SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving

Tiago Cortinhal[1], George Tzelepis[2] and Eren Erdal Aksoy[1,2]

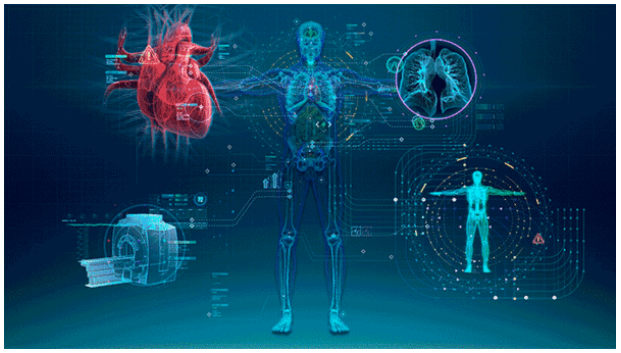
School of Information Technology
Halmstad University, Sweden

Environment Perception, Vehicle Automation
Volvo Group Trucks Technology, Sweden

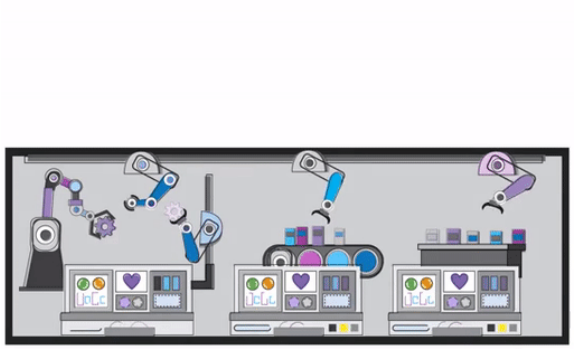
Deep Learning is Everywhere

- *Deep Neural Networks (DNNs)* have become the most popular approach to developing *Artificial Intelligence (AI)* solutions in many domains.

Autonomous Driving



Medical Diagnosis



Predictive Maintenance

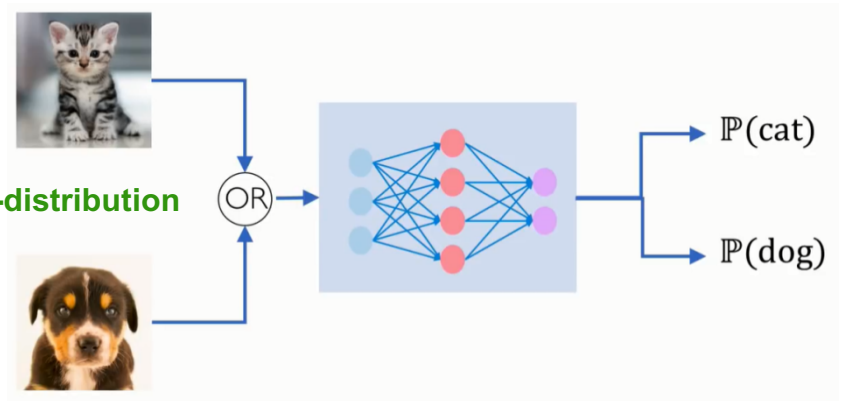
- All these applications are ***safety-critical***
- Thus, there are high requirements regarding ***safety, robustness and security***

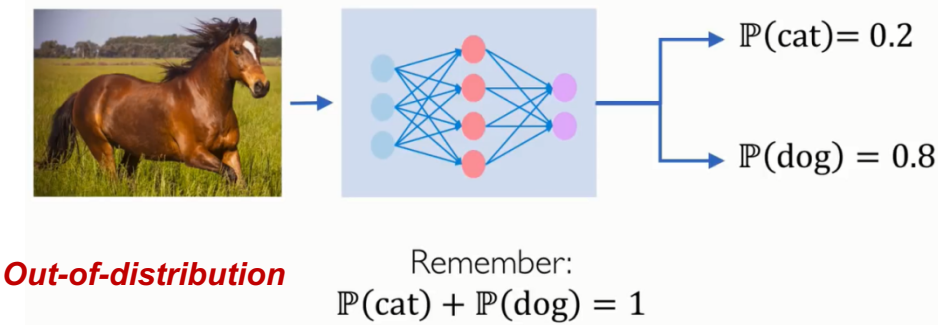
Deep Learning Limitation

- Neural networks ***cannot look beyond their horizon:***

  "Neural networks don't know, when they don't know!"

- If you have a DNN that learns to distinguish only two classes, the output probability will sum up to one for both classes. The network must return one of these classes.





- Therefore, neural networks produce ***overconfident predictions*** *for* ***out-of-distribution*** data.

- **Problem in safety-critical systems:** confidence of a network classifier is not ***reliable*** for triggering human intervention and/or transferring into a safe state of the system

- **Solution**: ***Uncertainty*** in Neural Networks

Semantic Segmentation

- Scene understanding is an essential prerequisite for autonomous vehicles.
- Semantic segmentation helps gaining a rich understanding of the scene by predicting a meaningful class label for each individual sensory data point.
- Safety-critical systems, such as self-driving vehicles, however, require not only highly accurate but also reliable predictions with a consistent measure of uncertainty.

Decision Making

Uncertainty Measure

LiDAR

Camera

road sidewalk parking car pole
vegetation terrain trunk building
other-structure other-object

HALMSTAD UNIVERSITY

3D Semantic Segmentation Methods

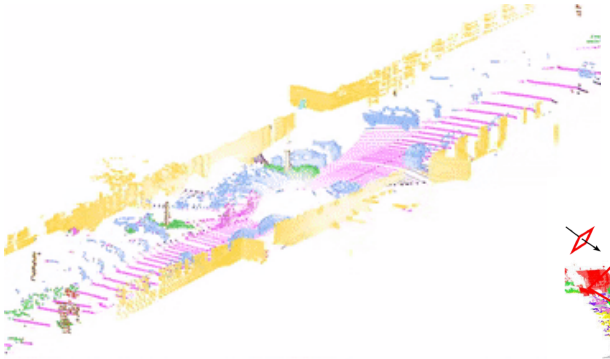
Pros

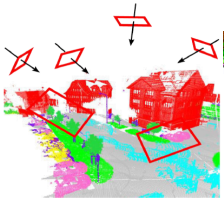
- ✓ Extremely accurate depth information
- ✓ 360 degrees of visibility (Increased range)

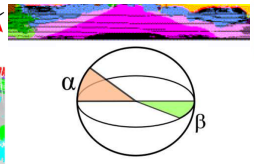
Cons

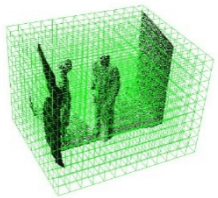
- ✗ Sparse and unstructured data
- ✗ Non-uniform sampling
- ✗ Scanning frequency ~ 10 Hz
- ✗ Expensive ~ 100.000 $

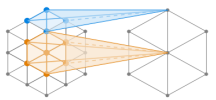
LiDAR







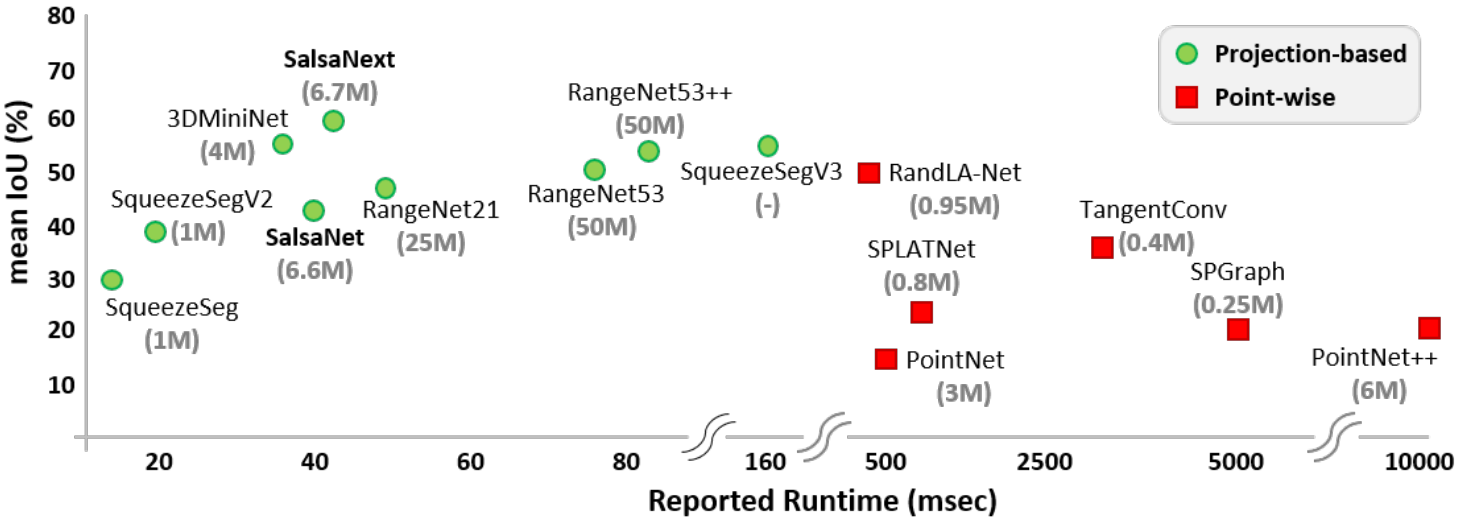




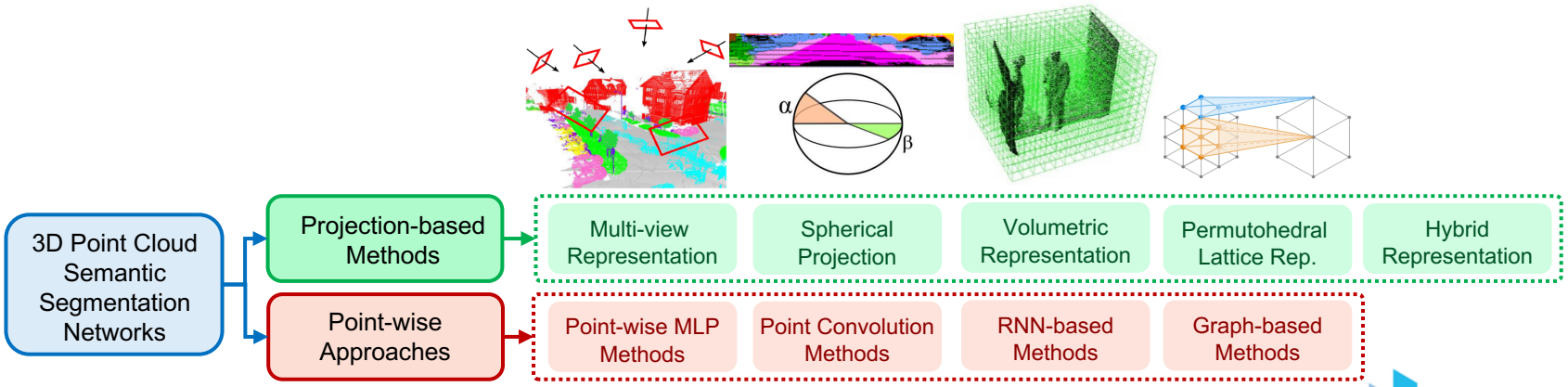
3D Point Cloud Semantic Segmentation Networks

- Projection-based Methods
  - Multi-view Representation
  - Spherical Projection
  - Volumetric Representation
  - Permutohedral Lattice Rep.
  - Hybrid Representation
- Point-wise Approaches
  - Point-wise MLP Methods
  - Point Convolution Methods
  - RNN-based Methods
  - Graph-based Methods

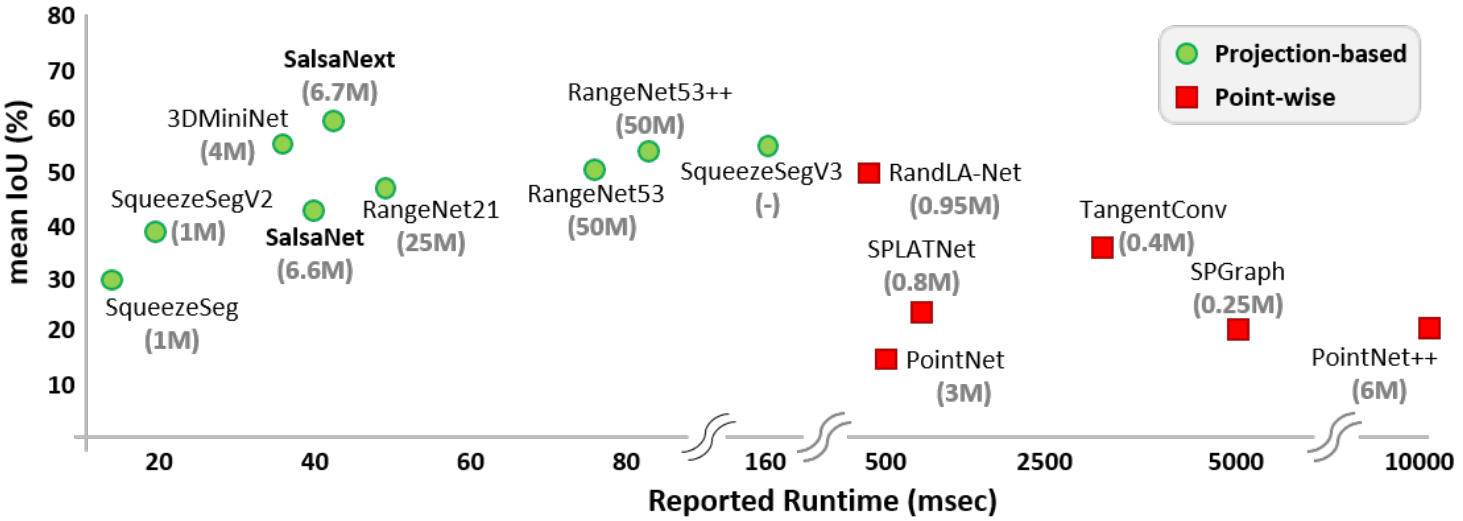
3D Semantic Segmentation Methods



***Mean IoU** versus **runtime** plot for the state-of-the-art 3D point cloud semantic segmentation networks on the Semantic-KITTI dataset. Inside parentheses are given the total number of network parameters in Millions.*



3D Semantic Segmentation Methods

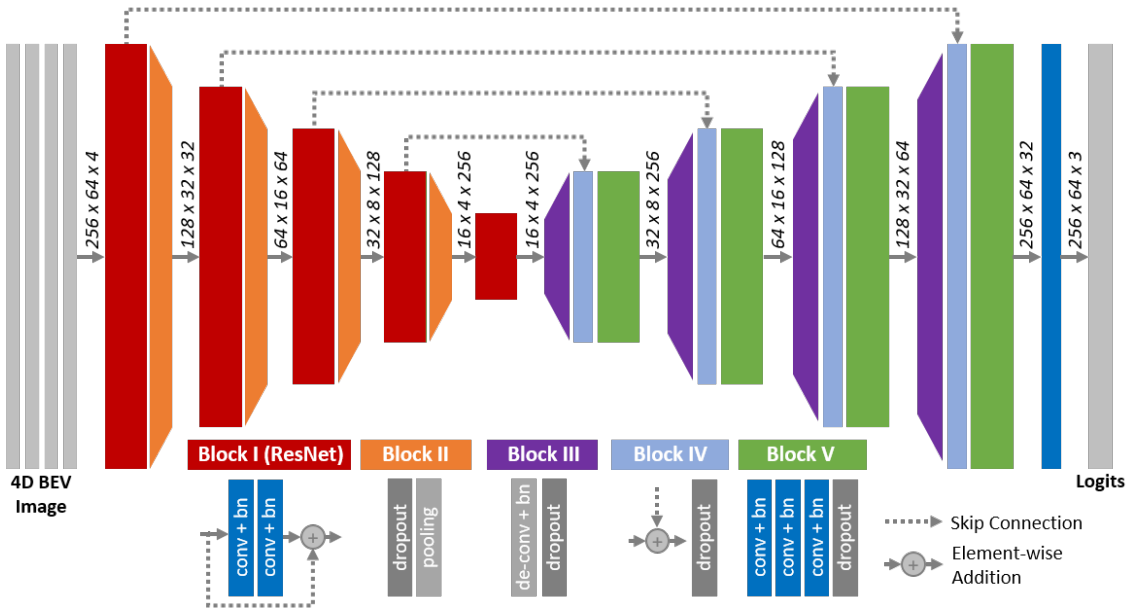


***Mean IoU** versus **runtime** plot for the state-of-the-art 3D point cloud semantic segmentation networks on the Semantic-KITTI dataset. Inside parentheses are given the total number of network parameters in Millions.*

- There is a clear split between the projection-based and point-wise approaches in terms of accuracy, runtime and memory consumption.
- Both point-wise and projection-based approaches in the literature lack uncertainty measures, i.e. confidence scores, for their predictions.

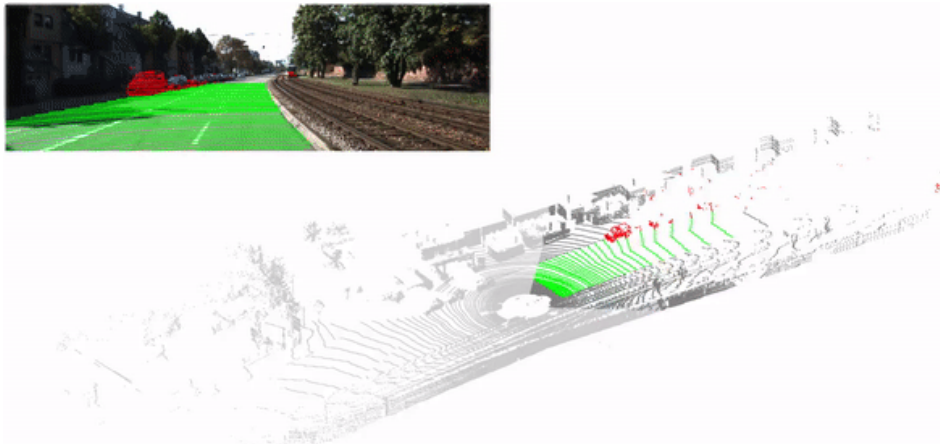
Goal: We aim at achieving a ***fine-grained semantic segmentation*** of 3D point clouds with ***reliable confidence estimates*** in ***real-time*** to reinforce the concept of safe autonomy

SalsaNext: The Original SalsaNet Model



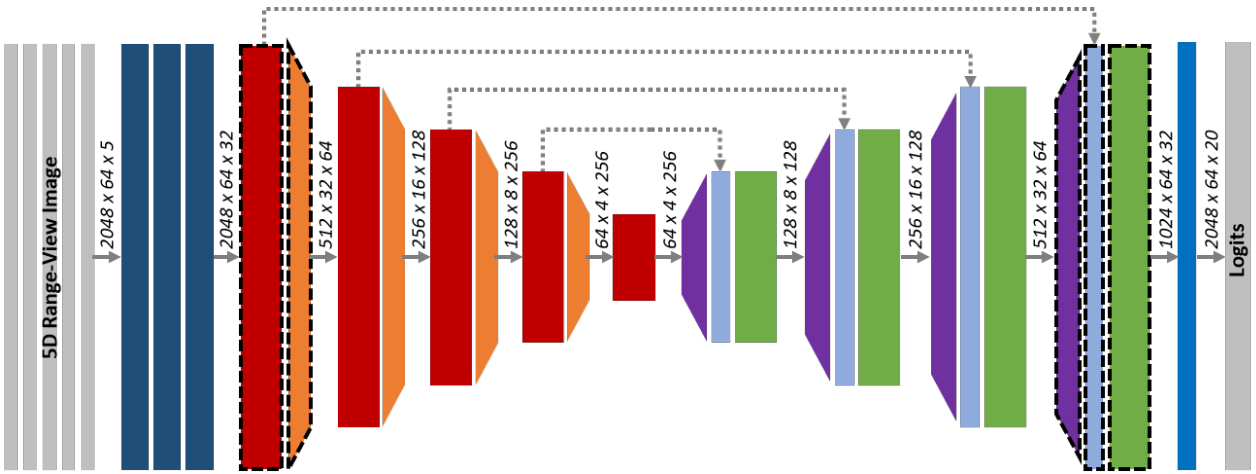
SalsaNet

- The original SalsaNet model has an encoder-decoder architecture with a bottleneck compression rate of 16.
- The SalsaNet encoder unit consists of a series of ResNet blocks and the decoder part upsamples and fuses features extracted in the residual blocks.
- To further exploit descriptive spatial cues, a stack of convolution is inserted after the skip connection.

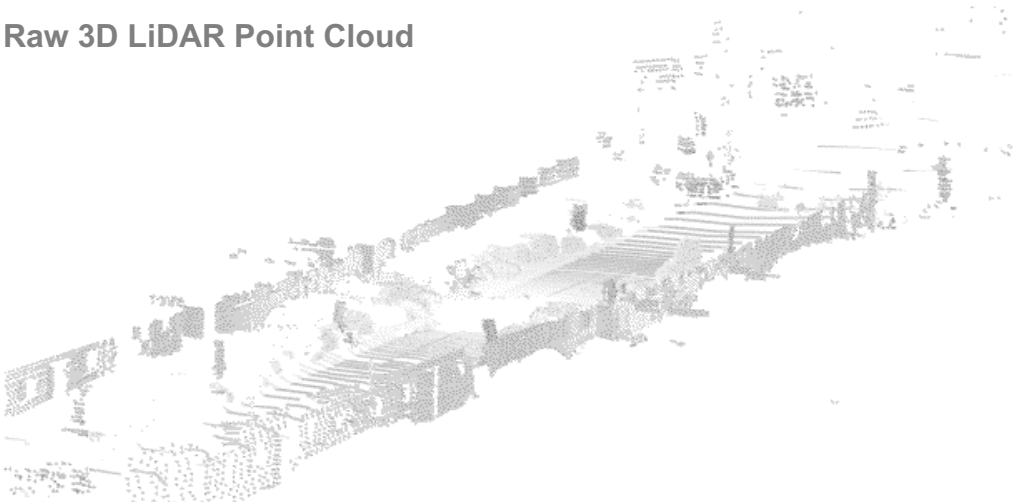


Aksoy et al., "Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving," in IEEE IV, 2020.

SalsaNext: Input



Raw 3D LiDAR Point Cloud



Range View Image

- ***Network Input:*** We ***project*** the ***3D LiDAR point cloud*** onto a spherical surface to generate the ***Range View image***:

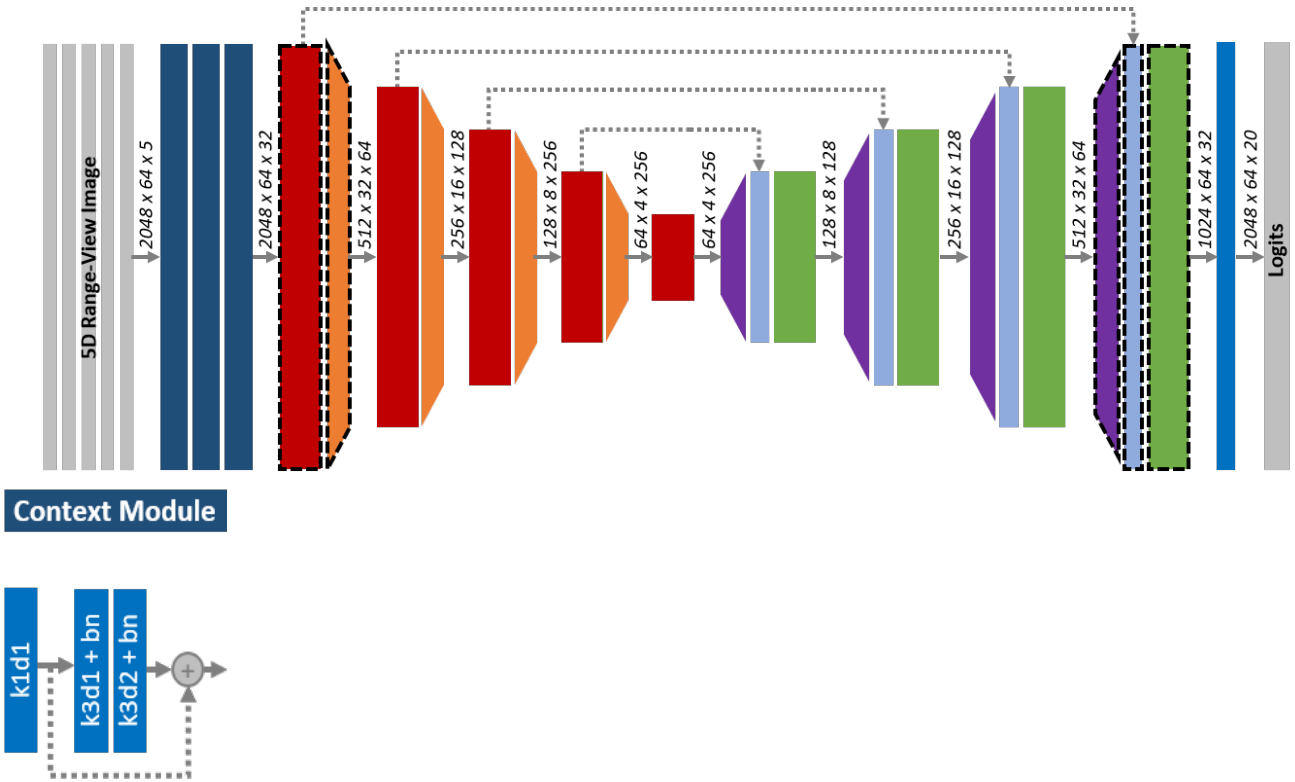
$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - \arctan(y,x)\pi^{-1}]w \\ [1 - (arcsin(z,r^{-1}) + f_{down})f^{-1}]h \end{pmatrix},$$

where h *and* w denote the height and width of the projected image, r is the range of each point as $r = \sqrt{x^2 + y^2 + z^2}$ and f defines the sensor vertical field of view as $f = |f_{down}| + |f_{up}|$.

- ***Range View image*** dimension is $[h\ x\ w\ x\ 5]$ where ***channels*** are $\boldsymbol{x, y, z, intensity, range}$.

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Context Module

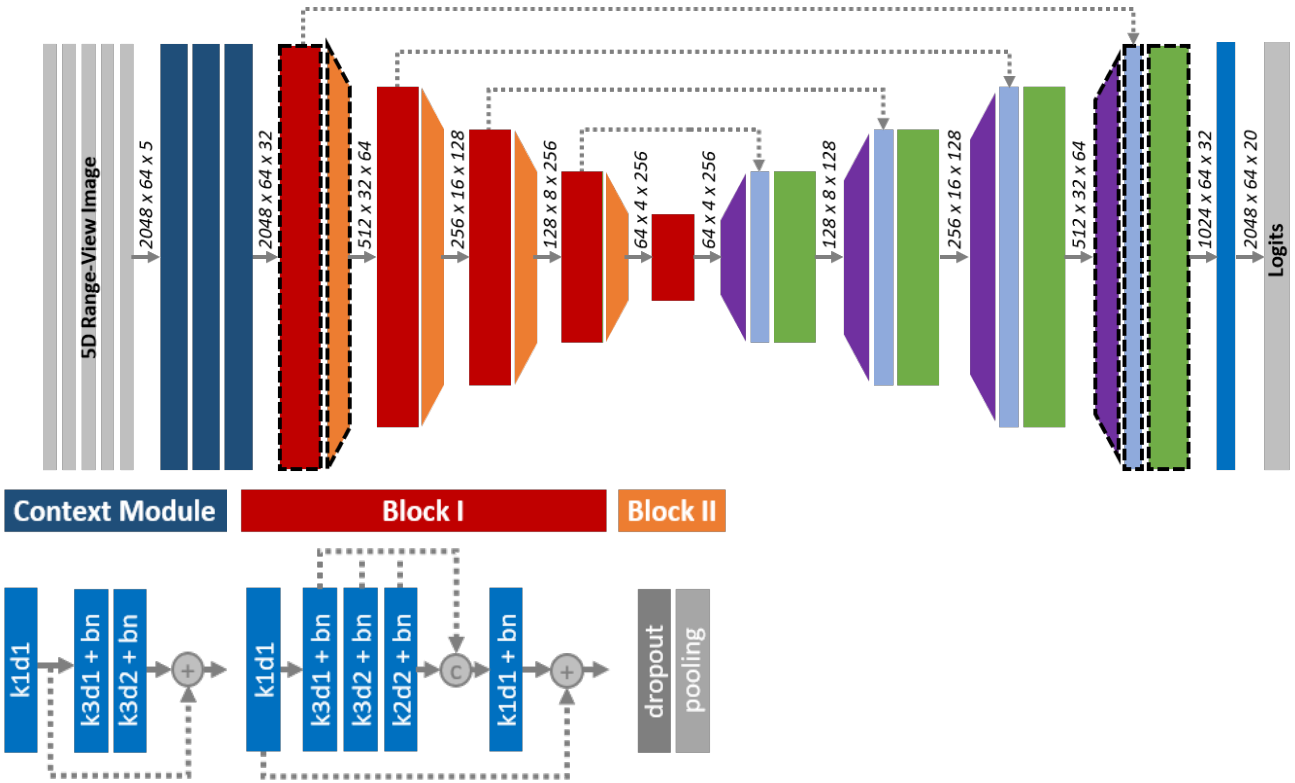


*The layer elements **k**, **d**, and **bn** represent the **kernel size**, **dilation rate** and **batch normalization**, respectively.*

- ***Global Multi-scale Context Module:*** To aggregate the context information in different regions, we place a ***residually connected dilated convolution stack*** that fuses a small receptive field (***1x1***) with a larger one (***3x3*** kernels with dilation rates=(1,2)) by applying element-wise addition.
- Starting with relatively small ***1x1 kernels*** helps aggregate ***channel-wise local spatial features*** while having ***3x3 kernels*** with different dilation rates captures various ***complex correlations*** between different segment classes.
- This helps focusing on more ***contextual information*** alongside with more detailed ***global spatial information*** via ***pyramid pooling*** (similar to Atrous Spatial Pyramid Pooling in DeepLabv3).

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Encoder

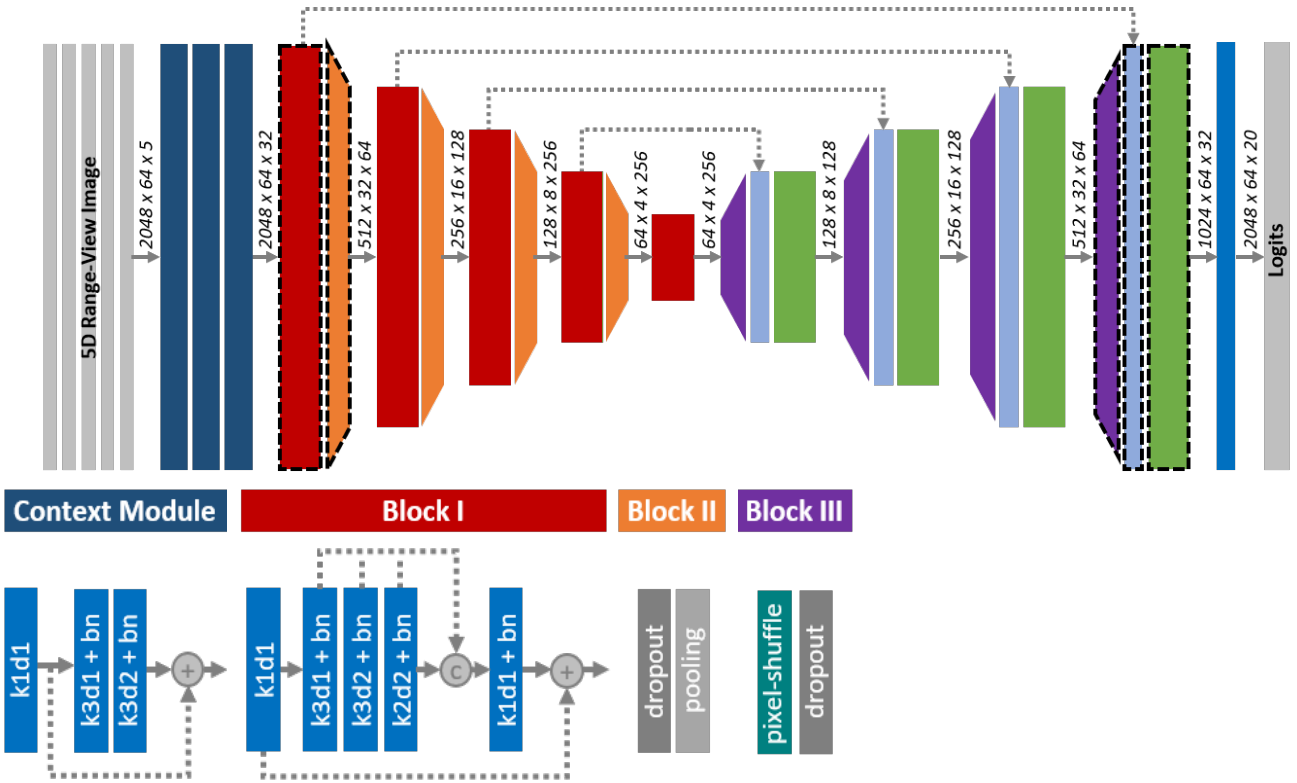


*The layer elements **k**, **d**, and **bn** represent the **kernel size**, **dilation rate** and **batch normalization**, respectively.*

- ***In the encoder,*** *we* replace the ***ResNet blocks*** in the original SalsaNet encoder with a novel combination of ***a set of dilated convolutions*** having effective ***receptive fields of 3, 5 and 7*** (see Block I).
- We further ***concatenate*** each dilated convolution output and apply a ***1x1 convolution*** followed by ***a residual connection*** to exploit more information from the fused features coming from various depths in the receptive field.
- Each of these new ***residual dilated convolution blocks*** (i.e. Block I) is followed by ***dropout and pooling layers*** as depicted in Block II.

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Decoder

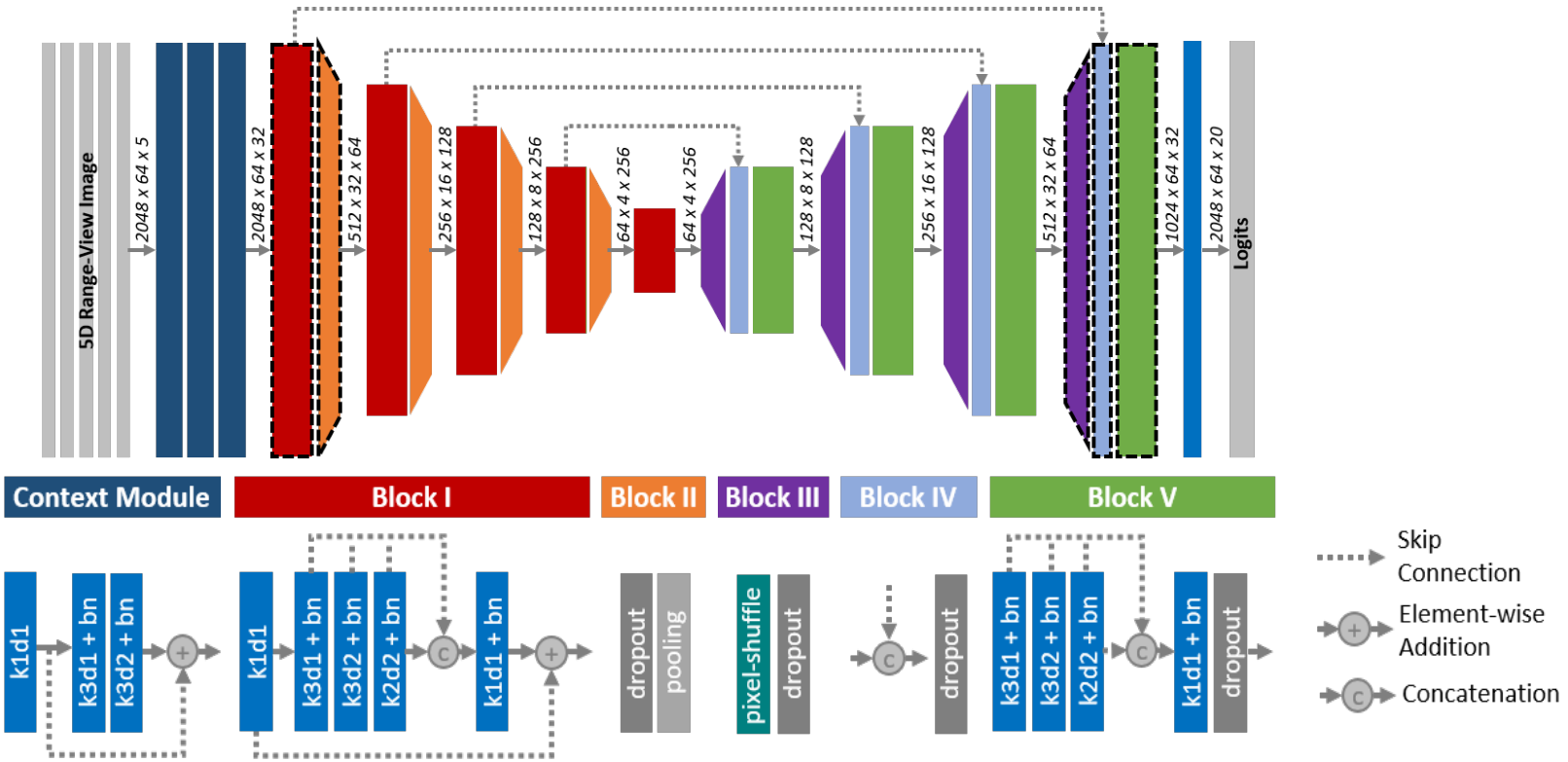


*The layer elements **k**, **d**, and **bn** represent the **kernel size**, **dilation rate** and **batch normalization**, respectively.*

- ***In the decoder,*** *we* replace the original ***transpose convolutions***, which are computationally expensive in terms of number of parameters, with ***pixel-shuffle layer*** (Block III) which directly leverages on the feature maps to upsample the input with less computation.
- ***Pixel-shuffle*** is a ***differentiable*** process which ***rearranges elements*** from depth dimension to spatial domain in a deterministic way.

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Decoder

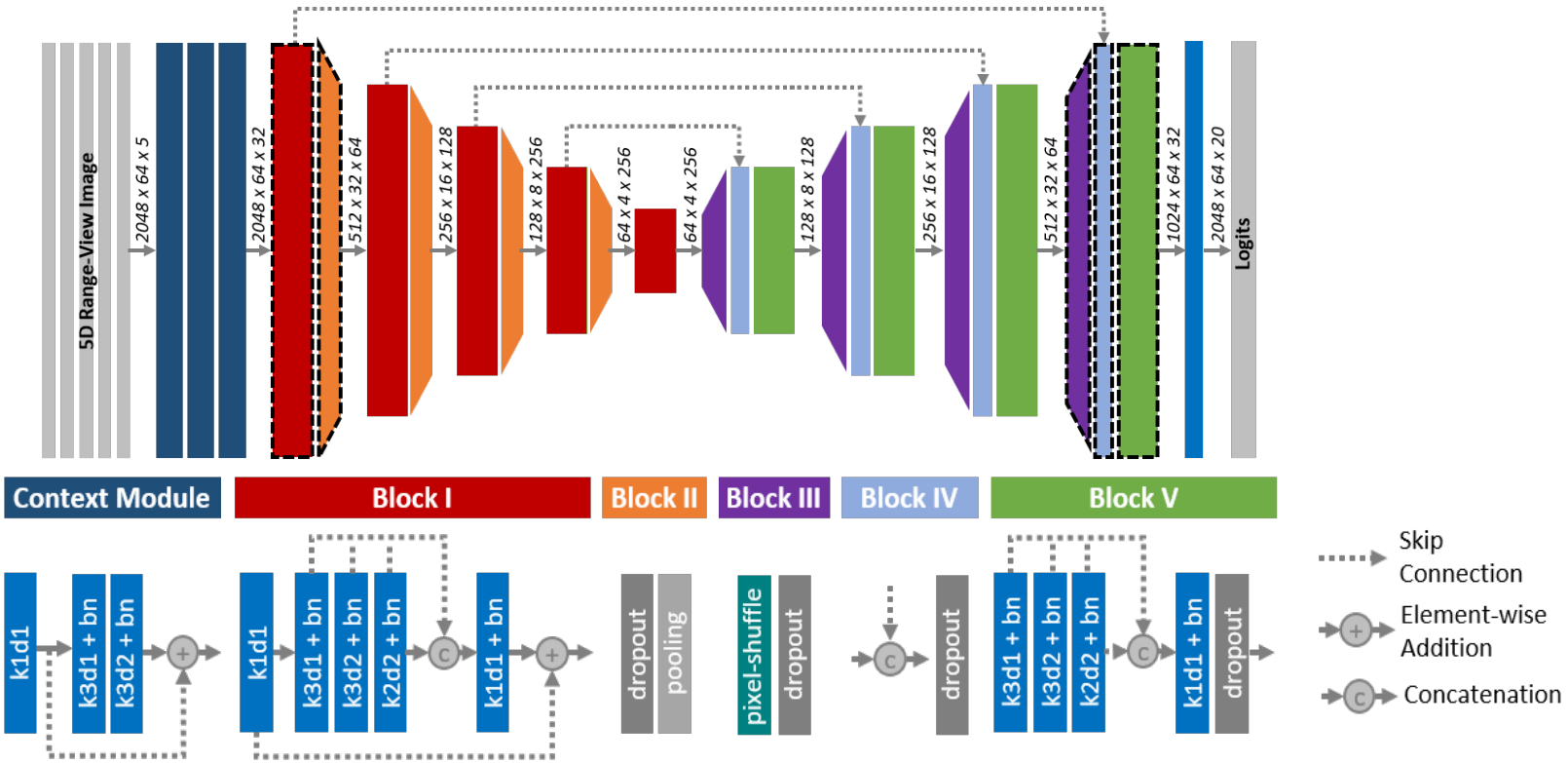


*The layer elements **k**, **d**, and **bn** represent the **kernel size**, **dilation rate** and **batch normalization**, respectively.*

- Consequently, ***shuffling pixels*** in the decoder leads to ***more accurate image reconstruction*** as it introduces fewer checkerboard artifacts ***with a vastly reduced number of parameters***.
- We additionally ***double the filters*** in the decoder side and ***concatenate the pixel-shuffle outputs*** with the skip connection (Block IV) before feeding them to the ***dilated convolutional blocks*** (Block V) in the decoder.

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Regularization

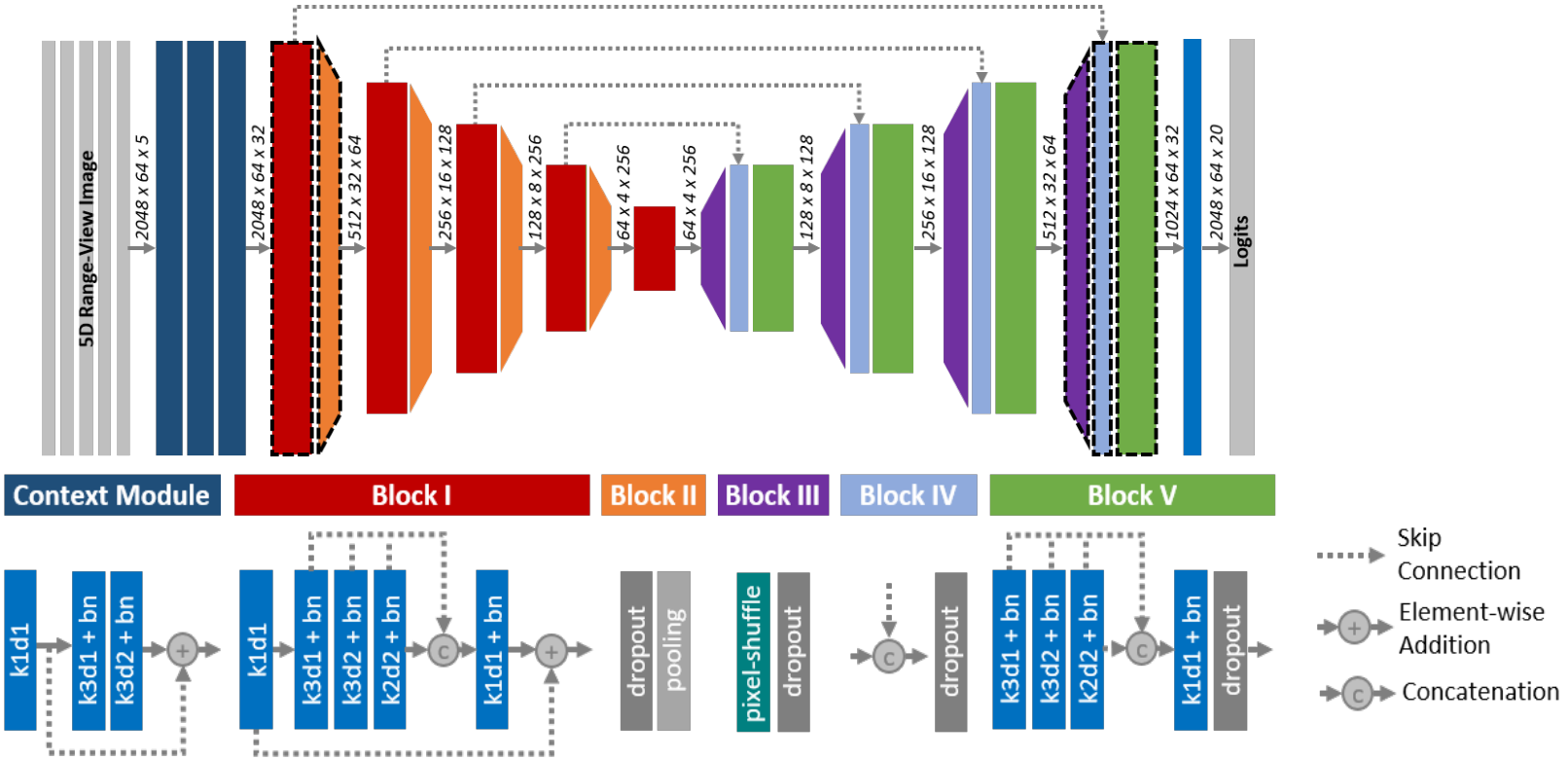


*The layer elements **k**, **d**, and **bn** represent the **kernel size**, **dilation rate** and **batch normalization**, respectively.*

- ***Central dropout treatment: To boost the roles of very basic features*** (e.g. edges and curves) in the segmentation process, we applied ***central dropout treatment*** by omitting the first and last network layers in the dropout process.
- ***Average pooling: To have a lighter model***, we also employed ***average pooling for down-sampling*** instead of having stride convolutions in the encoder.
- ***Post-processing***: We finally apply ***a kNN-based post-processing method*** to handle the ***information loss*** due to discretization errors when the network output, i.e. an RV image is ***re-projected*** back to the original 3D space.

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Loss Function



*The layer elements **k**, **d**, and **bn** represent the **kernel size**, **dilation rate** and **batch normalization**, respectively.*

- ***To cope with the imbalanced class problem***, we add more value to the underrepresented classes by ***weighting the softmax cross-entropy loss (L_{wce})*** with the inverse square root of class frequency.
- We further compute the ***Lovasz-Softmax loss (L_{ls})*** to directly ***optimize the Jaccard index.***

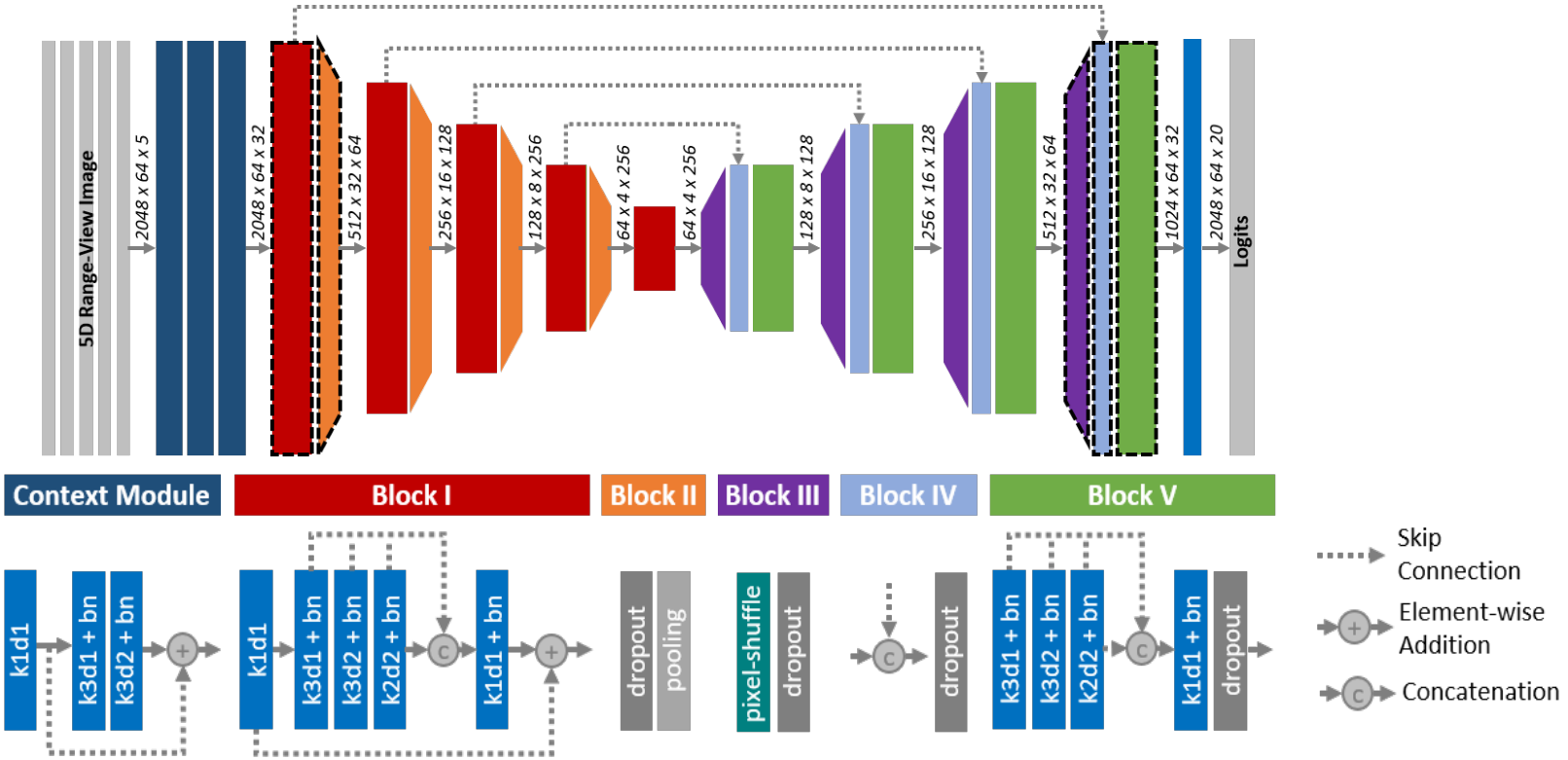
$$\mathcal{L} = \mathcal{L}_{wce} + \mathcal{L}_{ls}.$$

$$\mathcal{L}_{wce}(y, \hat{y}) = -\sum_i \alpha_i p(y_i) log(p(\hat{y}_i)) \quad with \quad \alpha_i = 1/\sqrt{f_i}$$

$$\mathcal{L}_{ls} = \frac{1}{|C|} \sum_{c \in C} \overline{\Delta_{J_c}}(m(c)) \ , \quad and \quad m_i(c) = \begin{cases} 1 - x_i(c) & \text{if} \quad c = y_i(c) \\ x_i(c) & \text{otherwise} \end{cases}$$

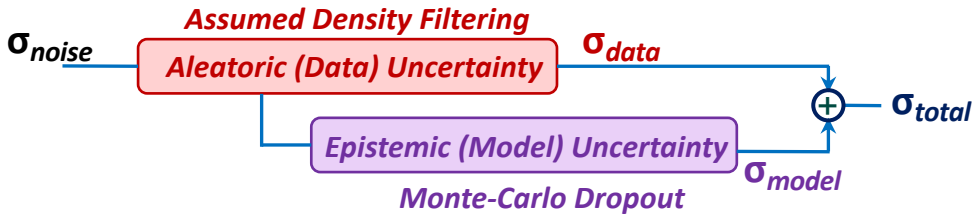
Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Uncertainty Estimation



*The layer elements **k**, **d**, and **bn** represent the **kernel size**, **dilation rate** and **batch normalization**, respectively.*

- ***Bayesian treatment*** for the uncertainty estimation

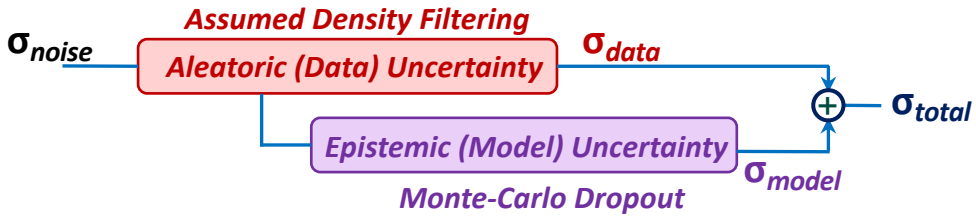


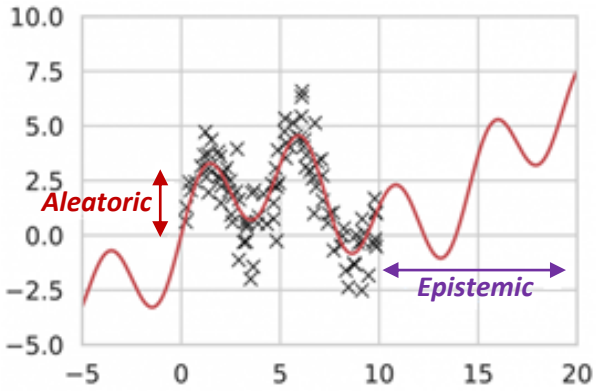
Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Uncertainty Estimation

- The ***predictive uncertainty*** comes from ***two sources***:
  - ***Aleatoric (Data) Uncertainty***
    - ***is measurement-based*** which arises from the sensor noise, data representation and label noise, etc.
    - deals with ***the potential intrinsic randomness of the sensor***, i.e. real data generating process.
    - is also called data or ***irreducible*** uncertainty since it might not be reduced by having more data.
    - e.g. may occur on ***segment boundaries*** or ***distant objects*** due to ***noisy sensor readings*** which are inherent in sensors.
  - ***Epistemic (Model) Uncertainty***
    - is ***model-based*** which arises from the ***uncertainty of model parameters***
    - reflects the ***limitation of the model*** on describing the biased training data
    - can be ***reduced*** by ***enlarging the dataset***
    - results from a lack of training data in certain areas of the input domain
    - e.g. a ***rare sample*** should have ***higher model uncertainty*** than a sample which appears more often in the training data (***imbalanced class problem***).

- ***Bayesian treatment*** for the uncertainty estimation

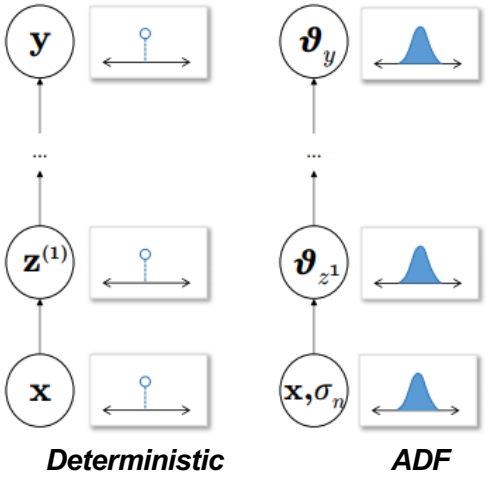




Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Uncertainty Estimation

- To compute the ***Aleatoric (Data) Uncertainty:***
  - we make use of ***prior information about the data***, e.g. sensor noise.
  - we forward-propagate ***sensor noise*** (i.e. the input uncertainty) through the network by using ***Assumed Density Filtering (ADF)*** which replaces each ***network activation***, including input and output, by ***probability distributions***



- ADF is a method for approximating the true posterior with a tractable parametric distribution in Bayesian networks.
  - When the posterior computed by Bayes' rule does not belong to the original parametric family, it can be approximated by a distribution belonging to the parametric family.
  - In ADF, the posterior is projected onto the closest distribution in the family of interest by minimizing the reverse KL divergence between the true posterior and approximate posterior.
  - In ADF, we choose a distribution that is easy for us to work with and project the true posterior after each measurement update onto this distribution family.

J. Gast and S. Roth, "Lightweight probabilistic deep networks," CVPR, 2018

- In ADF network, each intermediate layer ***i***, also outputs a distribution represented by some ***parameters*** v_{z^i} rather than a ***point estimate*** $z^{(i)}$.

- A forward pass in this ***ADF-based modified neural network*** generates output ***predictions*** ***μ*** with their respective ***aleatoric uncertainties*** $\sigma_{data.}$

- ADF basically ***approximates the distribution of activation functions***, instead of ***just weights***, which leads to a simple and effective method with a smaller computational footprint.

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

SalsaNext: Uncertainty Estimation

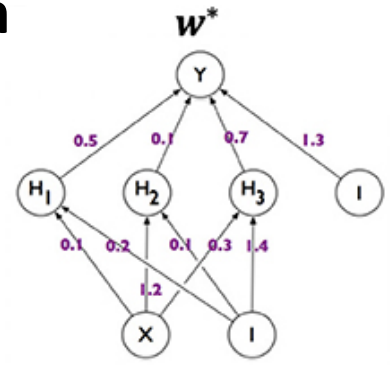
- To compute the ***Epistemic (Model) Uncertainty,*** we employ ***Monte Carlo sampling during inference***, i.e. use ***dropout*** to ***approximate the intractable weight posterior***.

- We run ***N*** trials and compute the average of the variance of the ***N*** predicted outputs:
  - At test time, a weight sample w_1 is drawn from the posterior distribution of the weights, and the resulting network is used to generate a prediction $p(y|x,w_1)$ for an example ***x***.
  - The same can be done for samples w_2 and w_3 , yielding predictions $p(y|x,w_2)$ and $p(y|x,w_3)$, respectively.
  - The three networks are treated as an ensemble and their ***predictions averaged.***

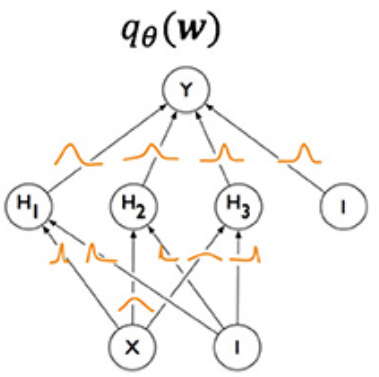
- For an ***already trained network***, the optimal dropout rate p is estimated by applying a grid search in the range [0; 1]. This means that the optimal dropout rates p will minimize

$$p = \arg\min_{\hat{p}} \sum_{d \epsilon D} \frac{1}{2}\log\left(\sigma_{tot}^{d}\right) + \frac{1}{2\sigma_{tot}^{d}}\left(y^{d} - y_{pred}^{d}(\hat{p})\right)^{2}$$

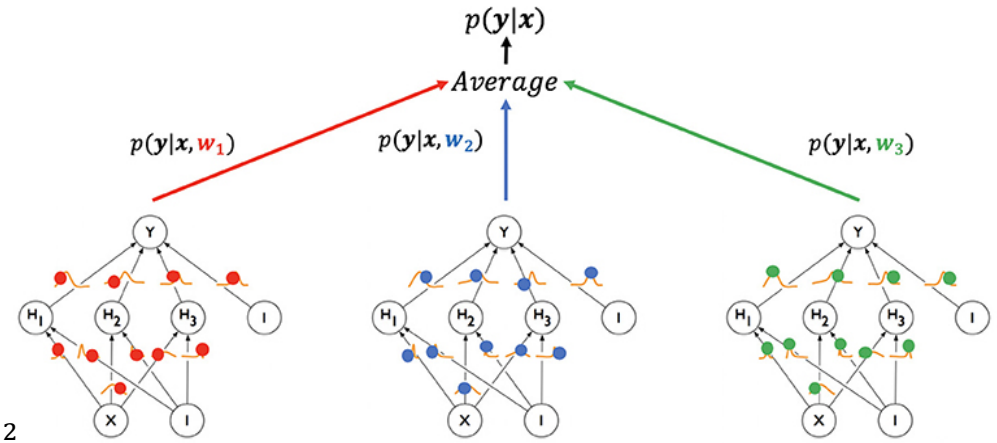
- This method ***does not require*** any change in the optimization or learning process, thus, can be directly applied to an already trained network.



In a standard deterministic NN, each weight has a fixed value ***(w*)*** as provided by classical backpropagation



In a BNN, each weight is assigned a posterior distribution, parameterized by theta ***(qθ(w)).***



Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in ICML, 2016.

A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," IEEE RA-L, 2020

Cortinhal et al., "Salsanext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving," in arXiv, 2020.

Quantitative Results on the Semantic-KITTI Test-set

	Approach	Size	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean-IoU
Point-wise	Pointnet [15]	50K pts	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.6
	Pointnet++ [16]		53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1
	SPGraph [17]		68:3	0.9	4.5	0.9	0.8	1.0	6.0	0.0	49.5	1.7	24.2	0.3	68.2	22.5	59.2	27.2	17.0	18.3	10.5	20.0
	SPLATNet [22]		66.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.4	0.8	41.5	0.0	68.7	27.8	72.3	35.9	35.8	13.8	0.0	22.8
	TangentConv [37]		86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2	35.9
	RandLa-Net [38]		**94.2**	26.0	25.8	**40.1**	**38.9**	49.2	48.2	7.2	90.7	60.3	73.7	**38.9**	86.9	56.3	81.4	61.3	**66.8**	49.2	47.7	53.9
	LatticeNet [23]		92.9	16.6	22.2	26.6	21.4	35.6	43.0	**46.0**	90.0	59.4	74.1	22.0	88.2	58.8	81.7	**63.6**	63.1	51.9	48.4	52.9
Projection-based	SqueezeSeg [6]	64×2048 pixels	68.8	16.0	4.1	3.3	3.6	12.9	13.1	0.9	85.4	26.9	54.3	4.5	57.4	29.0	60.0	24.3	53.7	17.5	24.5	29.5
	SqueezeSeg-CRF [6]		68.3	18.1	5.1	4.1	4.8	16.5	17.3	1.2	84.9	28.4	54.7	4.6	61.5	29.2	59.6	25.5	54.7	11.2	36.3	30.8
	SqueezeSegV2 [10]		81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	39.7
	SqueezeSegV2-CRF [10]		82.7	21.0	22.6	14.5	15.9	20.2	24.3	2.9	88.5	42.4	65.5	18.7	73.8	41.0	68.5	36.9	58.9	12.9	41.0	39.6
	RangeNet21 [7]		85.4	26.2	26.5	18.6	15.6	31.8	33.6	4.0	91.4	57.0	74.0	26.4	81.9	52.3	77.6	48.4	63.6	36.0	50.0	47.4
	RangeNet53 [7]		86.4	24.5	32.7	25.5	22.6	36.2	33.6	4.7	**91.8**	64.8	74.6	27.9	84.1	55.0	78.3	50.1	64.0	38.9	52.2	49.9
	RangeNet53++ [7]		91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	**91.8**	**65.0**	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9	52.2
	3D-MiniNet [27]		90.5	42.3	**42.1**	28.5	29.4	47.8	44.1	14.5	91.6	64.2	74.5	25.4	89.4	60.8	**82.8**	60.8	66.7	48.0	56.6	55.8
	SqueezeSegV3 [24]		92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9	55.9
	SalsaNet [1]	64×2048 pixels	87.5	26.2	24.6	24.0	17.5	33.2	31.1	8.4	89.7	51.7	70.7	19.7	82.8	48.0	73.0	40.0	61.7	31.3	41.9	45.4
	SalsaNext [Ours]		91.9	**48.3**	38.6	38.9	31.9	**60.2**	**59.0**	19.4	91.7	63.7	**75.8**	29.1	**90.2**	**64.2**	81.8	**63.6**	66.5	**54.3**	**62.1**	**59.5**

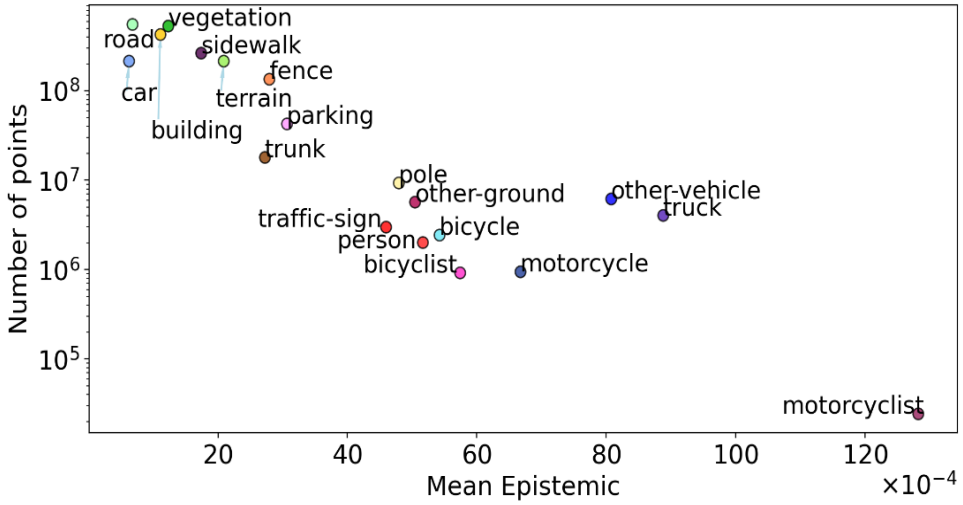
3.6%

- SalsaNext considerably outperforms the others by leading to ***the highest mean IoU score*** (***59.5%***) which is ***+3.6%*** over the previous state-of-the-art method.
- In contrast to the original ***SalsaNet***, we obtain more than ***14% improvement*** in the accuracy.
- When it comes to the performance of each individual category, SalsaNext performs the ***best in 9 out of 19 categories***. In most of these remaining 10 categories (e.g. road, vegetation, and terrain) SalsaNext has a comparable performance with the other approaches.

Quantitative Results on the Semantic-KITTI Test-set

	Approach	Size	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean-IoU
Point-wise	Pointnet [15]	50K pts	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.6
	Pointnet++ [16]		53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1
	SPGraph [17]		68:3	0.9	4.5	0.9	0.8	1.0	6.0	0.0	49.5	1.7	24.2	0.3	68.2	22.5	59.2	27.2	17.0	18.3	10.5	20.0
	SPLATNet [22]		66.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.4	0.8	41.5	0.0	68.7	27.8	72.3	35.9	35.8	13.8	0.0	22.8
	TangentConv [37]		86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2	35.9
	RandLa-Net [38]		**94.2**	26.0	25.8	**40.1**	**38.9**	49.2	48.2	7.2	90.7	60.3	73.7	**38.9**	86.9	56.3	81.4	61.3	**66.8**	49.2	47.7	53.9
	LatticeNet [23]		92.9	16.6	22.2	26.6	21.4	35.6	43.0	**46.0**	90.0	59.4	74.1	22.0	88.2	58.8	81.7	**63.6**	63.1	51.9	48.4	52.9
Projection-based	SqueezeSeg [6]	64×2048 pixels	68.8	16.0	4.1	3.3	3.6	12.9	13.1	0.9	85.4	26.9	54.3	4.5	57.4	29.0	60.0	24.3	53.7	17.5	24.5	29.5
	SqueezeSeg-CRF [6]		68.3	18.1	5.1	4.1	4.8	16.5	17.3	1.2	84.9	28.4	54.7	4.6	61.5	29.2	59.6	25.5	54.7	11.2	36.3	30.8
	SqueezeSegV2 [10]		81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	39.7
	SqueezeSegV2-CRF [10]		82.7	21.0	22.6	14.5	15.9	20.2	24.3	2.9	88.5	42.4	65.5	18.7	73.8	41.0	68.5	36.9	58.9	12.9	41.0	39.6
	RangeNet21 [7]		85.4	26.2	26.5	18.6	15.6	31.8	33.6	4.0	91.4	57.0	74.0	26.4	81.9	52.3	77.6	48.4	63.6	36.0	50.0	47.4
	RangeNet53 [7]		86.4	24.5	32.7	25.5	22.6	36.2	33.6	4.7	**91.8**	64.8	74.6	27.9	84.1	55.0	78.3	50.1	64.0	38.9	52.2	49.9
	RangeNet53++ [7]		91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	**91.8**	**65.0**	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9	52.2
	3D-MiniNet [27]		90.5	42.3	**42.1**	28.5	29.4	47.8	44.1	14.5	91.6	64.2	74.5	25.4	89.4	60.8	**82.8**	60.8	66.7	48.0	56.6	55.8
	SqueezeSegV3 [24]		92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9	55.9
	SalsaNet [1]	64×2048 pixels	87.5	26.2	24.6	24.0	17.5	33.2	31.1	8.4	89.7	51.7	70.7	19.7	82.8	48.0	73.0	40.0	61.7	31.3	41.9	45.4
	SalsaNext [Ours]		91.9	**48.3**	38.6	38.9	31.9	**60.2**	**59.0**	19.4	91.7	63.7	**75.8**	29.1	**90.2**	**64.2**	81.8	**63.6**	66.5	**54.3**	**62.1**	**59.5**

- ***Quantitative relationship*** between the ***epistemic uncertainty*** and the ***number of points that each class has*** in the entire Semantic-KITTI test set.
- Network becomes ***less certain*** about ***rare classes*** represented by low number of points (e.g. motorcyclist and motorcycle).
- There also exists, to some degree, *an inverse correlation* between the obtained ***uncertainty*** and the ***segmentation accuracy***, e.g. motorcyclist which has the lowest IoU score (19.4%).



Ablation Study

- The post processing step leads to a certain jump (around 2%) in the accuracy.
- Dilated convolution stack causes a peak in the model parameters, which is vastly reduced after adding the pixel-shuffle layers
- Switching to the pixel-shuffle layers yields 1% more accuracy while having 2.5M less parameters.

	mean IoU (w/o kNN)	mean IoU (+kNN)	Number of Parameters	FLOPs
SalsaNet [1]	43.2	44.4	6.58 M	51.60 G
+ context module	45.0	46.4	6.64 M	69.20 G
+ central dropout	48.5	50.8	6.64 M	69.20 G
+ average pooling	48.9	51.2	5.85 M	66.78 G
+ dilated convolution	50.6	52.3	9.25 M	161.60 G
+ Pixel-Shuffle	51.2	53.3	6.73 M	125.68 G
+ *Lovász-Softmax* loss	**56.4**	**59.9**	6.73 M	125.68 G

ABLATIVE ANALYSIS ON THE VALIDATION SET

- Combining the weighted cross-entropy loss with Lovasz-Softmax leads to the highest increment in the accuracy, since the Jaccard index which is the main metric to measure the segmentation accuracy is directly optimized.
- Consequently, we achieve the highest accuracy score of 59.9% by having only 2.2% (i.e. 0.15M) extra parameters compared to the original SalsaNet.

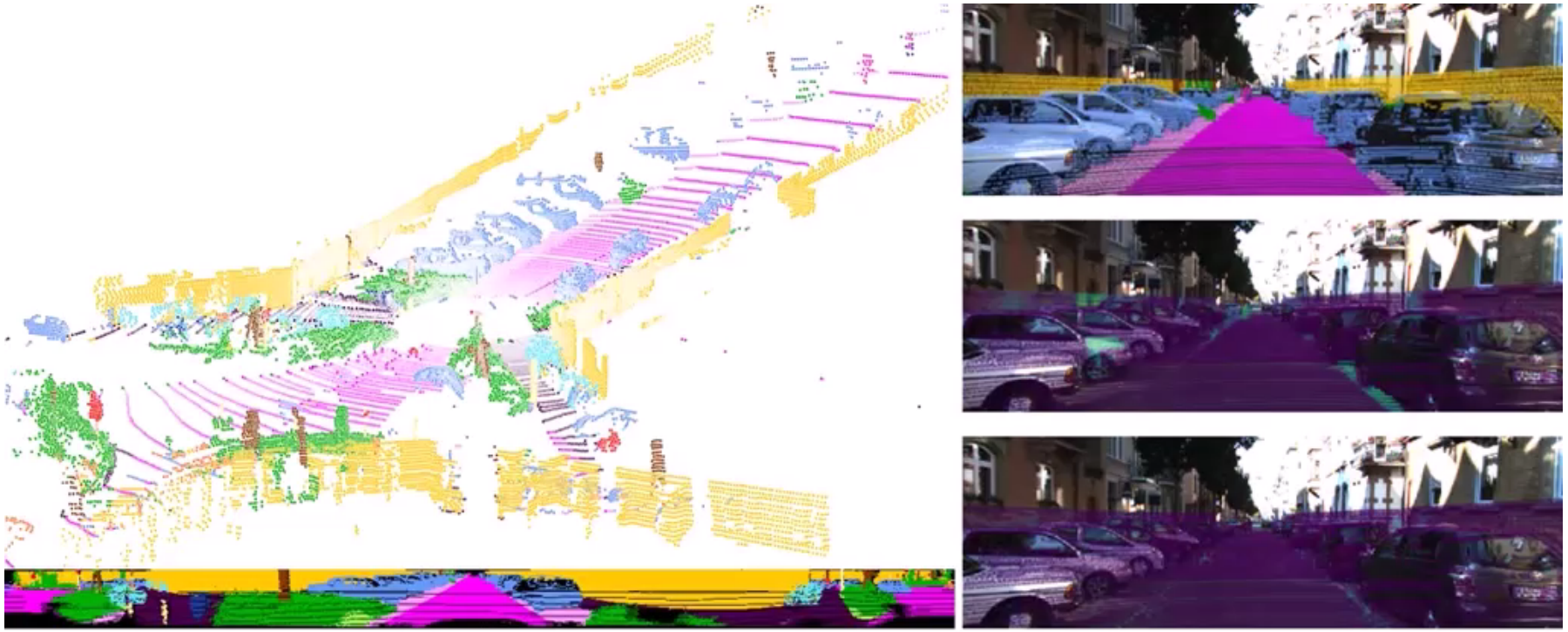
Runtime Evaluation

- SalsaNext exhibits better performance compared to RangeNet++ while having 7x less parameters.
- SalsaNext can run at 24 Hz when the uncertainty computation is excluded for a fair comparison with deterministic models.
- This speed is significantly faster than the sampling rate of mainstream LiDAR sensors which is typically 10 Hz.

	Processing Time (msec)				
	CNN	kNN	Total	Speed (fps)	Parameters
RangeNet++ [7]	63.51	2.89	66.41	15 Hz	50 M
SalsaNet [1]	35.78	2.62	38.40	26 Hz	6.58 M
SalsaNext [Ours]	38.61	2.65	41.26	24 Hz	6.73 M

RUNTIME PERFORMANCE

Qualitative Results on the Semantic-KITTI Test-set



Sample qualitative results: At the bottom the range-view image of the network response is shown. Camera images on the right are only for visualization purposes and have not been used in the training. The top camera image shows the projected segments whereas the middle and bottom images depict the projected epistemic and aleatoric uncertainties. The lighter the color is, the more uncertain the network becomes.

More info:

arXiv.org https://arxiv.org/pdf/2003.03653.pdf

YouTube https://youtu.be/MlSaIcD9ItU

PyTorch https://github.com/TiagoCortinhal/SalsaNext

tiago.cortinhal@hh.se eren.aksoy@hh.se

Conclusion

- We presented a new uncertainty-aware semantic segmentation network, named SalsaNext.
- Our contributions lie in the following aspects that can process the full 360 LiDAR scan in real-time.
- SalsaNext builds up on the SalsaNet model and can achieve over 14% more accuracy.
- In contrast to previous methods, SalsaNext returns +3.6% better mIoU score.
- Our network differs in that SalsaNext can also estimate both data and model-based uncertainty.

More info:

arXiv.org https://arxiv.org/pdf/2003.03653.pdf

YouTube https://youtu.be/MlSaIcD9ItU

PyTorch https://github.com/TiagoCortinhal/SalsaNext

tiago.cortinhal@hh.se eren.aksoy@hh.se

Questions & Comments

More info:

arXiv.org https://arxiv.org/pdf/2003.03653.pdf

YouTube https://youtu.be/MlSaIcD9ItU

PyTorch https://github.com/TiagoCortinhal/SalsaNext

tiago.cortinhal@hh.se eren.aksoy@hh.se