

Neutrino interaction event filtering at liquid argon time projection chambers using neural networks with minimal input model bias

Marta Babicz,^{a,*}[∩] Saúl Alonso-Monsalve^b and Stephen Dolan^b

^aIFJ-PAN,

ul. Radzikowskiego 152, Krakow, Poland

^bCERN

Espl. des Particules 1, Meyrin, Switzerland

E-mail: marta.babicz@ifj.edu.pl, saul.alonso.monsalve@cern.ch,
stephen.joseph.dolan@cern.ch

In current and future neutrino oscillation experiments using liquid argon time projection chambers (LAr-TPCs), a key challenge is identifying neutrino interactions from the pervading cosmic-ray background. Rejection of such background is often possible using traditional cut-based selections, but this typically requires the prior use of computationally expensive reconstruction algorithms. This work demonstrates an alternative approach of using 3D Convolutional Neural Networks (CNNs) trained on low-level timing information from only the scintillation light signal of interactions inside LAr-TPCs. We further present a means of mitigating biases from imperfect simulations by applying Domain Adversarial Neural Networks (DANNs). These techniques are applied to example simulations from the ICARUS detector, the far detector of the Short Baseline Neutrino program at Fermilab. The results show that cosmic background is reduced by up to 74% whilst neutrino interaction selection efficiency remains over 94%, even in cases where the simulation poorly describes the data.

37th International Cosmic Ray Conference (ICRC 2021)

July 12th – 23rd, 2021

Online – Berlin, Germany

*Presenter.

[∩]The work of M. Babicz was supported by the National Science Center, Poland, research project No. 2019/33/N/ST2/02874.

1. Introduction

The current and next generation of neutrino oscillation experiments offer a tantalising opportunity to explore physics beyond the Standard Model. However, as detectors grow larger and neutrino beams more powerful, a pre-filtering of relevant neutrino interaction data becomes increasingly important. This is particularly crucial for liquid argon time projection chambers (LAr-TPCs), which are commonly used for detecting neutrino interactions [1]. LAr-TPCs offer precise spatial and calorimetric measurements based on the electron drift signal from argon ionisation due to neutrino interaction products but typically suffer from the significant background, particularly from cosmic radiation. For example, at the ICARUS detector of the SBN experiment, cosmic rays are expected to outnumber neutrino interactions within the Booster Neutrino Beam's (BNB) spill window by more than three to one [2]. Even located 1.5 km underground, the DUNE far detectors will also still experience a comparable rate of cosmic rays and neutrinos [3].

In addition to the argon ionisation signal, readout by the TPC anode wires, the products of neutrino interactions in LAr-TPCs cause the emission of scintillation light, recorded by photodetectors. The most common choice for photodetectors are Photomultiplier Tube (PMT). Unlike the TPC signal, the PMT signal contains only relatively coarse information. Nevertheless, it provides an easily accessible means to classify events requiring little or no processing, which may help distinguish cosmic rays from neutrino interactions before running any reconstruction algorithms.

Cosmic-ray background rejection in LAr-TPCs typically starts at the online trigger based on PMT information to reject the events outside the neutrino beam spill window. However, it does not prevent the acceptance of all background events that enter the detector during the beam spill window. More sophisticated filtering involves multi-dimensional analyses, as the discrimination power stems from which PMTs received light at what time and for how long relative to all other PMTs inside the LAr-TPC. Machine learning methods, capable of analysing such high dimensional information, are therefore excellently suited to classifying events using the available PMT information. In particular, when detector data is represented as images, the use of Convolutional Neural Networks (CNNs) [4, 5] are especially effective. The use of CNNs for event classification is well established across the field of neutrino physics [6–8].

CNNs are the algorithms that learn how to classify images from a training sample (e.g., simulated neutrino events) to be later used to make accurate predictions on new images (e.g., detector data). They are a powerful tool for classifying events but, like many analysis methods, they can suffer from a strong reliance on the details of the input simulation such that minor changes can result in significantly altered classification performance. To address this problem, the CNN classifier may rely on domain adaptation (DA) techniques [9, 10] so that the classifier learned from the training domain (i.e., simulated data) can also be applied to the testing domain (i.e., eventual experimental data). This DA can be achieved through the application of Domain Adversarial Neural Networks (DANNs) [11], in which the detector data is used in an unsupervised (or semi-supervised) manner to prevent the CNN exploiting features that differ between data and simulation.

In this manuscript, we first use a CNN to classify simulated interactions in a LAr-TPC as either from incoming cosmic rays or neutrinos. For this analysis, we use the ICARUS detector of the SBN program, which is currently the world's largest LAr-TPC used in neutrino physics and operates close to the surface (thus, it is subject to a particularly challenging cosmic-ray background rejection). The

details of the ICARUS detector and simulation are summarised in Sec. 2. The CNN approach to event filtering is detailed and demonstrated in Sec 3. The application of a DANNs to reduce the CNN sensitivity to input simulation dependence, and a method of using “mock-data” studies to test their effectiveness, is then described and applied in Sec 4. Finally, the results and the main conclusions of this work are presented in Sec 5.

2. Event filtering at the ICARUS detector

The ICARUS detector [12] is a 760-ton LAr-TPC and is the far detector of the SBN program [2], positioned around 600 m away from the Booster Neutrino Beam (BNB) at FNAL. The detector consists of two identical adjacent modules, each housing two TPCs separated by a common cathode used to generate the electric field to drift the ionisation electrons to the readout anode wires. The prompt scintillation light from charged particles propagating within ICARUS is readout by 360 8 inch PMTs [13] arranged on the walls of the TPCs, placed as shown in Fig. 1. The PMT system provides the means to trigger the readout of signals within the $1.6 \mu\text{s}$ BNB spill windows, while enabling fast spatial localisation of neutrino beam associated events. The placement, performance and timing resolution of the PMTs are expected to allow the localisation of the associated charged particle track with accuracy better than 1 m [13–15]. The signals of all PMTs are continuously readout, where pairs of adjacent PMTs are typically used together within the ICARUS trigger system (the pairing scheme is also shown in Fig. 1). For each beam spill window, the ICARUS trigger system can access which PMTs observe a signal exceeding some predefined threshold, what time that signal was recorded with respect to the opening of the beam window and how many times the PMT recorded an “opening” above the threshold (the PMTs are “closed”, if open, every $0.16 \mu\text{s}$).

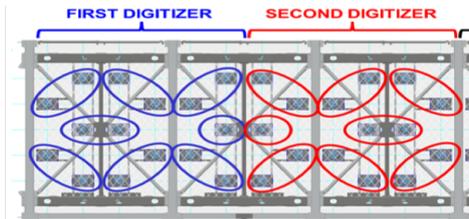


Figure 1: A schematic view of one wall segment of ICARUS’s TPCs showing how the PMTs are distributed. The PMTs are arranged in groups of 15 that are connected to the same digitiser board. The hollow ellipses show the pairing system of adjacent PMTs.

The BNB delivers neutrinos to ICARUS up to five times per second, with a neutrino arrival window at the detector of the order of microseconds. This window, compared to the TPC drift time (order of milliseconds), is very small. During the few millisecond drift time of the ionisation electrons, multiple incident cosmic rays will cross the detector. Hence, a typical event captured in coincidence with the neutrino beam has many cosmic particles reflected in the TPC data. The goal of the event filtering with the PMT information presented here is only to reject events where the signal within the beam window is caused by an incident cosmic ray rather than a neutrino interaction.

Since it is possible to better separate neutrino and cosmic-ray interactions after fully reconstructing the TPC signal, the primary aim of a low-level event filter is to achieve a modest rejection of cosmic rays whilst maintaining almost all neutrino interactions in order not to lose any physics

signal but to reduce the amount of data that must be processed for higher-level analyses. To allow the possibility of *online* event filtering, only the information available to the ICARUS event trigger is used. This means that information from each PMT pair is concatenated where the earliest opening time from each pair is stored alongside the total number of openings across both PMTs in the pair.

2.1 Simulation details

The cosmic-ray particles impinging the ICARUS detector were generated with CORSIKA event generator [16]. These particles are then propagated through the ICARUS detector and the surrounding material using GEANT4 [17] implementation in LArSoft [18]. Scintillation photons are then propagated to the PMTs using a parameterised model based on pre-calculated tables (also derived from GEANT4). A further parameterised PMT readout model, constrained from test data, is then used to simulate the digitised “data-like” signal from the detector.

The incoming flux of neutrinos is modelled using a GEANT4-based simulation of the BNB beamline [2, 19] whilst their interactions with nuclei (and electrons) within ICARUS are described using GENIE version 3 [20]. The particle propagation and detector response are simulated identically to the case of cosmic rays.

For this study, we simulate 396,400 PMT readout windows (*events*) containing cosmic rays and 113,240 containing a single neutrino interaction. Whilst it is possible to have a PMT readout window containing cosmic rays and a neutrino interaction or multiple neutrino interactions, this is not particularly common, and such details are beyond the scope of this study.

3. A CNN-based approach to event filtering using only PMT information

The goal of the CNN is to classify whether events are from neutrino interactions or cosmic rays. To train the CNN, the simulated PMT data is presented as images, where the position of each PMT pair, alongside its opening time and a number of openings, are stored. An example of the images provided as input is shown in Fig. 2. A total of 91,718 neutrino events and 36,134 cosmic-ray events are used for training the CNN, and the remaining 11,558 neutrino and 4,538 cosmic events are used as the test sample. Each image in the training sample is labelled as a cosmic ray or neutrino event.

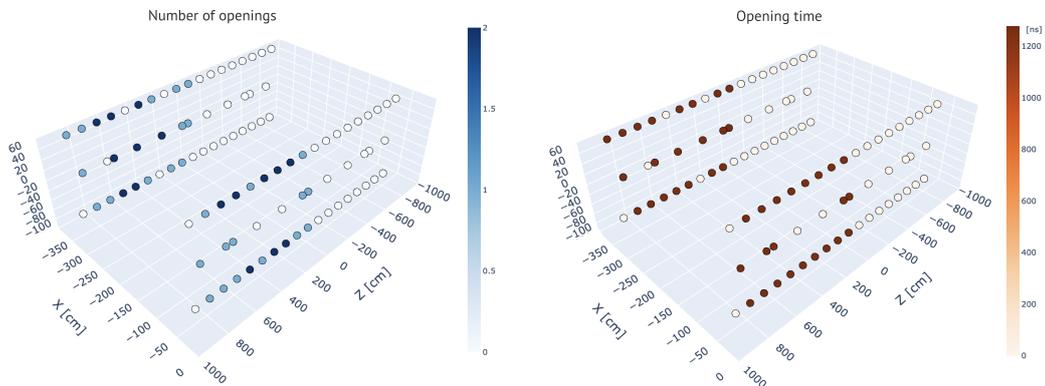


Figure 2: Example images input to the CNN. Each dot represents a PMT pair distributed across the walls of the TPCs. The colour of the dots represents the number of openings (left) or the opening time (right).

The main feature of CNNs is that they learn a series of filters (using convolutions), applied in sequence to extract increasingly powerful and abstract features that allow the CNN to learn a mapping between input images and target labels. Once the CNN is trained, it can be applied to new images to make accurate predictions on unseen examples during the training. The designed CNN architecture is depicted in Fig. 3 and uses Submanifold Sparse Convolutions [21] to deal with the sparsity of neutrino interactions. The CNN is trained using Python 3.6.9 and PyTorch 2.1.0 [22] using the Minkowski Engine version 0.5.4 [23] on NVIDIA Tesla V100 GPUs. Stochastic Gradient Descent (SGD) is used as the optimiser, with a mini-batch size of 32 events, a learning rate of 0.1 (divided by 10 when the error plateaus, as suggested in [24]), a weight decay of 0.0001, and a momentum of 0.9.

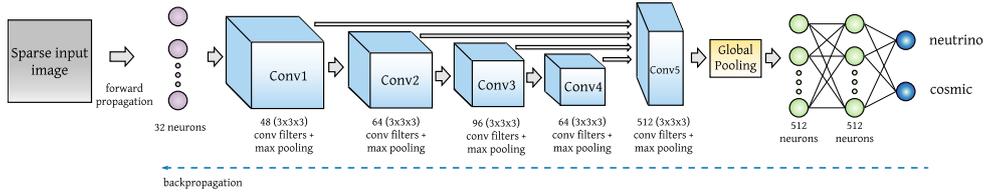


Figure 3: The CNN architecture used for this analysis.

3.1 CNN event filtering performance

Once trained, the output of the CNN is a score for each event between 0 (neutrino-like) and 1 (cosmic-like). The distribution of CNN scores for each true event type in the test sample is shown in Fig. 4 (left). If a selection of neutrino events is made by cutting at a CNN score of 0.5, a 99.6% selection efficiency is maintained whilst 74.2% of cosmic-ray backgrounds are rejected. The charged-current selection efficiency was found to be flat (i.e., unbiased by kinematics) in various tested observables. An example of outgoing lepton angle is shown in Fig. 4 (right).

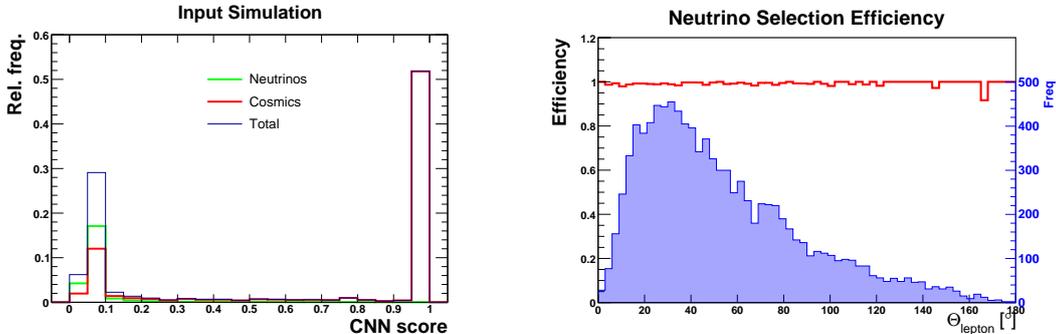


Figure 4: Left: The trained CNN’s classification of events in the test sample. Note that the relative normalisation of cosmic and neutrino events is fixed to approximate what would be expected in data. **Right:** The distribution of outgoing lepton angle (blue) with respect to the incoming neutrino from GENIE is shown alongside the neutrino selection efficiency of the CNN (red) following a cut at a score of 0.5.

4. Reducing model-dependence with DANN-based training

Whilst the CNN presented in Sec. 3.1 shows excellent performance, the results assume perfect modelling of the neutrino and cosmic-ray events, the particle propagation and the detector response.

If the CNN is trained with events that do not suitably represent what is in the real data, then the performance from the test sample will not be reliable. Machine learning methods can be especially susceptible to strong potential simulation dependent performances as it is unclear exactly which features are used for classification. Thus it is difficult ever to be sure that the pertinent aspects of the events are well modelled. To alleviate this issue, adversarial training methods can be employed to prevent neural networks from learning features that appear in simulations but not in the data. In this analysis, we show that it is possible to mitigate simulation dependence through the application of DANNs.

In DANNs, the neural network model is trained on examples from two domains: (a) the *source* domain, which consists of labelled simulated data; and (b) the *target* domain, which consists of unlabelled true experimental data. The goal is to learn a discriminator from the labelled source domain examples and use the unlabelled target domain examples to ensure the discriminator relies on only domain-invariant features to perform the predictions. Regarding the implementation of the neural network, the classifier architecture remains identical, and it can be seen as the combination of a feature extractor (i.e., the bulk of the CNN, in our case) and a label predictor (i.e., the fully connected layer(s) at the end). However, this alternative neural network has a second path, which connects the output of the feature extractor through a gradient reversal layer with a few fully connected layers that form a domain classifier. The gradient reversal layer performs an identity transformation during the forward propagation process but multiplies the gradient by a negative constant during the back-propagation, guarantying that the parameters learnt by the feature extractor are made similar for the source and target distributions. In other words, with this approach, the features learnt by this model are both discriminative - thanks to the label predictor - and domain-invariant - thanks to the domain classifier - at the same time. This behaviour is shown in Fig. 5.

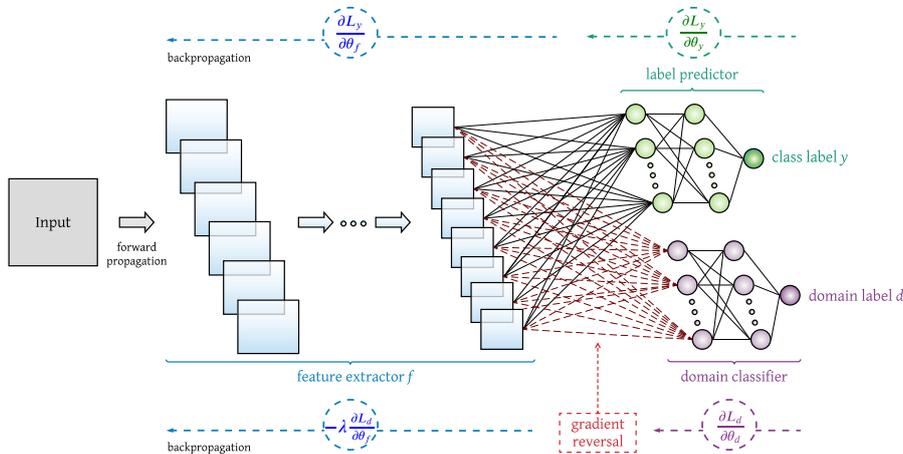


Figure 5: Domain-adversarial neural network architecture. The feature extractor (blue) and the label predictor (green) form a standard neural network classifier. The domain classifier (purple) provides the domain adaptation part since it is connected to the feature extractor through a gradient reversal layer, allowing the alignment of feature distributions across the source and target domains. Figure adapted from [11].

In order to test the effectiveness of DANNs as a method of reducing simulation dependence, we perform a series of “mock-data” studies. For these studies, a statistically independent simulation of events (from neutrinos and cosmic rays) is produced before being modified to simulate possible

mis-modelling bias. Since the coarse PMT information used in this analysis is likely not sensitive to the exact details of the neutrino interaction or cosmic ray production, we focus primarily on applying distortions to the simulated detector response. The details of the mock data are described below.

“Global noise” data: in this mock data noise, which is uncorrelated with the event content, is randomly added to each PMT with some pre-specified (“global”) probability that is common to all PMTs. The global noise probabilities considered are 2%, 5% and 10%. The timing of the noise is considered as uniform distribution, and if noise is simulated to arrive before a PMT is opened by a simulated signal, the opening time of the signal is overwritten by that of the noise.

“Local noise” data: similarly to the global noise data, this mock data set considers the addition of random noise to each PMT but where the probability of producing noise is different for every PMT. Noise probabilities per event for each PMT were generated randomly using a uniform distribution between 0 and either 2%, 5% or 10%.

For each mock data study, the DANN is trained as described in Sec. 3 but with an addition of 9,109 cosmic-ray mock data events and 37,724 neutrino mock data events, which are only labelled by the domain (i.e., mock data or MC) and not by event type (i.e., cosmic or neutrino). This method could equally be applied to real data.

Both the originally trained CNN (as described in Sec. 3) and the newly trained DANN are used to attempt to classify events from the original test sample and from a statistically independent set of events from the corresponding mock data sample. An example of the classification scores for each model applied to the original and mock data set is shown for two mock data studies in Fig. 6. A summary of the neutrino selection efficiency and the background rejection performance for each mock data set is shown in Tab. 1.

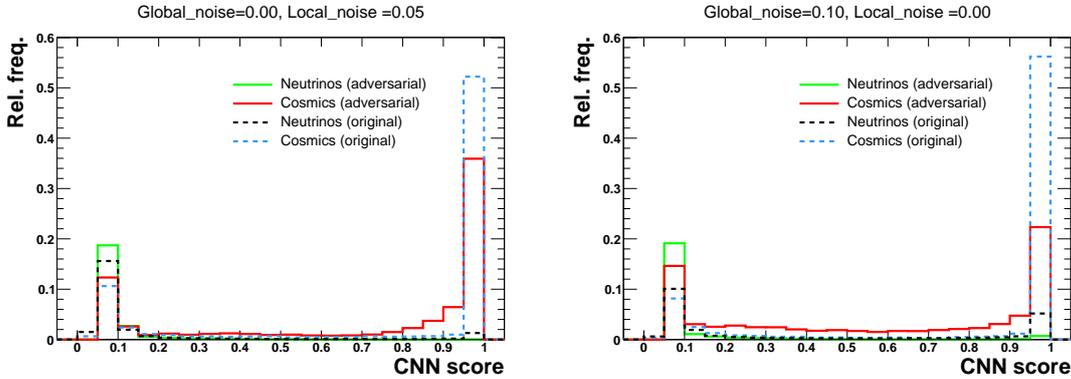


Figure 6: Results of the classification score of the nominal and adversarially trained CNN/DANN model applied to the original and: local noise mock data using a 5% noise spread (*left*) and the global noise mock data using a 10% noise spread (*right*).

From these results it can easily be seen that, without the adversarial training, the original CNN can reject a sizable portion of neutrino interactions in the mock data. However, once the adversarial training is used the network is able to mitigate the bias and maintain a very high neutrino selection efficiency (the main goal of the filter) whilst continuing to achieve a significant rejection of cosmic-ray backgrounds.

Dataset	Model type	\mathcal{E}_ν	\mathcal{R}_{cos}
Nominal	Original	99.26%	74.22%
	Adversarial	91.21%	75.52%
Global noise	2% Original	97.22%	74.42%
	2% Adversarial	80.29%	76.47%
	5% Original	98.17%	66.33%
	5% Adversarial	65.37%	79.47%
Local noise	10% Original	94.30%	56.20%
	10% Adversarial	95.11%	75.61%
	2% Original	98.41%	71.66%
	2% Adversarial	89.74%	75.37%
Global noise	5% Original	98.55%	70.82%
	5% Adversarial	81.14%	76.17%
	10% Original	97.21%	73.58%
	10% Adversarial		

Table 1: Efficiency (\mathcal{E}_ν) and proportion of rejected cosmic-ray background events (\mathcal{R}_{cos}) using the original and adversarially trained CNN/DANN to classify events in the nominal simulations and mock data studies.

5. Conclusion

The results in Sec. 3.1 and 4 demonstrate that using only low-level PMT information, effective filtering out of most cosmic ray events can be effectively achieved whilst almost no neutrino events are rejected. It is shown that mis-modelling in the input simulation can cause a trained CNN to reject neutrino events inadvertently. However, adversarial training via a DANN can mitigate the loss of efficiency at the cost of some reduced background rejection.

- [1] R. Acciarri et al. (ArgoNeuT), Phys. Rev. D **95**, 072005 (2017), [1610.04102](#).
- [2] M. Antonello et al. (MicroBooNE, LAr1-ND, ICARUS-WA104) (2015), [1503.01520](#).
- [3] B. Abi et al. (DUNE), JINST **15**, T08008 (2020), [2002.02967](#).
- [4] Y. LeCun et al., Neural Computation **1**, 541 (1989), ISSN 0899-7667.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Proceedings of the IEEE **86**, 2278 (1998).
- [6] A. Aurisano et al., JINST **11**, P09001 (2016), [1604.01444](#).
- [7] B. Abi et al. (DUNE), Phys. Rev. D **102**, 092003 (2020), [2006.15052](#).
- [8] C. Adams et al. (MicroBooNE), Phys. Rev. D **99**, 092001 (2019), [1808.07269](#).
- [9] S. Ben-David et al., Machine learning **79**, 151 (2010).
- [10] I. Redko et al., *Advances in domain adaptation theory* (Elsevier, 2019).
- [11] Y. Ganin et al., J. Mach. Learn. Res. **17**, 2096–2030 (2016), ISSN 1532-4435.
- [12] S. Amerio et al. (ICARUS), Nucl. Instrum. Meth. A **527**, 329 (2004).
- [13] M. Babicz et al. (ICARUS/NP01), JINST **13**, P10030 (2018), [1807.08577](#).
- [14] B. Ali-Mohammadzadeh et al. (ICARUS), JINST **15**, T10007 (2020), [2006.05261](#).
- [15] M. Babicz et al., Nucl. Instrum. Meth. A **958**, 162421 (2020).
- [16] D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw (1998).
- [17] S. Agostinelli et al. (GEANT4), Nucl. Instrum. Meth. A **506**, 250 (2003).
- [18] E. L. Snider and G. Petrillo, J. Phys. Conf. Ser. **898**, 042057 (2017).
- [19] A. A. Aguilar-Arevalo et al. (MiniBooNE), Phys. Rev. D **79**, 072002 (2009), [0806.1449](#).
- [20] C. Andreopoulos et al., Nucl. Instrum. Meth. A **614**, 87 (2010), [0905.2517](#).
- [21] B. Graham and L. van der Maaten (2017), [1706.01307](#).
- [22] M. Abadi et al., in *OSDI* (2016), vol. 16, pp. 265–283, [1605.08695](#).
- [23] C. Choy, J. Gwak, and S. Savarese (2019), [1904.08755](#).
- [24] K. He, X. Zhang, S. Ren, and J. Sun, CoRR (2015), [1512.03385](#).