PMFormer: Patch-Mix Transformer for Domain-Adaptive Semantic Segmentation

Vernon Toh[1*] **Benjamin Luo**[1*] **Shang Hong Sim**[1*]

[1]Singapore University of Technology and Design

Abstract

In this paper, we address the domain shift problem in semantic segmentation, focusing on the challenging task of domain adaptive semantic segmentation (DASS). Deep learning models, such as Vision Transformers (ViT), excel in various computer vision tasks but often struggle with generalization when faced with new domains. To tackle this issue, we propose PMFormer, an application of Patch-Mix Transformer to Unsupervised Domain Adaptation in semantic segmentation. PMFormer leverages a Patch-Mix module, the Mix Transformer encoder (MiT), and the SegFormer decoder to align source and target domains by constructing an intermediate domain through random patch sampling. We present experimental results on the GTA→Cityscapes and Synthia→Cityscapes datasets, demonstrating that PMFormer outperforms baseline models trained only on source images. Specifically, PMFormer improves from 40.56% to 42.22% mIoU on GTA→Cityscapes and from 31.98% to 35.87% mIoU on Synthia→Cityscapes. Additionally, we conduct ablation studies, revealing the significance of the MiT-B5 encoder in achieving superior DASS mIoU. Our findings emphasize the effectiveness of PMFormer in addressing domain shift challenges in semantic segmentation tasks. The code for the project is available at https://github.com/DidItWork/PMTrans_openMMSeg/tree/main.

1 Introduction

In recent years, the field of computer vision has witnessed a significant evolution, particularly with the advent of deep-learning models like Transformers. These models have shown remarkable success in various tasks such as image classification and semantic segmentation. However, a persistent challenge in deploying these models in practical scenarios is their tendency to underperform when encountering new, unseen domains. This phenomenon, known as the domain shift problem, arises when the source (training) and target (testing) images originate from different data distributions.

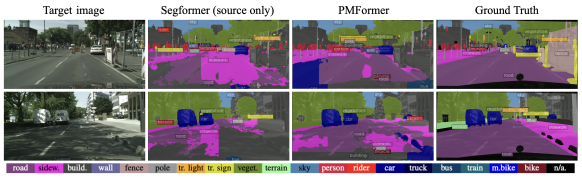

Figure 1: Predictions by PMFormer on Synthia→Cityscapes as compared to baselines.

Our project explores the area of semantic segmentation, with a specific focus on road image segmentation. Semantic segmentation, a critical task in computer vision, involves the detailed labeling of each pixel in an image, a process that is both labor-intensive and costly, especially for real-world road images. To address this, we turn to the concept of Domain Adaptive Semantic Segmentation (DASS). DASS aims to leverage the knowledge acquired from a labeled source domain (in our case, synthetic images) and apply it effectively to an unlabeled target domain (real-world road images).

In this report, we introduce PMFormer, an application of the Patch-Mix Transformer to Unsupervised Domain Adaptation in Semantic Segmentation. PMFormer is a novel architecture combining a Patch-Mix module, a Mix Transformer encoder (MiT), and a SegFormer decoder. The core idea of the Patch-Mix Transformer is to create an intermediate domain by randomly sampling patches from both the labeled source domain and the unlabeled target domain. This approach aims to mitigate the domain shift problem by aligning the source and target domains more effectively.

Our experimental results demonstrate that PMFormer achieves superior performance over existing baselines. Specifically, PMFormer shows notable improvements on the

[*]Equal contribution

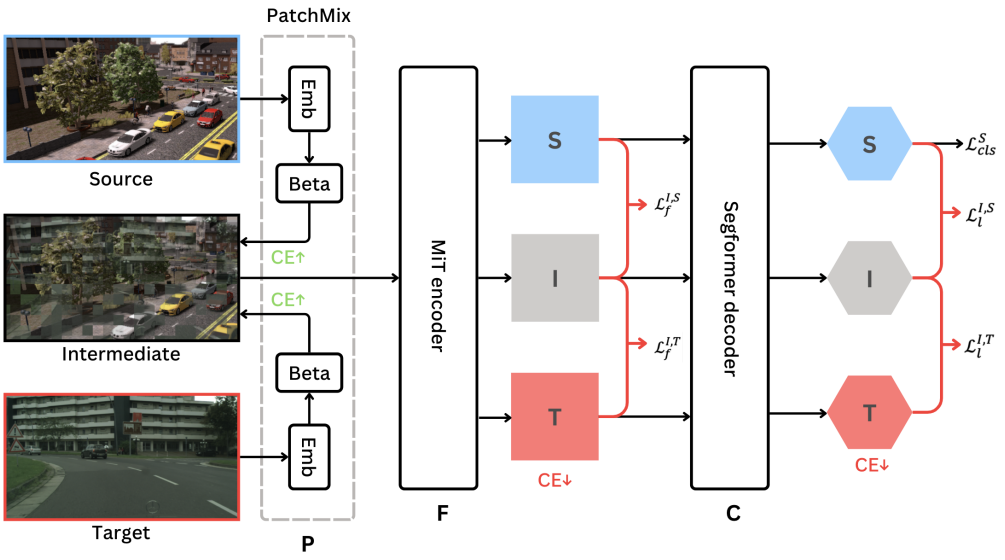

Figure 2: Overview of PMFormer framework with Patch-Mix module, MiT encoder, and SegFormer decoder.

GTA→Cityscapes (40.56% to 42.22% mIoU) and Synthia→Cityscapes (31.98% to 35.87% mIoU) benchmarks. Furthermore, our analysis reveals that the MiT-B5 encoder variant of PMFormer achieves the best results in DASS mIoU.

2 Related Works

Unsupervised Domain Adaptation (UDA) Unsupervised domain adaptation is the task of training a model with labeled source (synthetic) data and adapting it to unlabelled target (real world) data. Deep learning models such as transformers generalize poorly to new domains due to the domain shift problem. Self-training has been proposed to address this (Lee, 2013). In self-training, pseudo labels are generated for unlabelled target domains and then used to train the network. Pseudo labels can be generated offline (Sakaridis et al., 2018) or online (Araslanov and Roth, 2021) at training time. The retrained network is often more accurate than the original pseudo-labeller(Lee, 2013).

The other approach to tackle UDA is through using adversarial learning(Tsai et al., 2019) or game theory(Acuna et al., 2022) approaches to align source and target domains. For image classification, the Patch-Mix Transformer has been proposed to align source and target domain (Zhu et al., 2023). It does so by first sampling patches from both the labeled source and the unlabeled target domains randomly to build an intermediate domain. Then, it learns to sample domain-invariant features in the intermediate domain based on a game-theoretical framework and hence performs well in both the source and target domains.

Semantic Image Segmentation Semantic segmentation is the task of assigning a class label to every pixel in the image. The prevailing method to do semantic segmentation relies on Convolution Neural Networks (CNN) (LeCun et al., 1998; Long et al., 2015) with an encoder and decoder design (Badrinarayanan et al., 2017). However, baseline CNN methods such as Fully Convolution Network (FCN) suffer from low spatial resolution. To address this, skip connections (Ronneberger et al., 2015) or resolution-preserving architectures (Sun et al., 2019) were proposed. Inspired by the success of attention-based transformers in natural language processing, these techniques were modified for use in semantic segmentation (Liu et al., 2021), achieving state-of-the-art results. However, CNN was observed to be sensitive to distribution shifts such as domain shifts (Hendrycks et al., 2020) in image classification tasks. Transformers are more robust than CNN in this respect due to their focus on object shape rather than textures as in CNN (Bhojanapalli et al., 2021). Being more similar to how human vision works (Geirhos et al., 2022), transformers thus perform better on many computer vision tasks such as semantic segmentation and image classification. Hence, we chose to use transformer-based architectures in our approach.

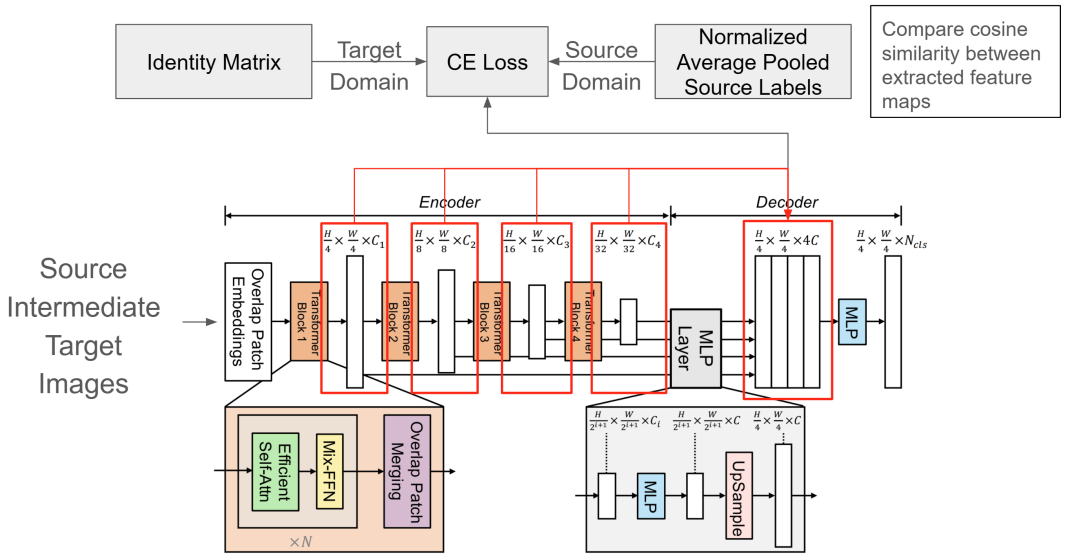

Figure 3: Segformer architecture (Xie et al., 2021).

3 Methodology

3.1 PMFormer Network Architecture

Overview. Figure 2 illustrates the framework of our proposed PMFormer, which consists of a Patch-Mix module, Mix Transformer encoder (MiT) and SegFormer decoder.

Patch Mixing. While the Patch-Mix Transformer mixes non-overlapping patches, the Mix-Vision Transformer in the SegFormer architecture uses overlapping patch embedding in its first layer with a patch size of 7 and a stride of 4 to preserve local continuity in patches. However, since patches are being mixed in the intermediate domain, the local continuity assumption between patches no longer holds and experiments have shown that the default parameters for SegFormer perform worse in PMFormer than if the patches were non-overlapping in the first patch embedding layer. Hence, the patch size is altered to be the same as the stride length in the first patch embedding layer of PMFormer.

Loss functions for PMFormer and Patch-Mix (Zhu et al., 2023):

$$\mathcal{L}_f(\boldsymbol{\omega_F}, \boldsymbol{\omega_P}) = \mathcal{L}_f^{I,S}(\boldsymbol{\omega_F}, \boldsymbol{\omega_P}) + \mathcal{L}_f^{I,T}(\boldsymbol{\omega_F}, \boldsymbol{\omega_P}) \tag{1}$$

$$\mathcal{L}_l(\boldsymbol{\omega}) = \mathcal{L}_l^{I,S}(\boldsymbol{\omega}) + \mathcal{L}_l^{I,T}(\boldsymbol{\omega}) \tag{2}$$

$$\mathrm{CE}_{i,s,t}(\boldsymbol{\omega}) = \mathcal{L}_f(\boldsymbol{\omega_F}, \boldsymbol{\omega_P}) + \mathcal{L}_l(\boldsymbol{\omega}) \tag{3}$$

$$J(\boldsymbol{w}) := \mathcal{L}_{cls}^{S}(\boldsymbol{w_F}, \boldsymbol{w_C}) + \alpha \mathrm{CE}_{s,i,t}(\boldsymbol{w}) \tag{4}$$

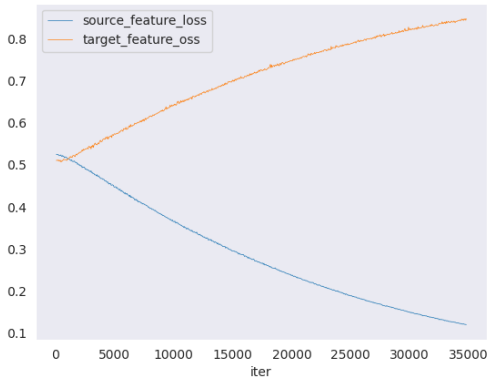

Figure 4: Feature losses between the intermediate domain and the source/target domains during training.

Feature Similarity is compared between the intermediate domain and the source/target domain. The feature selected for comparison is the $H/4 \times W/4 \times 4C$ tensor in the decoder of the SegFormer architecture as highlighted in Fig. 3. The feature tensor is the result of the concatenation of the up-scaled features of the four feature maps generated by the Mix-Vision Transformer Encoder. The feature tensor is then average pooled in the width and height dimensions to reduce computational complexity and flattened to produce a $4C \times (H/8 * W/8)$ tensor, where $4C$ is the number of channels of the feature vector at each patch location. The cosine distance between each patch and every other patch is then calculated to produce a $(H/8 * W/8) \times (H/8 * W/8)$ tensor where each value corresponds to the cosine similarity between

	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
GTA5 → Cityscapes																				
SegFormer$_{(\text{source only})}$	60.34	17.29	**82.7**	24.63	**25.6**	**28.65**	**41.06**	14.11	**86.15**	**39.29**	82.73	**61.03**	**18.5**	63.97	37.07	29.07	**0.04**	**33.18**	**25.18**	40.56
SegFormer$_{(\text{source + target})}$	78.31	0.0	64.72	0.0	0.0	0.02	13.92	0.0	64.44	6.0	**89.92**	36.13	0.0	36.38	0.0	16.32	0.0	0.0	0.0	21.38
PMFormer	**86.83**	**29.44**	82.47	**27.92**	24.47	27.72	37.4	**22.72**	82.77	35.35	86.37	59.59	9.11	**87.4**	**40.4**	**47.62**	0.0	14.51	0.0	**42.22**
Synthia → Cityscapes																				
SegFormer$_{(\text{source only})}$	37.3	16.59	78.6	**15.71**	**1.86**	**37.34**	**30.92**	15.53	79.63	-	77.86	**61.66**	19.41	69.13	-	34.48	-	16.09	15.48	31.98
SegFormer$_{(\text{source + target})}$	**83.07**	26.96	72.78	0.0	0.0	0.0	0.0	0.0	72.91	-	81.29	43.61	0.0	44.65	-	**42.41**	-	**32.47**	0.0	26.32
PMFormer	72.51	**29.57**	**81.08**	14.26	0.3	32.31	23.36	**17.61**	**84.74**	-	**85.86**	61.48	**23.14**	**82.78**	-	34.02	-	11.68	**26.86**	**35.87**

Table 1: Comparison with baselines for DASS. Baselines comprises of Segformer that is only trained on source labels and SegFormer that is trained on both source labels and pseudo target labels.

the two patches. In the same way, a similarity matrix is calculated from the labels of each image patch where patches with similar labels share a high similarity. Cross-entropy loss is then calculated between the two matrices to obtain the feature loss (Eq. 1). The label similarity in the source domain is produced from the ground-truth annotation map while in the target domain, it is the identity matrix due to the unreliability of the target pseudo labels.

Through our testing, it was revealed that the feature similarities in the source and target domains are not able to decrease simultaneously (Fig. 4). This could be because the assumption that each patch in the intermediate domain should only share similar features with the corresponding patch in the target domain does not hold in semantic segmentation as patches often have labels of the same class in the same image in both source and target domains.

Label Similarity is compared by taking the Cross-entropy Loss between the logits inferred from the intermediate image and the source labels/target pseudo-labels - similar to the usual decoder loss of semantic segmentation (Eq. 2).

Pseudolabel. In Patch-Mix Transformer, a supervised mixup loss was applied to the label space to measure the domain divergence based on the CE loss between the mixing logits and corresponding mixing labels. $\hat{\boldsymbol{y}^t}$ is the pseudo label for the unlabelled target data. We used offline pseudo-labelling where a SegFormer model trained on source data generates pseudo-labels for the target data.

4 Experiments

4.1 Implementation Details

Datasets For the target domain, we used the Cityscape street scene dataset (Cordts et al., 2016) with 2975 training images and 500 validation images. All images have a resolution of 2048x1024.

Cityscapes	
# Train Samples	2975
# Val Samples	500
Resolution	2048x1024
Synthia	
# Samples	9,400
Resolution	1280x760
GTA	
# Samples	24,966
Resolution	1914x1052

Table 2: Dataset statistics.

As per convention, we resized Cityscape images to 1024x512 pixels (Tsai et al., 2020). For the source domain, we used the Synthia (Ros et al., 2016) dataset, which contains 9,400 synthetic images with a resolution of 1280x760 and the GTA dataset (Richter et al., 2016) which contains 24,966 images with a resolution of 1914x1052. The statistics of the dataset are shown in Table 2.

Network Architecture Our implementation is based on the MMsegmentation framework (Contributors, 2020). For PMFormer architecture, we use the MiT-B5 encoder (Xie et al., 2021) and the SegFormer decoder with the Patch-Mix module.

Training We train PMFormer with the AdamW optimizer and a linear learning rate paramater scheduler. The learning rate starting at $5e-7$ with a linear increase to $6e-5$ and a subsequent decay with a coefficient of 0.01. All models are trained on a batch of two 256x256 random crops for 40k iterations.

4.2 Comparison to Baselines

We first compare our PMFormer with 2 baselines on GTA→Cityscapes and Synthia→Cityscapes in Table 1. Our first baseline is a SegFormer trained only with source labels and our second baseline is a Segformer trained with both source labels and

Enc.	Dec.	mIoU
GTA5 → Cityscapes		
MiT-B1	SegFormer	33.46
MiT-B2	SegFormer	36.14
MiT-B3	SegFormer	38.44
MiT-B4	SegFormer	40.55
MiT-B5	SegFormer	42.22
Synthia → Cityscapes		
MiT-B1	SegFormer	28.64
MiT-B2	SegFormer	30.99
MiT-B3	SegFormer	33.14
MiT-B4	SegFormer	33.62
MiT-B5	SegFormer	35.87

Table 3: Influence of the encoder on PMFormer performance.

pseudo target labels. In all cases, the models are evaluated on the Cityscapes validation set and the performance is provided as mIoU in %. We show that PMFormer generally outperforms baselines. On GTA→Cityscapes, it improves from 40.56 to 42.22 and on Synthia→Cityscapes from 31.98 to 35.87.

4.3 Influence of the size of encoder

We investigated the effect of increasing the size of the MiT encoder on the performance of DASS ranging from MiT-B0 to MiT-B5. Table 3 summarizes the mIoU scores for two tasks: GTA→Cityscapes and Synthia→Cityscapes. It can be seen that deeper models achieve a better performance demonstrating that deeper models generalize and adapt better to the new domain. Overall the best DASS mIoU is achieved by the MiT-B5 encoder.

4.4 Ablation Studies

An ablation study was conducted on the label and feature losses to analyze the impact each has on the model performance for the Synthia→Cityscapes task. As can be seen in Table 4, including just the feature loss produces the worst result, possibly due to the unsuitability of the intermediate-target feature loss for semantic segmentation as was highlighted before. However, the inclusion of both label and feature losses produces a better result than if only either was included.

Source Loss	Label Loss	Feature Loss	mIoU
✓	✓		22.43
✓		✓	11.77
✓	✓	✓	26.19

Table 4: Influence of label and feature losses on cross-domain performance (PMFormer MiT-b0) for Synthia→Cityscapes.

5 Conclusion

We introduced PMFormer, a network structure that utilizes a Patch-Mix module put forth by (Zhu et al., 2023), specifically tailored for Unsupervised Domain Adaptation (UDA) in classification. We extended its application to UDA in semantic segmentation. Our experiments demonstrated the efficacy of patch mix in comparison to source-only and naive self-training approaches, yielding a notable increase in mIoU of 4% over baseline in GTA5 to Cityscapes and 10% in Synthia to Cityscapes over our baseline models. For future works, we aim to implement a more robust pseudo-labeling pipeline for the target domain and improve the applicability of the feature space loss in semantic segmentation to achieve performance closer to State-of-the-Art UDA in Synthia/GTA to Cityscapes applications.

5.1 Work distribution

Shang Hong Sim Background research on UDA for Semantic Segmentation, documentation of PMFormer architecture, explored different methods to generate pseudolabels for UDA.

Benjamin Luo Implementation of PMFormer's architecture in MMsegmentation, documentation, and ablation studies.

Vernon Toh Training and testing baseline models, conducted experiments of different encoder sizes on the performance of PMFormer, result analysis.

References

David Acuna, Marc T Law, Guojun Zhang, and Sanja Fidler. 2022. Domain adversarial training: A game perspective.

Nikita Araslanov and Stefan Roth. 2021. Self-supervised augmentation consistency for adapting semantic segmentation.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495.

Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. 2021. Understanding robustness of transformers for image classification.

MMSegmentation Contributors. 2020. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2022. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. 2020. The many faces of robustness: A critical analysis of out-of-distribution generalization. *CoRR*, abs/2006.16241.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324.

Dong-Hyun Lee. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for data: Ground truth from computer games.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation.

German Ros, Laura Sellart, Joanna Materzynska, David Vázquez, and Antonio López. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. pages 3234–3243.

Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. 2018. Model adaptation with synthetic and real data for semantic dense foggy scene understanding.

Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. 2019. High-resolution representations for labeling pixels and regions.

Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. 2020. Learning to adapt structured output space for semantic segmentation.

Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. 2019. Domain adaptation for structured output via discriminative patch representations.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. 2021. Segformer: Simple and efficient design for semantic segmentation with transformers.

Jinjing Zhu, Haotian Bai, and Lin Wang. 2023. Patch-mix transformer for unsupervised domain adaptation: A game perspective.