Multipath Parameters Estimation in Physically Based Synthetic Environment Using Robust Deep Neural Regression

Thomas Gonzalez, Antoine Blais, Nicolas Couellan, Christian Ruiz

Multipath parameters estimation in physically based synthetic environment using robust deep neural regression

Thomas GONZALEZ[1,2], Antoine BLAIS[2], Nicolas COUËLLAN[2,3], and Christian RUIZ[1]

[1]Oktal-Synthetic Environment, France
[2]ENAC, Université de Toulouse, France
[3]Institut de Mathématiques de Toulouse, France

BIOGRAPHY

Thomas Gonzalez is a PhD student working with the Oktal-SE company and the ENAC (French Civil Aviation University) research laboratory in Optimization and Telecommunications. Engineer specialized in aerospace telecommunications and optimization for air transport, his current research focuses on the application of Deep Learning techniques to Global Navigation Satellite System (GNSS) interference issues.

Nicolas Couëllan is a professor with the Optimization and Machine Learning Team of the ENAC research laboratory. He is an associated member of the Institute of Mathematics of Toulouse (IMT). His research interests are at the interface of nonlinear optimization and machine learning with a special interest in robustness issues in machine learning and aeronautical applications.

Antoine Blais is assistant professor with the Telecommunication Team of the ENAC research lab. His research activities relate to GNSS receiver signal processing with an emphasis on the application of Machine Learning techniques to signal impairment detection, classification and mitigation.

Christian Ruiz is a research and development engineer responsible for Electromagnetic simulation activities within Oktal-SE company. His main fields of expertise cover radar signal simulation, RF processing and RF interaction with natural media phenomenology.

ABSTRACT

Global Navigation Satellite System (GNSS) signals reception may suffer from multipath propagation. Specifically, in deep urban environment the disturbance can be strong. It may lead to a degraded position solution at the expense of the final user. Despite the development of dedicated techniques like Multipath Estimating Delay Lock Loop (MEDLL) in Townsend et al. (1995), multipath error mitigation is difficult to achieve and still need to be improved. In this work, we propose a Deep Learning (DL) method to estimate the multipath parameters generated by a GNSS signal propagation simulator. As in Munin et al. (2020) we focus on multiple correlator outputs to construct the datasets. The correlator outputs form images containing both signal and multipath information. The number of correlator outputs is flexible in order to generate images with different size, so with various resolutions. We assume that a multipath can be entirely characterized by four parameters namely delay, Doppler frequency shift, magnitude and phase. The disturbed signals are retrieved from a synthetic environment, and more specifically from urban canyons where the multipath phenomenon is worse. A Convolutional Neural Network (CNN) model is trained on physically based synthetic data with various images sizes to assess the proposed CNN algorithm performances. Moreover, a soft-labelling method is applied to enhance the CNN regression task. The data labels take the form of distributions and a specific histogram loss function based on Kullback-Leibler (KL) divergence will be applied during the CNN training.

Keywords— Convolutional Neural Networks, correlator output images, distributional loss, GNSS, Kullback-Leibler divergence, multipath, multi-channel regression, synthetic environment, truncated Gaussian, Von Mises distribution.

I. INTRODUCTION

Multipath phenomenon has been and is still a matter of interest for the GNSS community. Its impact on the receiver can lead to serious deterioration on the positioning accuracy (Quan et al., 2018; Vigneau et al., 2006). Numerous investigations were conducted on multipath detection (Blais et al., 2022; Quan et al., 2018; Savas and Dovis, 2019; Suzuki et al., 2020), multipath

mitigation (Bhuiyan and Lohan, 2010; Tao et al., 2021; Phan et al., 2013) and multipath estimation (Fuchs, 1999; Phan et al., 2013).
The main motivation of this work is to estimate multipath parameters using DL algorithms. Few research works have already being conducted on applying Machine Learning (ML) techniques to multipath estimates. For example Support Vector Machines (SVM) have been employed to estimate the multipath code and carrier phase in Phan et al. (2013). Another ML application can be found in the work of Suzuki and Amano (2021) where a Non Line-Of-Sight (NLOS) multipath detection is based on Neural Networks (NN).

The vast majority of ML studies on multipaths focuses on constructing active features from the correlator output. Alternatively, we propose to apply DL directly on signal images. The images are constructed using correlator outputs. The idea is to compute more correlator outputs than usual to expand the information carried by the received signal after the correlation operation (Munin et al., 2020). The ultimate goal of this strategy is to accurately estimate the multipath components.
In this work, two main assumptions are made. First, we assume that the multipath estimation takes place during the tracking step. In other words the direct signal, or Line-Of-Sight (LOS) signal, components are accurately estimated. This also means that the images built are not acquisition grids. Secondly, we only consider a LOS signal and a single multipath in a first step.

Object classification on images by ML techniques has been done successfully in many application domains such as multi-scale detection in Lin et al. (2017), large scale recognition in Russakovsky et al. (2015) and even for audio application with a CNN structure in Hershey et al. (2017). CNN have proven to be very effective on images-like data due to their sparse connectivity and automatic feature extraction layers. They are able to extract high-level features from input images. The highly informative data are then processed into the fully-connected layers. Neurons in fully connected layers bring together the extracted information to form the final output.

CNN regression has been less studied than CNN classification. Among success of CNN regression applications, one can refer to the following works. Yi et al. (2014) have proposed based human age estimation on images analysis by CNNs, Human poses assessment by CNN was studied by Li and Chan (2014) and another example of a CNN research in regression domain was conducted for real time 2D/3D registration in Miao et al. (2016).

The CNN usually requires a large amount of labelled data to be trained. Multipath labelled data are scarce and difficult to obtain. Therefore one solution is to randomly generate multipath components as in Munin et al. (2020). It provides an estimate of the CNN estimation performance. To improve the model performance assessment, we propose to run experiments in 3D realistic environments. The multipath components generation-estimation chain will follow the diagram illustrated in Figure 1. Multipath signals will be generated from a specialized software in GNSS signal propagation in constrained environments. Implemented realistic urban environments give frame propitious to multipath phenomenon and all of them are then collected. The recovered multipath parameters are used as a reference base for multipath creation. Their reconstruction in the synthetic correlator outputs is made from the multipath parameters collected during the simulation. The CNN algorithm is fed with the generated dataset to carry out the multi-channel regression task.

The article organization is as follows. Section II explains the distinct elements contributing in the dataset generation chain. Section III deals with the DL used for the experiments. Section IV details the conducted experiments with numerical results. Section V concludes the article.

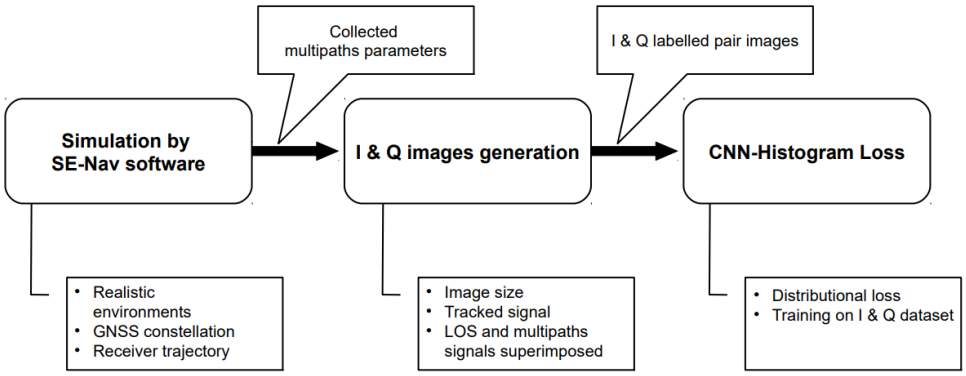


Figure 1: Multipath generation and estimation workflow

II. DATASET CONSTRUCTION

1. Physically based multipath simulator

In order to have enough labelled data to train the CNN, a solution is to generate synthetic multipaths via a generator. In this work, we propose to use the SE-Nav software (Oktal-SE, 2022). It simulates the GNSS signals propagation in constrained environments. The simulator generates realistic scenes like urban environments, suitable for multipath phenomenon. It computes the signal and multipaths propagation paths based on ray-tracing modeling. Multipaths parameters are available in the output files at the end of each simulation. The parameters are used in the synthetic correlator output generator to produce the I & Q image dataset. An alternative dataset construction, sharing the same correlator outputs generator, is illustrated in Figure 11 in Appendix A. Its process is detailed in subsection IV.1 a).

2. Correlator output images generator

In addition to the classical Early-Prompt-Late correlator points, other correlator output values are computed to form a delay x Doppler frequency shift 2D grid as illustrated in Figure 2. By adopting such an approach, additional information is aggregated to the CNN data on both I and Q channels. As said in section I, the signal is assumed to be tracked, meaning that only multipath signal parameters remain unknown. This correlator output image generation process is more precisely explained in Munin et al. (2020). In this work the dataset generator takes as inputs the multipath parameters from the GNSS signal propagation SE-Nav software. In realistic constrained environments which will be detailed in subsections IV.1 b) and IV.1 c), a moving receiver will collect the incoming LOS and its corresponding multipath signals.

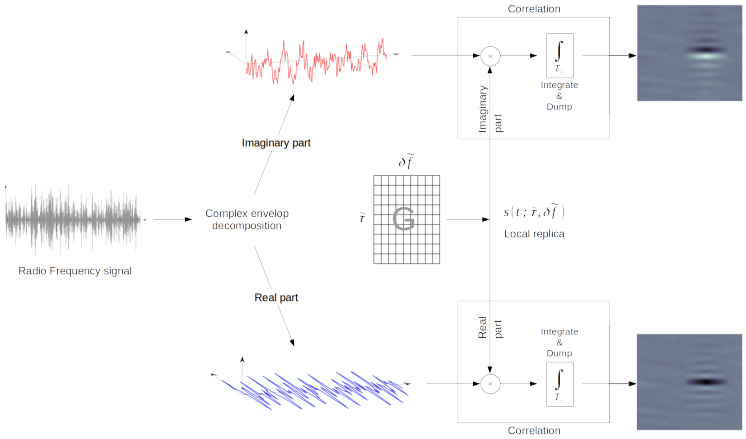


Figure 2: Correlator output images elaboration process. Firstly the received signal is divided into two orthogonal components. Then a grid G is generated from $\tilde{\tau}$ and $\delta\tilde{f}$ subset values. $\tilde{\tau}$ and $\delta\tilde{f}$ refer to the parameters of the local signal replica. The resulting 2D-images are used as an input for the CNN regression algorithm.

III. CNN REGRESSION AND DISTRIBUTION LOSS

The CNN automatic feature extraction mechanism is based on convolution processed by kernel filtering (see Goodfellow et al. (2016) for detailed explanations on filtering mechanism). The CNN model structure used in this work has been inspired by Visual Geometry Group (VGG)-16 architecture. VGG-16 networks have experimentally demonstrated their performances in image-recognition applications (Simonyan and Zisserman, 2014). The particularity of this network structure is the image size reduction while applying a growing number of filters. Figure 3 presents the CNN architecture we have used. The network can be divided in several convolutional blocks followed by dense layers. Each convolutional block is composed of consecutive convolutional layers which extract information from the images. A max pooling layer completes the convolutional block. It synthesizes the extracted information and accelerates data processing in the network. The fully-connected layers carry out the regression task on the extracted features from previous block.

The usual concept of regression task consists in building a model able to compute real responses from input x. The regression model N_θ, where θ represents the model parameter, has to be trained in order to predict values close to the known labels y. The

model parameter is calibrated by minimizing for example a square loss function $l(\theta,x,y) = (N_\theta(x) - y)^2$. The target value y, considered as the ground-truth label of a given input x, is experimentally obtained or measured. This presence of noise during the labelling of the data can lead to a regression on a label value that is false, and thus hinder the generalization process. The soft labelling technique starts from the premise that the measured value y is a random observation from a prior distribution $Y|x$. The main idea of this method is the estimation of the underlying distribution of the target $Y|x$ instead of estimating the sharp value y. Hence, the soft label of a given input x will be $Y|x$ with $y = \mathbb{E}[Y|x]$ where $Y|x$ is a prior distribution made on the target y.

The $Y|x$ distribution will undergo a discretization in order to be implemented in a ML algorithm. The resulting histogram formed on K bins is the discrete version of $Y|x$. As it can be observed in Figure 3, the CNN will be trained to reproduce the soft labels thanks to the multiple outputs in the last layer. In the sequel of the article, CNN-HL will refer to a CNN with a histogram distribution output. To minimize the difference between the target distribution $Y|x$ and the CNN-HL prediction, the square loss function is substituted by a histogram loss. Based on KL divergence, it computes the dissimilarities between two probability distributions. For two continuous probability distributions P and Q, the KL divergence is

$$D_{KL}(P||Q) = P\log(\frac{P}{Q}) \tag{1}$$

The histogram loss, noted HL, is the score of dissimilarities between the predicted an real distributions. Its expression is derived from equation 1 and can be simplified as:

$$HL(\theta,x) = -\sum_{i=1}^{K} p_i \log(q_i(\theta,x)) \tag{2}$$

with

- p_i the probability mass of the target y in bin i,
- $q_i(\theta,x)$ the i^{th} output of the network for an input x,
- K the number of outputs of the CNN-HL.

Next, we discuss two relevant choices for the prior distributions p.

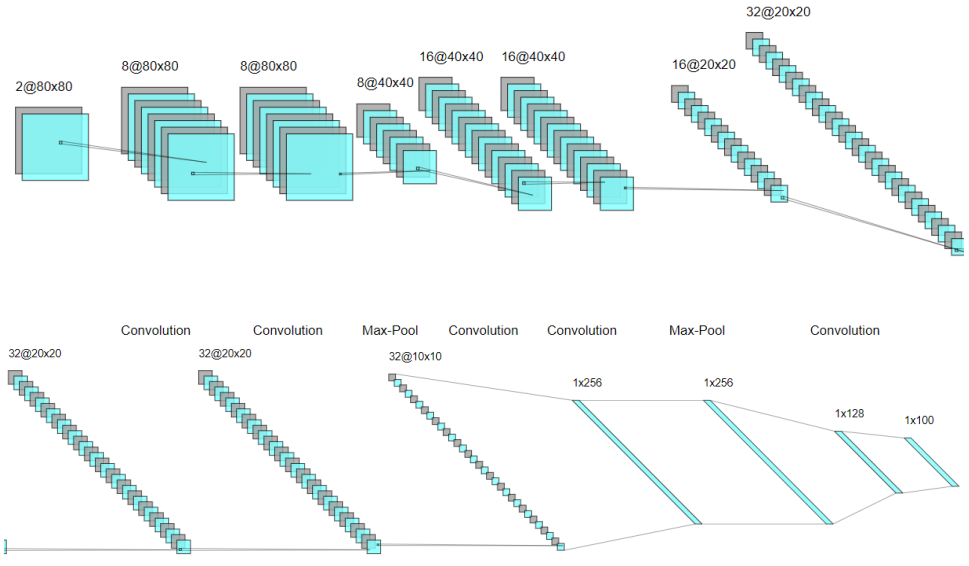


Figure 3: CNN architecture used for regression task with distributional outputs on 80x80 images. The top figure illustrates the first convolution blocks. Bottom figure shows the last convolution block followed by the dense layers with a multiple outputs layer. Images generated with NN-SVG LeNail (2019).

1. Truncated Gaussian distribution

Distinct probability distributions have been used in experiments by Imani and White (2018). However they have experimentally shown that the truncated Gaussian is a good default choice. This means that for an input x with its ground-truth label y_t, the probability density function p can be expressed as:

$$\forall y \in [a,b], p(y|x) = \frac{1}{Z\sqrt{2\pi\sigma^2}} e^{-\frac{(y-y_t)^2}{2\sigma^2}} \text{ with } Z = \frac{1}{2}(erf(\frac{b-y_t}{\sqrt{2\sigma^2}}) - erf(\frac{a-y_t}{\sqrt{2\sigma^2}})) \tag{3}$$

where

- $x \mapsto erf(x)$ is the Gauss error function,
- $[a,b]$ is the data targets support set,
- y_t the ground truth target value of the input x,
- σ^2 the variance parameter.

Figure 4 is an example with a truncated Gaussian distribution for a Doppler frequency shift target equal to 45Hz.

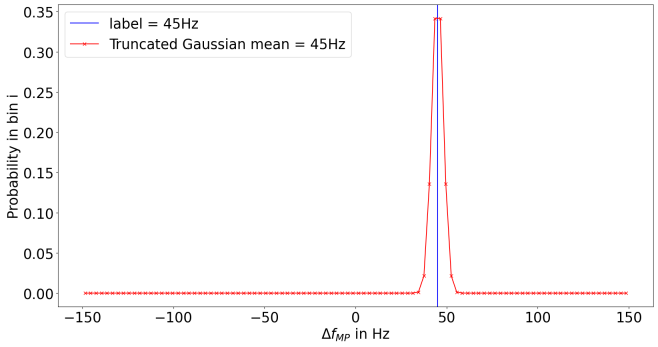


Figure 4: Truncated Gaussian probability distribution with Δf_D = 45 Hz.

2. Von Mises distribution

In prior research work (Gonzalez et al., 2022), it has been observed that the truncated Gaussian distribution show difficulties concerning the multipath phase estimation. It is not adapted for angular targets. To take in account the cyclical behaviour of angles, we opted for a Von Mises probability distribution. Figure 5 shows the difference between truncated Gaussian and Von Mises distributions for a multipath phase ground-truth value equals to 0°. The truncated Gaussian distribution has only one peak for 0° whereas the Von Mises distribution shows two peaks around 0° and 360°. It better reflects proximity of angles when a modulo 2π is applied. For this probability distribution, the probability density function p is:

$$p(y|x) = \frac{e^{\kappa cos(y-y_t)}}{2\pi I_0(\kappa)} \tag{4}$$

where

- I_0 is the modified Bessel function of order 0,
- $\frac{1}{\kappa}$ is analogous to σ^2 for a normal probability distribution.

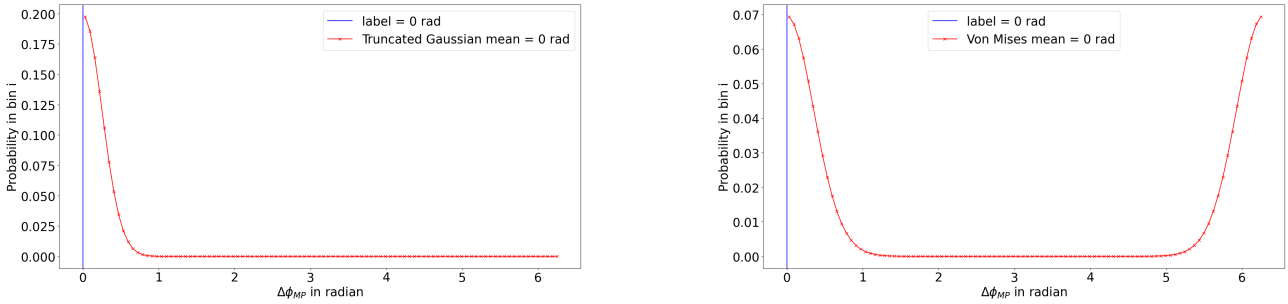


Figure 5: Comparison between truncated Gaussian (left) and Von Mises (right) distributions for $\Delta\phi_{MP} = 0$ rad.

IV. EXPERIMENTS

1. Experimental setup

Experiments were made using three distinct datasets. Each dataset contains 10 000 samples of labelled I and Q images pairs. Whitin a dataset the image size is constant. However, to observe the evolution in performance of our CNN-HL model with less correlator outputs (meaning decreasing image resolution), several variations of these datasets were created for 80x80, 40x40 and 20x20 pixels images. The datasets are further detailed in the following sections.

a) Dataset with randomly generated multipaths

A dataset with randomly generated multipaths is constructed as a baseline for CNN performance assessment. The random multipath parameters are generated as follow:

- $\alpha_{MP} \sim \mathscr{U}[0.1, 0.9]$ the attenuation coefficient of the multipath with respect to the LOS amplitude,
- $\Delta\tau_{MP} \sim \mathscr{U}[0, 1.5\text{Tc}^1]$ the code delay in excess compared with the useful signal,
- $\Delta f_{MP} \sim \mathscr{N}(0, \frac{125}{3})$ the difference in Doppler shift with the useful signal,
- $\Delta\phi_{MP} \sim \mathscr{U}[0, 2\pi]$ the phase difference with the useful signal.

All the multipaths in the 'Random multipaths' dataset were generated at a *C/N0* level fixed to 40dB − Hz to simulate a typical urban environment receiving condition.

b) Dataset with 'Toulouse center, France' 3D environment

The Capitole square of Toulouse is located in the center of the French city Toulouse. Characterized by its small streets surrounded by 3-4 floor buildings, Toulouse city center is propitious to multipath phenomenon. As it can be seen in Figure 6, main materials are brick and concrete for this scene. Four distinct trajectories were constructed for a mobile receiver to have massive signal collection with many cases of multipath. To supplement the dataset, the receiver follows those four trajectories twice, at different times to have distinct satellite constellation configurations.

The observed multipaths components distributions are plotted in Figure 7. Observations on multipath phase shift support the assumption made in section IV.1 a) with a uniform distribution of this parameter. Doppler frequency shift also follows a normal distribution, but with a different variance. The observations are interesting also for the other parameters since we can note that the delay in excess and the multipath attenuation seem to follow an exponential law. To form the various datasets, 10 000 multipaths are randomly picked among all the observed interference to reproduce in the synthetic correlator outputs image generator.

[1]Tc is the chip period, a basic defining parameter of this type of signal. Tc $= 1/1023$ ms for the Global Positioning System (GPS) L1 C/A signal.

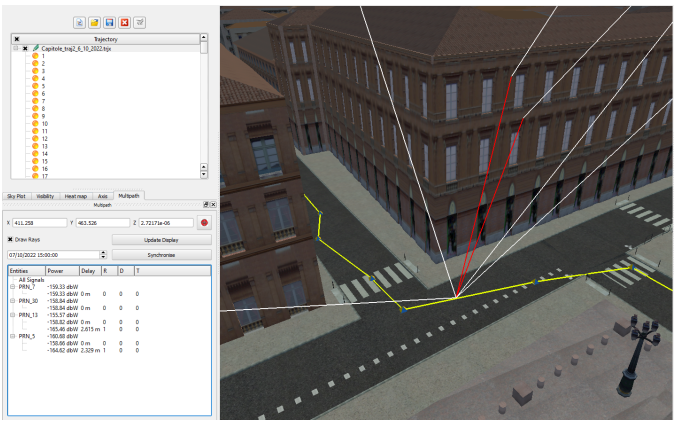


Figure 6: Capitole place (Toulouse center) multipath environment, SE-Nav software

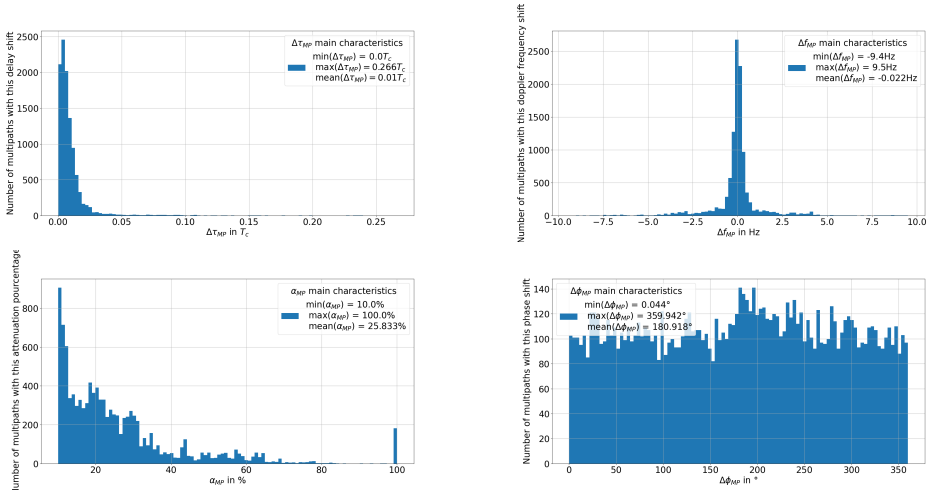


Figure 7: Multipaths components distribution in the Toulouse city receiving simulations (10 722 in total).

c) Dataset with 'Manhattan, NYC (New York City), United States of America' 3D environment

The Manhattan scene in SE-Nav software is an environment characterized by sky-scrapper buildings and straight-lined streets. Even if streets are not as narrow as in Toulouse city, multipath phenomenon is present due to the high reflection potential of the high buildings. The receiver environment is mainly composed of glass and concrete (as it is shown in Figure 8). The observed multipaths distributions are different compared with the Toulouse collect. In Manhattan, multipaths are stronger and with larger delays than the multipaths generated in the French city. For the same reason than for Toulouse city dataset, the signals collect is made over four different trajectories at two distinct times of day.

Doppler frequency shifts in NYC environment have a similar distribution (Figure 9) compared with the multipaths generated in Toulouse environment. Nevertheless, two parameters raise questions on them: the multipath magnitude coefficient seems to be attracted around two extreme values, 10% and 100% of the LOS signal. The amount of multipaths with a 100% attenuation coefficient (meaning with the same magnitude as the LOS signal), could be explained by the receiver local environment, surrounded by a large amount of metal structure and allowing a near 'perfect' reflection on those surfaces. The second parameter catching attention is the phase shift, with a uniform distribution of the multipath phases except between 225° and 275°, where the multipaths with such phase shift form a larger group.

2. Experiment setup

The figures or values provided in the next section are results averaged over 10 runs. For each run, the CNN-HL dataset is formed as follow: 8000 samples for the training set and 1000 samples for the validation set (both are used during the CNN-HL

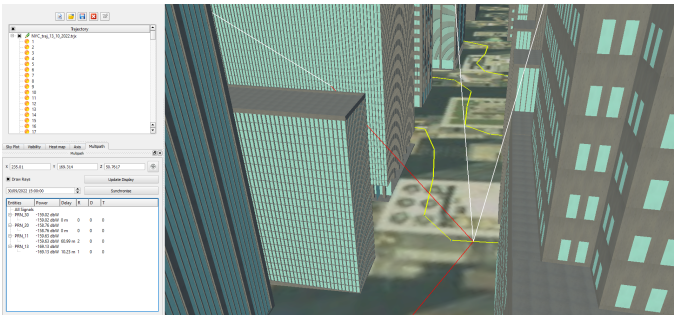


Figure 8: Manhattan multipath environment, SE-Nav software

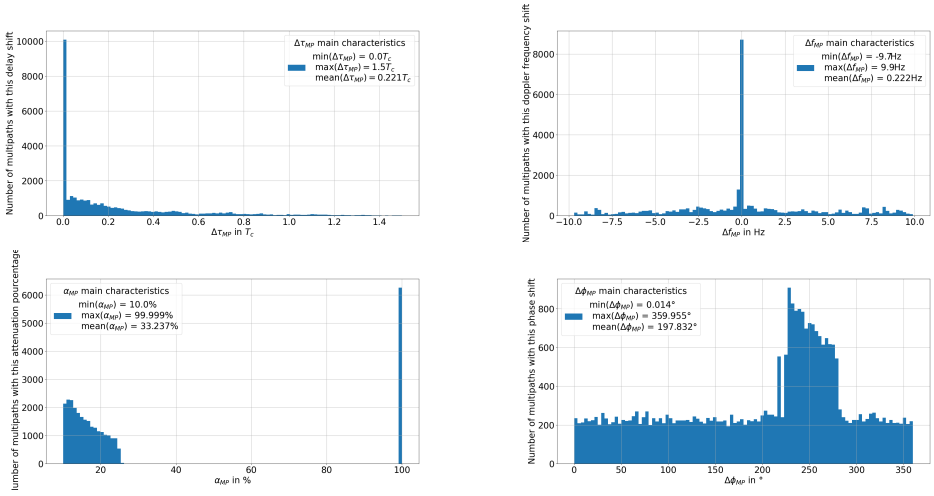


Figure 9: Multipaths components distribution in the NYC receiving simulations (30 676 in total)

training phase). Then the last 1000 samples are to assess the performances of the trained network. The dataset is shuffled between each run to give distinct training conditions to the CNN-HL. The network is trained on 100 epochs with a batch size of 1000 samples. The CNN-HL learning rate is empirically fixed to 10^{-3}. Convolutional filters are fixed to a 3x3 size while max-pooling kernels are 2x2 size as it is shown in Figure 3. The number of outputs of the neural network is fixed to K = 100 bins. Preliminary experiments were made on 80x80 images of all dataset to determine the 'best' σ and κ hyper-parameters for the estimation task. The corresponding tables and figures are available in Appendix C. Then the experiments on various image sizes were made with those optimal hyper-parameters.

3. Results

Performance evaluation is based on Mean Absolute Error (MAE) results averaged over 10 runs and computed as follows:

$$\frac{1}{L}\sum_{l=1}^{L}\left|\tilde{\mathbb{E}}[N_\theta(x_l)]-y_l\right| \quad \text{with} \quad \tilde{\mathbb{E}}[N_\theta(x_l)]=\sum_{i=1}^{K} q_i(\theta,x_l).c_i$$

and

- L is the number of samples in the test dataset,
- x_l is the l-th test sample,
- y_l is the target value for the l-th sample.
- $q_i(\theta,x_l)$ is the probability of the i-th softmax output neuron associated to x_l,
- c_i is the center of the i-th bin for $i \in [1,K]$ (where the interval $[a,b]$ has been partitioned into K equal subdivisions).

Tables 1,2,3 and 4 gather the average MAE performances of the CNN-HL on the four multipath components for the various datasets. For those experiments, σ and κ values where set following the Figure 10 results. The plots in Figure 10 illustrate the optimal σ and κ parameters for CNN-HL on each 80x80 dataset and for each multipath parameter. The x-axis is labelled according to the bin width w since $\sigma = \beta w$, $\beta \in \mathbb{R}^+$. The estimation performances are plotted according to increasing σ for delay, Doppler frequency and attenuation, and κ for the phase. Tables 5, 6, 7 and 8 in Appendix C show the numeric values of the plots. The Figure 10 gives an idea of the prior distributions parameter impact on estimation performances.

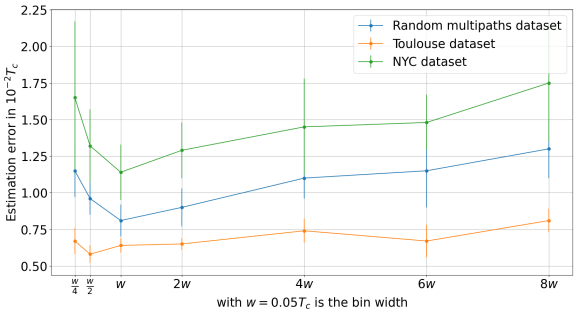
4. Discussion

a) Tests on various σ and κ values

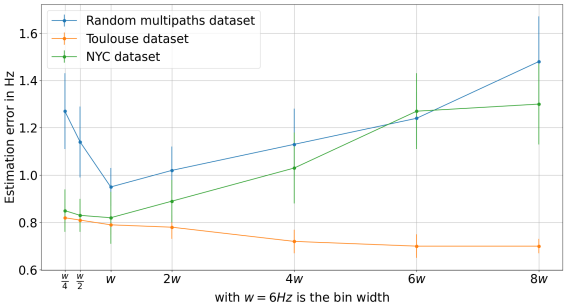
As it can be seen in Figure 10, σ values have an impact on the averaged parameter estimation accuracy. For example in Figure 10.a, a minimum in the delay estimation error performance can be observed for both 'Random multipaths' and NYC datasets whereas there is no significant influence on the delay estimation on the Toulouse multipath set. Figure 10.b shows lower Doppler frequency estimation errors for small σ on 'Random' and NYC datasets. On the contrary, the Doppler frequency estimation performs better for high σ values when estimating multipaths collected in Toulouse dataset. In Figures 10.c and 10.d, small σ and κ values lead to a raise of the parameter estimation error. For κ adjustment, the estimation error variance is consequent and does not allow the finding of an optimal κ value for the phase estimation. It can also be observed that for a same σ or κ value, the estimation performances can significantly vary depending on the multipath generation. An example of this is illustrated in Figure 10.a where the delay estimation error on Toulouse multipaths is always lower than the delay estimation error on NYC multipaths.

b) Multipath parameters estimation

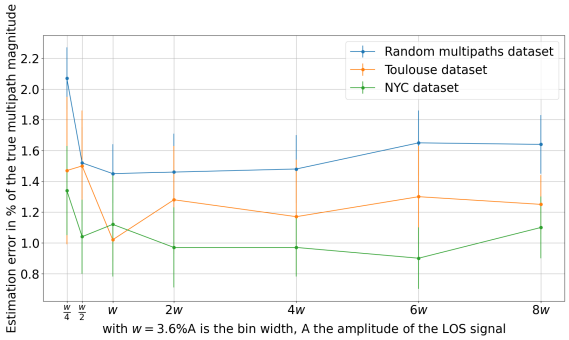
The results on the tables show the impact of the image resolution on the parameter estimation errors. The error raises as the resolution of the input images deteriorates, with the example of the delay estimation in Table 1 on 'Random multipaths' and NYC datasets. We notice a steadiness in delay estimation for Toulouse multipaths where the estimation performance has a lower error compared with the Random multipaths. This consistency in the estimation error is also observable for the Doppler frequency when estimating Toulouse multipaths. A similar behaviour is noticeable for the multipath magnitude error performance for NYC dataset. For the delay and Doppler frequency estimation, we believe that Toulouse estimation performances are better due to the dataset distribution (in Figure 7 in section IV.1 b), $\Delta\tau_{MP} \in [0, 0.3\text{Tc}]$). Generally, the CNN-HL estimation performances are similar between randomly generated multipaths and interference collected on simulations.



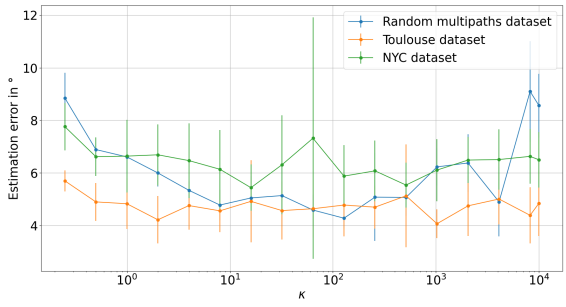
(a) Averaged $\Delta\tau_{MP}$ estimation performances for 80x80 pixel images with respect to the bin width w.



(b) Averaged Δf_D estimation performances for 80x80 pixel images with respect to the bin width w.



(c) Averaged α_{MP} estimation performances for 80x80 pixel images with respect to the bin width w.



(d) Averaged ϕ_{MP} estimation performances for 80x80 pixel images with respect to κ.

Figure 10: Averaged estimation performances for 80x80 pixel images with respect to σ and κ parameters

Table 1: Average MAE $\Delta\tau_{MP}$ in 10^{-2}Tc for different simulated environments.

Dataset	Image size	CNN-HL
Random multipaths	80x80	0.81 ± 0.11
	40x40	1.20 ± 0.21
	20x20	2.21 ± 0.29
SE-Nav Toulouse	80x80	0.64 ± 0.05
	40x40	0.71 ± 0.07
	20x20	0.62 ± 0.06
SE-Nav New York City	80x80	1.14 ± 0.19
	40x40	1.53 ± 0.31
	20x20	2.69 ± 0.36

Table 2: Average MAE Δf_D in Hz for different simulated environments.

Dataset	Image size	CNN-HL
Random multipaths	80x80	0.95 ± 0.08
	40x40	1.10 ± 0.13
	20x20	2.10 ± 0.28
SE-Nav Toulouse	80x80	0.79 ± 0.04
	40x40	0.74 ± 0.06
	20x20	0.72 ± 0.05
SE-Nav New York City	80x80	0.82 ± 0.11
	40x40	0.85 ± 0.13
	20x20	1.29 ± 0.09

Table 3: Average MAE α_{MP} in % for different simulated environments.

Dataset	Image size	CNN-HL
Random multipaths	80x80	1.45 ± 0.19
	40x40	1.60 ± 0.14
	20x20	2.21 ± 0.28
SE-Nav Toulouse	80x80	1.02 ± 0.14
	40x40	1.14 ± 0.11
	20x20	1.59 ± 0.26
SE-Nav New York City	80x80	1.12 ± 0.34
	40x40	0.86 ± 0.22
	20x20	1.08 ± 0.14

Table 4: Average MAE $\Delta\phi_{MP}$ in degrees for different simulated environments.

Dataset	Image size	CNN-HL
Random multipaths	80x80	5.07 ± 0.69
	40x40	5.43 ± 0.49
	20x20	6.63 ± 0.40
SE-Nav Toulouse	80x80	5.13 ± 1.96
	40x40	4.24 ± 0.59
	20x20	4.36 ± 0.57
SE-Nav New York City	80x80	5.54 ± 0.85
	40x40	5.37 ± 0.82
	20x20	6.30 ± 0.59

V. CONCLUSION

In this work we have proposed to estimate the multipath components using a CNN model that embeds a soft-labelling mechanism. Synthetic correlator output images were generated based on GNSS signal propagation simulations. The constructed I & Q images pair gathers information about the tracked signal and one multipath. Regression task on images is a difficult task for a DL algorithm. Soft labelling technique was applied to improve the multi-channel regression performances of the network. This was possible with a softmax output layer estimating the distribution of the parameters to estimate. The histogram loss function used during the training was constructed from the Kullback-Leibler divergence. It allowed to compare the target histograms and the predicted distributions. The experiments in section IV were conducted on several datasets in order to measure the stability and robustness of the algorithm facing multipaths from various environments. 'Random multipath' data was generated according to assumed hypothesis while NYC and Toulouse city sets were constructed on signal propagation simulator results. The achieved averaged MAE demonstrates the capacity of the CNN-HL to estimate the delay, Doppler frequency, attenuation and phase of the multipath. The CNN-HL shows similar performances when estimating randomly generated multipaths or as outputs of physically-based signal propagation software. The experiments also shown the importance of the hyper parameters values to reach the optimal performances of the CNN-HL estimation. In addition, we have proposed the use of the prior Von Mises distribution for angular labels. Comparing with the results on the same dataset in (Gonzalez et al., 2022), the new prior distribution gives improved performances. Results on phase estimation by CNN-HL based on Von Mises distribution may encourage the use of this technique in angular regression problems. From a model perspective, future investigations should focus on more realistic dataset with the estimate of several multipaths at the correlator outputs.

REFERENCES

Bhuiyan, M. Z. H. and Lohan, E. S. (2010). Advanced multipath mitigation techniques for satellite-based positioning applications. *International Journal of Navigation & Observation*.

Blais, A., Couellan, N., and Munin, E. (2022). A novel image representation of GNSS correlation for deep learning multipath detection. *Array*, 14:100167.

Fuchs, J.-J. (1999). Multipath time-delay detection and estimation. *IEEE transactions on signal processing*, 47(1):237–243.

Gonzalez, T., Blais, A., Couëllan, N., and Ruiz, C. (2022). Distributional loss for convolutional neural network regression and application to GNSS multi-path estimation. *arXiv preprint arXiv:2206.01473*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., et al. (2017). CNN architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE.

Imani, E. and White, M. (2018). Improving regression performance with distributional losses. In *International Conference on Machine Learning*, pages 2157–2166. PMLR.

LeNail, A. (2019). NN-SVG: Publication-ready neural network architecture schematics. *J. Open Source Software*, 4(33):747.

Li, S. and Chan, A. B. (2014). 3D human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Miao, S., Wang, Z. J., and Liao, R. (2016). A CNN regression approach for real-time 2D/3D registration. *IEEE transactions on medical imaging*, 35(5):1352–1363.

Munin, E., Blais, A., and Couellan, N. (2020). Convolutional neural network for multipath detection in GNSS receivers. In *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pages 1–10. IEEE.

Oktal-SE (2010-2022). SE-Nav software. 2022.1.22206 version.

Phan, Q.-H., Tan, S.-L., McLoughlin, I., and Vu, D.-L. (2013). A unified framework for gps code and carrier-phase multipath mitigation using support vector regression. *Advances in Artificial Neural Systems*, 2013.

Quan, Y., Lau, L., Roberts, G. W., Meng, X., and Zhang, C. (2018). Convolutional neural network based multipath detection method for static and kinematic GPS high precision positioning. *Remote Sensing*, 10(12):2052.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Savas, C. and Dovis, F. (2019). Multipath detection based on k-means clustering. In *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, pages 3801–3811.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Suzuki, T. and Amano, Y. (2021). NLOS multipath classification of GNSS signal correlation output using machine learning. *Sensors*, 21(7):2503.

Suzuki, T., Kusama, K., and Amano, Y. (2020). NLOS multipath detection using convolutional neural network. In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, pages 2989–3000.

Tao, Y., Liu, C., Chen, T., Zhao, X., Liu, C., Hu, H., Zhou, T., and Xin, H. (2021). Real-time multipath mitigation in multi-GNSS short baseline positioning via CNN-LSTM method. *Mathematical Problems in Engineering*, 2021.

Townsend, B., Fenton, P., Van Dierendonck, K. J., and Van Nee, D. R. (1995). Performance evaluation of the multipath estimating delay lock loop. *Navigation*, 42(3):502–514.

Vigneau, W., Nouvel, O., Manzano-Jurado, M., Sanz, C. C., Abdulkader, H., Roviras, D., Juan, J., and Holsters, P. (2006). Neural networks algorithms prototyping to mitigate GNSS multipath for LEO positioning applications. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, pages 1752–1762.

Yi, D., Lei, Z., and Li, S. Z. (2014). Age estimation by multi-scale convolutional network. In *Asian conference on computer vision*, pages 144–158. Springer.

APPENDIX A. MULTIPATH GENERATION PROCESS

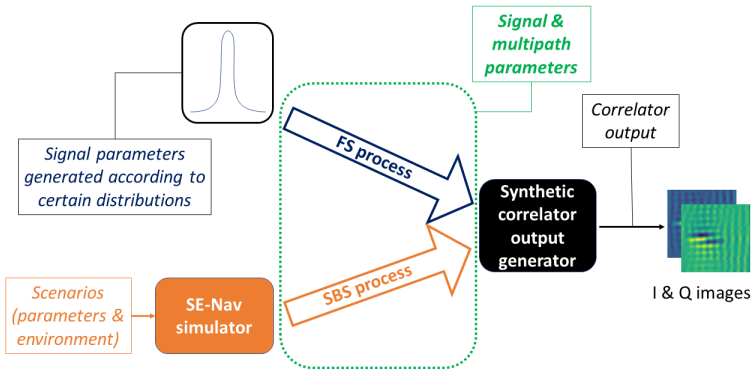


Figure 11: Representation of the different multipath I and Q images generation processes. The Fully Synthetic (FS) is the method employed for 'Random multipaths' dataset. The second approach, the Scenario Based Synthetic (SBS) method, uses a distinct parameters generation scheme. Based on the multipath parameters collected on the SE-Nav simulations, this strategy was used for Toulouse and Manhattan datasets.

APPENDIX B. HISTOGRAMS DEPENDING ON SIGMA VALUE

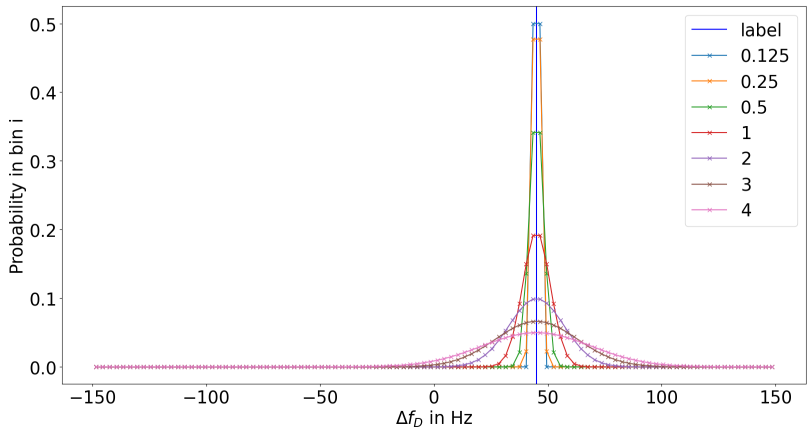


Figure 12: Different histograms based on the truncated Gaussian distribution are plotted according to σ values for a $\Delta f_D = 45$Hz. As it can be seen, the distribution is wider as σ parameter is larger.

APPENDIX C. AVERAGE ESTIMATION PERFORMANCES FOR 80X80 PIXEL IMAGES WITH RESPECT TO σ AND κ PARAMETERS

Table 5: Averaged MAE $\pm\sigma$ for the $\Delta\tau_{MP}$ estimation in Tc.

σ	Random multipaths	SE-Nav (SE-Nav) datasets	
		Toulouse	NYC
$\frac{w}{4}$	1.15 ± 0.18	0.67 ± 0.09	1.65 ± 0.52
$\frac{w}{2}$	0.96 ± 0.11	$\mathbf{0.58 \pm 0.06}$	1.32 ± 0.25
w	$\mathbf{0.81 \pm 0.11}$	0.64 ± 0.05	$\mathbf{1.14 \pm 0.19}$
$2w$	0.9 ± 0.13	0.65 ± 0.04	1.29 ± 0.19
$4w$	1.10 ± 0.14	0.74 ± 0.08	1.45 ± 0.33
$6w$	1.15 ± 0.25	0.67 ± 0.11	1.48 ± 0.19
$8w$	1.30 ± 0.20	0.81 ± 0.08	1.75 ± 0.40

Table 6: Averaged MAE $\pm\sigma$ for the Δf_D estimation in Hz.

σ	Random multipaths	SE-Nav datasets	
		Toulouse	NYC
$\frac{w}{4}$	1.27 ± 0.16	0.82 ± 0.04	0.85 ± 0.09
$\frac{w}{2}$	1.14 ± 0.15	0.81 ± 0.10	0.83 ± 0.07
w	$\mathbf{0.95 \pm 0.08}$	0.79 ± 0.04	$\mathbf{0.82 \pm 0.11}$
$2w$	1.02 ± 0.10	0.78 ± 0.05	0.89 ± 0.09
$4w$	1.13 ± 0.15	0.72 ± 0.05	1.03 ± 0.15
$6w$	1.24 ± 0.11	0.70 ± 0.05	1.27 ± 0.16
$8w$	1.48 ± 0.19	$\mathbf{0.70 \pm 0.03}$	1.30 ± 0.17

Table 7: Average MAE $\pm\sigma$ for the α_{MP} estimation.

σ	Random multipaths	SE-Nav datasets	
		Toulouse	NYC
$\frac{w}{4}$	2.07 ± 0.20	1.47 ± 0.48	1.34 ± 0.29
$\frac{w}{2}$	1.52 ± 0.23	1.50 ± 0.36	1.04 ± 0.24
w	$\mathbf{1.45 \pm 0.19}$	$\mathbf{1.02 \pm 0.14}$	1.12 ± 0.34
$2w$	1.46 ± 0.25	1.28 ± 0.35	0.97 ± 0.26
$4w$	1.48 ± 0.22	1.17 ± 0.37	0.97 ± 0.19
$6w$	1.65 ± 0.21	1.30 ± 0.33	$\mathbf{0.90 \pm 0.20}$
$8w$	1.64 ± 0.19	1.25 ± 0.19	1.10 ± 0.20

Table 8: Average MAE $\pm\sigma$ for the ϕ_{MP} estimation.

κ	Random multipaths	SE-Nav datasets	
		Toulouse	NYC
0.25	8.85 ± 0.96	5.70 ± 0.40	7.76 ± 0.90
0.5	6.89 ± 0.45	4.90 ± 0.72	6.62 ± 0.73
1	6.61 ± 0.63	4.83 ± 0.96	6.64 ± 1.38
2	6.00 ± 0.52	4.22 ± 0.90	6.69 ± 1.16
4	5.34 ± 0.37	4.76 ± 0.91	6.47 ± 1.42
8	4.78 ± 0.57	4.56 ± 0.81	6.14 ± 1.49
16	5.05 ± 0.82	4.92 ± 1.56	$\mathbf{5.44 \pm 0.88}$
32	5.14 ± 1.17	4.57 ± 1.10	6.31 ± 1.88
64	4.59 ± 0.58	4.64 ± 1.09	7.32 ± 4.59
128	$\mathbf{4.28 \pm 0.38}$	4.78 ± 1.19	5.88 ± 1.18
256	5.08 ± 1.67	4.70 ± 0.81	6.08 ± 1.15
512	5.07 ± 0.69	5.13 ± 1.96	5.54 ± 0.85
1024	6.23 ± 0.69	$\mathbf{4.07 \pm 0.55}$	6.11 ± 1.18
2048	6.38 ± 1.09	4.75 ± 1.14	6.49 ± 0.87
4092	4.90 ± 1.31	5.01 ± 0.97	6.51 ± 1.15
8196	9.10 ± 1.91	4.39 ± 1.07	6.63 ± 1.04
10000	8.56 ± 1.21	4.85 ± 1.24	6.50 ± 1.05