ViterbiNet: A Deep Learning Based Viterbi Algorithm for Symbol Detection (2020)

Nir Shlezinger, Nariman Farsad, Yonina C. Eldar, and Andrea J. Goldsmith

EE4140 Mini Project

Aditya Mallick

Background

- Task of receiver - symbol detection.
- Conventional detection algorithms rely on the the instantaneous CSI (channel state information) for detection.
- This needs to be accurately estimated, otherwise leading to poor detection performance.
- Channel estimation overhead also decreases the data transmission rate – need a solution for effective symbol detection

Motivation for ML Based Approach

- ML doesn’t rely on known stochastic models, becomes effective when the model is unknown or hard to estimate.
- ML often captures information hidden in complex data better than traditional methods.
- ML techniques often converge faster than iterative model-based approaches, even when the model is known.

Related Work

- DNN for decoding structured codes (Gruber et al., 2017)
- Neural Belief Propagation - decoding linear codes (Nachmani et al., 2018)
- VAE for equalizing linear multipath channels (Caciularu et al. 2018)
- Sliding Bidirectional RNN for sequence detection (Farsad et al., 2018)
- RNN decoders for sequential codes (Kim et al. 2018)

These approaches yield good performance when sufficient training is available.

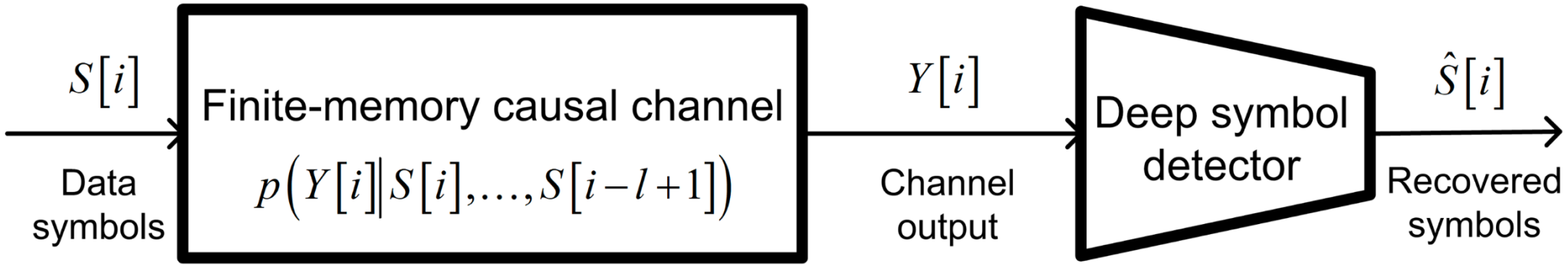
Scope for Improvement

- Previous approaches treated the network as a black box
- Depended on exhaustive training and relied on methods that were developed to treat secondary tasks such as image processing.

Q. Can we achieve gains in performance, complexity, or training size, by combining channel-model-based methods, such as the Viterbi algorithm, with ML-based techniques?

Mathematical Formulation

The channel can be modelled as a conditional PDF between input and output.



Courtesy of: Shlezinger et Al., “ViterbiNet: A deep learning based Viterbi algorithm for symbol detection,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Mathematical Formulation

Probability of receiving the output sequence Y_{k1}^{k2} given a sequence of input symbols s^t is:

$$p_{\boldsymbol{Y}_{k_1}^{k_2}|\boldsymbol{S}^t}\left(\boldsymbol{y}_{k_1}^{k_2}|\boldsymbol{s}^t\right) = \prod_{i=k_1}^{k_2} p_{Y[i]|\boldsymbol{S}_{i-l+1}^{i}}\left(y[i]|\boldsymbol{s}_{i-l+1}^{i}\right)$$

Courtesy of: Shlezinger et Al., “ViterbiNet: A deep learning based Viterbi algorithm for symbol detection,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Review of the Viterbi Algorithm

- One of the most common symbol detection algorithms
- Minimum error probability symbol detector for channels obeying a Markovian input-output stochastic relationship, which is encountered in many practical channels.
- Requires instantaneous CSI.

Mathematical Formulation of VA

Since VA gives minimum probability of error, we know that for an equiprobable constellation this corresponds to a maximum likelihood decision rule:

$$\hat{\boldsymbol{s}}^t\left(\boldsymbol{y}^t\right) \triangleq \underset{\boldsymbol{s}^t \in \mathcal{S}^t}{\arg\max}\, p_{\boldsymbol{Y}^t|\boldsymbol{S}^t}\left(\boldsymbol{y}^t|\boldsymbol{s}^t\right)$$

$$= \underset{\boldsymbol{s}^t \in \mathcal{S}^t}{\arg\min} - \log p_{\boldsymbol{Y}^t|\boldsymbol{S}^t}\left(\boldsymbol{y}^t|\boldsymbol{s}^t\right)$$

Courtesy of: Shlezinger et Al., “ViterbiNet: A deep learning based Viterbi algorithm for symbol detection,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Mathematical Formulation of VA

We will define the log-likelihood function below

$$c_i\left(\boldsymbol{s}\right) \triangleq -\log p_{\boldsymbol{Y}[i]|\boldsymbol{S}_{i-l+1}^{i}}\left(\boldsymbol{y}[i]|\boldsymbol{s}\right), \qquad \boldsymbol{s} \in \mathcal{S}^{l}$$

Then, using the channel model, we can write the following:

$$\log p_{\boldsymbol{Y}^t|\boldsymbol{S}^t}\left(\boldsymbol{y}^t|\boldsymbol{s}^t\right) = \sum_{i=1}^{t} c_i\left(\boldsymbol{s}_{i-l+1}^{i}\right)$$

Courtesy of: Shlezinger et Al., “ViterbiNet: A deep learning based Viterbi algorithm for symbol detection,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Mathematical Formulation of VA

Finally, the problem reduces to:

$$\hat{\boldsymbol{s}}^t\left(\boldsymbol{y}^t\right) = \underset{\boldsymbol{s}^t \in \mathcal{S}^t}{\arg\min} \sum_{i=1}^{t} c_i\left(\boldsymbol{s}_{i-l+1}^{i}\right)$$

Can be solved recursively by treating the possible combinations of transmitted symbols at each time instance as *states* and iteratively updating a *path cost* value for each state.

Courtesy of: Shlezinger et Al., “ViterbiNet: A deep learning based Viterbi algorithm for symbol detection,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Pseudocode for VA

Algorithm 1 The Viterbi Algorithm

1: Input: Block of channel outputs $\boldsymbol{y}^t$, where $t > l$.

2: Initialization: Set $k = 1$, and fix the initial path cost $\tilde{c}_0(\tilde{\boldsymbol{s}}) = 0$, for each state $\tilde{\boldsymbol{s}} \in \mathcal{S}^l$.

3: For each state $\tilde{\boldsymbol{s}} \in \mathcal{S}^l$, compute the path cost via $\tilde{c}_k(\tilde{\boldsymbol{s}}) = \min_{\boldsymbol{u} \in \mathcal{S}^l : \boldsymbol{u}_2^l = \tilde{\boldsymbol{s}}^{l-1}} (\tilde{c}_{k-1}(\boldsymbol{u}) + c_k(\tilde{\boldsymbol{s}}))$.

4: If $k \geq l$, set $(\hat{\boldsymbol{s}})_{k-l+1} := (\tilde{\boldsymbol{s}}_k^{\mathrm{o}})_1$, where $\tilde{\boldsymbol{s}}_k^{\mathrm{o}} = \arg\min_{\tilde{\boldsymbol{s}} \in \mathcal{S}^l} \tilde{c}_k(\tilde{\boldsymbol{s}})$.

5: Set $k := k + 1$. If $k \leq t$ go to Step 3.

6: Output: decoded output $\hat{\boldsymbol{s}}^t$, where $\hat{\boldsymbol{s}}_{t-l+1}^t := \tilde{\boldsymbol{s}}_t^{\mathrm{o}}$.

Courtesy of: Shlezinger et Al., "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

VA Analysis

- CSI is required in the VA only to compute the log-likelihood function $c_i(s)$, which is used as a path metric in its trellis-based implementation.
- $c_i(s)$ can be pre-computed for each $s \in S^l$ and each received output $y[i]$.
- After this, VA only requires knowledge of the memory length l of the channel to recover the transmitted symbols.

Integrating ML into the Viterbi Algorithm

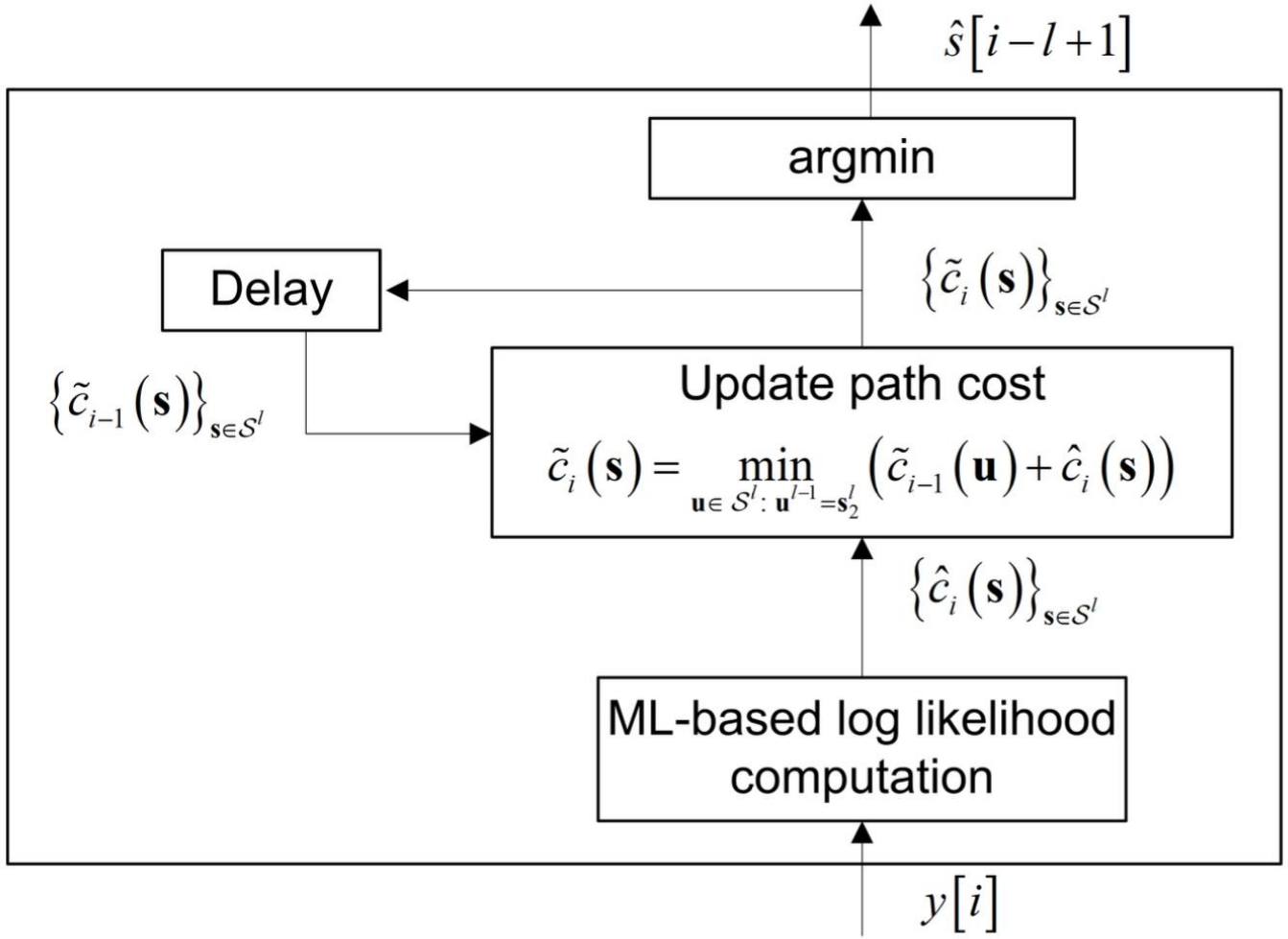
- We will henceforth assume that the channel is modelled as a causal LTI filter with a fixed memory length known to the receiver.

- Since channel is stationary, the log-likelihood function $c_i(s)$ depends only on the values of $y[i]$ and of s, and not on the time index i.

- Thus, we can use ML to learn the log-likelihood from training data.

Integrating ML into the Viterbi Algorithm

- For input $y[i]$, the output we need an estimate of the likelihood $\hat{c}(s)$ for each $s \in S^l$.
- The rest of the Viterbi algorithm remains same.
- ML is only used to compute the log-likelihood estimates.



Courtesy of: Shlezinger et Al., "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Integrating ML into the Viterbi Algorithm

- Using a neural network to compute $c_i(s)$ has a caveat.
- Classification DNNs with softmax layer at the end give output as the posterior distribution $\mathrm{P}(S|y[i])$ for a DNN input $y[i]$.
- But VA requires the likelihood of receiving $y[i]$ for each state, i.e $\mathrm{P}(y[i]||s)$. Note that this doesn't sum to one over all states, unlike the posterior PDF.
- Thus, we must transform the DNN output to the desired likelihood.

Integrating ML into the Viterbi Algorithm

Due to the equiprobable nature of transmitted symbols, we can express the desired PDF in terms of marginal and posterior PDFs by Baye's rule:

$$P(y|s) = \frac{P(s|y) \cdot P(y)}{m^{-l}}$$

m – constellation size, l – memory size

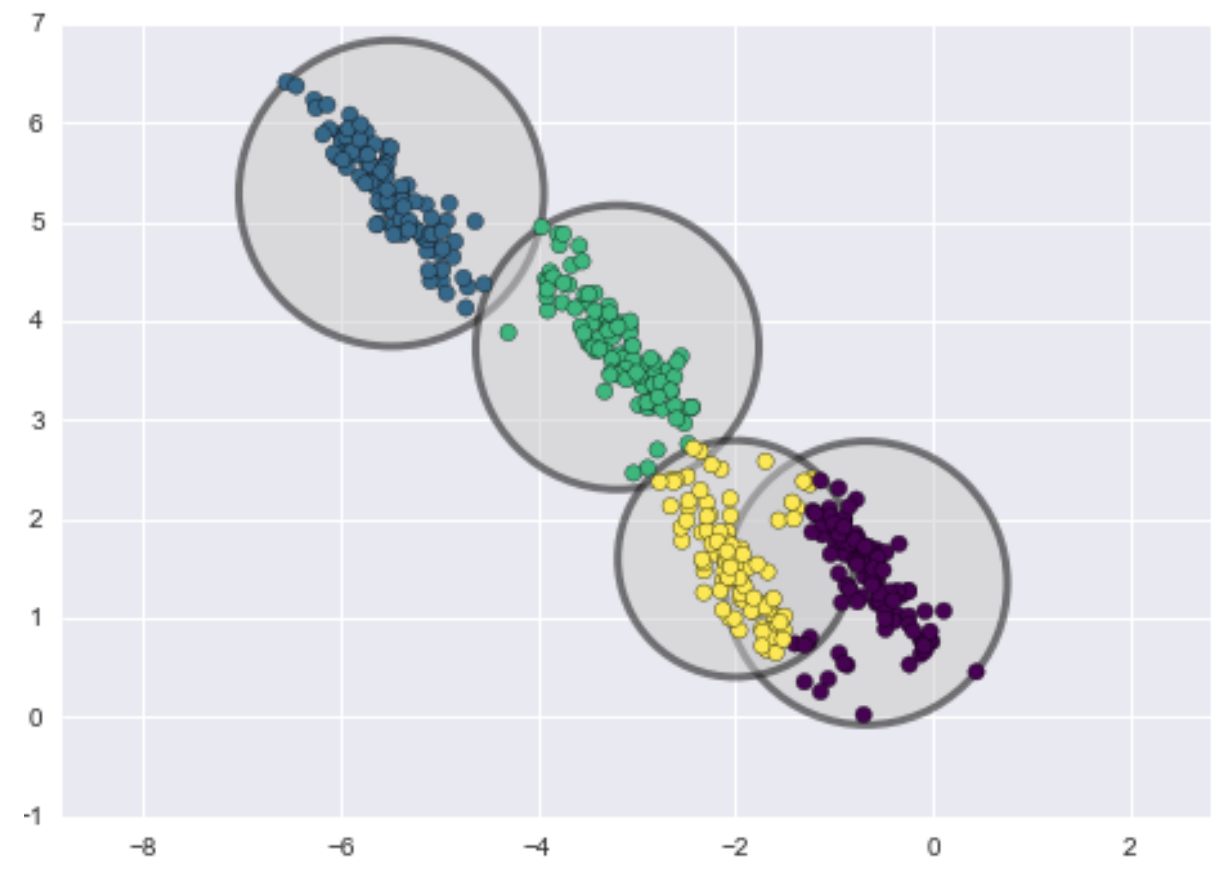
Integrating ML into the Viterbi Algorithm

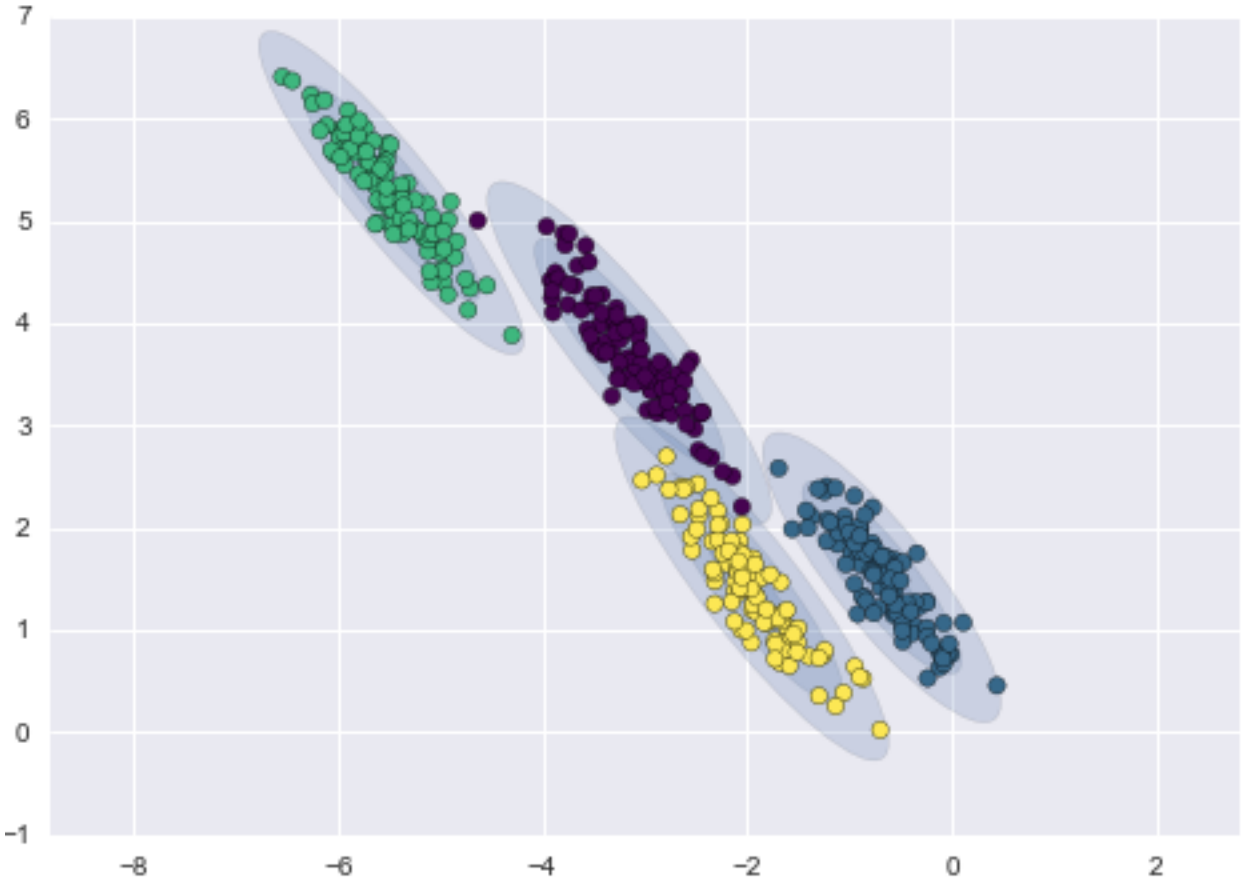
- $P(s|y[i])$ can be estimated by training a standard classification DNN.
- The marginal PDF of $Y[i]$ may be estimated from the training data by using kernel density estimation methods.
- Since $Y[i]$ is a stochastic mapping of S_{i-l+1}^{i} , its distribution can be approximated as a mixture model of m^l kernel functions.
- For our experiments, we use a Gaussian Mixture Model (GMM) to estimate the marginal PDF.

Gaussian Mixture Model

- Assumes all data points are generated from mixture of a finite number of Gaussian distributions with unknown parameters.
- Iteratively finds mean, variance and weight of each distribution in the mixture by using the expectation-maximization (EM) algorithm.
- Similar to k-means clustering but in addition to the mean, also incorporates information about the shape of the distribution (covariance).

K-Means Clustering vs GMM

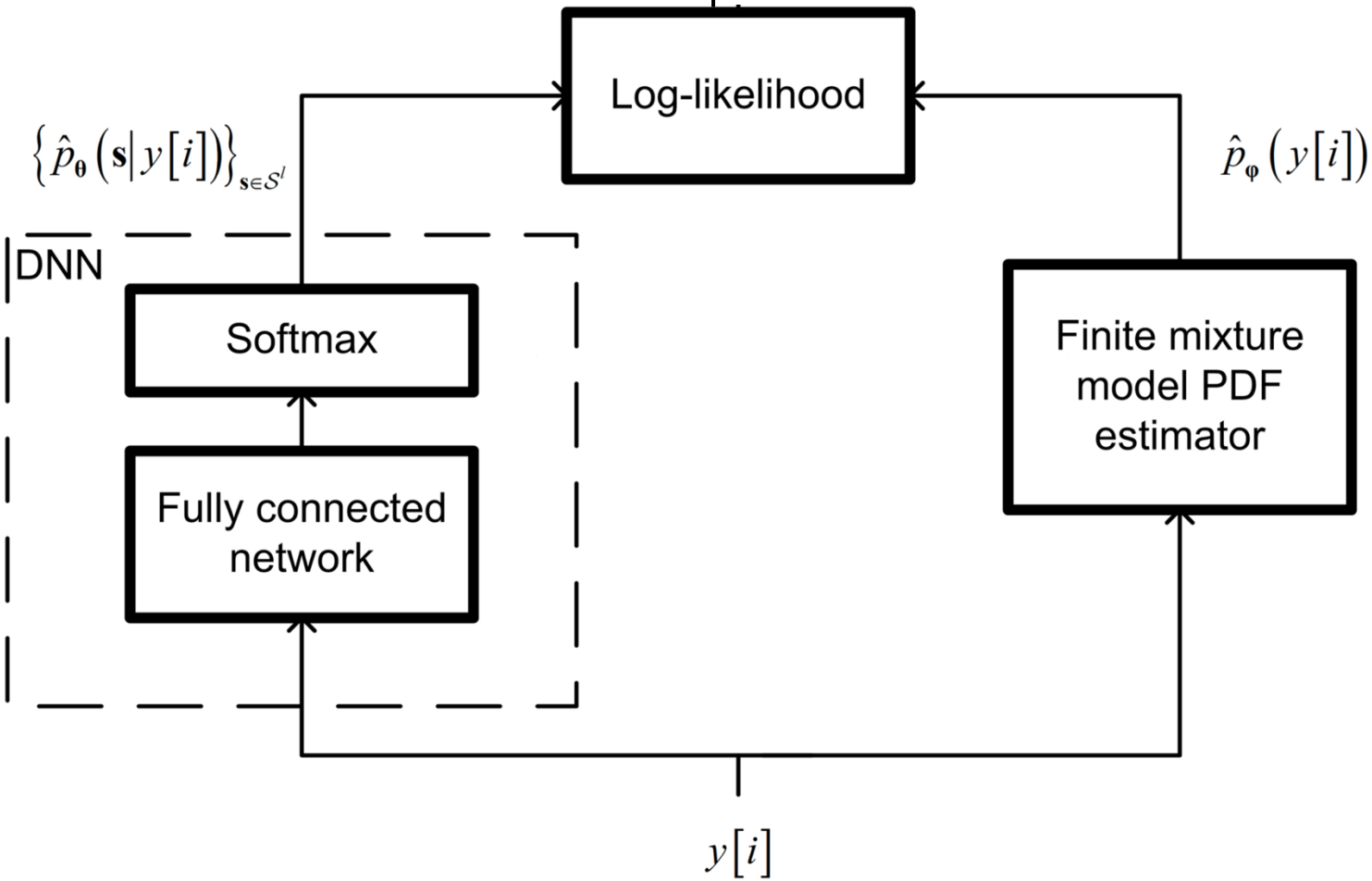




Courtesy of: https://www.kaggle.com/code/vipulgandhi/gaussian-mixture-models-clustering-explained

Final Workflow

$$\widehat{c_i}(s) = -\log \frac{\hat{P}_\theta(s|y[i]) \cdot \hat{P}_\varphi(y[i])}{m^{-l}}$$



Courtesy of: Shlezinger et Al., "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Experiments

We have considered an ISI Channel with exponential decay profile to model real-world fading phenomena:

$$(\boldsymbol{h})_\tau \triangleq e^{-\gamma(\tau-1)}$$

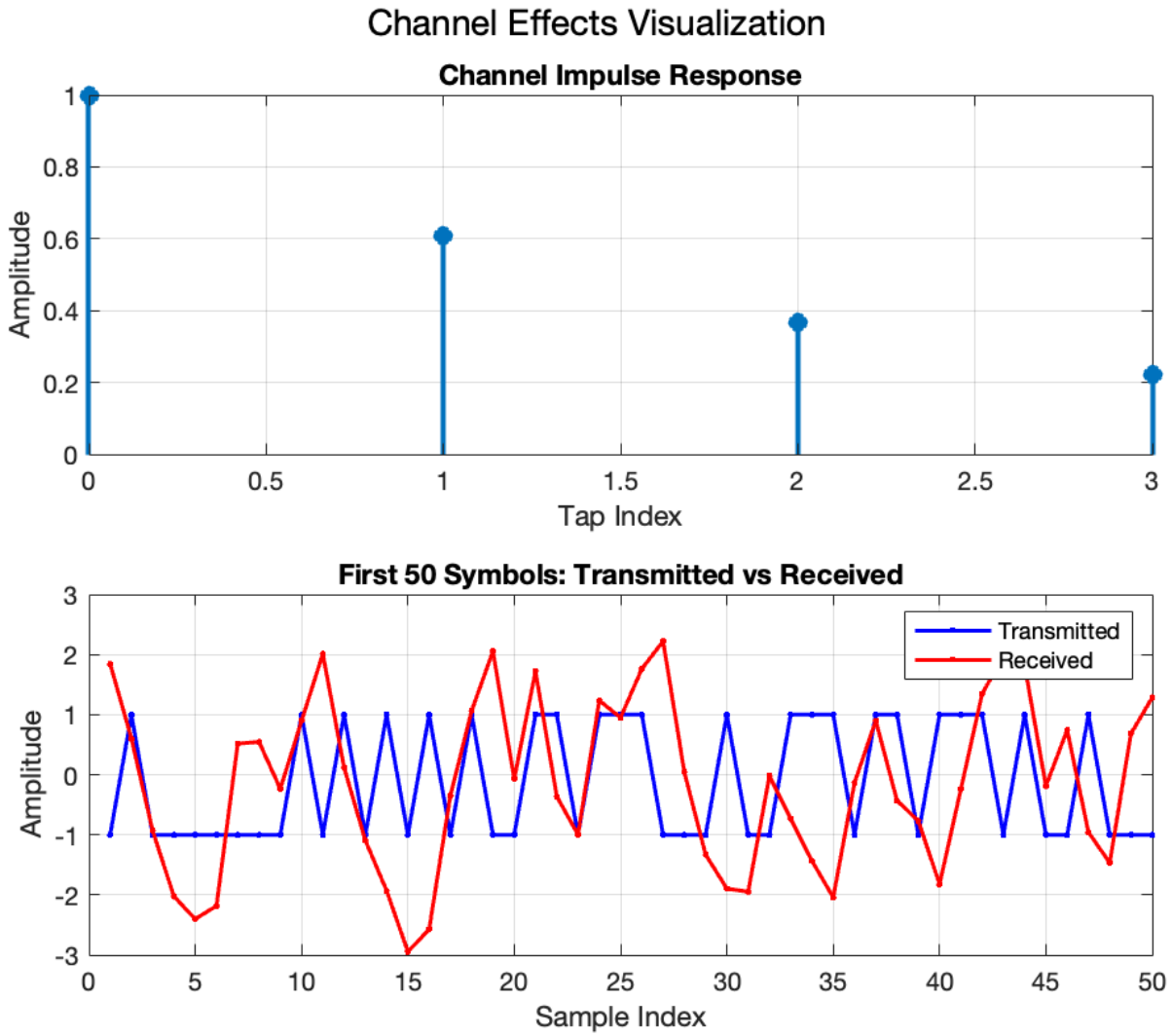
AWGN is also present, giving us the following output expression:

$$Y[i] = \sqrt{\rho} \cdot \sum_{\tau=1}^{l} (\boldsymbol{h}(\gamma))_\tau \, S[i-\tau+1] + W[i]$$

Courtesy of: Shlezinger et Al., “ViterbiNet: A deep learning based Viterbi algorithm for symbol detection,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.

Experiments

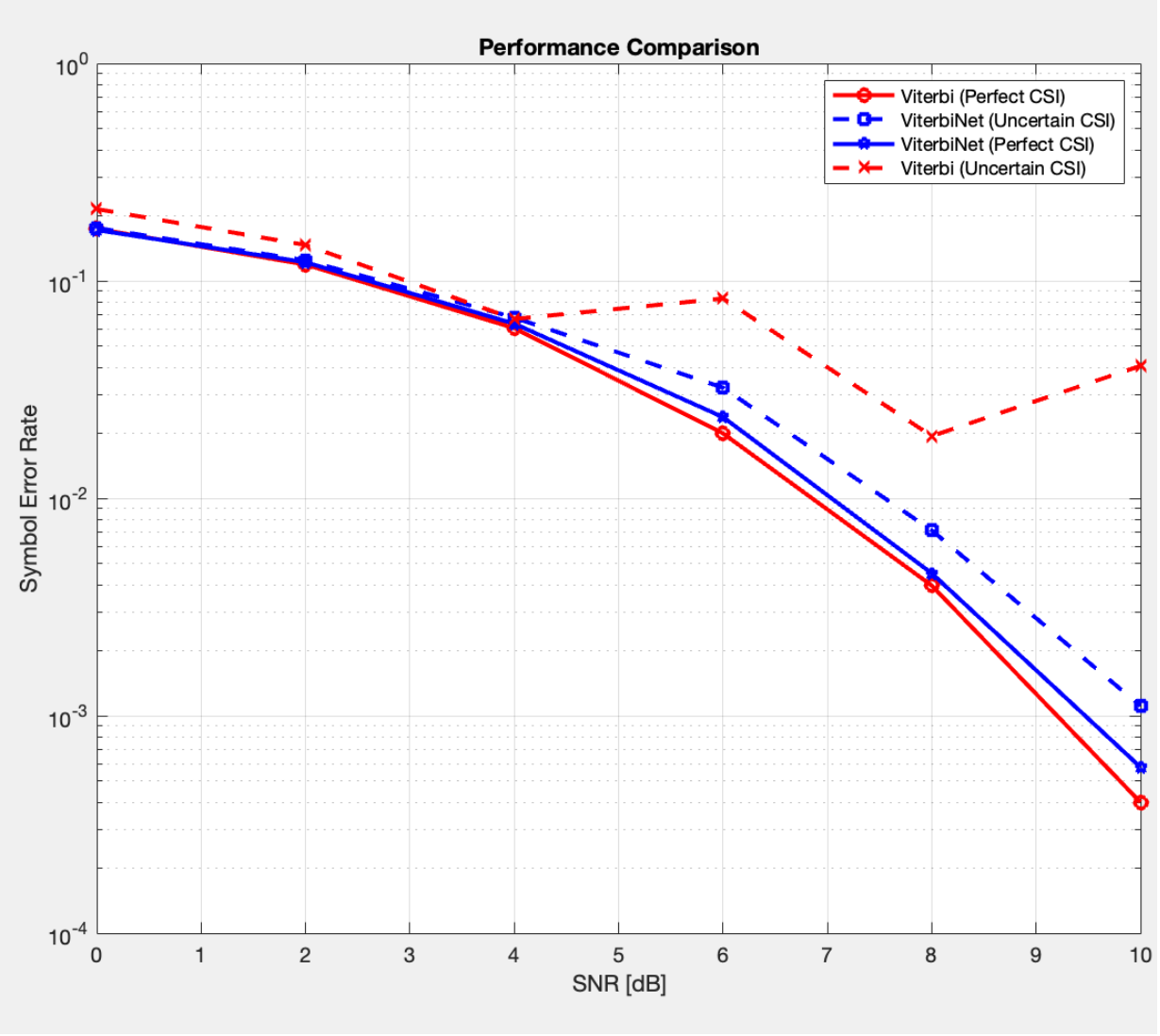
- BPSK modulated symbols, 4-tap real-valued channel.
- Monte-carlo simulations averaged over 50k symbols
- 3 – layer DNN (1x100 sigmoid, 100x50 Relu and 50x16 softmax).



Experiments

Comparison of ViterbiNet and Viterbi Algorithm in 2 settings:

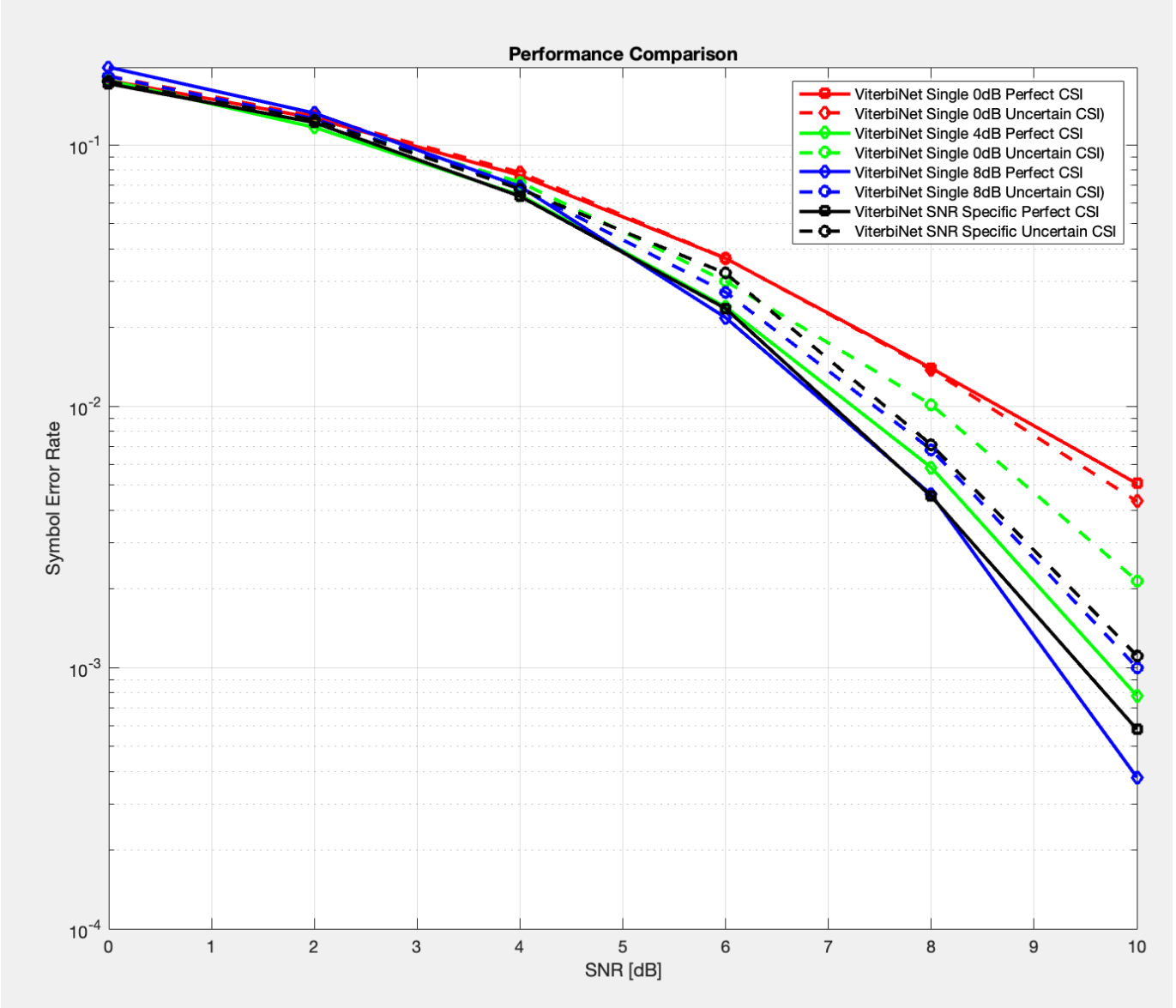
- Perfect CSI
- Uncertain CSI – noisy channel estimate for VA, VN trained using multiple noisy channel estimates.
- VN almost matches VA performance in full CSI setting.
- VN performs much better than VA in uncertain CSI scenario, only slightly worse than Perfect CSI.



Experiments

Comparison of ViterbiNet using general SNR training and specific SNR training:

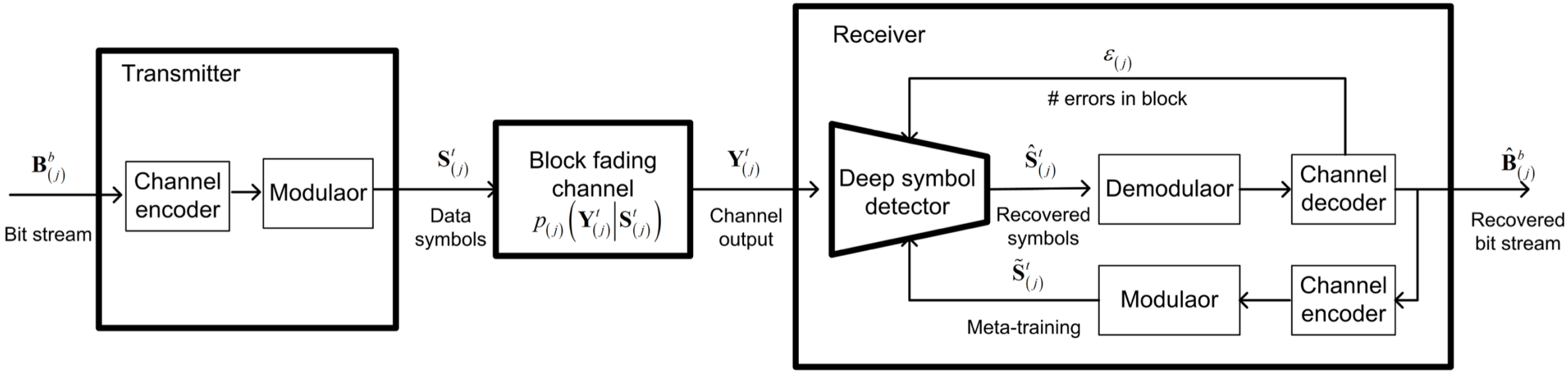
- ViterbiNet generalizes well to multiple SNR levels even when trained with single SNR data, sometimes even better than SNR specific training!
- Lesser training cost for same performance



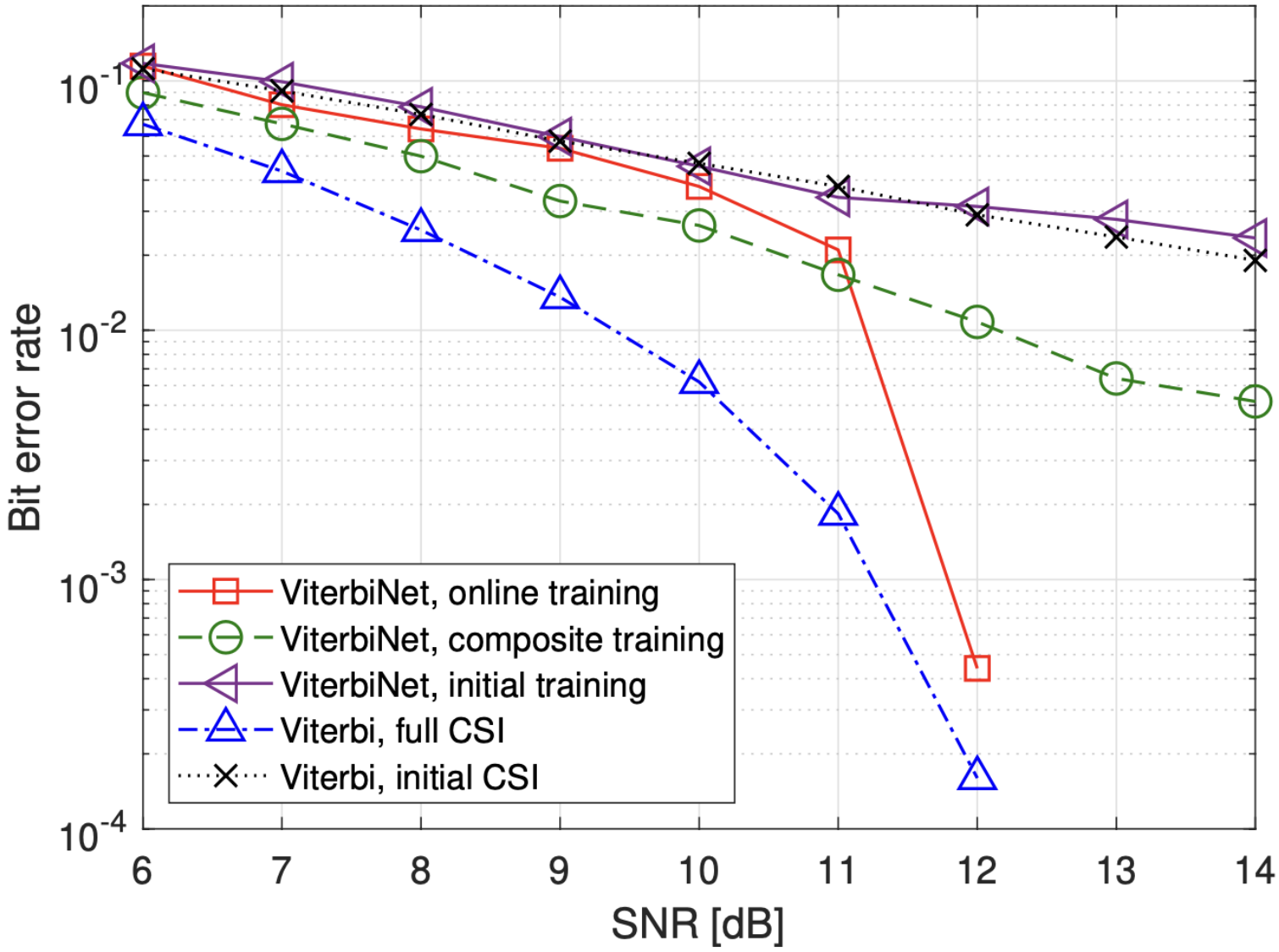
Online Training in Block-Fading Channels

- Block faded communications refers to channel changing with every block of data received.
- Channel coding can allow ViterbiNet to track channel variations and train online.
- The receiver will use accurately estimated transmitted symbols to re-train in real-time.
- A FEC code combined with error detection code like checksum, CRC will give best results.

Online Training in Block-Fading Channels



Courtesy of: Shlezinger et Al., "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3319–3331, 2020.



Paper result: Coded BER versus SNR, block-fading ISI channel with AWGN.

Main Takeaways

- ViterbiNet approaches the optimal VA and outperforms previous ML-based symbol detectors using a small amount of training data.
- VN can operate in the presence of CSI uncertainty, where the VA fails.
- VN can adapt via online learning in block-fading channel conditions.

Drawbacks and Future Directions

- ViterbiNet still suffers from inherent drawbacks of VA.
- Large constellation size and channel memory increases complexity of ViterbiNet, since the state space of the DNN output grows exponentially.
- We assumed that the receiver has either accurate knowledge or a reliable upper bound on channel memory l.
- The extension of the decoder to account for unknown memory length is left for future exploration.

Q&A

Code available at https://github.com/aditya2331/ViterbiNet-Testbench