DeepSig

Pioneering Deep Learning for Wireless

Ben Hilburn

DEEPSIG

Universal Approximation

Let $\varphi(\cdot)$ be a nonconstant, bounded, and monotonically-increasing continuous function. Let I_m denote the m-dimensional unit hypercube $[0,1]^m$. The space of continuous functions on I_m is denoted by $C(I_m)$. Then, given any $\varepsilon > 0$ and any function $f \in C(I_m)$, there exist an integer N, real constants $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$, where $i = 1, \cdots, N$, such that we may define:

$$F(x) = \sum_{i=1}^{N} v_i \varphi\left(w_i^T x + b_i\right)$$

as an approximate realization of the function f where f is independent of φ; that is,

$$|F(x) - f(x)| < \varepsilon$$

for all $x \in I_m$. In other words, functions of the form $F(x)$ are dense in $C(I_m)$.

DEEPSIG

Universal Approximation

"The theorem thus states that simple neural networks can *represent* a wide variety of interesting functions when given appropriate parameters; however, it does not touch upon the algorithmic learnability of those parameters."

https://en.wikipedia.org/wiki/Universal_approximation_theorem

DEEPSIG

Machine Learning

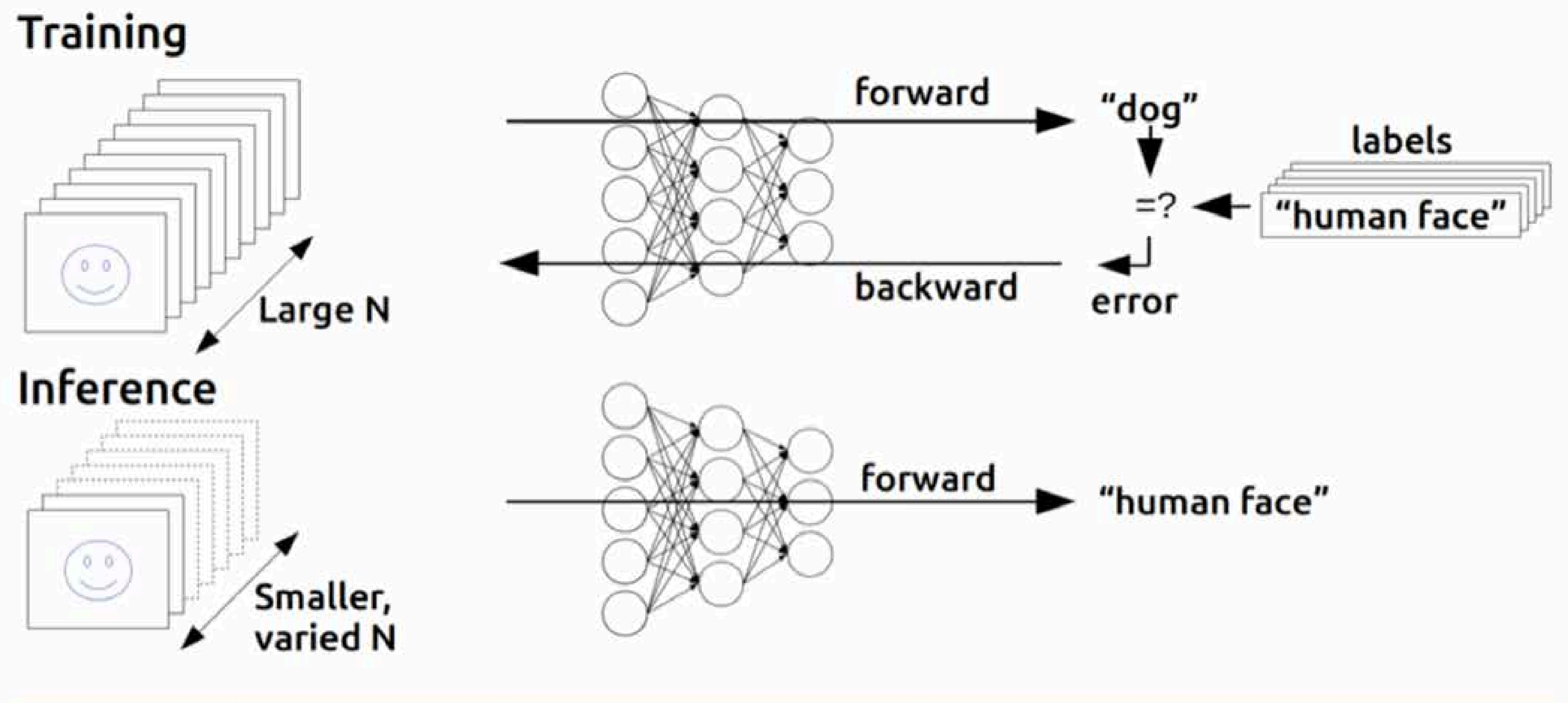


Figure 1: Deep learning training compared to inference. In training, many inputs, often in large batches, are used to train a deep neural network. In inference, the trained network is used to discover information within new inputs that are fed through the network in smaller batches.

Image Credit: https://devblogs.nvidia.com/inference-next-step-gpu-accelerated-deep-learning/

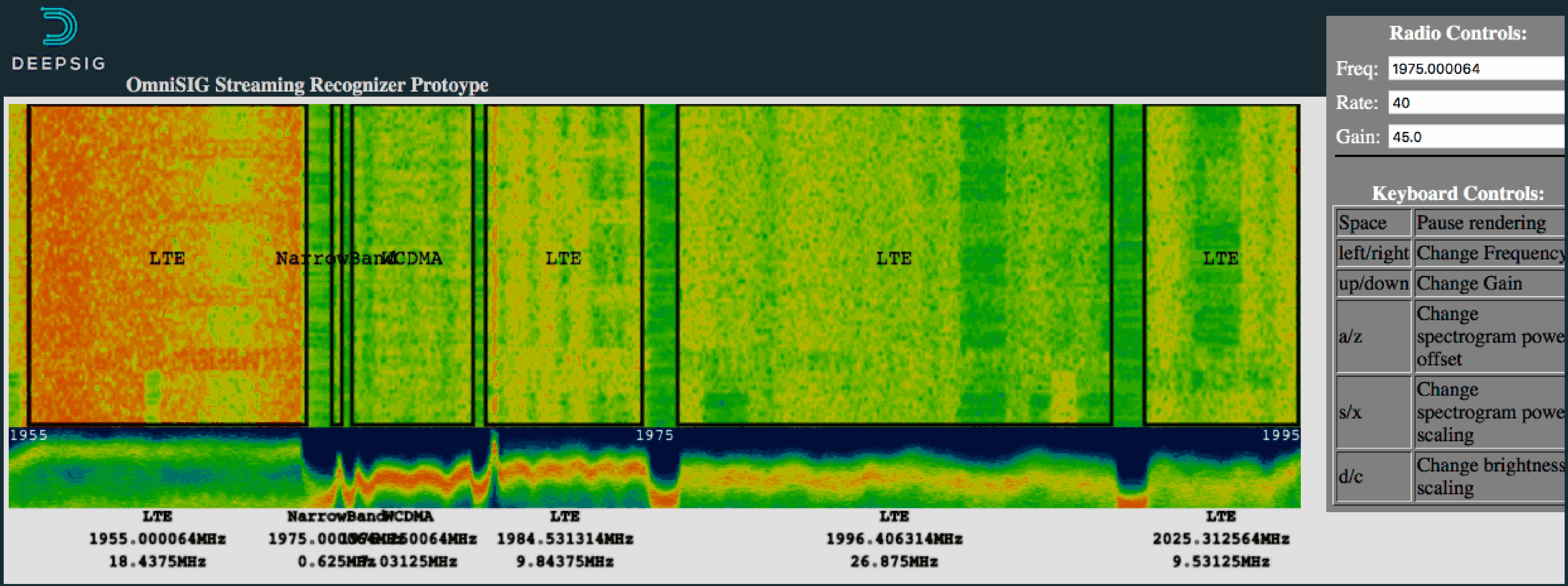
DEEPSIG

Machine Learning for RF

- Replacing signal processing with machine learning
  - Applying the concept of Software 2.0 to RF systems
- In all deep learning applications, the data is the key.
  - "…in the future your data is your company's source code." – J. Huang, NVIDIA CEO
- Two primary product areas: sensing and *learned* physical layers

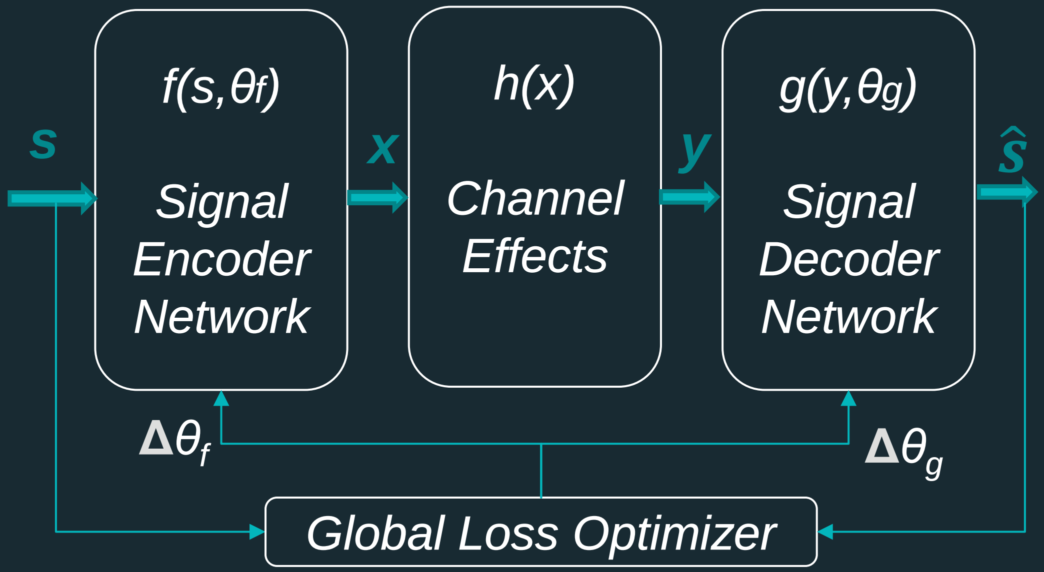
DEEPSIG

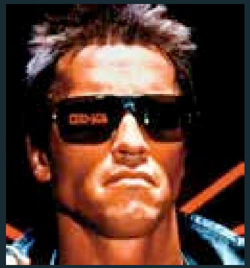
Data for Sensing



Learned Physical Layers

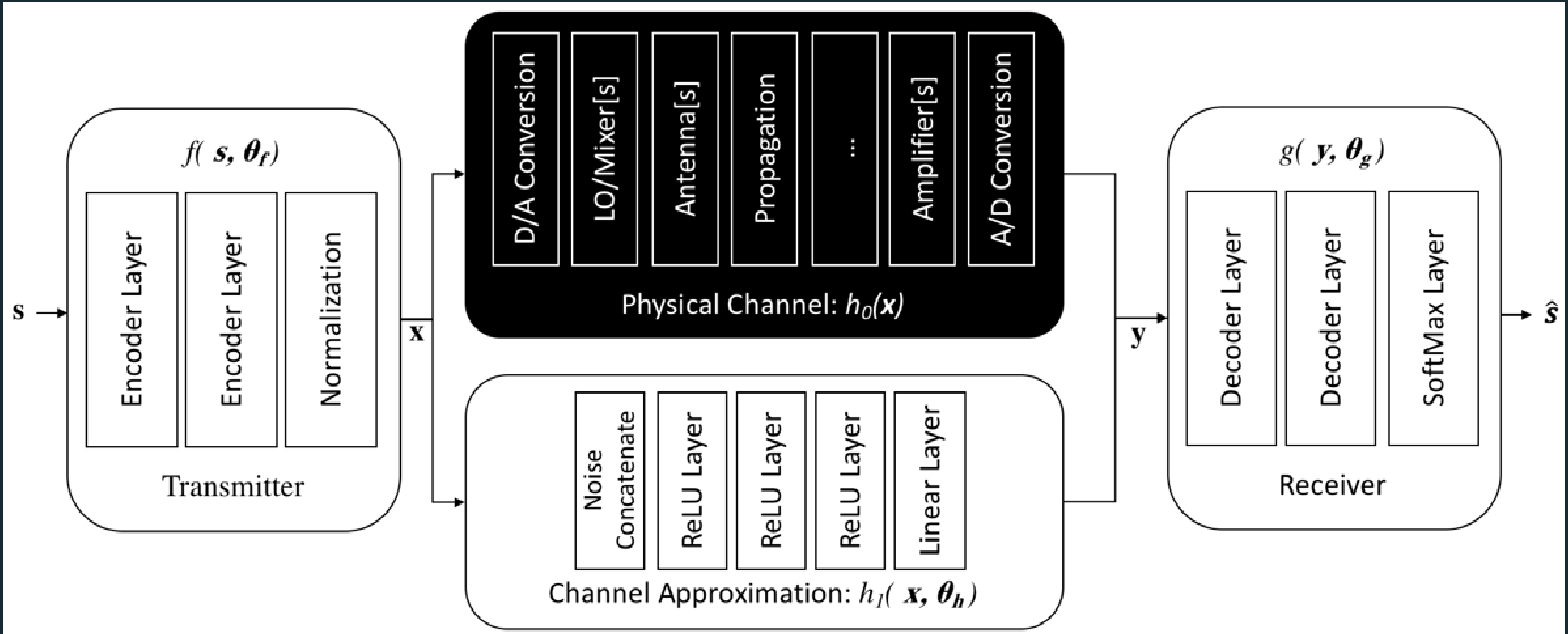
- Creating a *learned physical layer* means training over a channel or channel model.
- One of our techniques is *learning* a channel model on which to train a PHY.
- Put differently, we use Deep Learning to approximate channel models.
    - The machines outperform us.





DEEPSIG

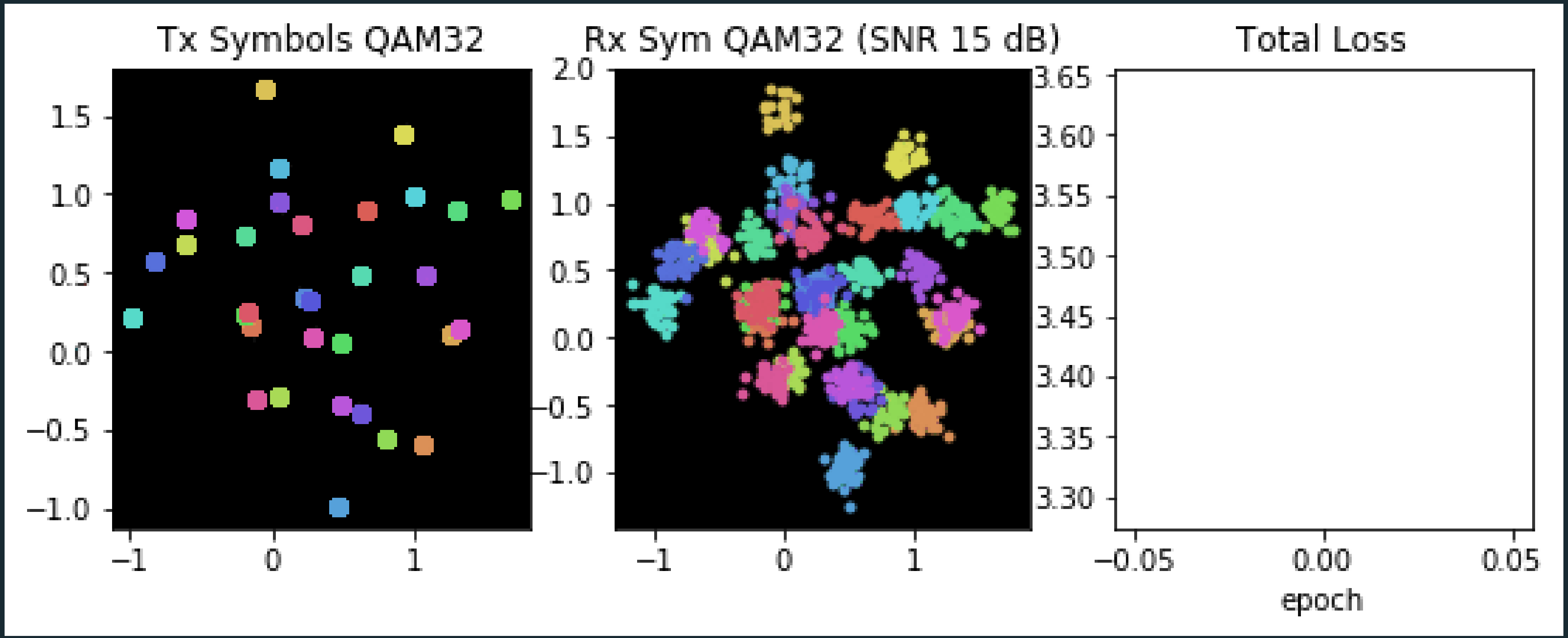
Learned Channel Models



DEEPSIG

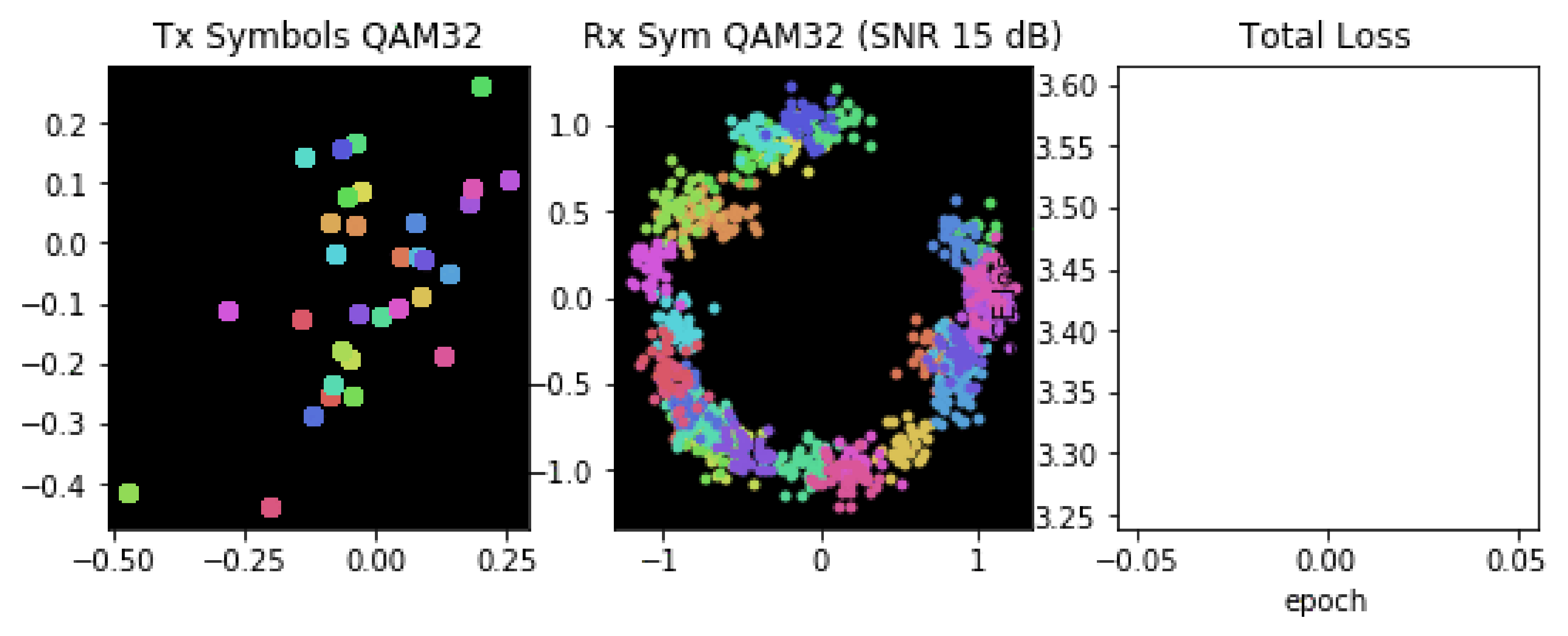
Training over Simple Channel Models

Training a simple 32-QAM autoencoder for an AWGN channel



Training over Insane Channel Models

Training a 32-QAM system over harsh TDRSS TWTA non-linearities



DEEPSIG

(Ballmer voice) Data, Data, Data!

- We are using the Signal Metadata Format (SigMF) for everything
  - https://sigmf.org
  - Disclaimer: I'm the lead developer of SigMF, but I'm totally not biased.
  - Specification for describing recordings of digital samples with JSON
- Based on our experience thus far, *real data™* is critical
  - Actually making useful datasets with *real data™* turns out to be rather difficult
  - We are collecting, processing, and labeling live captures