Supplementary Material
DeepPruner: Learning Efficient Stereo Matching via Differentiable PatchMatch

Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hui and Raquel Urtasun
Uber ATG

Abstract

This supplementary material provides more details and thorough analysis of our deep pruner model. We hope the readers can gain more insights into our efficient stereo matching approach. We first quantitatively evaluate the effectiveness of our uncertainty estimation in Sec. 1. Next, we visualize the predicted confidence range under various scenarios and demonstrate how uncertainty can improve the overall quality of point cloud aggregation in Sec. 2. Finally, we provide the network architecture as well as the training details in Sec. 3 and Sec. 4. Alongside this material, we also provide a video to showcase the qualitative results of our model on KITTI Odometry dataset.

1. Quantitative Uncertainty Estimation

To assess the correlation between the predicted uncertainty and the outliers, we prune the uncertain pixels sequentially, starting from pixels whose confidence range is large (i.e., more uncertain), and re-compute the metric. As shown in Fig. 1, our best model and our fast model reduces the outliers ratio by 38% and 27% respectively after removing 6% of the uncertain pixels.

2. Qualitative Uncertainty Estimation

To gain more insights into our predicted uncertainty, we visualize the confidence bound and the predicted disparity along a particular scanline for different images. As shown in Fig. 2, the confidence bound (uncertainty) is small for most pixels. We also compare the predicted disparity and uncertainty between our best model and our fast model in Fig. 3. As expected, our best model is able to predict better and sharper uncertainty modes at the edges compared to the fast model.

To further showcase the effectiveness of the predicted uncertainty, we exploit it to improve the quality of 3D point cloud aggregation. Specifically, we project the certain pixels to 3D using the estimated disparities and aggregate them with ground truth poses from the KITTI Odometry dataset. As shown in Fig. 4, pruning uncertain pixels drastically reduce the smearing effect that happens frequently at the object boundaries.

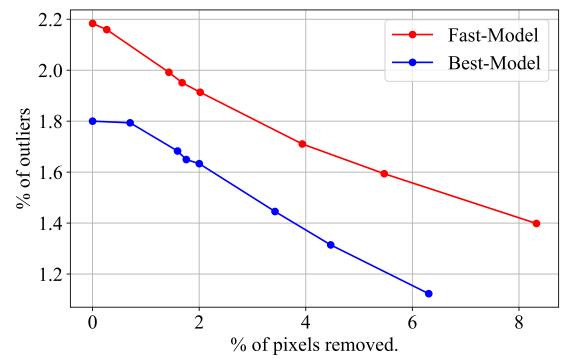

Figure 1: Outliers (%) vs Uncertain Pixel removal (%). Each dot in the plot refers to one particular threshold that we used to define uncertainty. Specifically, if the confidence range of a pixel is larger than the threshold, we treat such pixel as uncertain pixel and prune it out. The threshold value monotonically decrease from the left to the right, with the first dot representing the maximum possible disparity and the last dot representing a threshold of 3.

Samples in PatchMatch (before CRP)	Inference Runtime	All(%) bg	fg	all
9-samples	141 ms	1.75	3.0	1.95
11-samples	152 ms	1.6	3.2	1.85
14-samples	172 ms	1.6	2.9	1.8

Table 1: Quantitative Results vs PatchMatch Samples.

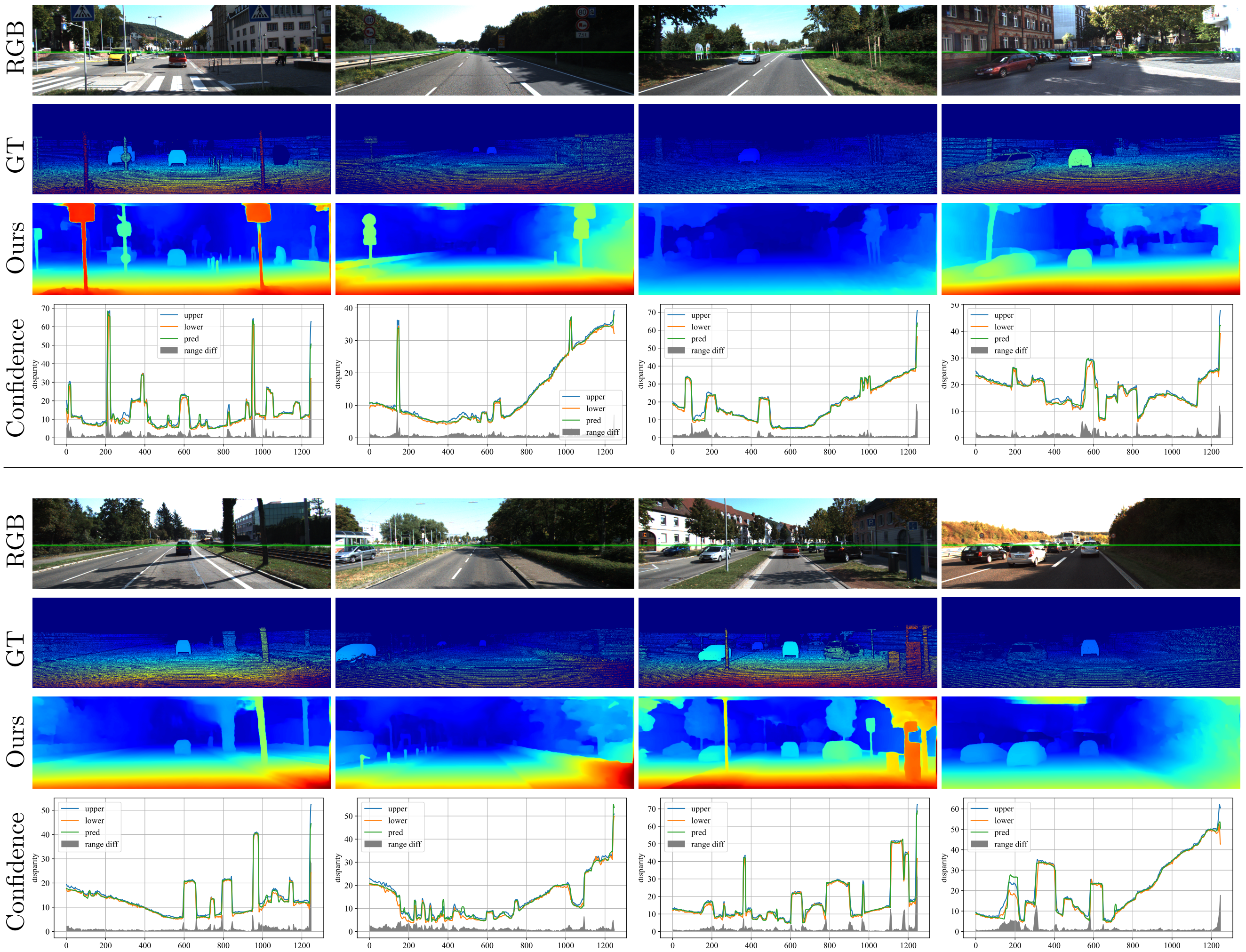

Figure 2: Qualitative results on KITTI 2015 validation set.(best-model) 2 blocks of visualizations, where each block contains(from top to bottom): input, gt, our prediction, our confident range prediction along the green horizontal scanline as marked in the RGB images.

3. Model Architecture

In this section, we describe the detailed architecture of the proposed model, starting from the overall architecture to each module.

3.1. Overall Architecture

We present the full end-to-end architecture in Tab. 2. There are two major differences between 'ours-best' model and 'ours-fast' model. Specifically, scale 'S' in Tab. 2 is set to 4 for 'ours-best' and 8 for 'ours-fast'. Also, unlike 'ours-best', we adopt the refinement module at 2 different scales ($\times 2$ and $\times 4$) in a coarse-to-fine manner for 'ours-fast' model. Next we will discuss detailed implementation for each component to ensure the reproducibility.

3.2. Feature Extractor

The detailed architecture of the Feature Extractor is shown in Tab. 3. The main difference between 'ours-fast' and 'ours-best' model lies in the feature extractor. The output resolution of the fast model is half of the best model. This is achieved by breaking down the $RB3$ residual block into one down-sampling block and two residual blocks at the same resolution (similar to residual blocks $RB2\_1$ and $RB2\_2$). Furthermore, since the receptive filed is automatically enlarged by reducing the feature-scale for 'ours-fast', we remove the $SPP1$ branch and reduce the dilation of the last residual block to 1. We note that unlike the best model, the feature extractor of the fast model outputs feature maps at 3 different scales. "ConvBn" in the tables

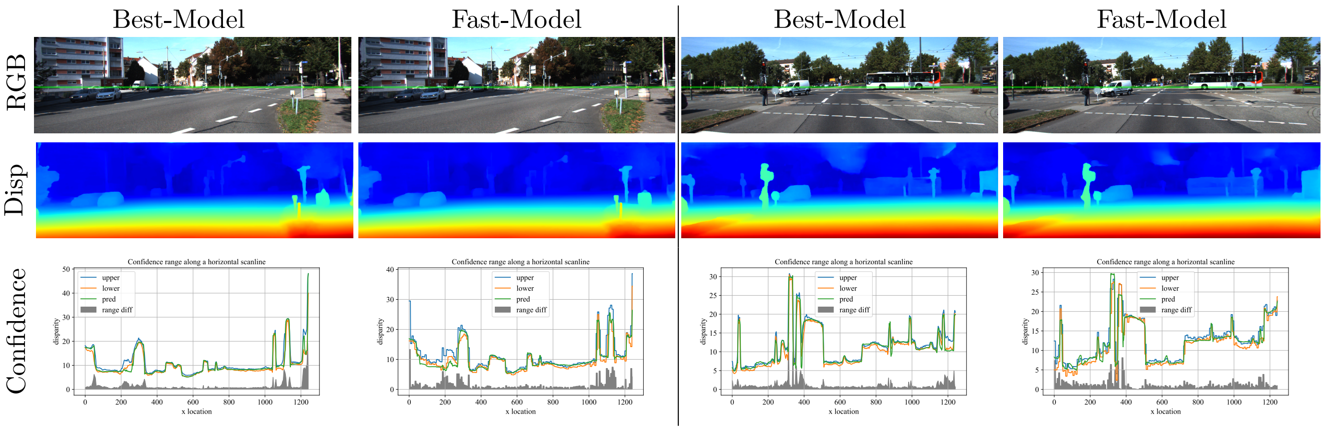

Figure 3: Qualitative comparison between best and fast models on KITTI 2015 validation set. (From top to bottom:) input, gt, our prediction, our confident range prediction along the green horizontal scanline as marked in the RGB images.

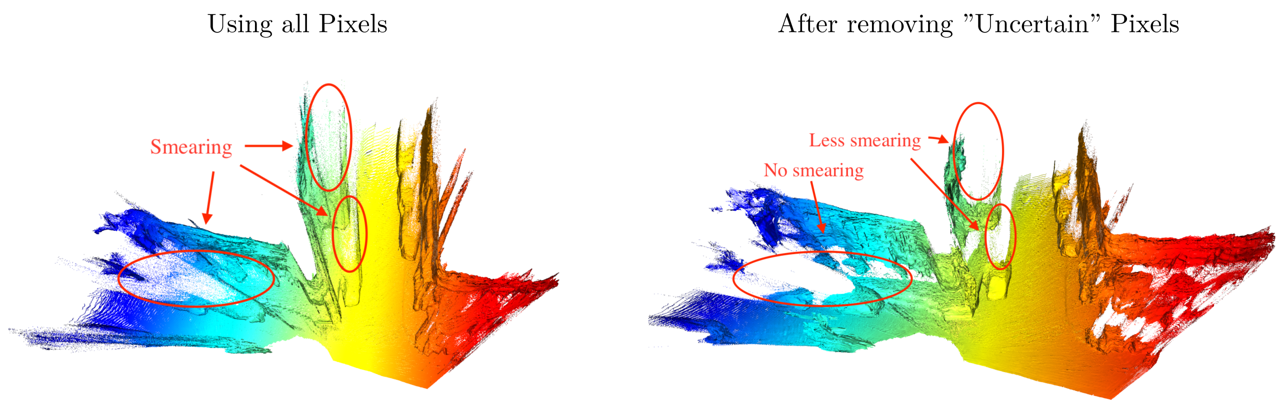

Figure 4: 3D maps created from concatenated point clouds across multiple frames of KITTI Odometry sequence 7.

refers to a Convolution operation followed by a BatchNorm and a LeakyReLU ($\alpha = 0.1$) layer. We do not use BatchNorm and LeakyReLU for the last convolution layer in the hourglass blocks and refinement network.

3.3. HourGlass Block

We revisit the detailed computation graph of a hourglass block [3] in Table 5. It is a crucial component used in Confidence Range Predictor and the Cost Aggregator of the proposed model. F in the table refers to the number of input features, with $F = 16$ in our model. The depth dimension is decided by the input number of intervals/samples drawn in the previous PatchMatch stage, in which $D_1 = 14$ and $D_2 = 9$.

3.4. PatchMatch

We discuss the implementation details of the differentiable PatchMatch module. For propagation, we adopt spatial separable one-hot filters as shown in Fig. 3 in the main paper. We unroll PatchMatch two times for each stage, with 14 samples in stage-1 and 9 samples in stage-2. The intuition behind this choice is that we want to ensure diversity at the beginning of the search in stage-1 while improving computational efficiency at stage-2 when the model is more certain. We show the performance and runtime on KITTI w.r.t. # of samples in PatchMatch stage-1 in Tab. 1. We observe that by reducing 5 samples in stage-1, we can improve the speed by $30ms$ at the minimal cost of increasing the outliers ratio by 0.15%.

Input	Layer Type	Layer Description	Output	Dimensions
Left-Image	FeatureExtractor		LeftFeat,LF2	$H/S \times W/S \times 32$
Right-Image	FeatureExtractor		RightFeat,RF2	$H/S \times W/S \times 32$
		PatchMatch Stage-1		
LeftFeat, RightFeat	PatchMatch		PM-Samples	$H/S \times W/S \times D_1$
		Min-Max-Disparity Predictor		
PM-Samples LeftFeat RightFeat	Concat		PM1	$D_1 \times H/S \times W/S \times F_{CRP}$ $(F_{CRP} = 65)$
PM1	ConvBn3d	$[64 \times F_{CRP} \times 1 \times 3 \times 3]$	CRP1	$D_1 \times H/S \times W/S \times 64$
CRP1	ConvBn3d	$[32 \times 64 \times 1 \times 3 \times 3]$	CRP2	$D_1 \times H/S \times W/S \times 32$
CRP2	ConvBn3d	$[16 \times 32 \times 1 \times 3 \times 3]$	CRP3	$D_1 \times H/S \times W/S \times 16$
CRP3	ConvBn3d	$[16 \times 16 \times 1 \times 3 \times 3]$	CRP4	$D_1 \times H/S \times W/S \times 16$
CRP4	HourGlass	Hourglass(CRP4, PM1)	MinDisp MinFeat MinConf	$H/S \times W/S \times 1$ $H/S \times W/S \times D_1$ $H/S \times W/S \times D_1$
CRP4	HourGlass	Hourglass(CRP4, PM1)	MaxDisp MaxFeat MaxConf	$H/S \times W/S \times 1$ $H/S \times W/S \times D_1$ $H/S \times W/S \times D_1$
		PatchMatch Stage-2		
LeftFeat, RightFeat	PatchMatch		PM-Samples-2	$H/S \times W/S \times D_2$
		Cost-Aggregator		
PM-Samples-2 LeftFeat, RightFeat MinFeat MaxFeat	Concat		PM2	$D_2 \times H/S \times W/S \times F_{CA}$ $(F_{CA} = 93)$
PM2	ConvBn3d	$[64 \times F_{CA} \times 1 \times 3 \times 3]$	CA1	$D_2 \times H/S \times W/S \times 64$
CA1	ConvBn3d	$[32 \times 64 \times 1 \times 3 \times 3]$	CA2	$D_2 \times H/S \times W/S \times 32$
CA2	ConvBn3d	$[16 \times 32 \times 1 \times 3 \times 3]$	CA3	$D_2 \times H/S \times W/S \times 16$
CA3	ConvBn3d	$[16 \times 16 \times 1 \times 3 \times 3]$	CA4	$D_2 \times H/S \times W/S \times 16$
CA4	HourGlass	Hourglass(Input, PM2)	CADisp CAFeat CAConf	$H/S \times W/S \times 1$ $H/S \times W/S \times D_2$ $H/S \times W/S \times D_2$
CADisp	Upsample	Biliner + Conv2d$[1 \times 5 \times 5]$	$2H/S \times 2W/S \times 1$	CADisp
CAFeat	Upsample	Biliner + Conv2d$[D_2 \times 5 \times 5]$	$2H/S \times 2W/S \times D_2$	CAFeat
		Refinement		
CAFeat LF2, CADisp	Concat		RFC0	$2H/S \times 2W/S \times F_{RM}$ $(F_{RM} = 42)$
RFC0	ConvBn2d	$[32 \times F_{RM} \times 3 \times 3]$	RFC1	$2H/S \times 2W/S \times 32$
RFC1	ConvBn2d	$[32 \times 32 \times 3 \times 3]$	RFC2	$2H/S \times 2W/S \times 32$
RFC2	ConvBn2d	$[32 \times 32 \times 3 \times 3]$	RFC3	$2H/S \times 2W/S \times 32$
RFC3	ConvBn2d	$[16 \times 32 \times 3 \times 3]$	RFC4	$2H/S \times 2W/S \times 16$
RFC4	ConvBn2d	$[16 \times 16 \times 3 \times 3]$	RFC5	$2H/S \times 2W/S \times 16$
RFC5	ConvBn2d	$[16 \times 16 \times 3 \times 3]$	RFC6	$2H/S \times 2W/S \times 16$
RFC6	Conv2d	$[1 \times 16 \times 3 \times 3]$	RFC7	$2H/S \times 2W/S \times 1$
RFC7 CADisp	Ele-wise Addition	CADisp + ReLU(RFC7)	RefinedDisp*	$2H/S \times 2W/S \times 1$

Table 2: Overview of the proposed architecture

4. KITTI Dataset Training Details

Following [4], we leverage all available image pairs from KITTI 2012 [1] & KITTI 2015 [2] (394 images in total). We held out 40 images from KITTI 2015 for validation. All experiments are cross-validated across 5 folds. We adopt different learning rate (lr) scheduler according to the number of samples in the PatchMatch module. Specifically, for 9-samples model, we use an initial lr of 7×10^{-5} and reduce it to 3×10^{-5} after 500 epochs, while for 14-samples we use an initial lr of $10^{-4}$and reduce it to 5×10^{-5} after 500 epochs.

5. Supplementary Video

We also include a supplementary video to showcase the qualitative results. Specifically, we run the proposed stereo estimation model over one of the KITTI Odometry sequence (sequence 7). As demonstrated in the video, our model produces high-quality disparity estimation. Most of the "uncertain" regions happen at significant object boundaries (e.g. boundary of the ve-

Input	Layer Type	Layer Description		Output Dimension	Layer Tag
Image	ConvBn2d	$[\ 32 \times 3 \times 3 \times 3\ ]$		$H/2 \times W/2 \times 32$	C1
C1	ConvBn2d	$[\ 32 \times 32 \times 3 \times 3\ ]$	$\times 2$	$H/2 \times W/2 \times 32$	C2
C2	ConvBn2d	$\begin{bmatrix} 32 \times 32 \times 3 \times 3 \\ 32 \times 32 \times 3 \times 3 \end{bmatrix}$	$\times 3$	$H/2 \times W/2 \times 32$	RB1
RB1	ConvBn2d	$\begin{bmatrix} 64 \times 32 \times 3 \times 3 \\ 64 \times 64 \times 3 \times 3 \end{bmatrix}$	$\times 3$	$H/4 \times W/4 \times 64$	RB2_1
RB2_1	ConvBn2d	$[\ 64 \times 64 \times 3 \times 3\ ]$	$\times 15$	$H/4 \times W/4 \times 64$	RB2_2
RB2_2	ConvBn2d	$\begin{bmatrix} 128 \times 128 \times 3 \times 3 \\ 128 \times 128 \times 3 \times 3 \end{bmatrix}$	$\times 3$	$H/4 \times W/4 \times 128$	RB3
RB3	ConvBn2d (dilation=2)	$\begin{bmatrix} 128 \times 128 \times 3 \times 3 \\ 128 \times 128 \times 3 \times 3 \end{bmatrix}$	$\times 3$	$H/4 \times W/4 \times 128$	RB4
RB4	SPP-Block(64)			$H/4 \times W/4 \times 32$	SPP_1
RB4	SPP-Block(32)			$H/4 \times W/4 \times 32$	SPP_2
RB4	SPP-Block(16)			$H/4 \times W/4 \times 32$	SPP_3
RB4	SPP-Block(8)			$H/4 \times W/4 \times 32$	SPP_4
SPP_*	Concat			$H/4 \times w/4 \times 320$	SPP
SPP	ConvBn2d	$\begin{bmatrix} 128 \times 320 \times 3 \times 3 \\ 32 \times 128 \times 1 \times 1 \end{bmatrix}$		$H/4 \times W/4 \times 32$	Feat

Table 3: Architecture of Feature Extractor

Input	Layer Type	Layer Description	Output Dimension	Layer Tag
		$H/S \times W/S \times 128$	SPPInput	
SPPInput	AveragePool		$H/S \times W/S \times 128$	SPPB1
SPPB1	ConvBn2d	$[32 \times 128 \times 1 \times 1]$	$H/S \times W/S \times 32$	output

Table 4: Architecture of a SPPBlock.

hicles) as well as heavy-textured regions (e.g. bushes).

References

[1] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In CVPR, 2012.

[2] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In CVPR, 2015.

[3] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016.

[4] Z. Yin, T. Darrell, and F. Yu. Hierarchical discrete distribution decomposition for match density estimation. 2019.

Input	Layer Type	Layer Description	Output Dimension	Layer Tag
	Input		$H/S \times W/S \times F$	Input
	Disp		$D \times H/4 \times W/4 \times 1$	Disp
Input	ConvBn3d	$[\ 2F \times F \times 1 \times 3 \times 3\ ]$	$D \times H/S \times W/S \times 2F$	E1_1
E1_1	ConvBn3d	$[\ 2F \times 2F \times 1 \times 3 \times 3\ ]$	$D \times H/S \times W/S \times 2F$	E1_2
E1_2	ConvBn3d	$[\ 4F \times 2F \times 1 \times 3 \times 3\ ]$	$D \times H/2S \times W/2S \times 4F$	E2_1
E2_1	ConvBn3d	$[\ 4F \times 4F \times 1 \times 3 \times 3\ ]$	$D \times H/2S \times W/2S \times 4F$	E2_2
E2_2	ConvBn3d	$[\ 8F \times 4F \times 1 \times 3 \times 3\ ]$	$D \times H/4S \times W/4S \times 8F$	E3_1
E3_1	ConvBn3d	$[\ 8F \times 8F \times 1 \times 3 \times 3\ ]$	$D \times H/4S \times W/4S \times 8F$	E3_2
E3_1	ConvTransposeBn3d	$[\ 4F \times 8F \times 1 \times 3 \times 3\ ]$	$D \times H/2S \times W/2S \times 4F$	D3
D3	ConvTransposeBn3d	$[\ 2F \times 4F \times 1 \times 3 \times 3\ ]$	$D \times H/2S \times W/2S \times 2F$	D2
D2	ConvTransposeBn3d	$[\ F \times 2F \times 1 \times 3 \times 3\ ]$	$D \times H/S \times W/S \times F$	D1
D1	Conv3d	$\begin{bmatrix} 2F \times F \times 1 \times 3 \times 3 \\ 1 \times 2F \times 1 \times 3 \times 3 \end{bmatrix}$	$D \times H/4 \times W/4 \times 1$	Feat
Feat	SoftMax		$D \times H/4 \times W/4 \times 1$	Score
Score Disp	Mul-Reduce	Score * Disp	$H/4 \times W/4 \times 1$	Pred

Table 5: Architecture of a Hourglass Block.