID and KID) used in our image-to-image translation task.

Firstly, Fréchet-Inception-Distance (FID) [76] mainly contrasts the statistics of generated samples against real samples. FID fits a Gaussian distribution to the hidden activations of InceptionNet for each compared image set and then computes the Fréchet distance (also known as the Wasserstein-2 distance) between those Gaussians. Lower FID is better, indicating that the generated images are more similar to the real ones.

TABLE 10: We compare training multimodal features in a parallel manner with different parameter sharing settings. Results of the proposed fusion method are reported in the last column. Evaluation metrics are FID/KID ($\times 10^{-2}$). We observe that the convolutional layers can be shared as long as we leave individual INs for different modalities, achieving even better performance.

Modality	Network stream	Unshared Convs unshared INs	Shared Convs shared INs	Shared Convs unshared INs	Multi-modal fusion
Shade +Texture →RGB	Shade	102.21 / 5.25	112.40 / 5.58	100.69 / 4.51	72.07 / 2.32
	Texture	98.19 / 4.83	102.28 / 5.22	93.40 / 4.18	65.60 / 1.82
	Ensemble	92.72 / 4.15	96.31 / 4.36	87.91 / 3.73	62.63 / 1.65
Shade +Texture +Depth →RGB	Shade	101.86 / 5.18	115.51 / 5.77	98.49 / 4.07	69.37 / 2.21
	Texture	98.60 / 4.89	104.39 / 4.54	95.87 / 4.27	64.70 / 1.73
	Depth	114.18 / 5.71	121.40 / 6.23	107.07 / 5.19	71.61 / 2.27
	Ensemble	91.30 / 3.92	100.41 / 4.73	84.39 / 3.45	58.35 / 1.42
Shade +Texture +Depth +Normal →RGB	Shade	100.83 / 5.06	131.74 / 7.48	96.98 / 4.23	68.70 / 2.14
	Texture	97.34 / 4.77	109.45 / 4.86	94.64 / 4.22	63.26 / 1.69
	Depth	114.50 / 5.83	125.54 / 6.48	109.93 / 5.41	70.47 / 2.09
	Normal	108.65 / 5.45	113.15 / 5.72	99.38 / 4.45	67.73 / 1.98
	Ensemble	89.52 / 3.80	102.78 / 4.67	86.76 / 3.63	57.19 / 1.33

TABLE 11: An Instance-Normalization layer consists of four components, including scaling factors γ, offsets β, running mean μ and variance σ. Following Table 5, we further compare the evaluation results with only unshared γ, β, or only unshared μ, σ. Evaluation metrics are FID/KID ($\times 10^{-2}$). We observe that using unshared scaling factors and offsets seems to be more important. ℓ_1 regulation and channel exchanging are not applied throughout these experiments.

Modality	Network stream	Unshared Convs unshared INs	Shared Convs unshared INs	Shared Convs,γ,β unshared μ,σ	Shared Convs,μ,σ unshared γ,β
Shade +Texture +Depth →RGB	Shade	101.86 / 5.18	98.49 / 4.07	107.86 / 5.53	105.29 / 5.29
	Texture	98.60 / 4.89	95.87 / 4.27	105.46 / 5.25	102.90 / 5.06
	Depth	114.18 / 5.71	102.07 / 4.89	118.35 / 6.07	114.35 / 5.80
	Ensemble	91.30 / 3.92	84.39 / 3.45	96.30 / 4.41	92.25 / 4.02
Shade +Texture +Depth +Normal →RGB	Shade	100.83 / 5.06	96.98 / 4.23	113.56 / 5.65	102.74 / 5.17
	Texture	97.34 / 4.77	94.64 / 4.22	105.36 / 5.32	97.53 / 4.56
	Depth	114.50 / 5.83	109.93 / 5.41	119.31 / 6.20	112.73 / 5.60
	Normal	108.65 / 5.45	99.38 / 4.45	108.01 / 5.06	100.34 / 4.53
	Ensemble	89.52 / 3.80	86.76 / 3.63	95.56 / 4.64	89.26 / 3.91

Secondly, Kernel-Inception-Distance (KID) [77] is a metric similar to the FID score but uses the squared Maximum-Mean-Discrepancy (MMD) between Inception representations with a polynomial kernel. Unlike FID, KID has a simple unbiased estimator, making it more reliable especially when there are much more inception feature channels than image numbers. Lower KID indicates more visual similarity between real and generated images. Regarding our implementation of KID, the hidden representations are derived from the Inception-v3 pool3 layer.

APPENDIX B
ADDITIONAL DISCUSSIONS AND RESULTS

Dividing channels into sub-parts. Here we provide additional descriptions of why dividing channels into M sub-parts and individually applying sparsity constraints. We first use the case with 2 modalities for example. As shown in Fig. 11, we divide channels into 2 disjoint sub-parts and apply sparsity constraints. During training, each certain channel belongs to one of the three categories: **A** (where

TABLE 12: Additional experiments on the NYUDv2 dataset based on RefineNet (ResNet101) to evaluate the importance of the exchanging process. Results include multimodal fusion on image translation (to RGB) with $2 \sim 4$ input modalities. Evaluation metrics are FID/KID ($\times 10^{-2}$) and lower values indicate better performance.

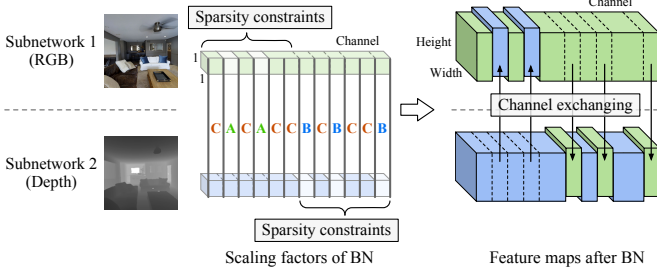| For modality m, replacing (the feature map) at a zeroed-out channel (channel index i) with:	Depth+Normal	Depth+Normal +Texture	Depth+Normal +Texture+Shade
A zero embedding (only fusion by ensemble)	107.32 / 5.39	96.90 / 4.75	95.50 / 4.68
The i-th channel from another one modality $m' \neq m$	Same with CEN	63.14 / 1.73	62.76 / 1.69
Average of evenly spaced channels [a] (Fig. 10) (beyond sparsity constraints) from the same modality m	106.62 / 5.29	95.63 / 4.64	95.90 / 4.71
One random channel in the region [b] (Fig. 10) (beyond sparsity constraints) from the same modality m	109.71 / 5.62	97.06 / 4.92	96.64 / 4.85
Average of evenly spaced channels [c] (Fig. 10) from other modalities $\forall m' \neq m$	89.52 / 3.56	68.11 / 2.06	66.15 / 1.91
Average of channels including both [c] and [d] (Fig. 10) from other modalities $\forall m' \neq m$	85.08 / 2.92	64.73 / 1.82	61.99 / 1.64
Average of unused channels with the largest scaling factors from other modalities $\forall m' \neq m$	88.61 / 3.13	68.09 / 2.10	68.87 / 2.15
Weighted average of the i-th channels [d] (Fig. 10) from other modalities $\forall m' \neq m$	Same with CEN	61.07 / 1.59	58.26 / 1.40
Concatenation (followed by a 1×1 Conv) of the i-th channels [d] (Fig. 10) from other modalities $\forall m' \neq m$	Same with CEN	63.32 / 1.73	61.70 / 1.64
Average of the i-th channels [d] (Fig. 10) from other modalities $\forall m' \neq m$ (our CEN)	**84.33 / 2.70**	**60.90 / 1.56**	**57.19 / 1.33**

Fig. 11: An illustration of CEN. The sparsity constraints on scaling factors are applied to disjoint channel regions of different modalities. As annotated, each channel is categorized to **A**, **B**, or **C**, based on its γ_{rgb} and γ_{depth}.
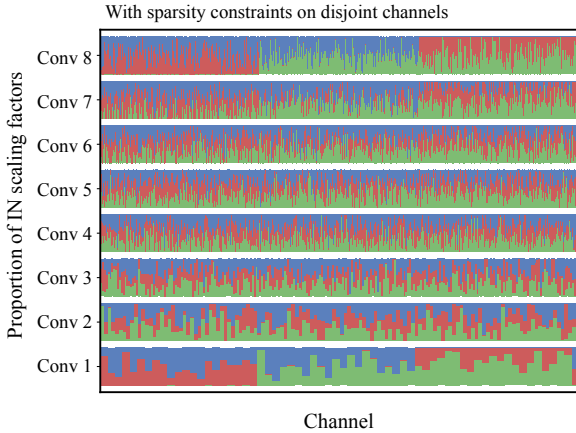

Fig. 12: We adopt shared Convs and unshared INs, and plot the proportion of scaling factors for each modality at each Conv-layer, *i.e.* $\gamma_{m,c}/(\gamma_{1,c} + \gamma_{2,c} + \gamma_{3,c})$, where $m = 1, 2, 3$ being Shade, Texture and Depth, respectively. Proportion of scaling factors in all Conv-layers, and sparsity constraints are applied on disjoint channels.

$\gamma_{rgb} \approx 0, \gamma_{depth} > 0$), **B** (where $\gamma_{rgb} > 0, \gamma_{depth} \approx 0$), and **C** (where $\gamma_{rgb} > 0, \gamma_{depth} > 0$). There won't be ($\gamma_{rgb} \approx 0, \gamma_{depth} \approx 0$) as we apply sparsity constraints on disjoint sub-parts. However, if we apply sparsity constraints on all scaling factors for each modality (without dividing 2 sub-parts), there is likely be a portion of channels with close-to-zero scaling factors *w.r.t.* both modalities, *i.e.*,

TABLE 13: Multimodal fusion on image translation (to RGB) with or without (w/o) dividing channels into M sub-parts. Evaluation metrics are FID/KID ($\times 10^{-2}$). Lower values indicate better performance for both metrics.

Method	Depth+Normal	Depth+Normal +Texture	Depth+Normal +Texture+Shade
Dividing M sub-parts (default)	84.33 / 2.70	60.90 / 1.56	57.19 / 1.33
W/o dividing M sub-parts	87.63 / 3.49	65.12 / 1.90	64.87 / 1.85

($\gamma_{rgb} \approx 0, \gamma_{depth} \approx 0$). These channels are considered to be unimportant/redundant for both modalities. Regarding multimodal fusion, it is kind of waste of channels, which might lead to the decline of model capacity. Besides, it is hard to decide the exchanging direction on these channels according to Eq. 4 (main paper).

Similarly, when there are 3 (or M) modalities as input, dividing the whole channels into 3 (or M) sub-parts avoids a channel from being redundant for all modalities. As a result, we divide channels into M sub-parts and apply sparsity constraints on one sub-part for each modality. As an example, for Shade+Texture+Depth$\rightarrow$RGB image-to-image translation with shared Convs and unshared INs, channels are evenly divided into three sub-parts. We plot the proportion of IN scaling factors at each Conv-layer in the encoder of U-Net in Fig. 12.

Comparison results to support the channel dividing have been provided in Table 1 of our main paper: semantic segmentation "All-channel" (49.8) vs "Half-channel" (51.1). Additional results for image-to-image translation are shown in Table 13. All these results indicate the superiority of applying sparsity constraints on sub-parts of channels.

Effect of network sharing. In Table 10, we verify that sharing convolutional layers (Convs) but using individual Instance-Normalization layers (INs) allows 2~4 modalities trained in a single network, and even achieve better performance than training with individual networks. Again, if we further share INs, there will be an obvious performance drop. More detailed comparison is provided in Table 11.

Visualization of indoor experiments. We provide additional visualizations of the image-to-image translation task in Fig. 13, as a complement to Fig. 5 (main paper). Regarding baseline implementation in all these visualizations, we adopt all-layer fusion (fusion at all eight Conv-layers in the encoder) for concatenation and self-attention methods, and
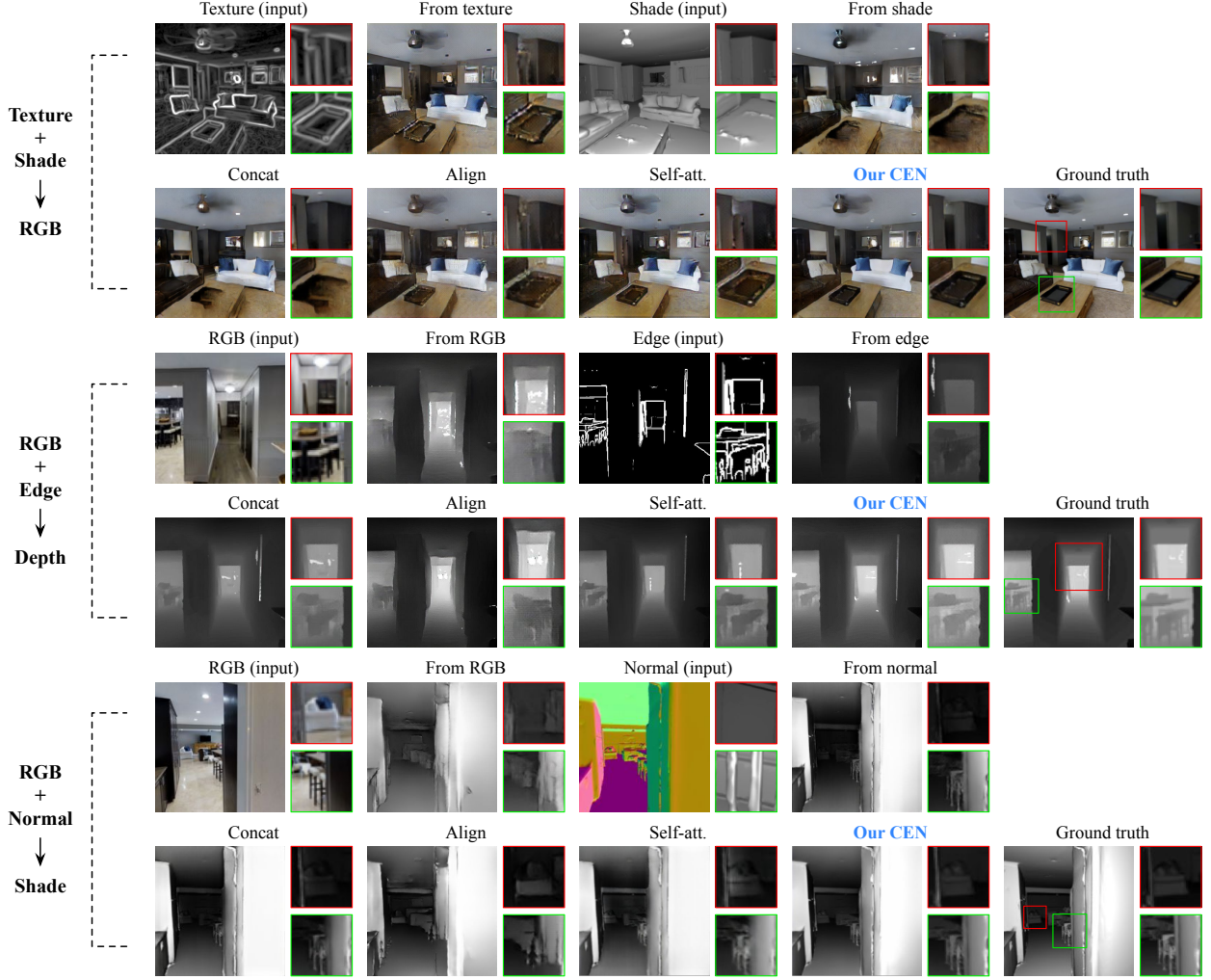
Fig. 13: Additional visualization results of image-to-image translation including Texture+Shade→RGB (top group), RGB+Edge→Depth (middle group), and RGB+Normal→Shade (bottom group), respectively. The resolution of each predicted image is 256×256.

adopt middle fusion (fusion at the 4th Conv-layer) for the alignment method. These settings achieve relatively high performance regarding baseline methods according to their numerical results.

Visualization of outdoor experiments. In this part, we additionally conduct outdoor semantic segmentation experiments on the Cityscapes dataset [83] and provide the visualization comparison. Cityscapes is an outdoor dataset containing images from 27 cities in Germany and neighboring countries. The dataset contains 2,975 training, 500 validation and 1,525 test images. There are 20,000 additional coarse annotations provided by the dataset, which are not used for training in our experiments. All results are obtained with the backbone PSPNet (ResNet101) of single-scale evaluation for test. These visualizations are provided in Fig 14.

Evaluation for unsupervised learning. Apart from the common supervised training settings for dense image prediction [99], [100], we show the potential of CEN for unsupervised learning, *e.g.*, [27], [47], [82]. As an example shown in Fig. 15 and Table 14, we apply CEN to the Saliency Network in [82] for RGB-Depth unsupervised saliency detection, which also achieves promising results, indicating

TABLE 14: Quantitave results of applying CEN to the unsupervised RGB-D saliency detection. We follow the training settings in [82]. Evaluation datasets include NJUD [84], NLPR [85], STERE [86], and DUTLF-Depth [87]. We adopt Mean Absolute Error (MAE) [88] as the evaluation metric following [82]. Lower values indicate better performance.

Method	NJUD	NLPR	STERE	DUTLF-Depth
MST [89]	.281	.199	.269	.279
BSCA [90]	.216	.178	.179	.181
GP [91]	.204	.144	.182	-
CDB [92]	.200	.108	.166	-
SE [93]	.164	.085	.143	.196
DCMC [94]	.167	.196	.148	.243
MB [95]	.202	.089	.178	.156
CDCP [96]	.181	.114	.149	.159
USD [97]	.163	.119	.146	.157
DeepUSPS [98]	.159	.088	.124	.149
SP [82] (RGB-D)	.135	.065	.099	.107
CEN (RGB-D)	**.132**	**.059**	**.095**	**.103**

the effectiveness of CEN in this case. Further discussion of unsupervised learning is left for future exploration.

Enlarging the sampled dataset. In our image-to-image translation experiments on the Taskonomy dataset, we find that CEN with 1,000 training images already achieves
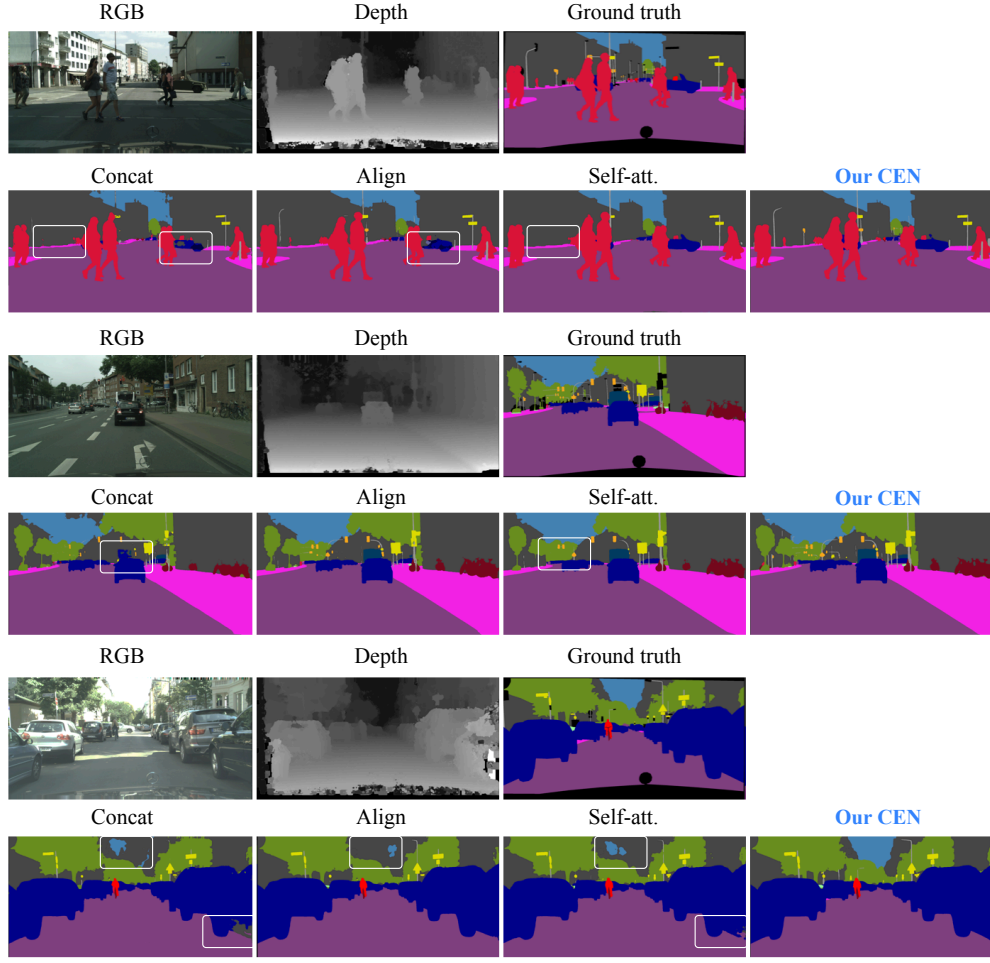
Fig. 14: Visualization for the semantic segmentation on Cityscapes [83]. For the baseline methods, we use white frames to highlight the regions with poor prediction results. We observe that when the light intensity is high, baseline methods are weak in capturing the boundary between the sky and buildings using the depth information. Besides, the concatenation and self-attention methods do not preserve fine-grained objects, *e.g.* traffic signs, and are sensitive to noises of the depth input (see the rightmost vehicle in the last group). In contrast, the prediction of our CEN is better in these aspects.

Fig. 15: Visualization of applying CEN to the unsupervised RGB-D saliency detection. We compare our method with another RGB-D-based method Promoting Saliency (PS) [82] which recently achieves SOTA.

promising results. Here, we enlarge the sampled set with 15,000 training images and conduct the experiments. Since the training cost becomes quite large given the 15× expansion of the default sampled training dataset, we choose typical experiments to evaluate our CEN. Results provided in Table 15 include multimodal fusion, cycle multimodal fusion, multitask learning, and multimodal multitask learning based on 15,000 sampled training images. By comparison, we observe that mostly, training with 1,000 images and training with 15,000 images achieve similar relative improvements of CEN over baselines. These results indicate that using 1,000 images for training already demonstrates the general advantages of CEN.

REFERENCES

[1] T. Baltrusaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. PAMI*, 2019. 1, 2

[2] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Processing Magazine*, 2017. 1

[3] G. Lin, F. Liu, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for dense prediction," *IEEE Trans. PAMI*, 2019. 1, 6, 8

[4] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *IJCV*, 2020. 1, 2, 7, 8, 14

[5] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding," in *CVPR*, 2018. 1

TABLE 15: Comparison on multimodal fusion, cycle multimodal fusion, multitask learning, and multimodal multitask learning with 15,000 sampled training images (instead of 1,000 for other experiments). Evaluation metrics are FID/KID ($\times 10^{-2}$) for RGB predictions and MAE ($\times 10^{-2}$)/MSE ($\times 10^{-2}$) for other predictions. Lower values indicate better performance for all metrics. "Curve", "SemSeg", and "X-TC" are abbreviations for the principle curve, semantic segmentation, and Cross-Task Consistency [65] respectively. More abbreviation details follow the captions of Table 4, 6, 7, 9 in our paper.

Multimodal fusion	Concat	Average	Align	Self-att.	Our CEN
Shade+Texture → RGB	75.39 / 2.77	75.46 / 2.82	86.20 / 3.92	68.65 / 2.23	**56.94 / 1.45**
Depth+Normal → RGB	91.64 / 3.38	93.81 / 3.50	97.05 / 3.99	88.60 / 3.02	**79.68 / 2.59**

Cycle multimodal fusion	CEN (IN×6, enc×3, dec×3)	CEN-random (IN×6, enc×1, dec×3)	CEN-cycle (IN×6, enc×1, dec×3)	CEN-cycle (IN×6, enc×1, dec×1)
RGB+Shade → Texture	1.59 / 2.80	1.94 / 4.07	**1.36 / 2.21**	1.45 / 2.34
RGB+Texture → Shade	13.77 / 22.78	16.93 / 24.02	**12.91 / 21.77**	13.13 / 22.09
Shade+Texture → RGB	56.94 / 1.45	66.51 / 2.18	**55.26 / 1.38**	56.03 / 1.43
RGB+Depth → SemSeg	18.49 / 33.20	20.03 / 35.44	**15.90 / 30.41**	16.72 / 31.03
RGB+SemSeg → Depth	3.72 / 7.54	4.29 / 8.02	**3.50 / 7.07**	3.54 / 7.31
Depth+SemSeg → RGB	93.95 / 3.57	96.40 / 3.65	**91.72 / 3.30**	92.44 / 3.51
Total params. (M)	Gen: 163.3; Dis: 8.3	Gen: 124.2; Dis: 8.3	Gen: 124.2; Dis: 8.3	Gen: **54.5**; Dis: 8.3

Multitask learning	Indiv (IN×2, enc×2, dec×2)	Indiv (IN×2, enc×1, dec×2)	X-TC [65] (IN×2, enc×1, dec×2)	CEN-dec (IN×2, enc×1, dec×2)	CEN-dec + X-TC [65] (IN×2, enc×1, dec×2)
RGB → SemSeg	23.13 / 36.95	22.93 / 36.51	20.19 / 35.09	19.44 / 34.36	**18.79 / 33.75**
RGB → Depth	4.93 / 8.75	4.72 / 8.50	4.17 / 7.60	3.99 / 7.78	**3.70 / 7.56**
RGB → Normal	15.30 / 31.19	15.92 / 32.04	14.35 / 30.28	13.19 / 28.61	**12.02 / 27.66**
RGB → Curve	5.04 / 14.42	4.97 / 14.22	4.24 / 13.05	4.05 / 12.86	**3.32 / 11.29**
Total params. (M)	Gen: 108.7; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3

Multimodal multitask learning	Indiv (IN×4, enc×2, dec×2)	CEN-enc (IN×4, enc×1, dec×2)	CEN-dec (IN×4, enc×1, dec×2)	CEN-enc & dec (IN×4, enc×1, dec×2)
RGB, Depth → SemSeg	23.55 / 37.23	18.68 / 33.71	20.52 / 35.79	**17.86 / 32.94**
RGB, Depth → Curve	4.94 / 14.22	4.25 / 13.30	4.63 / 13.87	**4.04 / 12.68**
RGB, Depth → Nomal	15.71 / 31.60	11.35 / 27.13	13.64 / 29.19	**10.62 / 25.28**
RGB, Depth → Shade	7.09 / 11.43	6.13 / 9.97	6.94 / 10.65	**5.83 / 9.25**
Total params. (M)	Gen: 108.7; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3	Gen: 89.3; Dis: 8.3

[6] N. C. Garcia, P. Morerio, and V. Murino, "Modality distillation with multiple stream networks for action recognition," in *ECCV*, 2018. 1

[7] S. Song, J. Liu, Y. Li, and Z. Guo, "Modality compensation network: Cross-modal adaptation for action recognition," *IEEE Trans. Image Process.*, 2020. 1, 2

[8] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: visual question answering," in *ICCV*, 2015. 1

[9] I. Ilievski and J. Feng, "Multimodal learning and reasoning for visual question answering," in *NIPS*, 2017. 1

[10] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T. Kim, "Pose guided RGBD feature learning for 3d object pose estimation," in *ICCV*, 2017. 1

[11] W. Jin, K. Yang, R. Barzilay, and T. S. Jaakkola, "Learning multi-modal graph-to-graph translation for molecule optimization," in *ICLR*, 2019. 1

[12] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *ICCV*, 2019. 1

[13] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowledge and Data Engineering*, 2021. 1

[14] D. Zhou, J. Wang, B. Jiang, H. Guo, and Y. Li, "Multi-task multi-view learning based on cooperative multi-objective optimization," *IEEE Access*, 2017. 1

[15] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *CVPR*, 2016. 1, 3

[16] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *CVPR*, 2019. 1

[17] P. Guo, C.-Y. Lee, and D. Ulbricht, "Learning to branch for multi-task learning," in *ICML*, 2020. 1

[18] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *ICML*, 2020. 1

[19] X. Sun, R. Panda, R. Feris, and K. Saenko, "Adashare: Learning what to share for efficient deep multi-task learning," *NeurIPS*, 2020. 1

[20] J. Andreas, D. Klein, and S. Levine, "Modular multitask reinforcement learning with policy sketches," in *ICML*, 2017. 1

[21] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *ICRA*, 2018. 1

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015. 1, 8

[23] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. PAMI*, 2018. 1

[24] S. Huang, Z. Lu, R. Cheng, and C. He, "Fapn: Feature-aligned pyramid network for dense image prediction," in *ICCV*, 2021. 1

[25] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017. 1, 6

[26] X. Huang, M. Liu, S. J. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018. 1

[27] J. Ma, Y. Ma, and C. Li, "Infrared and visible image fusion methods and applications: A survey," *Inf. Fusion*, 2019. 1, 13, 16

[28] H. Lee, H. Tseng, Q. Mao, J. Huang, Y. Lu, M. Singh, and M. Yang, "DRIT++: diverse image-to-image translation via disentangled representations," *IJCV*, 2020. 1

[29] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture," in *ACCV*, 2016. 1, 2, 8

[30] J. Zeng, Y. Tong, Y. Huang, Q. Yan, W. Sun, J. Chen, and Y. Wang, "Deep surface normal estimation with hierarchical RGB-D fusion," in *CVPR*, 2019. 1, 2, 7

[31] J. Wang, Z. Wang, D. Tao, S. See, and G. Wang, "Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks," in *ECCV*, 2016. 1, 2, 7, 14

[32] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *CVPR*, 2017. 1, 3

[33] S. Chennupati, G. Sistu, S. Yogamani, and S. A Rawashdeh, "Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning," in *CVPR Workshops*, 2019. 1, 3

[34] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *AAAI*, 2019. 1, 3

[35] D. Du, L. Wang, H. Wang, K. Zhao, and G. Wu, "Translate-to-recognize networks for RGB-D scene recognition," in *CVPR*, 2019. 1, 2

[36] S. Lee, S. Park, and K. Hong, "Rdfnet: RGB-D multi-level residual feature fusion for indoor semantic segmentation," in *ICCV*, 2017. 1, 2, 6, 8

[37] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," *arXiv preprint arXiv:1810.04650*, 2018. 1

[38] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *ICCV*, 2017. 2, 3

[39] J. Ye, X. Lu, Z. Lin, and J. Z. Wang, "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers," in *ICLR*, 2018. 2, 3

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015. 2, 3

[41] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016. 2

[42] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *ECCV*, 2012. 2, 6

[43] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *CVPR*, 2015. 2, 6

[44] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *CVPR*, 2018. 2, 6

[45] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *ICML*, 2011. 2

[46] W. Zhou, J. Yuan, J. Lei, and T. Luo, "Tsnet: Three-stream self-attention network for RGB-D indoor semantic segmentation," *IEEE Intell. Syst.*, 2021. 2

[47] H. Xu, J. Ma, J. Jiang, X. Guo, and H. Ling, "U2fusion: A unified unsupervised image fusion network," *IEEE Trans. PAMI*, 2022. 2, 3, 13, 16

[48] D. Lin, G. Chen, D. Cohen-Or, P. Heng, and H. Huang, "Cascaded feature network for semantic segmentation of RGB-D images," in *ICCV*, 2017. 2, 8

[49] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel two-sample test," in *JMLR*, 2012. 2

[50] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *NIPS*, 2016. 2

[51] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia systems*, 2010. 2

[52] E. Bruni, N.-K. Tran, and M. Baroni, "Multimodal distributional semantics," in *Journal of Artificial Intelligence Research*, 2014. 2

[53] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, 1997. 2

[54] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *ACM MM*, 2005. 2

[55] A. Lazaridou, E. Bruni, and M. Baroni, "Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world," in *ACL*, 2014. 2

[56] Y. Wang, F. Sun, M. Lu, and A. Yao, "Learning deep multimodal feature representation with asymmetric multi-layer fusion," in *ACM MM*, 2020. 2

[57] H. de Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *NIPS*, 2017. 2, 3, 8

[58] H. De Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville, "Guesswhat?! visual object discovery through multi-modal dialogue," in *CVPR*, 2017. 2

[59] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. d. Vries, A. Courville, and Y. Bengio, "Feature-wise transformations," in *Distill*, 2018. 2

[60] M. Long, Z. Cao, J. Wang, and P. S. Yu, "Learning multiple tasks with multilinear relationship networks," *arXiv preprint arXiv:1506.02117*, 2015. 3

[61] M. Suteu and Y. Guo, "Regularizing deep multi-task networks using orthogonal gradients," *arXiv preprint arXiv:1912.06844*, 2019. 3

[62] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, "Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction," in *CVPR*, 2019. 3

[63] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017. 3

[64] T. Standley, A. R. Zamir, D. Chen, L. J. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *ICML*, 2020. 3, 11, 12

[65] A. R. Zamir, A. Sax, N. Cheerla, R. Suri, Z. Cao, J. Malik, and L. J. Guibas, "Robust learning through cross-task consistency," in *CVPR*, 2020. 3, 11, 18

[66] S. Vandenhende, S. Georgoulis, and L. V. Gool, "Mti-net: Multi-scale task interaction networks for multi-task learning," in *ECCV*, 2020. 3

[67] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017. 3

[68] W. Shao, S. Tang, X. Pan, P. Tan, X. Wang, and P. Luo, "Channel equilibrium networks for learning deep representation," in *ICML*, 2020. 3

[69] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *CVPR*, 2018. 3

[70] W. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *CVPR*, 2019. 4

[71] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *CVPR*, 2017. 5

[72] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *CVPR*, 2013. 6

[73] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017. 6

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 6

[75] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *IJCV*, 2015. 6

[76] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*, 2017. 6, 14

[77] M. Binkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD gans," in *ICLR*, 2018. 6, 14

[78] X. Hu, K. Yang, L. Fei, and K. Wang, "ACNET: attention based network to exploit complementary features for RGBD semantic segmentation," in *ICIP*, 2019. 8

[79] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for RGBD semantic segmentation," in *ICCV*, 2017. 8

[80] D. Lin, R. Zhang, Y. Ji, P. Li, and H. Huang, "SCN: switchable context network for semantic segmentation of RGB-D images," *IEEE Trans. Cybern.*, 2020. 8

[81] X. Sun, R. Panda, R. Feris, and K. Saenko, "Adashare: Learning what to share for efficient deep multi-task learning," in *NeurIPS*, 2020. 11, 12

[82] W. Ji, J. Li, Q. Bi, C. Guo, J. Liu, and L. Cheng, "Promoting saliency from depth: Deep unsupervised RGB-D saliency detection," in *ICLR*, 2022. 13, 16, 17

[83] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016. 16, 17

[84] R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu, "Depth saliency based on anisotropic center-surround difference," in *ICIP*, 2014. 16

[85] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "RGBD salient object detection: a benchmark and algorithms," in *ECCV*, 2014. 16

[86] Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *CVPR*, 2012. 16

[87]   Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu, "Depth-induced multi-scale recurrent attention network for saliency detection," in *ICCV*, 2019. 16

[88]   A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, 2015. 16

[89]   W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *CVPR*, 2016. 16

[90]   Y. Qin, H. Lu, Y. Xu, and H. Wang, "Saliency detection via cellular automata," in *CVPR*, 2015. 16

[91]   J. Ren, X. Gong, L. Yu, W. Zhou, and M. Ying Yang, "Exploiting global priors for RGB-D saliency detection," in *CVPR Workshops*, 2015. 16

[92]   F. Liang, L. Duan, W. Ma, Y. Qiao, Z. Cai, and L. Qing, "Stereoscopic saliency model using contrast and depth-guided-background prior," *Neurocomputing*, vol. 275, 2018. 16

[93]   J. Guo, T. Ren, and J. Bei, "Salient object detection for RGB-D image via saliency evolution," in *ICME*, 2016. 16

[94]   R. Cong, J. Lei, C. Zhang, Q. Huang, X. Cao, and C. Hou, "Saliency detection for stereoscopic images based on depth confidence analysis and multiple cues fusion," *IEEE Signal Processing Letters*, 2016. 16

[95]   C. Zhu, G. Li, X. Guo, W. Wang, and R. Wang, "A multilayer backpropagation saliency detection algorithm based on depth mining," in *ICCAIP*, 2017. 16

[96]   C. Zhu, G. Li, W. Wang, and R. Wang, "An innovative salient object detection using center-dark channel prior," in *ICCV Workshops*, 2017. 16

[97]   J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley, "Deep unsupervised saliency detection: A multiple noisy labeling perspective," in *CVPR*, 2018. 16

[98]   T. Nguyen, M. Dax, C. K. Mummadi, N. Ngo, T. H. P. Nguyen, Z. Lou, and T. Brox, "DeepUSPS: Deep robust unsupervised saliency prediction via self-supervision," in *NeurIPS*, 2019. 16

[99]   Y. Wang, W. Huang, F. Sun, T. Xu, Y. Rong, and J. Huang, "Deep multimodal fusion by channel exchanging," in *NeurIPS*, 2020. 16

[100] Y. Wang, X. Chen, L. Cao, W. Huang, F. Sun, and Y. Wang, "Multimodal token fusion for vision transformers," in *CVPR*, 2022. 16