Referee Report for:
"How to GAN Event Subtraction"

The authors of the paper present a proposal to use a Generative Adversarial Networks (GAN) for even subtraction and addition. They compare this with bin by bin subtraction and claim that with a GAN they can reduce the additive quadratic error that is introduced in a bin by bin subtraction method. To show this, the authors give two simple toy examples in 1D space and two real examples, one of a background subtraction and the other of a collinear subtraction. My observations are as follows.

Specific Remark:

1. I do not think a GAN is necessary for such a simple task. GANs are very powerful neural nets that are required for generative tasks that would otherwise require a very large training time for other generative networks. I will not go into further details since its common knowledge and the details can be found in Goodfellow et. al., Generative Adversarial Networks and subsequent works.

   This task can be easily performed with much simpler and faster machine learning algorithms like decision trees or SVM or with a deep neural network. Figure 1 has been produced with two discriminative deep neural networks with an architecture of 1-5-5-1 with linear-sigmoid-sigmoid-linear activation and mse loss optimized with a RMSprop optimizer with a learning rate of 0.003. An early callback has been used as a regularizer which usually caps the number of epochs to less than 1000. It has been implemented with TensorFlow. The networks take less than 5 minutes to train on a quadcore cpu and give errors comparable to Figure 2 of the paper. The training was done with 1000 simulation points for each network. One can question the behaviour at the lower tail, but it should disappear if one uses a larger simulation set as the authors did. The DNN can be generalized to arbitrary input dimensions as is well known.

While the work is pertinent, it can be done with very simple regression tools and requires

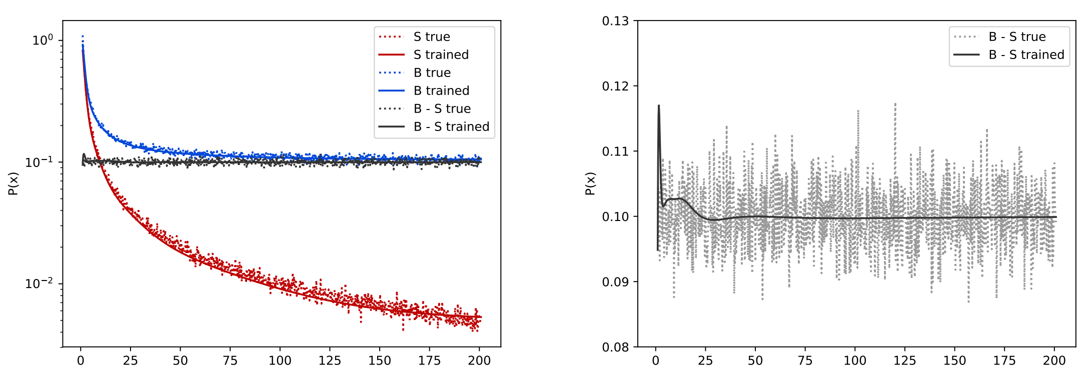

Figure 1: Event subtraction with DNN

minimal coding and tuning of hyperparameters. It can only be accepted as significant work if it is part of a bigger physics program. For this reason I cannot recommend the paper for publication. In case the authors consider submitting it to another journal, I have a few comments below.

General Remarks:

1. I do not think ref. 11 uses a generative network. They use a DNN and show that they perform better than ref. 12 which uses a GAN. The authors can maybe take a deeper look into these papers.

2. The authors do not provide the code that they use or any details about it or what framework they used (PyTorch/Sci-Kit Learn/ TensorFlow etc.). The authors also do not provide the data they used for the training. While this is not necessary, it is useful to have it if someone wants to reproduce their results. I would suggest the authors provide all these details (possibly in a public repository) and also an example code since the work is primarily computational.

3. The authors do not make explicit the training times and the hardware used for training the GANs. This is useful to benchmark it against other regression methods.

4. The authors do not describe how they get the error-bars in the left panels of fig.2, fig. 3, etc. Are they from eq. 13?