Deep learning for passive source detection in presence of complex cargo.

W. Baines$^1$, P. Kuchment$^2$, and J. Ragusa$^3$

$^1$ Mathematics Department, Texas A&M University, College Station, TX, USA
$^2$ Mathematics Department, Texas A&M University, College Station, TX, USA
$^3$ Nuclear Engineering Department, Texas A&M University, College Station, TX, USA

Abstract. Methods for source detection in high noise environments are crucial for single-photon emission computed tomography (SPECT) medical imaging and especially for homeland security applications, which is our main interest. In the latter case, one deals with detecting the presence of low emission nuclear sources with significant background noise (with Signal To Noise Ratio ($SNR$) 1% or less). Direction sensitive detectors are needed to achieve this goal. Collimation, used for that purpose in standard $\gamma$-cameras, is not an option. Instead, Compton cameras can be utilized. Backprojection methods suggested before enable detection in the presence of a random uniform background. In most practical applications, however, cargo packing in shipping containers and trucks creates regions of strong absorption, while leaving streaming gaps open. In such cases the background will not be uniform, which renders backprojection methods ineffective. A deep neural network is implemented for the source detection in 2D, which exhibits higher sensitivity and specificity than the backprojection techniques in a low scattering case and works well when presence of cargo makes the latter fail.

Submitted to: *Inverse Problems*
1. Introduction

Checking for presence of illicit nuclear materials (most probably in small quantities and shielded) at border crossings is an important homeland security task. Ideally, one would try to reconstruct from the detected signals the source distribution function inside the cargo. When the data is sufficiently well behaved (e.g., in SPECT), analytic reconstruction is possible [26]. However, in a very low SNR environment, as in the case of illicit nuclear source detection, this is impossible. The saving grace is that in this case one is mostly interested in the presence of a source, rather than its exact location.

In order to have any hope of detecting the small fluctuations in background noise produced by the presence of a small signal, direction sensitive detectors are necessary. Indeed, otherwise the data measured is two-dimensional, which is insufficient for recovery of three-dimensional information. The following options for obtaining directional sensitivity are available:

- **Mechanical collimation**, when only rays incident along (or close to) a certain line are allowed to reach the detector (see section 2). This, while determining the incoming photon’s direction, significantly reduces the signal strength and thus becomes unsuitable for low SNR.

- **Compton type cameras** are a novel type of gamma detectors that determine a surface cone of possible incident trajectories, rather than their exact directions.

Backprojection techniques introduced in [5,29] for detecting not only the presence, but even the location of the source do reasonably well in the absence of complex cargo. They utilize the existence of a certain quantity of ballistic (non-scattered) particles from the source reaching the detector. However, they start failing if such cargo is present [6]. However, there seemed to be a reason to believe (see [6]) that the data may still contain a signature of the source presence. Indeed, when the method of [5] was applied to some cases of complex cargo in [6], despite its failure to locate the source, such signatures (e.g., the pathways between cargo boxes highlighted differently) seemed to appear only when a source was present (See Fig. 1 below). However, no detection algorithm was found for such cases. Fortunately, one is mostly interested in the presence of a source, rather than its exact location, so we may safely eschew the complete imaging task. This motivated us to apply deep learning for the source inference.

Deploying neural networks is an inherently data driven approach. Its success is predicated upon our access to sufficient data for neural network training. We address the process we used for simulating the data in section 6, the design of the network in section 7, and present the results in section 8. Additional remarks can be found in section 9. Acknowledgements are provided in section 10. Some auxiliary tables are located in the Appendix.

‡ Albeit based on different physics principles, neutron detectors that produce similar information are being developed. We thus will not emphasize the type of particles detected.
Deep learning for source detection

Figure 1. (Left): Example of complex cargo configuration for which backprojection methods fail. The red spot denotes the source location, the grey area represents iron and the white area represents air. (Middle): Backprojection results in absence of source. (Right): Backprojection results in presence of source.

2. Collimated $\gamma$-Cameras

Mechanical collimators (see Fig. 2) can be installed in front of a direction insensitive $\gamma$-camera to block all particles but those incident along (or close to) a desired trajectory. Mechanical collimators are widely used in medical imaging. They, however, significantly attenuate the signal and require rotating the detector (or the object). In the applications with sufficiently high $SNR$, this additional data loss is not such a problem. In dealing with low $SNR$ signals however, it renders recovery of weak signals impossible. For this reason one can consider Compton type cameras instead.

3. Compton Type Cameras

The Compton camera is a rather recently introduced type of $\gamma$-particle detector that does not attenuate the incident particles, but provides less detailed direction information.
surface cone of possible incoming directions is measured rather than a precise trajectory. In the absence of mechanical collimation, signal strength is effectively maintained, although the directional information is less precise and thus data analysis becomes more complex. On the other hand, the data provided is significantly over-determined (five-dimensional space of cones versus the unknown distribution being three-dimensional). This turns out not a bad thing at all, but rather a blessing (see [26]).

As we have mentioned before, novel neutron detectors (albeit based upon a different physics rather than Compton scattering) that provide mostly similar cone information are currently being developed.

4. Inversion and Backprojection Techniques

We define here the notion of the Compton (or cone) transform arising from employing Compton cameras and describe briefly some of its properties (see [26] for details and further references).

To be precise, data measured by Compton cameras is the Compton transform of the unknown source distribution function \( F(x) \), as described below.

**Definition 1.** Given a set \( M \subset \mathbb{R}^n \) of Compton detection sites and a compactly supported function \( f \), the **Compton Transform** \( Cf \) of \( f \) is defined as follows:

\[
Cf(\vec{u}, \vec{\beta}, \psi) := \int \mathcal{C}(\vec{u}, \vec{\beta}, \psi) f(x) dS(x) \tag{1}
\]

where \( \mathcal{C}(\vec{u}, \vec{\beta}, \psi) \) is the surface cone with vertex \( \vec{u} \in M \), central axis \( \vec{\beta} \in \mathbb{S}^{n-1} \) and (half-)opening angle \( \psi \in [0, \pi] \) and \( dS(x) \) is the surface measure on the cone.

![Figure 3. Surface cone produced by Compton camera from particle detection](image)

Using a Compton camera, one samples the Compton Transform of the emission sources distribution inside the cargo container. Now the question arises of extracting information about possible presence of a internal source.
4.1. Inversion and backprojection

A variety of exact inversion formulas of filtered-backprojection and other types have been developed and implemented (see [26] and references therein). The choices are much more diverse than for the usual Radon transform inversions (see [21]). The reason is that the Compton data is highly overdetermined. It was shown that this feature can be used to get high quality of reconstructions in SPECT in presence of 50% noise and higher. However, this is a far cry from the low SNRs encountered in the homeland security problems described above.

In [5, 6, 29], a backprojection technique was used, which employed the assumption that the possible source would be geometrically small and existence of a sufficient number of ballistic (i.e., unscattered) particles from the source to the detector. The idea is that when backprojected, the ballistic trajectories would pass through the same small geometric region, where the source is located, and this accumulation might be statistically significant to detect the source and its locations.

4.2. Cargo Difficulties

This technique worked extremely well when a uniform random background noise is assumed. However, in the presence of scattering and absorbing cargo, one cannot rely upon ballistic particles, which might be none at all. Hence, as is shown in [6], the technique (as well as its crude mathematical justification there) fails.

We therefore switch to the deep learning approach, which should naturally start with producing a large quantity of training and testing data.

5. Simulating Cargo Scenarios

In order to obtain rich training data for a neural network, at least thousands (better hundreds of thousands or millions) cargo configurations are needed. The more training data can be obtained, the better. Due to the high computation costs, we were restricted to several thousands of samples. However, our results (see section 8) already show a success in detection.

To start, we randomly produce several thousand cargo configurations and compute forward radiation data simulations for each one.

5.1. Procedural Generation of Cargo Configurations

A square cargo hold of size of $2.4m \times 2.4m$ is assumed and partitioned into $2.4cm \times 2.4cm$ cells (the possible source would occupy one of them). Each cell can be indexed via a pair of row and column indices, $(i, j)$, with $1 \leq i, j \leq 100$ and is assigned a material identification number $ID_{i,j}$. These numbers correspond to a variety of materials according to the Table 1 below.
Table 1. Material Identifications

| ID | Material                | Molecular Composition                                                                 |
|----|------------------------|---------------------------------------------------------------------------------------|
| 1  | Air                    | N (78.1%), O (20.9%), Ar (0.94%), H (0.06%)                                         |
| 2  | Concrete               | H(0.5%), O (48.7%), Na (1.7%), Mg (2.5%), Al (4.5%), Si (30.8%), K (1.9%), Ca (8.1%), Fe (1.2%), traces of Th-232 and U-238 and their daughters |
| 3  | Highly Enriched Uranium| Uranium (100%)                                                                         |
| 4  | Iron                   | Fe (100%)                                                                             |
| 5  | Cotton                 | O (25%), C (25%), H (50%)                                                             |
| 6  | Wood                   | O (25%), C (25%), H (50%)                                                             |
| 7  | Plastic                | C (33.3%), H (66.7%)                                                                  |
| 8  | Fertilizer             | K (13.9%), N (46.8%), O (30%), P (9.3%)                                               |

Real cargo typically consists of several large distinct boxes with small spaces in between. In order to emulate this, an algorithm is implemented to generate different cargo configurations, which consists of three main steps:

- A network of several horizontal and vertical “corridors” between boxes with random widths and locations is generated. The number of corridors $c$ is selected randomly in a desired range $c_{\text{min}} \leq c \leq c_{\text{max}}$.
- The resulting configurations are unlikely to be symmetric, while real cargo might happen to be symmetric. To check whether symmetry plays any role in detectability, a portion of the samples produced are “symmetrized” by enforcing various (rotation and mirror) symmetry rules.
- Connected components of the rest of the space are identified as distinct “cargo boxes.” Then material contents are assigned to all boxes. In a subset of (rather than all) symmetric cargo configurations, material contents are also “symmetrized” according to the corresponding rule.

Generating the corridors between pieces of cargo is performed using a modification of the procedure outlined in [9] for generating road networks. Our modification is restricted to networks consisting of horizontal and vertical segments.

Remark. Instead of selecting corridor locations uniformly randomly, road locations are selected according to a probability distribution generated from a type of gradient noise developed by Ken Perlin in 1983 [22] in order to automate the production of realistic looking textures in computer graphics. Perlin noise $P(x)$ can be generated for a square domain such as $[0, 1]^d$ as follows:

- fix a regularly spaced lattice of points \( \{x_k\}_{k=1}^n \in [0, 1]^d \),
- assign a random unit vector $u_k$ to each grid point $x_k$ in the lattice,
• for each point $x \in [0, 1]^d$ compute the dot product $u_k \cdot (x - x_j)$, where $x_j$ is one of the $2^d$ grid points closest to $x$,
• compute $P(x)$ by interpolating between the $2^d$ dot products corresponding to $x$ and normalizing to a desired range.

This process is shown in Fig. 4.

![Perlin noise generation procedure. (Left): Randomly generated unit vectors at grid points. (Middle): Dot products with nearest grid points. (Right): Interpolation of dot products.](https://creativecommons.org/licenses/by-sa/4.0/deed.en)

Identification of connected components ("boxes") is performed using SciPy’s (Scientific Python, a popular Python package for scientific computing [14]) implementation of the algorithms outlined in [28].

A variety of symmetry rules, including rotational symmetry and mirror symmetry are applied randomly to the resulting configurations. Materials are then assigned randomly to the boxes. A subset of the symmetric configurations have material assigned according to the associated symmetry rule. The entire generation procedure is summarized in Algorithm 1 in the Appendix (See section 11).

**Remark.** For this work, we randomly select, prior to cargo generation, which symmetry rules to apply. Each cargo symmetry rule has equal selection probability, and each material symmetry rule has equal selection probability.

### 6. Forward radiation simulations

To simulate training and testing data, a massive radiation transport computation is required

#### 6.1. Physics Preliminary

We are interested in detecting presence (or absence) of a shielded Uranium source against a significant background noise. Uranium-238 (U-238) photons from the 1.001 MeV emission line have mean-free-path in high-Z materials sufficiently high to be detected outside the container (13.3mm mean-free-paths) [25]. In our application, sources of background radiation include a concrete base located some distance below the container §. These background

---

§ Cosmic rays and other natural sources can be easily included.
sources radiate at much higher energies than 1.001 MeV, including 1.461 MeV from Potassium-40, 1.12 MeV and 1.76 MeV from Bismuth-214, and 2.61 MeV from Thallium-208 (Bismuth and Thallium are products of the decay of Uranium-238 and Thorium 232 respectively, and are present in trace amounts in concrete). Gamma photons which downscatter from these sources into the energy group surrounding the 1.001 MeV line account for the noise in our signal. Gamma photons from the source will also undergo scattering and absorption within the volume of the container. Absorption and scattering within the container will reduce the number of ballistic source particles reaching the detectors placed around the container, thus weakening the signal.

6.2. Mathematics of the forward radiation data simulation

We describe briefly the mathematical model and numerical scheme employed.

The radiation transport within the cargo container is modeled by the linear Boltzmann equation, given below using the multigroup approximation:

$$\vec{\Omega} \cdot \vec{\nabla} + \Sigma^g_t(\vec{r})\Psi^g(\vec{r}, \vec{\Omega}) = \sum_{g'=1}^G \sum_{l=0}^L \sum_{s,l} \Sigma^g_{s,l} \Phi^{g'}_{l,m}(\vec{r}) + Q^g(\vec{r}, \vec{\Omega})$$

where $\vec{r} \in \mathcal{D}$ is the position, $\vec{\Omega} \in S^2$ the set of discrete directions and $g \in [1,G]$ the energy group. $\mathcal{D}$ is the volume of the cargo container, $S^2$ is the unit sphere, $G$ is the total number of energy groups, $\Psi^g$ is the photon angular flux in the energy group $g$, $\Sigma^g_t$ is the total interaction cross section in group $g$, $\Sigma^g_{s,l}$ is the $l^{th}$-Legendre moment of the scattering cross section from group $g'$ to group $g$, $L$ is the maximum anisotropy expansion order, and $Q^g$ is the volumetric source of photons in group $g$ (stemming from the U-238 source). The moments of the angular flux are given by

$$\Phi^g_{l,m}(\vec{r}) = \int_{4\pi} Y_{l,m}(\vec{\Omega}) \Psi^g(\vec{r}, \vec{\Omega}) d\Omega$$
Deep learning for source detection

where $Y_{l,m}$ is the spherical harmonic of order of $l$ and degree $m$. Eq. (2) is supplied with boundary conditions:

$$\Psi^q(\vec{r}, \vec{\Omega}) = h^q(\vec{r}, \vec{\Omega}) \quad \forall \vec{r} \in \partial D^-$$  \hspace{1cm} (4)

where $\partial D^-$ is the incoming boundary defined as $\partial D^- = \{\vec{r} \in \partial D \text{ such that } \vec{\Omega} \cdot \vec{n}(\vec{r}) < 0\}$ with $\vec{n}(\vec{r})$ the outward unit normal vector at position $\vec{r}$. The function $h^q$ describes the background radiation due to a large concrete slab underneath the container, as previously described. Cross sections for various materials were generated using NJOY-99 [20]. The materials’ compositions are given in Table 1. The multigroup structure employed ranges from 1.00099 MeV to 2.61449 MeV with narrow bands centered at the radiation lines of the background and U-238. See Table 6 in the Appendix for details (Section 11).

For the purposes of this paper, calculations are carried out in two-dimensional space and only the energy group corresponding to the 1.001 MeV line is considered after solving Eq. (2). The photon transport equation, Eq. (2), is discretized using standard techniques:

(i) $S_n$ product Gauss-Legendre-Chebychev angular quadrature [24] is employed (only a small number of polar angles are needed, but a very high number of azimuthal angles are needed to resolve properly the angular distribution in the 2D domain.)

(ii) Spatial discretization based on a standard bilinear discontinuous finite element technique with upwinding at cell interfaces. [23, 27]

(iii) Transport sweeps and Source Iteration are employed to solve the resulting system. [18]

Once the transport equation, Eq. (2), has been solved, the outgoing angular photon flux at any boundary edge in 2D is recorded for use in Deep Learning and Backprojection.

Once configurations have been generated, a radiating source emitting an expected 8042.17 photons per second at 1.001 MeV is randomly placed, a forward radiative transfer equation is solved, and from its solution the radiation angular flux distribution on the boundary of the cargo is collected.

7. Convolutional Neural Network

In order to solve the source detection problem, we construct a deep convolutional neural network (CNN). Convolutional networks (which are much cheaper to compute) were chosen due to the mostly locally correlated features in the data set. This is the reason why CNNs find ubiquitous and effective use in a wide range of image classification and signal analysis problems [19]. They also serve as a suitable candidate architecture for the source identification problem. The suggested CNN architecture is summarized in Fig. 6 below. The network is trained on 2100 simulated cargo configurations with randomly placed source. The output is a probability measure $P$ on $\{0, 1\}$. A source is determined to be present if $P(x = 1) > 0.5$, and absent otherwise. The loss function used for training is the binary cross-entropy loss:

$$\mathcal{L}(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$ \hspace{1cm} (5)
Where \( y \) is that network prediction and \( \hat{y} \) is the target value (See [12]). Two networks were trained on simulations of a localized source in the presence of high uniform background noise (\( SNR = 0.01 \)) corresponding to high exposure times (In other words, sufficiently high particle counts). In all cases, early stopping is used to halt training before over-fitting. The various hyper-parameter values used in training are summarized in Table 2 below. The CNN is implemented using Keras with Tensorflow as its backend. Keras is a high level API for interfacing with machine learning toolkits such as Tensorflow, Theano, and Microsoft Cognitive Toolkit. It helps streamline the construction and training of neural networks [10]. Tensorflow is Google’s machine learning toolkit and was chosen due to its scalability, wide range of features, and the wide range of documentation and tutorials available [1]. Any parameters not explicitly mentioned here were set to default values.

![Figure 6](image)

**Figure 6.** CNN architecture used for source detection. The left-most cell shows an example of the detector data input to the CNN. 2×2 Max pooling layers are placed after every second convolutional layer.

| Optimization Method | Adam (See [15]) |
|---------------------|----------------|
| Activation          | RELU (Softmax at output) |
| Bias                | True |
| Convolution Window Size | 3x3 |
| Learning Rate       | \( 2.0 \times 10^{-5} \) |
| Learning Rate Decay Rate | 0 |
| Batch Size          | 4 |
| Early Stopping Patience | 3 epochs |
| Loss                | Binary Cross-Entropy |

**Table 2.** Hyper-parameters used during training
8. Results

In 500 randomly chosen cargo configurations a source with 1% SNR was randomly placed and particle detections are simulated for an exposure time corresponding to expected background detection levels of 100000 particles and 10000 particles. The data were fed into the trained CNN for source presence prediction. The results obtained, which clearly confirm our expectations (see Section 4.2), are summarized in Table 3 below. While at the level of the total of 10000 particles the detectability fails, for 100000 particles CNN succeeds extremely well and beats hands down the backprojection technique. We remind the reader that sensitivity, or true positive rate, shows the success of determining the presence of a source (i.e., few false negatives), while specificity reflects how well the absence of the source is detected (i.e., few false positives).

The perfect specificity of the CNN at the low particle count should not be surprising. Indeed, the presence of very few sources was detected, and there were nearly no false positives.

In order to see whether there is any influence of presence or absence of a symmetry, the results were classified in terms of the symmetry type:

| Symmetry Type (s)                  | Sensitivity | Specificity | Count |
|------------------------------------|-------------|-------------|-------|
| Total (symmetric and not)          | 0.954       | 0.988       | 500   |
| Asymmetric                         | 0.959       | 1.00        | 145   |
| Rotational Arrangement             | 0.984       | 0.984       | 64    |
| Rotational Material                | 0.922       | 1.00        | 51    |
| X-Mirrored Arrangement             | 0.925       | 1.00        | 67    |
| X-Mirrored Material                | 0.944       | 0.982       | 54    |
| Y-Mirrored Arrangement             | 0.970       | 0.985       | 66    |
| Y-Mirrored Material                | 0.962       | 0.981       | 53    |

Table 4. Sensitivity and specificity CNN for 10000 particles and several different symmetry types

The level of around 600000 particles was required in [5] and [29] for stable detection, even without complex cargo being involved. High specificity was hardwired into the BP techniques [5, 6], it was only the sensitivity that was questionable.
The accuracy of the prediction generally increases with particle count (and thus observation time), and sufficient particle counts are required for successful detection.

A second network was trained with several heavy iron configurations to try to capture more difficult scenarios. An additional 2000 configurations were added to the training set (for a total of 4600 training samples) and the network retrained. It was then tested on 160 heavy iron configurations and 500 “normal” ones. Additionally, several particle count levels were tested to determine how performance varies with decreased exposure time. The results are summarized in Table 5 below.

| Expected Particle Count | Sensitivity | Specificity |
|-------------------------|-------------|-------------|
| 100000                  | 0.947       | 0.975       |
| 90000                   | 0.945       | 0.967       |
| 80000                   | 0.941       | 0.979       |
| 70000                   | 0.924       | 0.967       |
| 60000                   | 0.920       | 0.948       |
| 50000                   | 0.924       | 0.911       |
| 40000                   | 0.927       | 0.839       |
| 30000                   | 0.964       | 0.577       |
| 20000                   | 0.998       | 0.0742      |
| 10000                   | 1.000       | 0.000       |

Table 5. Sensitivity and specificity of retrained network.

Backprojection results are omitted in Table 5 since the high absorption of iron causes the uniformity assumptions of the backprojection algorithm to be violated to such a large extent that it is no longer feasible. At the ten thousand particle count level the neural network always indicates the presence of a source, which leads to perfect sensitivity with no specificity. This renders the result unusable.

8.1. Example Scenarios

The flux measured at the boundary is a function of the placement of materials in the cargo, as well as the source strength and position, and the background level. Accordingly, to reach the number of particle measurements necessary for source detection shown in Tables 3 and 5, different exposure times are required for different cargo. These exposure times can vary from times on the order of a fraction of a second, when mostly low scattering or absorbing materials are present, to as high as a week and beyond when a high proportion of the container’s volume is filled with iron. In the following subsections, a selection of example scenarios are presented to give an idea of what exposure times to expect in different scenarios.

8.1.1. Example #1
This configuration as well as backprojection with source present is shown in Figure 7 below. It is evident that the corridor highlighting phenomenon observed in [6] is exhibited in the backprojection for this configuration, while the backprojection algorithm still fails to detect the source. The network, on the other hand, succeeds in detecting the source. This is one of the heavy iron configuration from the testing set which has a shorter exposure time (18 seconds for 101,180 background particles).

![Figure 7. Left: Backprojection with source present. Right: Cargo configuration with source location indicated by arrow. 101,180 particles detected, 100,185 background particles and 995 source particles. Exposure time is 18 seconds.](image)

8.1.2. Example #2

Next we consider the scenario in Figure 8 where backprojection fails to detect the source but the network succeeds. Here the exposure time is significantly longer. In this configuration a long thick iron slab effectively blocks one side of the detectors. Smaller chunks of iron spread throughout the container further attenuate the signal along certain trajectories. As a result, it takes 9 hours and 26 minutes to detect 101,092 particles.

8.1.3. Example #3

Here we consider perhaps a more tenable scenario in Figure 9 where backprojection fails to detect the source, yet the network succeeds. In this case the exposure time is 50 minutes and 17 seconds for 100,866 particles. In this configuration several large blocks of iron are periodically tiled in the container, with the source located within one of the blocks.

8.1.4. Example #4

Now we consider a somewhat extreme scenario where both approaches succeed in detecting the source. In this case the exposure time is 3 days and 12 hours for 101,272 particles. In this configuration one very large block of iron in the center of the container
surrounds the source. The source is still localized relatively well by backprojection for this scenario, which suggests it is one of the borderline cases where backprojection may still work with proper choice of parameters. Most of the cargo is filled with a homogeneous material, which explains why backprojection did not fail. The configuration can be seen in Figure 10 below.

8.1.5. Example #5

Finally, we consider a rather easy scenario where both backprojection and the network succeed. In this case the exposure time is 276 milliseconds for 100,898 particles. In this configuration several small blocks of different materials are spread throughout the container.
15

Figure 10. Left: Backprojection with source present. Right: Cargo configuration with source location indicated by arrow. 101,272 particles detected, 100,328 background particles and 944 source particles. Exposure time is 3 days and 12 hours.

Only an insignificant amount of particles are scattered, so backprojection recovers the source distribution extremely well. The configuration can be seen in Figure 11 below.

Figure 11. Left: Backprojection with source present. Right: Cargo configuration with source configuration indicated by arrow. 100,898 particles detected, 99,911 background particles and 987 source particles. Exposure time is 276 milliseconds.

9. Remarks and Conclusions

- Our work shows that the deep learning approach significantly improves over detection by backprojection techniques of [6, 26, 29] and works for complex attenuating and scattering cargo scenarios, where the latter fails completely. This confirms the opinion expressed in [6] that some information about source presence was there. However, there are various
further improvements that one should attempt, some of which are addressed below.

- Producing many more training data is a serious stumbling block in 2D, and especially in 3D case.
- The CNN architecture should be improved for reaching shorter observation time and even lower SNR levels.
- We are working on moving to the more realistic 3D situation. The significant difference here is, first, the much higher dimensionality of the data (5D) and corresponding much more massive computations that are needed. Second, in 3D, unlike 2D, the Compton data differ significantly from the usual Radon ones. In particular, an issue arises even when entering the data in a way that makes use of CNNs plausible.
- The neural network approach should be tested on real data, which the authors currently do not have. However, the radiative transport forward computations we used seem to be very realistic and involve realistic material parameters.
- The approach we describe indicates presence of a source, but not its location (at least in the heavy iron cargo case). One wonders whether location can also be attempted.
- The exposure time required to reach a certain level of particle detections is a function of the configuration of the cargo, including source location, material composition, and material placement. This makes it difficult to predict boundary flux rates, even if the configuration is known, without solving the Boltzmann equation \( \Box \). As a general rule of thumb, the more high-Z materials present, such as iron, the lower will be the boundary flux. Figure 12 reflects the result of several thousand configuration runs for detecting the presence of a source emitting on the order of 1000 particles. This is a histogram of the runs vs. time required for detection. Generally speaking, the large bin on the left-hand side corresponds to configurations with less high-Z materials, and the larger bins on the right-hand side correspond to configurations with more high-Z materials. It can certainly become unrealistic to detect many source particles in some of the latter cases. Nevertheless, as is evidenced by some of the examples presented previously, certain configurations of high-Z material exist where many source particles can be detected in a reasonable amount of time. These lower exposure time scenarios would be the most appropriate cases for detecting illicit nuclear materials at a border crossing. Some of the longer exposure times (on the order of several minutes to perhaps several days) would be appropriate for detection of illicit nuclear materials in shipping containers on cargo ships, where scanning can be done while the container is in transit.

10. Acknowledgements

The first author acknowledges the support from the NSF DMS grant # 1816430 and from Texas A&M Cyclotron Institute. The second author was supported by the National Science Foundation DMS grants # 1906664, # 1816430, and # 1211463. The third author has been partially supported through a grant by the Department of the Defense, Defense Threat
Figure 12. Histogram of the number of runs vs. exposure times required for detection.

Reduction Agency under Award No. HDTRA1-18-1-0020. The content of the information does not necessarily reflect the position or the policy of the federal government, and no official endorsement should be inferred.

References

[1] Abadi M, et. al. 2015 Tensorflow: Large-Scale Machine Learning on Heterogeneous Systems [https://www.tensorflow.org/]

[2] Adams M P, Adams M L, Hawkins W D, Smith T, Rauchwerger L, Amato N M, Bailey T S and Falgout R D 2013 Provably optimal parallel transport sweeps on regular grids International Conference on Mathematics and Computational Methods Applied to Nucl. Sci. & Eng. 4 2535

[3] Adams M P, Adams M L, McGraw C N and Till A T 2015 Provably optimal parallel transport sweeps with non-contiguous partitions ANS MC2015-Joint International Conference on Mathematics and Computation (M&IC), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method 2 1218

[4] Adams M P, Adams M L, Hawkins W D, Smith T, Rauchwerger L, Amato N M, Bailey T S, Falgout R D, Kunen A and Brown P 2020 Provably optimal parallel transport sweeps on semi-structured grids Journal of Computational Physics 01/2020 109234

[5] Allmaras M, Darrow D, Hristova Y, Kanschat G and Kuchment P 2010 Detecting small low emission radiating sources Inverse Problems & Imaging, 7(1) 47

[6] Allmaras M, Ciabatti A, Hristova Y, Kuchment P, Olson A and Ragusa J 2016 Passive Detection of Small Low-Emission Sources: Two-Dimensional Numerical Case Studies Nuclear Sci. and Eng. 184(1) 125

[7] Bell G I and Glasstone S 1970 Nuclear reactor theory. (Malabar, FL: Krieger Publishing Company)

[8] Carlson B G 1955 Solutions of the Transport Equation by SN Approximations (Los Alamos, NM: Los Alamos Scientific Laboratory)
[9] Chen G, Esch G, Wonka P, Müller P and Zhang E 2008 Interactive Procedural Street Modeling ACM Trans. Graph. 27 1
[10] Chollet F 2015 Keras GitHub https://github.com/fchollet/keras
[11] Duderstadt J J and Martin W R 1979 Transport theory (New York, NY: John Wiley & Sons)
[12] Goodfellow I, Bengio Y and Courville A 2016 Deep Learning (Cambridge, MA: MIT Press)
[13] Hawkins W D, Bailey T S, Adams M L, Brown P N, Kunen A J, Adams M P, Smith T, Amato N M and Rauchwerger L 2014 Validation of Full-Domain Massively Parallel Transport Sweep Algorithms Trans. Amer. Nucl. Soc. 111 699
[14] Jones E, et. al. 2001 SciPy: Open Source Scientific Tools for Python http://www.scipy.org/
[15] Kingma D P and Ba J 2015 Adam: A Method for Stochastic Optimization International Conference on Learning Representations
[16] Lamarsh J 1966 Introduction to nuclear reactor theory. (Boston, MA: Addison-Wesley series in nuclear engineering. Addison-Wesley Pub. Co.)
[17] Lesaint P and Raviart P A 1974 On a finite element method for solving the neutron transport equation. Mathematical Aspects of Finite Elements in Partial Differential Equations, 33 89
[18] Lewis E E and Miller W F 1984 Computational methods of neutron transport (United States: John Wiley and Sons, Inc.)
[19] Liu W, Wang Z, Liu X, Zeng N, Liu Y and Alsaadi F 2017 A survey of deep neural network architectures and their applications, Neurocomputing 234 11
[20] MacFarlane R and Muir D 1994 The NJOY nuclear data processing system version 91 (Los Alamos, NM: Los Alamos Scientific Laboratory)
[21] Natterer F 2001 Mathematics of computerized tomography (Philadelphia, PA: SIAM: Society for Industrial and Applied Mathematics)
[22] Perlin K 1985 An Image Synthesizer Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques 10 287
[23] Reed W H and Hill T R 1973 Triangular mesh methods for the neutron transport equation. National topical meeting on mathematical models and computational techniques for analysis of nuclear systems Los Alamos Report LA-UR-73-479
[24] Sánchez R and Ragusa J 2011 On the construction of galerkin angular quadratures Nuclear Science and Engineering 169(2) 133
[25] Santi P A 2013 Passive nondestructive assay of nuclear materials (Los Alamos NM: Los Alamos Scientific Laboratory) Los Alamos Report LA-UR-73-479
[26] Terzioglu F, Kuchment P and Kunyansky L 2018 Compton camera imaging and the cone transform : A brief overview Inverse Problems 34(5) 054002.
[27] Wareing T, McGhee J, Morel J and Pautz S 2001 Discontinuous finite element Sn methods on three-dimensional unstructured grids Nuclear Science and Engineering, 138(3) 256
[28] Wu K, Otoo E and Shoshani A 2005 Optimizing connected component labeling algorithms Medical Imaging 2005: Image Processing 5747 1965
[29] Xun X, Mallick B, Carroll R and Kuchment P 2011 Bayesian approach to detection of small low emission sources Inverse Problems 27(11) 115009
11. Appendix

Algorithm 1: Procedural Cargo Configuration

Generate Perlin noise in cargo;
Initialize \( n_x \) and \( n_y \) to desired number of vertical and horizontal boundaries (numbers can be chosen randomly);
Sum Perlin noise over rows and columns to produce noise function on edge of cargo;
Randomly select \( n_x \) distinct \( x \)-coordinates for vertical boundaries and \( n_y \) distinct \( y \)-coordinates for horizontal boundaries according to edge noise functions. Store in \( x \) and \( y \) respectively.
\( n_{iter} = 0 \);

\[
\textbf{while } n_x > 0 \text{ or } n_y > 0 \text{ do } \\
\quad \textbf{if } n_{iter} \text{ is even and } n_x > 0 \text{ then } \\
\quad \quad \text{Determine all existing boundary points along the line } (x[n_{iter}/2], y). \\
\quad \quad \text{Randomly select a starting point } y_s \text{ and ending point } y_e \text{ from among the existing boundary points according to previously generated Perlin noise. Set all points between } (x[n_{iter}/2], y_s) \text{ and } (x[n_{iter}/2], y_e) \text{ to boundary points.}; \\
\quad \quad n_x = n_x - 1; \\
\quad \quad n_{iter} = n_{iter} + 1; \]
\[
\quad \textbf{else if } n_{iter} \text{ is odd and } n_y > 0 \text{ then } \\
\quad \quad \text{Determine all existing boundary points along the line } (x, y[(n_{iter} - 1)/2]). \\
\quad \quad \text{Randomly select a starting point } x_s \text{ and ending point } x_e \text{ from among the existing boundary points according to previously generated Perlin noise. Set all points between } (x_s, y[(n_{iter} - 1)/2]) \text{ and } (x_e, y[(n_{iter} - 1)/2]) \text{ to boundary points.}; \\
\quad \quad n_y = n_y - 1; \\
\quad \quad n_{iter} = n_{iter} + 1; \\
\quad \textbf{end} \]
Identify connected components (Scipy.Measure.Label);
\[
\textbf{if } \text{Rotational Symmetry Desired} \text{ then } \\
\quad \text{Copy one quadrant of the configuration over all others with appropriate rotation}; \]
\[
\textbf{if } \text{Mirror Symmetry Desired} \text{ then } \\
\quad \text{Copy one side of the configuration over the other with mirroring }; \]
\[
\ldots \\
\text{Randomly assign material identification to each connected component}; \\
\text{Save configuration to file}; \\
\textbf{Result: } \text{Single cargo configuration}
| Group Index | Lower Bound (MeV) | Upper Bound (MeV) |
|-------------|------------------|------------------|
| 1           | 2.61449          | 2.61451          |
| 2           | 2.56137          | 2.61449          |
| 3           | 2.50824          | 2.56137          |
| 4           | 2.45512          | 2.50824          |
| 5           | 2.402            | 2.45512          |
| 6           | 2.34887          | 2.402            |
| 7           | 2.29575          | 2.34887          |
| 8           | 2.24262          | 2.29575          |
| 9           | 2.1895           | 2.24262          |
| 10          | 2.13638          | 2.1895           |
| 11          | 2.08325          | 2.13638          |
| 12          | 2.03013          | 2.08325          |
| 13          | 1.977            | 2.03013          |
| 14          | 1.92388          | 1.977            |
| 15          | 1.87076          | 1.92388          |
| 16          | 1.81763          | 1.87076          |
| 17          | 1.76451          | 1.81763          |
| 18          | 1.76449          | 1.76451          |
| 19          | 1.71391          | 1.76449          |
| 20          | 1.66333          | 1.71391          |
| 21          | 1.61275          | 1.66333          |
| 22          | 1.56217          | 1.61275          |
| 23          | 1.51159          | 1.56217          |
| 24          | 1.46101          | 1.51159          |
| 25          | 1.46099          | 1.46101          |
| 26          | 1.40421          | 1.46099          |
| 27          | 1.34743          | 1.40421          |
| 28          | 1.29065          | 1.34743          |
| 29          | 1.23387          | 1.29065          |
| 30          | 1.17709          | 1.23387          |
| 31          | 1.12031          | 1.17709          |
| 32          | 1.12029          | 1.12031          |
| 33          | 1.09047          | 1.12029          |
| 34          | 1.06065          | 1.09047          |
| 35          | 1.03083          | 1.06065          |
| 36          | 1.00101          | 1.03083          |
| 37          | 1.00099          | 1.00101          |

Table 6. Energy Group Structure