Training of Multiple and Mixed Tasks With A Single Network Using Feature Modulation

Mana Takeda, Gibran Benitez, and Keiji Yanai

The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 Japan
{takeda-m,gibran,yanai}@mm.inf.uec.ac.jp

Fig. 1. The proposed network can learn multiple mixed tasks (Right) as well as multiple tasks (Left) on a single model with negligibly small task-specific parts.

Abstract. In recent years, multi-task learning (MTL) for image translation tasks has been actively explored. For MTL image translation, a network consisting of a shared encoder and multiple task-specific decoders is commonly used. In this case, half parts of the network are task-specific, which brings a significant increase in the number of parameters when the number of tasks increases. Therefore, task-specific parts should be as small as possible. In this paper, we propose a method for MTL image translation using a single network with negligibly small task-specific parts, in which we share not only the encoder part but also the decoder part. In the proposed method, activation signals are adjusted for each task using Feature-wise Linear Modulation (FiLM) which performs affine transformation based on task conditional signals. In addition, we tried to let a single network learn mixing of heterogeneous tasks such as a mix of semantic segmentation and style transfer. With several experiments, we demonstrate that a single network is able to learn heterogeneous image translation tasks and their mixed tasks by following our proposed method. In addition, despite its small model size, our network achieves better performance than some of the latest baselines in most of the individual tasks.

1 Introduction

Deep Convolutional Neural Network (CNN) has achieved great success in various image transformation tasks such as semantic segmentation, style transfer,

and coloring of grayscale photos. In general, each of these tasks is trained with a single network independently, which is called as "Single Task Learning (STL)." On the other hand, by employing Multi Task Learning (MTL), multiple tasks can be trained with one network. Therefore, MTL is considered to be desirable in actual applications that need to process multiple tasks with limited resources. However, MTL models currently proposed for image translation tasks [19,29,23] require task-specific parts, which are sometime about half portion of the whole network, in addition to the parts shared by all the tasks. Typically, the MTL image translation network consists of a shared encoder and multiple task-specific decoders. In this case, the number of network parameters greatly increases as the number of tasks increases. Learning of multiple tasks with a single network consisting of a shared encoder and a shared decoder would have the advantage that the size of the network does not depend on the number of tasks involved. However, the distributions of the activations generally differ from task to task when multiple tasks are trained with a single network. It is necessary to adjust the distribution of the activations to the task-dependent distribution for each task. One of the techniques to enable adjusting the distribution of activations is Feature-wise Linear Modulation (FiLM) [32,5], which is a formalization of the conditional affine transformation. By applying the affine transformation based on the conditional signal to the network, the activation distribution can be adjusted for each task.

Furthermore, Dumoulin et al. [6], took advantage of the FiLM characteristics by proposing a multiple style transfer. They demonstrated that it was possible to mix multiple styles by providing a mixed conditional vector in addition to the conditional single style selection. Their results implied the possibility of extending mixing of multiple different styles to mixing of multiple different image transformation tasks. Therefore, in our work, we explore learning of mixing of multiple heterogeneous image translation tasks with a single network using FiLM, as well as learning of multiple tasks. We call this "mixed-task learning." Mixed-task learning is an unsolved problem in the existing MTL works. If mixed-task learning is possible with a single network, the trained network can process new tasks by combining existing tasks, which can be regarded as the first step to realize a general purpose image translation network. This is similar to arbitrary style transfer [10] which performs fast style transfer with any unknown styles by adaptive mixing of trained known styles. Note that in this paper, we assume two kinds of mixing of heterogeneous tasks: (1) sequential mixing and (2) mixing by region masking. "Sequential mixing" is mixing of multiple image translation tasks sequentially such as applying denoising first and applying style transfer next, while "mixing by region masking" is masking out the output image processed by one task using region masks estimated by the semantic segmentation task such as style transfer on only specific object region. Fig.1 shows some results on learning of both multiple and mixed tasks by our method.

The major contributions of this paper can be summarized as follows: (1) We propose a single network capable of learning multiple heterogeneous image translation tasks with negligibly small task-specific parts, which is based on FiLM. (2) We enable mixed-task learning with the proposed network using synthesized mixed-task training samples. (3) We demonstrate the effectiveness of our proposed network on several experiments, which achieves better performance than

the latest baselines (SGN [2] and Piggyback [25]), even with a smaller number of parameters.

2 Related Work

Multi-task Learning (MTL): In MTL, it has been shown that joint learning of multiple tasks leads to an improvement in accuracy [11,13,33,7,8,38]. However, feature sharing among multiple tasks can have a negative or positive effect depending on the combination of tasks (task interference) [23,34]. In our work, we use Feature-wise Linear Modulation (FiLM) [32,5] as a method to reduce task interference. By using FiLM, we apply a conditional transformation to activation signals, and realize a network which changes the operation dynamically for each task.

In MTL of image translation tasks, most of the networks have a shared encoder, and task-specific decoders or task-specific parts [3,21,28,37]. Cross-Stitch Network [29] includes a feed-forward network for each task, and uses cross-stitch units to share features between tasks. UberNet [23] proposes an image pyramid approach for processing images across multiple resolutions. At each resolution, an additional task-specific layer is formed on top of the shared network. Maninis et al. [26] proposes a network that modifies behavior based on task-specific features and attention. Task attention uses task-specific residual adapter branch and Squeeze-and-Excitation modulation. In addition, they also use adversarial training to force shared parts to be statistically indistinguishable across tasks. Strezoski et al. [35] proposes a task-specific binary mask which is applied to the activations of each channel, and the activations corresponding to each task were extracted. Bragman et al. [1] proposes the similar idea with stochastic assignments of convolution channels to tasks instead of binary assignments. Liu et al. [24] have proposed a multi-task attention network (MTAN) that can learn the attention of task-specific feature levels. MTAN consists of a single shared network including global feature sharing and a soft-attention module for each task. In the above-mentioned networks, it is necessary to train a network by replacing part of the network for each task. Therefore, more network parameters are required compared to a single task network, and the number of parameters of task-specific parts increases in proportion to the number of tasks involved.

No mixed-task learning with heterogeneous image translation tasks has been proposed as far as we know. One of the most similar work is Sym-parameterized Generative Network (SGN) [2]. In SGN, the distribution of multiple domains is learned by changing the weighted loss function dynamically. Thus, an input image can be dynamically translated to a mixed domain. SGN is similar to our work in that it uses a single network and performs learning of mixed loss functions. However, unlike our work, the tasks for which SGN mixed learning was applied were style transfer and domain transfer, both of which are similar tasks to each other. Heterogeneous task mixing such as mixing of inpainting and style transfer was not examined.

Continual Learning: Continuous learning [30] is a learning method that, when there is a series of n tasks $t_1, ..., t_n$, learns those tasks one by one and does not reduce the accuracy of the tasks learned in the past. However, catastrophic forgetting, which degrades the performance of $t_1, ..., t_{n-1}$ by learning t_n,

is a major problem. Many studies have been proposed to address this problem. Piggyback [25] is the methods that transforms the output by applying a learned weight mask. Although in the original paper [25], Piggyback was applied to only image classification tasks, Matsumoto et al. [27] showed that Piggyback was able to be applied to image transformation tasks with an encoder-decoder network. Piggyback is similar to our work in that multiple tasks can be learned with a single network. However, it is not able to perform mixed-task learning.

Feature-wise Linear Modulation (FiLM): FiLM [32,5] is a formalization of the conditional affine transformation. By applying the affine transformation to the network based on the condition, the effect on the output of the network is learned. Specifically, as shown in Eq.(1), we learn the functions f_γ and f_β of an input conditional signal $\mathbf{c}$ that output the scaling coefficient γ_i and the shift coefficient β_i. The subscript i means the feature or feature map number. As shown in Eq.(2), γ_i and β_i regulate network activations.

$$\gamma_i = f_{\gamma,i}(\mathbf{c}) \quad , \quad \beta_i = f_{\beta,i}(\mathbf{c}) \tag{1}$$

$$FiLM(\mathbf{F}_i|\gamma_i, \beta_i) = \gamma_i \mathbf{F}_i + \beta_i \tag{2}$$

Various methods using FiLM have been proposed for image translation tasks. Dumoulin et al. [6] made it possible to combine multiple styles by applying FiLM to Fast Style Transfer [18]. AdaIN [15] is another study that applied FiLM to Style Transfer. The AdaIN module is widely used in various image translation tasks as well as style transfer because of the ability to manipulate network outputs using AdaIN parameters [16,20,31]. In our work, we propose using FiLM as a method for learning multiple tasks and mixed tasks using a single network. In the proposed method, we extend the work of Dumoulin et al. from multiple mixed style transfer to multiple mixed image transformation tasks. For FiLM generators, we use StyleGAN's Mapping Network [20] mechanism, which can generate AdaIN parameters from the conditional vectors with a sequence of fully-connected layers.

3 Proposed Method

In our work, Feature-wise Linear Modulation (FiLM) [32,5] is used for learning multiple image translation tasks using a single encoder-decoder network. Its effectiveness has been demonstrated in various image translation tasks. In other words, it can be said that FiLM has high versatility in image transformation tasks and is expected to be able to learn various image transformation tasks on a single network. Therefore, in our work, we propose learning of different kinds of image transformation tasks with a single network using FiLM, which is the first objective of this paper. The second objective is to accomplish mixed-task learning of multiple heterogeneous image translation tasks by a single network using FiLM, which is inspired by the work on mixing of multiple styles by Dumoulin et al. [6].

3.1 Task Conditional Vector

A task conditional vector is used for specifying tasks at the time of both training and inference of the network. At training time, the task conditional vector

corresponding to the task to be learned by the network is given. Similarly, at inference time, the task conditional vector corresponding to the task to be executed is given. If the number of tasks is n, the task conditional vector $\mathbf{c}$ is defined as a n-dimensional vector $[c_1, ..., c_n]$. At inference time, c_i takes values in the range of 0.0 to 1.0, while c_i can be 0.0 or 1.0 at training time. Note that a zero conditional vector, $\mathbf{c} = [0.0, 0.0, 0.0]$, represents the identity transformation in which the output is the same as the input. Including the identity transformation in the training is needed to control the degree of the transformation, which can be explicitly defined by providing intermediate values between 0.0 and 1.0 in the task conditional vector.

3.2 FiLM-based Network Architecture

We propose an architecture consisting of the FiLM generator and the FiLM network, as shown in Fig.2. For the FiLM network, we adopted the Encoder-Decoder CNN with five Resblocks proposed by Johnson et al. [18]. This network is typically used in image translation tasks such as Pix2Pix [17] and CycleGAN [39]. As normalization layers, we used Instance Normalization (IN) [36] instead of Batch Normalization as used in the original network. We inserted FiLM layers after all the layers except for the last layer. The combination of IN and FiLM is equivalent to AdaIN [15]. The FiLM layer receives the FiLM parameters from the FiLM generator, and controls the operation of the network by affine transformation based on the FiLM parameters.

The FiLM generator is built with only Fully Connected (FC) layers, referring to the architecture of StyleGAN's mapping network [20]. The FC layers generate the FiLM parameters, γ_i and β_i, where i depicts an index of the FiLM layer. Although one FC layer is illustrated in Fig.2, we compared one, three and five FCs for the FiLM generator in the pre-liminary experiments.

The computation of the FiLM layers is carried out based on Eq.1 and Eq.2 by using the FiLM parameters, γ_i and β_i, obtained by the FiLM generator after IN layers, as shown in Fig.3. Here, γ represents scaling parameters, and β represents bias parameters.

As explained above, most part of the proposed network are shared with all the tasks. Only the number of input elements of the first FC layer in the FiLM generator depends on the number of tasks, which is the only task-specific part. Therefore, the number of the task-specific parameters is negligibly small.

3.3 Training of Mixed Tasks

We examine the possibility of a FiLM-based network for mixed-task learning in addition to learning of multiple tasks. We prepared three methods for mixed-task learning.

In Method 1, we train only single tasks individually. We expect the inference on mixed tasks is performed implicitly, in the same way as the FiLM-based multiple style transfer [6].

In Method 2, we train mixed tasks explicitly with the total loss functions of the multiple tasks we want to mix. We use a simple mixed loss, $L_{mixed} = L_{taskA} + L_{taskB}$. To calculate L_{mixed}, we calculate L_{taskA} and L_{taskB} with training samples of task A and B, respectively, and sum up both in a single function.

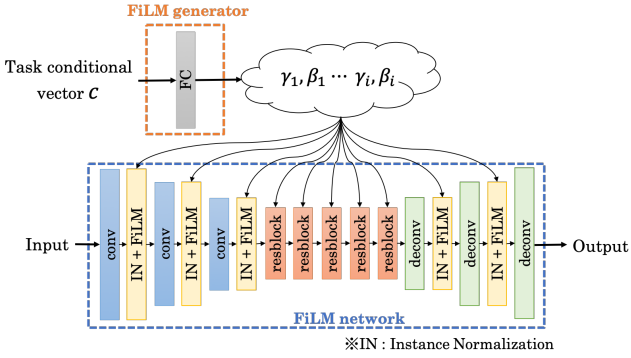


Fig. 2. Network Architecture. The FiLM network consists of three convolution layers, five residual blocks, and three deconvolution layers. FiLM layers are inserted after all the instance normalization layers including resblocks.

For training, we add mixed tasks to a task set originally consisting of only single individual tasks. In the training loop, we select randomly one task from an extended task set at every mini-batch. Note that in Method 2 and 3, we use a mixed-task conditional vector in the form of $c = [1.0, 1.0]$ when a mixed task is selected.

In Method 3, we create new training samples for mixed tasks. Fig.4 shows how to create mixed-task samples. We prepare two ways to create mixed-task samples as shown in Fig.4: (a) sequential mixing and (b) mixing by masking. Sequential mixing is a standard mixing way of multiple tasks, while mixing by masking is a combination of semantic segmentation with a different task. In (a) we need image translation networks in the second translation step, while we can use training ground truth (GT) samples instead of translation processing in the first step. In (b) we need the trained network of the fast style transfer if a mixed task contains style transfer, since no GT samples exist for this task. Therefore, in both ways, first, we train all the tasks except for semantic segmentation with a single FiLM-based network. Note that we can substitute a set of single-task networks for the multi-task network.

Next, we generate mixed-task samples. In (a), we pick up GT samples of the first task, and apply the trained model of the second task using the corresponding task conditional vectors. In the example of Fig.4(a), the sequentially-translated image by Denoising and Style Transfer is generated. Thus, the synthesized image is used as a GT image (mixed GT) of the mixed-task of "Denoising + Style Transfer." We use another mixing way, (b) mixing by masking, for the case that mixed tasks contain semantic segmentation since we want to cut out the region belonging to specific objects as shown in Fig.4(b). In (b), we pick up the segmentation mask from the training dataset of semantic segmentation, and the corresponding GT sample of the second task. If GT samples are not available in the second task, we apply the trained network to generate it. Next, we mask out the output of the second task with the segmentation mask. In the example of Fig.4(b), we generate a mixed-task sample by combining Style Transfer and semantic segmentation on the horse image. To train mixed task samples, we use L2 loss. We compare these three methods in the experiments.

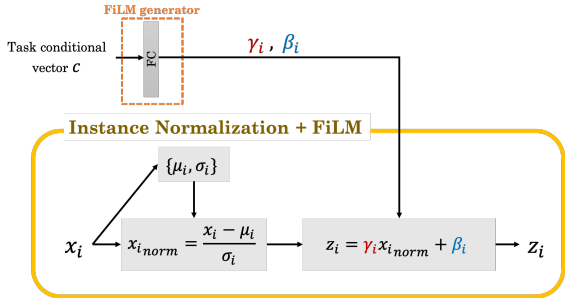
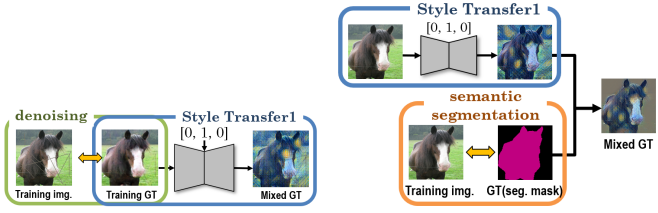


Fig. 3. Computation in a IN+FiLM layer. After normalizing the input features by Instance Normalization, affine transformation with FiLM parameters is applied. These parameters are obtained by the FiLM generator.



(a) Sequential mixing (b) Mixing by masking

Fig. 4. Two ways to generate mixed-task training samples. (a) Denoising and Style Transfer are mixed sequentially. (b) The output of Style Transfer is masked out by the segmentation mask obtained Semantic Segmentation Task.

4 Experiments

We performed three kinds of experiments to verify the effectiveness of the proposed method: (1) learning of multiple different tasks, (2) learning of mixed tasks with the three training methods, and (3) comparison with the baselines.

In (1), we verified whether it is possible to learn multiple different image translation tasks with a single FiLM-based network. In (2), we verified whether the proposed method could learn mixed image transformation tasks with a single FiLM-based network. To do that, we qualitatively compared the three methods explained in Sec.3.3. Finally, in (3), we quantitatively compared the proposed method with several baselines by evaluating their performance on all the tasks.

The dataset used for the experiments was Pascal VOC [9], a dataset containing general images such as people, vehicles, and animals with 20-class pixel-level annotation. In the experiment, out of 11,355 images on Pascal VOC 2011 for which Hariharan et al. [12] created the pixel-level annotation data, 8,498 were used as training data and 2,857 were used as test data.

As the network, we used the FiLM-based network explained in Sec. 3.2. In all the training, Adam [22] with a learning rate of 1e-4 was used as the optimization method, and the batch size were 32. To training multiple tasks with one network, one task is randomly selected from a task set at every mini-batch. So we set the number of epochs as $300n$, where n is the total number of all the tasks to be learned. We did not use any weighting for loss functions, since several existing works [24,2] showed that no weighting or equal weighting was enough for training

Table 1. The task list with the loss functions.

Task number	Task name	Loss function
Task 0	reconstruction (identity)	L2 loss
Task 1	inpainting	L2 loss
Task 2	denoising	L2 loss
Task 3	semantic segmentation	L2 loss + adversarial loss
Task 4	style transfer 1 (Gogh) 	perceptual loss
Task 5	style transfer 2 (Munk) 	perceptual loss

of multiple image translation tasks. Regarding the FiLM generator, we used one FC layer in the experiments, because we obtained good enough results with only one FC layer, which helped to keep the total model size small.

4.1 Task Sets for the Experiments

Tab.1 shows six tasks and their loss functions used in the experiments. For all the tasks, both the inputs and the outputs are three-channel images.

Task 0 is the reconstruction of an input image which is equivalent to auto-encoder or identity transformation. In Task 1 (the inpainting task), we used training images masked with small square regions, as shown in the second column of Fig.5, and in Task 2 (the denoising task) we used training images to which scratches were randomly added, as shown in the third column of Fig.5.

Regarding Task 3 (the semantic segmentation task), the semantic segmentation network usually generates a semantic segmentation map (segmentation masks) with the number of channels equal to the number of classes, where a cross-entropy loss is used for training. However, in our work, to mix segmentation with a second task, the output of semantic segmentation is a three-channel image in which the region of the specified classes is cut out and the others are filled out. The target image used for learning of semantic segmentation was artificially created based on the annotation mask of the dataset. To train the semantic segmentation task, we used both L2 loss and adversarial loss instead of the conventional cross-entropy loss. In the experiment, 20 classes except for the background class conformed the set of the cut-out classes. In other words, the output of semantic segmentation is an image in which the background is filled with the background color. In this experiment, the RGB value of the background part of the semantic segmentation was set to (0,0,0) for the input image normalized by the ImageNet [4] mean and standard deviation.

Task 4 and 5 are Fast Neural Style Transfer with different styles using perceptual loss in the same way as Johnson et al. [18].

4.2 Experiment 1: Learning of Multiple Different Tasks

In Experiment 1, we qualitatively evaluate the proposed FiLM-based network to learn multiple different image translation tasks at the same time. We trained the proposed FiLM-based network with the six tasks shown in Tab.1. To train the network and to get the results of all the tasks, we used one-hot vectors except for Task 0, which uses a zero vector. Fig.5 shows the results of all the six tasks. From the figure, it can be seen that most of the tasks are successfully executed

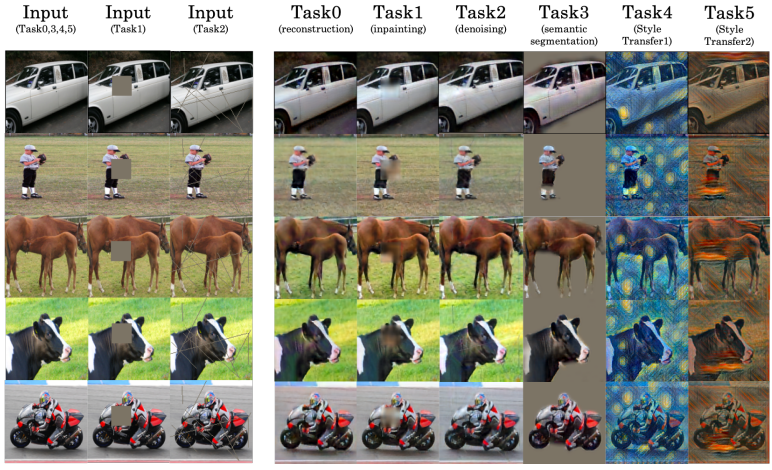


Fig. 5. The results of all the six tasks generated by one trained FiLM-based network.

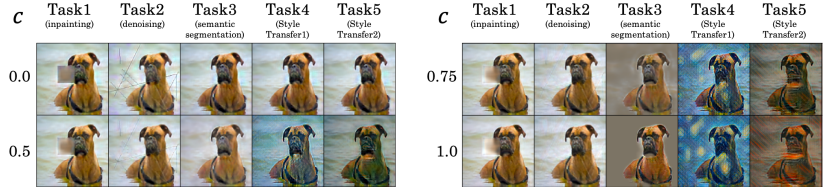


Fig. 6. Controlling the degree of each of the tasks. c on the vertical axis indicates the value of the element of the task conditional vector corresponding to each task.

by changing only the task conditional vector with a single trained network. No prominent task degradation and interference is not observed except for inpainting. Note that the result for inpainting (Task 1) is not perfect, partly because 8,498 training samples might be not enough for this task, which explicitly include images with square-masking. From these results, we conclude that multiple different image translation tasks can be learned with a single FiLM-based model.

4.3 Experiment 2: Learning of Mixed Tasks

In Experiment 2, we examine if it is possible to learn mixing of different image transformation tasks using the proposed method. As explained in Sec.3.3, we prepared three methods:(1) training only individual tasks, (2) training mixed tasks using a compound loss function, and (3) training using synthesized mixed-task samples. For verification of mixed-task learning, four tasks were used: reconstruction, denoising, semantic segmentation, and Style Transfer 1, shown in Tab.1. Then, experiments were performed on all the combinations of these patterns.

For Method 3, we used sequential mixing (Fig.4(a)) for the pair of the denoising and the style transfer tasks, and mixing by masking (Fig.4(b)) for the other two pairs. In the training time of Method 3, we repeat training of a FiLM-based network twice. First, we train individual single tasks with a single FiLM-based network, and after generating mixed samples, we train again both single tasks and mixed tasks with the FiLM-based network. To train mixed task samples, we use L2 loss between an output image and a synthesized mixed-task sample.

Table 2. Task conditional vectors for mixed-task learning.

	Mixed tasks	conditional vector	mixing for Method 3
Mix 1	denoising + Style Transefer1	[1.0, 0.0, 1.0]	sequential
Mix 2	denoising + semantic segmentation	[1.0, 1.0, 0.0]	masking
Mix 3	semantic segmentation + Style Transfer1	[0.0, 1.0, 1.0]	masking

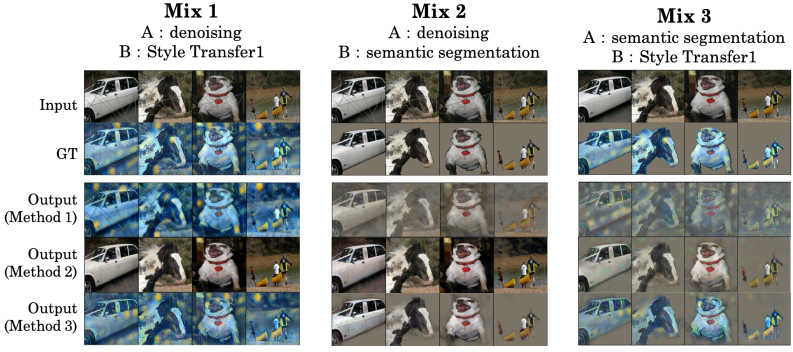


Fig. 7. The outputs of learning of the mixed tasks with Method 1, 2, and 3.

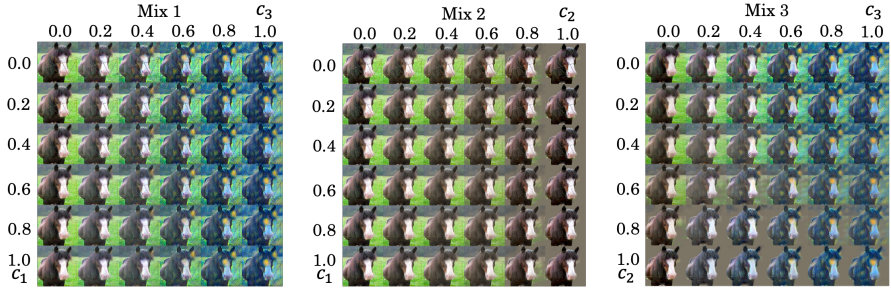


Fig. 8. The transitional results of mixed-task learning with Method 3 by changing the conditional weights by 0.2.

To train mixed tasks with Method 2 and 3, we used mixed-task conditional vectors shown in Tab.2. Although the summed value of the conditional vector is commonly set as 1.0, the degree of both task in the mixed task should be normal in case of sequential mix and mixing by masking. Therefore, we set 1.0 to the elements corresponding to both tasks, as shown in the table.

Fig.7 shows the experimental results. In case of Method 1, the results of Mix 2 and 3 look like weak combination of two tasks, while the result of Mix 1 remains many scratches, which means only style transfer without denoising was performed. In case of Method 2, the outputs are biased toward either output of the mixed task. For Mix 1 and 2, only denoising was carried out, while for Mix 3 only segmentation was performed. Therefore, we can conclude that with Method 1 in which only individual tasks are trained, and with Method 2 in which mixed-task was learned using the summed loss function, it is not possible to learn mixed tasks with the FiLM-based network. On the other hand, with Method 3, the results are almost the same as ground truth (GT), which means mixed-task learning succeeded. We found that mixed task learning of heterogeneous image translation tasks is possible when we use synthesized mixed-task training samples and L2 loss for training of mixed tasks.

In addition, we found an interesting characteristic of mixed-task learning with Method 3. Fig.8 shows the three kinds of mixed-task results of Method 3 by changing the conditional weights by 0.2. The elements of a task conditional vector corresponding to denoising, semantic segmentation, and Style Transfer1 is represented by c_1, c_2 and c_3, respectively. The figure shows that by performing mixed-task learning with mixed task conditions shown in Tab.2 and synthesized mixed training samples, a natural transition from identity transformation to mixed-task image translation is possible by gradually changing the conditional vector at inference time, which covers both single task translations of the mixed tasks. This result indicates that the objective space between two tasks can be regarded as a linear combination space of the two tasks by using the proposed method, and the degree of the mixed task can be controlled over the 2D linear combination space among two target tasks during inference.

4.4 Experiment 3: Comparison to the Baselines

In Experiment 3, we compare the proposed method with the baselines both qualitatively and quantitatively. In this experiment, the basic network architecture of all the baselines was the same as the proposed method.

The first baseline is the Sym-parameterized Generative Network (SGN) [2], which adopts Conditional Channel Attention Module (CCAM) as an injection method of conditional signals. Inspired by SENet [14], CCAM controls feature channels based on Sigmoid attention by integrating both conditional signals and average-pooled feature map activations. The second baseline is the CCAM of SGN with bias control. The normal CCAM uses only scaling to control feature channels with Sigmoid function. Therefore, more flexible feature channel control is expected to be performed by adding bias control to CCAM. Note that no Sigmoid was used for the bias. Although this baseline is similar to FiLM [32,5], it uses the activation signals in addition to the task conditional vector to generate scaling and bias parameters. Besides it uses instance normalization as normalization layers and CCAM were inserted on only three parts of the network according to the SGN original paper [2].

The third baseline is Piggyback [25], which is a method for fixing the parameters of the base network learned first and learning the task-specific binary mask each time a new task is added. For the initialization of the real number mask, all parameters were initialized to 1e-2, as in the Piggyback's paper, and the threshold of the binary mask was set to 5e-3 in the experiment. We created two models, Piggyback1 and Piggyback2, which used different image translation tasks for learning the base network. For Piggyback1, inpainting was selected, and for Piggyback2, semantic segmentation was selected as the task used for learning the base network. In the second and subsequent tasks, we learned a binary mask that selects effective weights for the newly added task from the base network. Since it is not possible to perform mixed-task learning with Piggyback, only the results of single tasks alone were compared.

First, we quantitatively evaluated the proposed method and the baselines. Fig.9 shows their outputs when learning of multiple heterogeneous image translation tasks. From the figure, it can be seen that in Piggyback1, much noise appears in the background part in the semantic segmentation. In Piggyback2, the outputs in the denoising still have a lots of scratches. In SGN and SGN+bias,

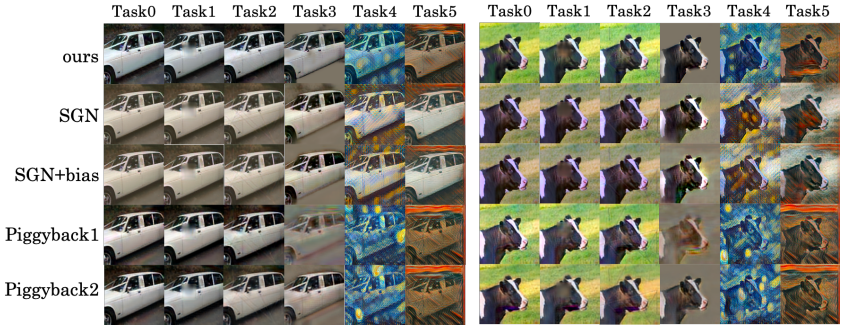


Fig. 9. The results of the proposed method and the baselines.

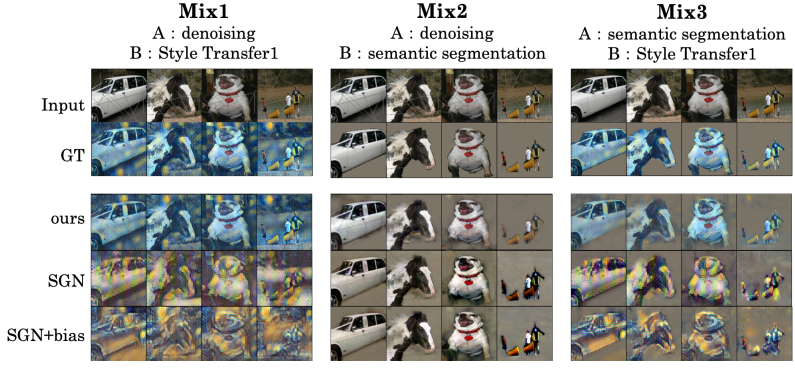


Fig. 10. Comparison of the results of mixed-task learning.

in Style Transfer 1, the blue hue of the style was not transferred, although the style with yellow spots was transferred. Similarly, style transfer failed in Style Transfer2.

Fig.10 compares the output of the proposed method and the baseline for mixed-task learning. Following Method 3 explained in Sec.3.3, we used synthesized mixed-task samples to train the networks of SGN and SGN+bias as well as the proposed network, since Method 1 and 2 did not work between heterogeneous image translation tasks in Experiment 2. From the figure, we can see that mix learning is possible at the two baselines as well as the proposed method. However, compared to the proposed method, the two baselines showed more noise. In Mix 2 and Mix 3 in which semantic segmentation is included, the contrast of the cut-out part is increased, the output is dark overall, and noise in the background part is conspicuous. In Mix 1 and Mix 3, in which Style Transfer1 is included, in addition to the blue hue of the style not being transferred, white noise is seen in SGN, and orange noise is seen in SGN+bias. From Fig.9 and Fig.10, it can be said that the proposed method was superior to the baselines qualitatively, especially in the tasks of semantic segmentation and Style Transfer.

Furthermore, we quantitatively evaluate the proposed method and the baselines. In this evaluation, we added three additional baselines: (1) a set of single models trained with six tasks independently (Single), (2) a model consisting of a shared encoder and task-specific ResBlocks/decoders (SharedEnc), and (3) the modified FiLM-based network in which the encoder part uses no FiLM but

Table 3. Comparison of the proposed method with the baselines.

		Ours	EncIN	SGN	SGN+bias	Piggyback1	Piggyback2	Single	SharedEnc
Task0	reconstruction	**0.1408**	0.1862	0.4182	0.4191	0.1598	0.1619	0.1222	0.1797
	(MSE⇓, SSIM⇑)	**0.9870**	0.9820	0.9579	0.9578	0.9854	0.9860	0.9892	0.9844
Task1	inpainting	**0.0932**	0.1041	0.4524	0.4518	0.1230	0.1535	0.0771	0.2375
	(MSE⇓, SSIM⇑))	0.9931	0.9920	0.9553	0.9554	**0.9939**	0.9885	0.9945	0.9806
Task2	denoising	**0.0839**	0.1038	0.4300	0.4309	0.1742	0.2022	0.0890	0.2045
	(MSE⇓, SSIM⇑)	**0.9936**	0.9916	0.8703	0.9572	0.9868	0.9830	0.9931	0.9825
Task3	semantic segmentation	**0.2112**	0.2298	0.5942	0.5892	0.3657	**0.2112**	0.2289	0.2516
	(MSE⇓, SSIM⇑, IoU⇑)	0.9798	0.9775	0.9473	0.9500	0.9630	**0.9801**	0.9775	0.9753
		0.5907	0.5689	0.4299	0.4454	0.4030	**0.6014**	0.5639	0.5545
Task4	Style Transfer1 (ST1) (FID⇓)	**281.3**	318.3	299.0	307.1	333.4.8	331.4	186.7	324.4
Task5	Style Transfer2 (ST2) (FID⇓)	**235.3**	250.6	263.6	250.1	297.6	323.3	163.9	251.6
Mix1	denoising + ST1 (FID⇓)	**304.3**	361.3	349.1	343.4	-	-	-	-
Mix2	denoising +	**0.2130**	0.2166	0.5699	0.5663	-	-	-	-
	semantic segmentation	**0.9794**	0.9791	0.9524	0.9519	-	-	-	-
	(MSE⇓, SSIM⇑, IoU⇑)	**0.5755**	0.5752	0.4700	0.4625	-	-	-	-
Mix3	semantic segmentation + ST1	**313.0**	320.3	339.0	348.6	-	-	-	-
	(FID⇓, IoU⇑)	**0.5457**	0.5403	0.5011	0.4555	-	-	-	-
	Model size (num. of prams)	1,698,435	1,695,747	1,765,363	1,902,243	**1,688,835**	**1,688,835**	10,075,410	9,572,370

standard INs (EncIn). Thus, (1) is the strong baseline, (2) is the standard MTL network for multi-task image translation, and (3) is a variant of the proposed FiLM-based network with a shared encoder. As evaluation indices, Frechet Inception Distance (FID) is used for tasks including Style Transfer, while Mean Square Error (MSE) and Structural Similarity (SSIM) are used for other tasks. For the tasks including semantic segmentation, Intersection over Union (IoU) is used as well.

Tab.3 compares the performance of the proposed method and the baselines in the case of learning of both multiple single tasks and mixed tasks. From the table, we can see that despite its small model size, our method achieved the best evaluation scores in almost all the tasks compared to the MLT methods. Note that Piggyback1 initially started training on the single task of inpainting, while Piggyback2 started on the semantic segmentation. That is why both achieved the best score on each of these tasks, respectively. Besides, our proposal presents comparable results with the strong baseline (Single), even though its model size is the biggest (more than five times bigger). On the other hand, Piggyback has the smallest model size. However, it cannot learn mixed task.

Between the proposed method and SGN, the performance of most of the tasks was lower in SGN than in our proposed method in learning of multiple tasks and mixed tasks. One of the possible reasons is that CCAM are inserted on only three parts of the whole network, which is expected to make it hard to adopt the network to heterogeneous image translation tasks, such as style transfer and semantic segmentation. In addition, SGN uses Sigmoid functions to generate attention maps and the output values of attention maps are limited from 0.0 to 1.0, which is expected to restrict the adaptability of the network for various kinds of tasks. On the other hand, in our network, the combination of Instance Normalization and FiLM layers (IN+FiLM) without Sigmoid functions are inserted after all the convolutional layers except the last layer. Moreover, compared with EncIN, in which the encoder part uses no FiLM but standard non-conditional INs, the proposed network outperformed EncIN on all the tasks. Therefore, we can conclude that IN+FiLM should be used after all the convolutional layers except the last layer regardless of encoder and decoder parts.

5 Conclusions

In this work, we performed learning of multiple different image translation tasks and their mixed tasks with the single FiLM-based network. The experimental results showed that mixed-task learning using synthesized training images of mixed tasks is possible in addition to learning of multiple individual tasks. Furthermore, it was found that the objective space of those expressions could be complemented by changing the task conditional vector during inference, even though the intermediate representation of each task and the mutual tasks between multiple tasks were not learned. We also compared the proposed method with other baselines and showed its effectiveness.

In future work, we plan to add more tasks such as various kinds of image domain translation tasks and mix them. We aim to build a more practical network and task mixing by verifying learning of mixing of various tasks. In addition, we like to reduce task interference and improve accuracy by devising the network architecture. We will examine the effectiveness of the proposed network for image classification tasks as well. Extending the method to incremental learning is also one of the interesting topics.

References

1. Bragman, F.J., Tanno, R., Ourselin, S., Alexander, D.C., Cardoso, J.: Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In: ICCV (2019) 3
2. Chang, S., Park, S., Yang, J., Kwak, N.: Sym-parameterized dynamic inference for mixed-domain image translation. In: ICCV (2019) 3, 7, 11
3. Chen, Z., Badrinarayanan, V., Lee, C., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: ICML (2018) 3
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009) 8
5. Dumoulin, V., Perez, E., Schucher, N., Strub, F., Vries, H.d., Courville, A., Bengio, Y.: Feature-wise transformations (2018). https://doi.org/10.23915/distill.00011, https://distill.pub/2018/feature-wise-transformations 2, 3, 4, 11
6. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: ICLR (2017) 2, 4, 5
7. Dvornik, N., Shmelkov, K., Mairal, J., Schmid, C.: Blitznet: A real-time deep network for scene understanding. In: ICCV (2017) 3
8. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV (2015) 3
9. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. In: IJCV (2015) 7
10. Ghiasi, G., Lee, H., Kudlur, M., Dumoulin, V., Shlens, J.: Exploring the structure of a real-time, arbitrary neural artistic stylization network. In: BMVC (2017) 2
11. Girshick, R.: Fast R-CNN. In: ICCV (2015) 3
12. Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV (2011) 7
13. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) 3
14. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018) 11

15. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV (2017) 4, 5
16. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV (2018) 4
17. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017) 5
18. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016) 4, 5, 8
19. Kaiser, L., Gomez, A.N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., Uszkoreit, J.: One model to learn them all. arXiv:1706.05137 (2017) 2
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019) 4, 5
21. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: CVPR (2018) 3
22. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 7
23. Kokkinos, I.: UberNet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: ICCV (2017) 2, 3
24. Liu, S., Johns, E., Davison, A.J.: End-to-end multi-task learning with attention. In: CVPR (2019) 3, 7
25. Mallya, A., Lazebnik, S.: Piggyback: Adding multiple tasks to a single, fixed network by learning to mask. In: ECCV (2018) 3, 4, 11
26. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: CVPR (2019) 3
27. Matsumoto, A., Yanai, K.: Continual learning of an image transformation network using task-dependent weight selection masks. In: ACPR (2019) 4
28. Michelle, G., Albert, H., De-An, H., Serena, Y., Li, F.F.: Dynamic task prioritization for multitask learning. In: ECCV (2018) 3
29. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: CVPR (2016) 2, 3
30. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. arXiv:1802.07569 (2018) 3
31. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR (2019) 4
32. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: FiLM: Visual reasoning with a general conditioning layer. In: AAAI (2018) 2, 3, 4, 11
33. Rosenfeld, A., Biparva, M., Tsotsos, J.K.: Priming neural networks. arXiv:1711.05918 (2017) 3
34. Standley, T., Zamir, A.R., Chen, D., Guibas, L.J., Malik, J., Savarese, S.: Which tasks should be learned together in multi-task learning? arXiv:1905.07553 (2019) 3
35. Strezoski, G., van Noord, N., Worring, M.: Many task learning with task routing. In: ICCV (2019) 3
36. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: The missing ingredient for fast stylization. arXiv:1607.08022 (2016) 5
37. Vandenhende, S., Brabandere, B.D., Gool, L.V.: Branched multi-task networks: Deciding what layers to share. arXiv:1904.02920 (2019) 3
38. Zhao, X., Li, H., Shen, X., Liang, X., Wu, Y.: A modulation module for multi-task learning with applications in image retrieval. In: ECCV (2018) 3
39. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017) 5