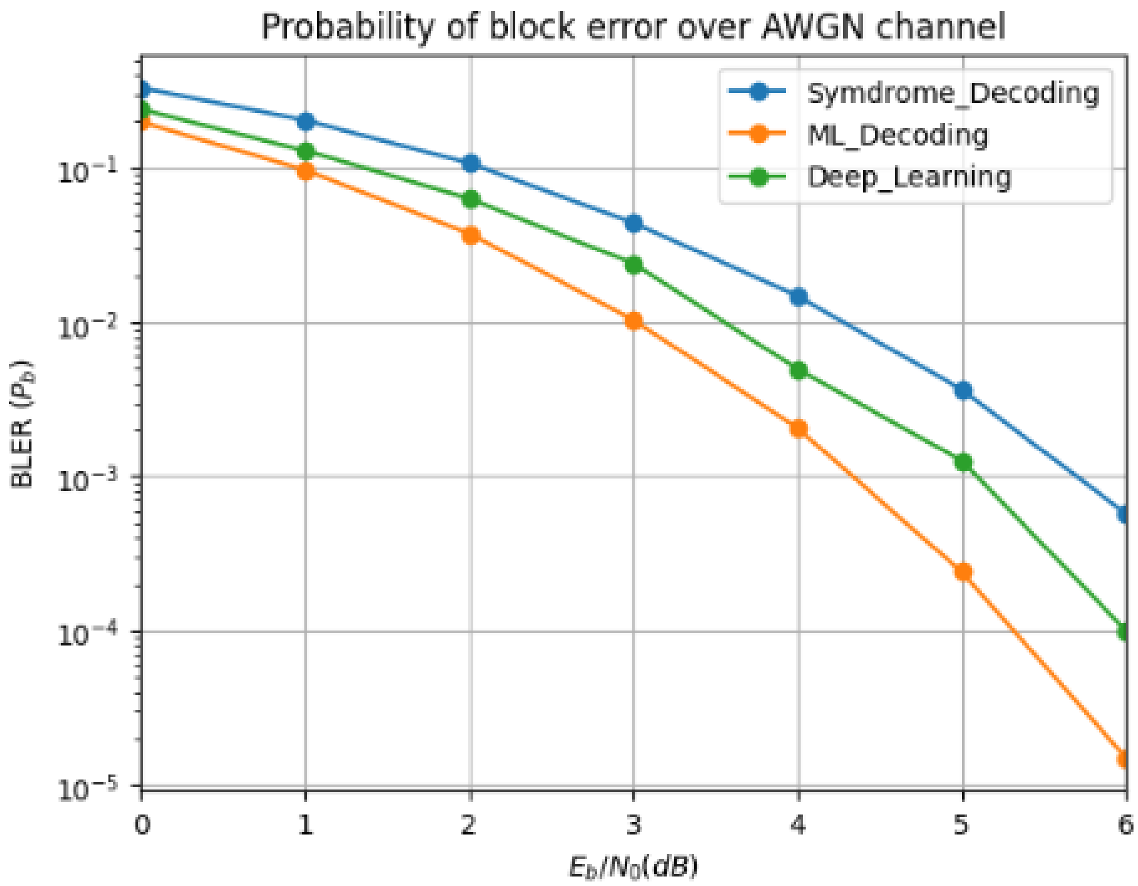
Mini Project for Module 1

111511245 朱昱安

1、The method to decoding in Mini Project is DNN, and the BLER rate of the method is as follows：

```
Block Error Rate: [2.41133333e-01 1.29866667e-01 6.34666667e-02 2.41000000e-02
 4.96666667e-03 1.26666667e-03 1.00000000e-04]
```

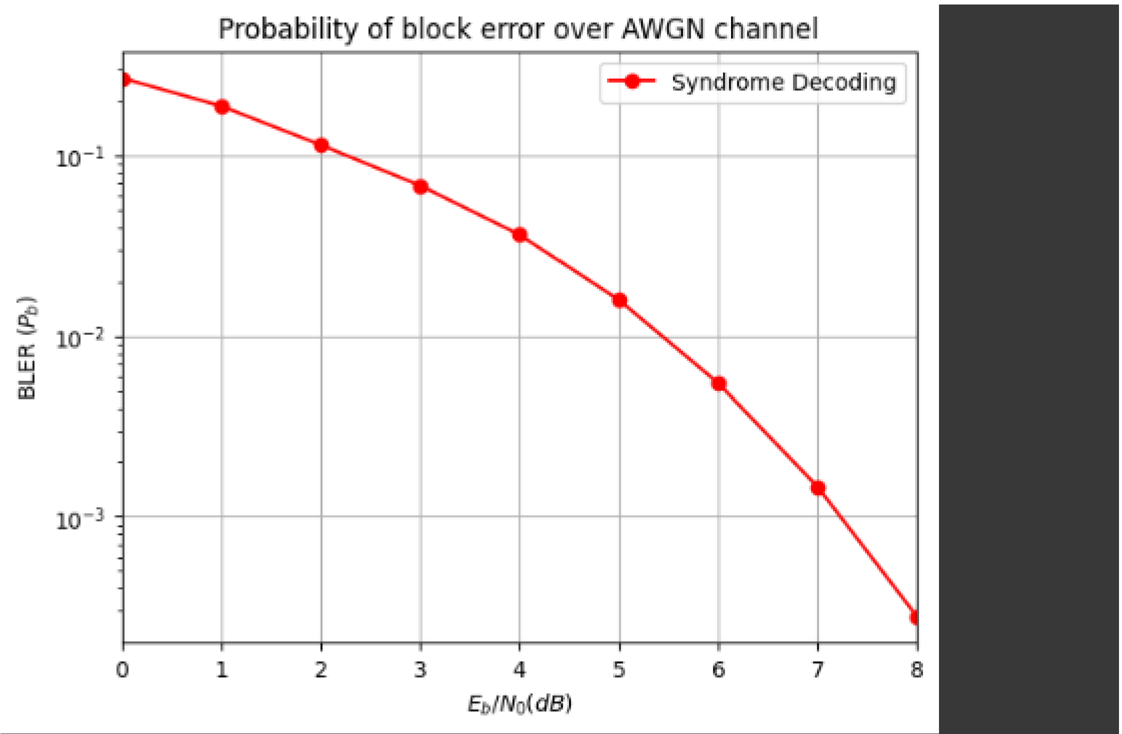


It can be observed that DNN (Deep Neural Networks) is an efficient method to decode, and its accuracy is between Syndrome decoding and ML decoding. Just like TA mentioned in the course, it is because that ML decoding is theoretically optimal, which computes the likelihood of all possible codewords and selects the most probable one, leading to the lowest error rate. And the effectiveness of DNN decoding is influenced by the amount and quality of training data and model architecture. If trained well, it can approach ML performance. As for Syndrome decoding, it relies on algebraic error correction rather than probability-based optimization, making it less effective in reducing errors. Hence, DNN is more accurate than Syndrome decoding, and it can enhance its performance by improving its layer structures. In conclusion, ML decoding provides the best performance, while DNN decoding is promising but not yet superior to ML. Traditional Syndrome decoding is the least effective, making it less suitable for high-performance communication systems.

2、Description and discussion of all decoding method in Module 1

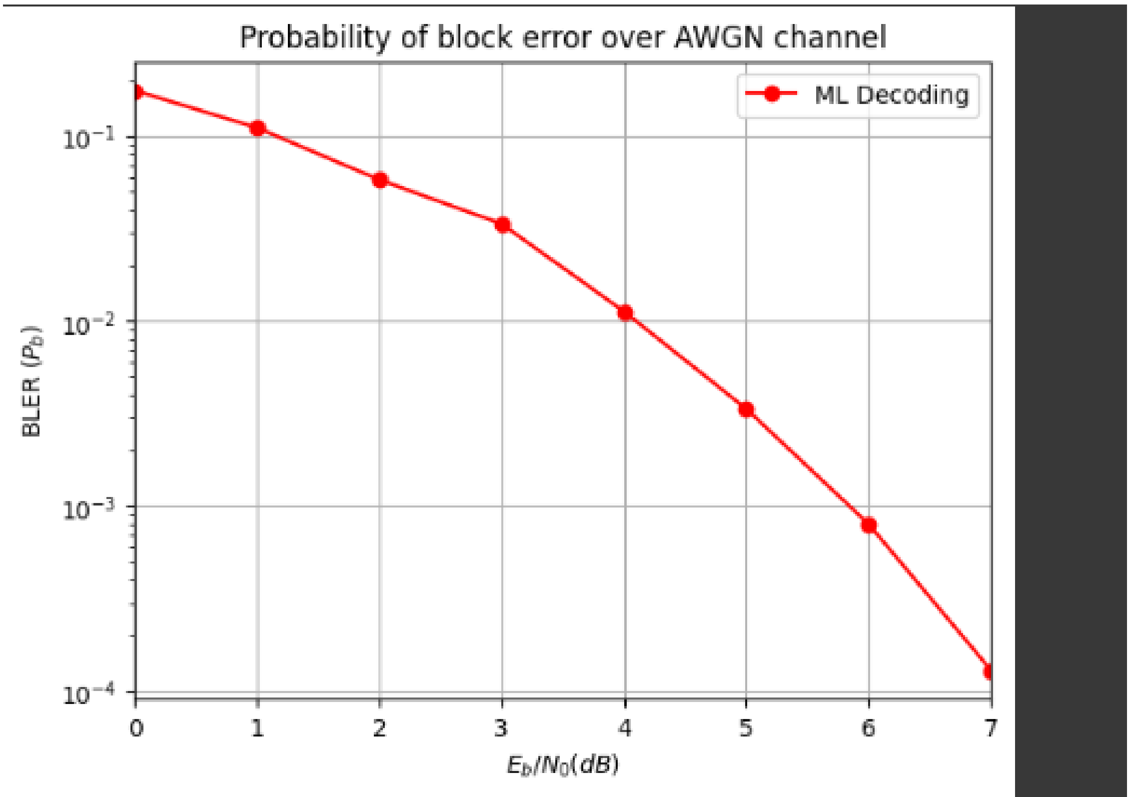
The following is the BLER result for all decoding methods in Module 1：

- Syndrome Decoding




```
Symbol Error Rate: [0.26725  0.187     0.114     0.068     0.036375 0.01578  0.005475 0.00147
 0.000282]
```

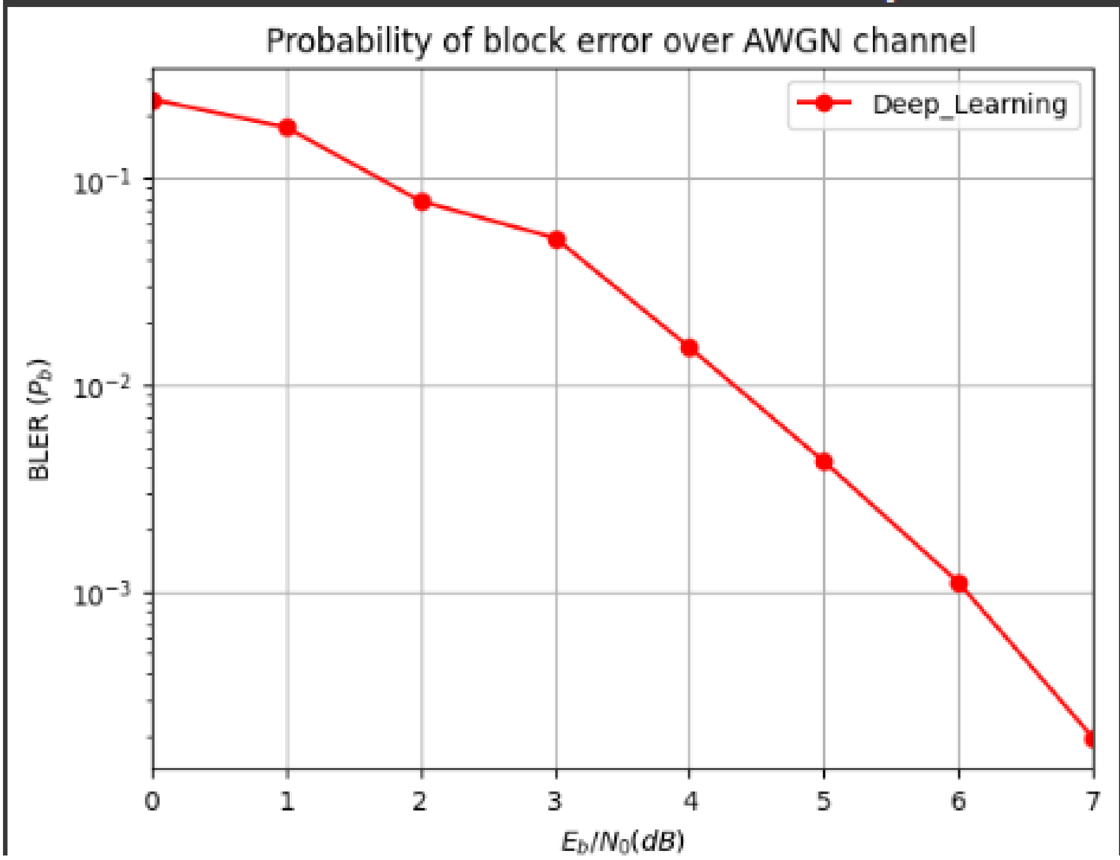
- ML Decoding



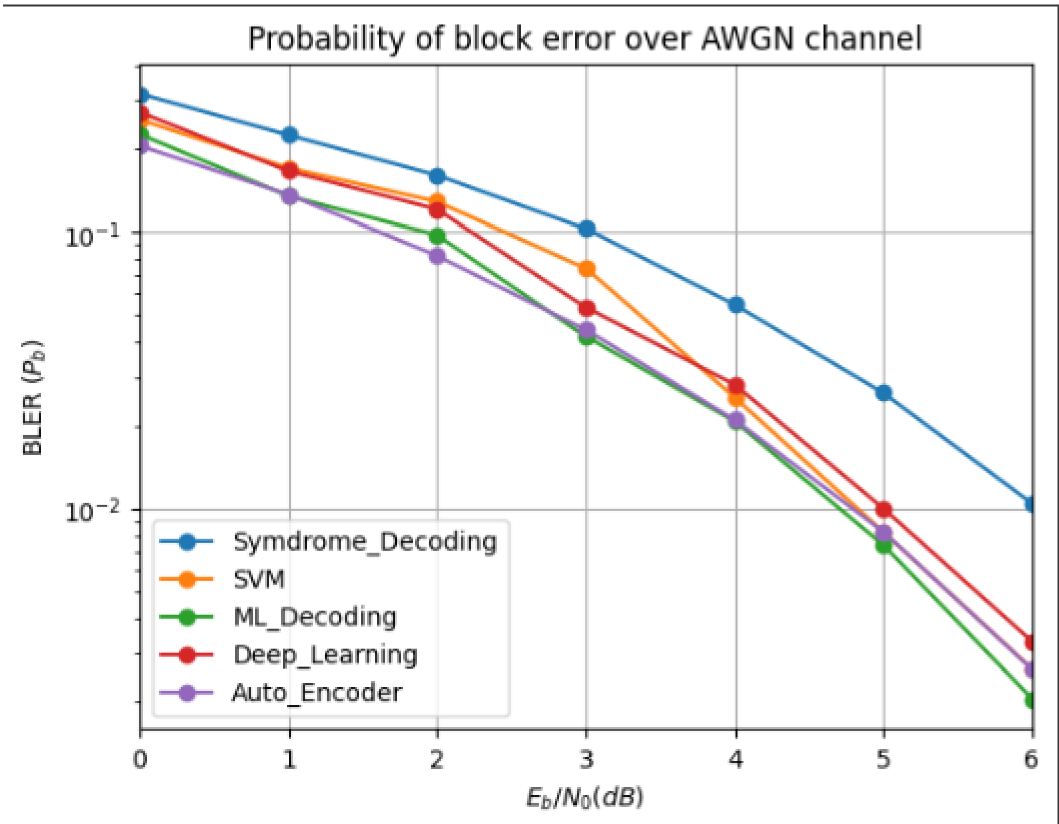

```
Block Error Rate: [1.7625e-01 1.1075e-01 5.8000e-02 3.3500e-02 1.1275e-02 3.3550e-03
 8.0000e-04 1.3000e-04]
```

- Deep Learning

```
Block Error Rate: [2.41428571e-01 1.76428571e-01 7.75000000e-02 5.14285714e-02
 1.51785714e-02 4.25714286e-03 1.10714286e-03 1.96428571e-04]
```

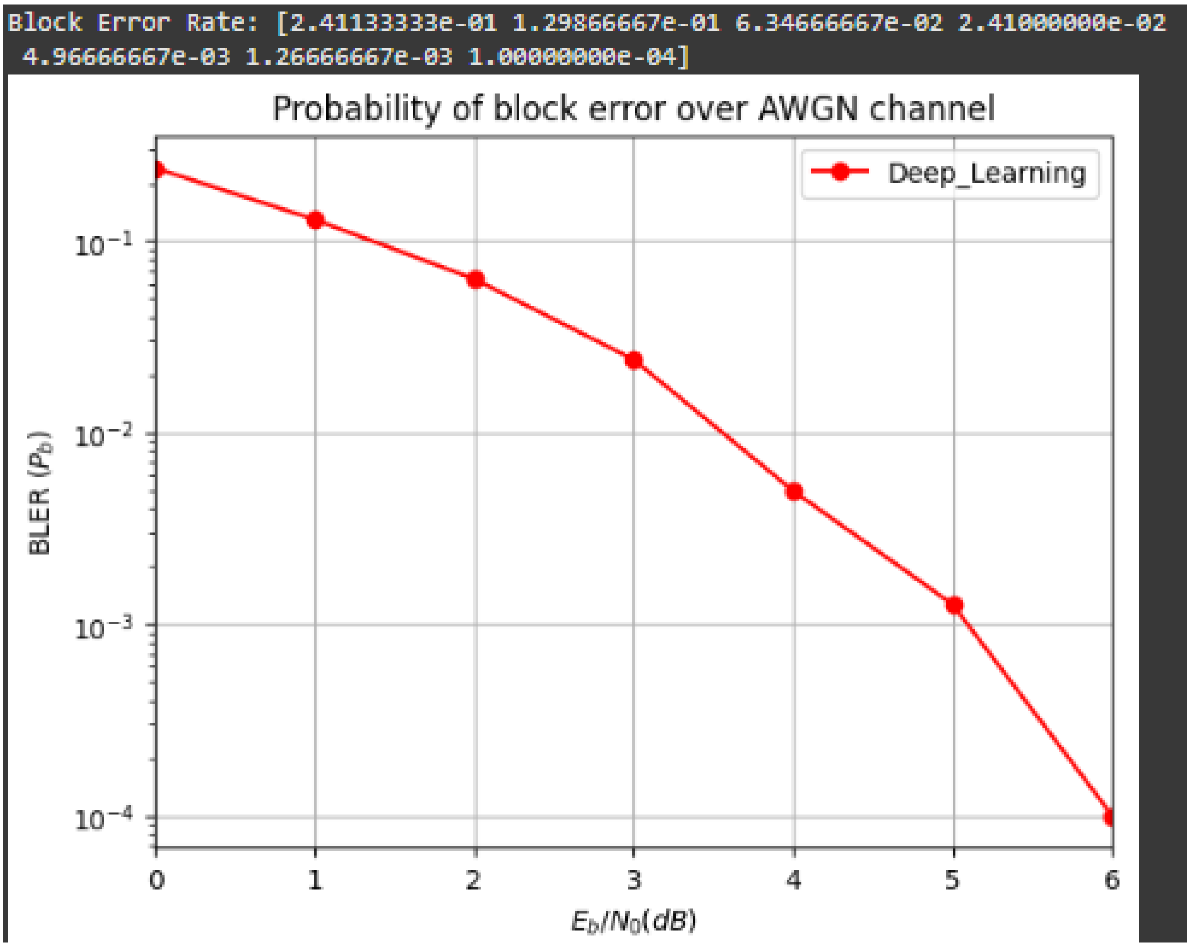


- Auto Encoder




```
Block Error Rate: [0.20456667 0.13631667 0.08218333 0.04438333 0.02095     0.00823333
 0.00261667]
```

- DNN



(The name of Deep_Learning curve is kept the same as it is in the lab)

Here's a description of each decoding method shown in the graph and their mechanisms:

1. <u>Syndrome Decoding</u>
    - A traditional error-correction technique used with linear block codes
    - Works by computing the syndrome of the received codeword, which identifies the error pattern
    - Uses lookup tables or algebraic methods to map syndromes to error patterns
    - Computationally efficient for smaller codes but becomes impractical for large block lengths
    - It may be more inaccurate in comparison to other decoding methods
2. <u>ML Decoding (Maximum Likelihood Decoding)</u>
    - A decoding principle that selects the codeword closest to the received signal
    - Makes decisions by finding the codeword with the highest probability of having been transmitted
    - Optimal in terms of minimizing error probability
    - Often computationally intensive, especially for long codes
    - Since it directly counts the error and Euclidean distance between the received signal and each of the codewords, it may take more time to decode for larger data.

3. Deep Learning

- Use deep neural networks to learn decoding functions
- Typically employs fully connected or recurrent neural networks
- Trained on examples of codewords and their noisy versions
- Can adapt to different channel conditions through training
- More general approach than DNN, possibly using various neural architectures

4. Auto Encoder

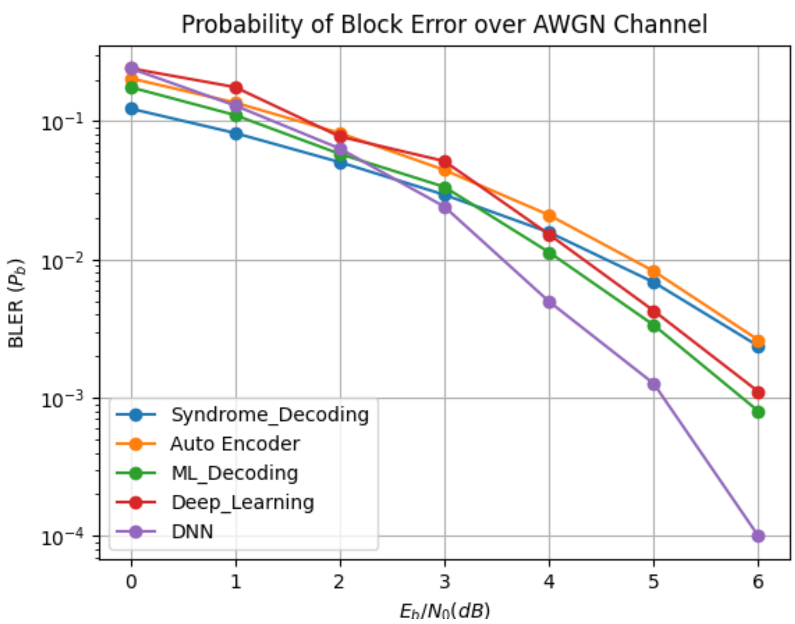
- An end-to-end communication system based on neural networks
- Both encoder and decoder are implemented as neural networks and trained jointly
- The encoder learns to map messages to codewords optimized for the channel
- The decoder learns to recover the original message from potentially corrupted codewords
- Doesn't require explicit code design but learns representations directly from data

5. DNN (Deep Neural Network)

- A specific implementation of deep learning for decoding
- Uses multiple layers of neurons to learn complex mappings
- Often specialized for codes or channel models
- May incorporate specific architectural elements like attention mechanisms or residual connections
- Need to be defined under different circumstances

While traditional methods like Syndrome decoding perform adequately, neural network-based approaches (especially DNN) achieve superior error correction performance at higher signal-to-noise ratios. This suggests that learning-based methods can discover more effective representations and decision boundaries than analytically designed approaches, particularly in challenging noise conditions. However, when it comes to accuracy, ML decoding is still the best solution to decoding.

By combining all the BLER data in the previous labs, we can conclude the above results：



We can observe from the graph that Syndrome decoding and Auto Encoder are not as good as other methods. The former is due to its error pattern may be not that accurate, and the latter one is that there will be two process of encode and decode, which may cause higher error rate and problems in gradient.

As for Deep learning and DNN, it is better than the two methods mentioned above. Since it can be designed in consideration of customized issues, it can largely increase performance on error reduction. Though it seems like DNN is even better than ML decoding, it is because that the data in DNN lab is easier to decode in comparison to those in ML decoding. According to what I've looked up, ML is the theoretically best solution to decoding. Even customized DNN is not that easy to surpass.

The performance of Syndrome decoding and Auto encoders is relatively weaker compared to others like Deep Learning (DNN) or Maximum Likelihood (ML) decoding. Syndrome decoding can be less effective due to its error pattern and its reliance on the error-correcting code's structure, which may not always be accurate for certain error patterns. This can lead to suboptimal error correction, especially when the channel noise is high, or the code isn't designed to handle specific types of noise efficiently.

Autoencoders, while useful in many machine learning applications, can struggle with the issue of two encoding and decoding processes. The error correction might not be as robust because the process is trained in a way that does not necessarily optimize it for the same type of noise encountered in a communication channel. Additionally, the backpropagation gradient might not flow as efficiently, leading to higher error rates during training.

DNN-based approaches tend to outperform syndrome decoding and autoencoders because they can be designed to specifically address the issues related to the noise and channel characteristics of a given system. These models can be customized for error reduction in a way that improves decoding performance.

In some scenarios, DNNs may outperform Maximum Likelihood decoding (ML), but this is often due to the easier-to-decode nature of the data used in the DNN approach. However, the ML decoding method remains the theoretically optimal solution. Even highly customized DNN models are not always able to surpass the performance of ML decoding because ML considers the complete likelihood of all possible codewords and finds the most probable one, making it the ideal choice in terms of error minimization.

In conclusion, while deep learning methods, especially DNNs, can provide significant improvements over traditional methods like syndrome decoding and autoencoders, Maximum Likelihood decoding remains the best theoretical approach for decoding in terms of error reduction and reliability. However, it may be less efficient for large and specific data.