Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Eléa VELLARD
Advanced Statistical Inference
EURECOM
elea.vellard@eurecom.fr

1 Introduction

Uncertainty estimation is a key limitation of standard deep learning models, which typically provide point predictions without indicating confidence, which could be critical in high-stakes areas like healthcare. In 2016, Gal and Ghahramani proposed a practical solution by reinterpreting dropout as approximate Bayesian inference and allowing uncertainty estimation while using standard architectures and with minimal changes.[1]

In this project, we aim to understand this method and reproduce key results from the paper to observe how it is implemented in practice. Through various experiments, we evaluate the method's ability to provide both accurate predictions and meaningful uncertainty estimates.

The code and results are available at `https://github.com/elea-vellard/ASI-project_VELLARD_CHAROLOIS.git`.

2 Context and theoretical foundations

2.1 Context and challenges

Deep neural networks typically return only a single predicted value for each input, without any built-in assessment of how certain or uncertain the prediction is. As a result, they can be very miscalibrated, confidently assigning high probability to incorrect labels or regressing to extreme values on data that lie outside their training distribution. In critical applications like medical diagnosis or autonomous navigation, such errors can have direct consequences, enforcing the need for models that not only predict accurately but also quantify their own uncertainty.

Many Bayesian methods exist to quantify uncertainty, but they often scale poorly to large networks. In their paper "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning", Gal and Ghahramani aimed to establish how dropout can serve as an efficient and simple approximate Bayesian inference method to represent uncertainty in any large neural networks.

2.2 Key contributions of the paper

2.2.1 Context

The goal of Bayesian neural networks is to learn a probability distribution over possible neural networks by modeling the posterior distribution $p(\theta|X, Y)$ over the network given the observed data (X, Y).

However, most of the time, those distributions are intractable due to the high parameter space. To work around this, Gal and Ghahramani propose using dropout to define a much simpler variational distribution $q(\omega)$. Dropout is a regularization technique that randomly deactivates a subset of neurons

Preprint. Under review.

at each training step, effectively sampling from an ensemble of smaller networks. Specifically, dropout defines this distribution $q(\omega)$ as composed of multiple networks with randomly deactivated units.

Every time we apply dropout in training, we effectively draw one sample $\hat{\omega}$ from q. We then choose q so that it is as close as possible to the true posterior by minimizing the Kullback–Leibler divergence:

$$KL(q(\omega)||p(\omega|X,Y)) \tag{1}$$

Moreover, if you carry out this minimization using a single Monte Carlo sample per example, two familiar terms emerge naturally: a data fit term, which is just the usual training loss (e.g. mean squared error or cross-entropy) evaluated with dropout; and a KL penalty term, which can be interpreted as a standard weight-decay when you assume a simple Gaussian prior over weights.

Putting these together gives exactly the familiar dropout objective:

$$\mathcal{L}_{\text{dropout}} = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(x_i;\hat{\theta})) + \lambda \sum_j \|\theta_j\|^2, \tag{2}$$

Minimizing this loss is therefore equivalent to making our dropout-induced $q(\omega)$ as close as possible to the true posterior, thereby approximating Bayesian inference with no change to standard dropout training.

2.2.2 Uncertainty estimation using MC Dropout

Another key contribution of the paper is using MC Dropout to estimate uncertainty: unlike standard practice which disables dropout at test time, keeping it active during inference allows to capture uncertainty. Concretely, we perform T stochastic forward passes, each with a different dropout mask $\hat{\theta}_t$, and use these samples $f(x^*;\hat{\theta}_t)$ to approximate the predictive distribution. The predictive mean and variance are then estimated as:

$$\mathbb{E}[y^*|x^*] \approx \frac{1}{T}\sum_{t=1}^{T} f(x^*;\hat{\theta}_t), \tag{3}$$

$$\text{Var}[y^*|x^*] \approx \tau^{-1}I + \frac{1}{T}\sum_{t=1}^{T} f(x^*;\hat{\theta}_t)^2 - \left(\frac{1}{T}\sum_{t=1}^{T} f(x^*;\hat{\theta}_t)\right)^2, \tag{4}$$

where τ is the model precision. The $\tau^{-1}I$ term captures irreducible data noise, while the rest of the variance expression reflects the model's uncertainty, which diminishes as more data are used in training. This simple procedure lets MC Dropout distinguish between high-confidence and low-confidence regions without any modifications to the network architecture.

3 Experiments and reproduction of the paper's results

3.1 Experiment 1: MC Dropout for regression and uncertainty visualization

3.1.1 Experiment overview

Our first experiment is designed not to replicate a specific result from the paper but rather to familiarize ourselves with MC Dropout and demonstrate how it can estimate predictive uncertainty. We use the California Housing dataset to predict home prices, a regression task that allows us to visualize predictive uncertainty.

We define a simple feed-forward neural network with two hidden layers of 100 units each, ReLU activations, and a dropout rate of 0.1. We train for 2,000 epochs using the Adam optimizer (learning rate 0.01) and mean squared error loss. At inference time, we keep dropout active and perform 100 stochastic forward passes for each input. From these samples, we compute the predictive mean and variance, which yield our uncertainty estimates.

For visualization, we produce two plots:

- Predictions with uncertainty bands: Predicted prices for 300 test samples as well as the $\pm 2\sigma$ uncertainty band, σ being the estimated standard deviation (see figure 1).
- Uncertainty distribution: a histogram of the standard deviations across all test predictions, showing how uncertainty is distributed and the overall confidence of the model (see figure 2).

3.1.2 Experiment results

The model achieved an MSE of 0.58 on the test set, indicating strong predictive performance.

In figure 1, each test point has its own uncertainty band. Wider bands correspond to higher predictive uncertainty, demonstrating how MC Dropout adapts its confidence per sample.

Figure 2 shows that most predictions carry low uncertainty, while a small number has notably higher variance, probably due to unclear inputs. This confirms that the model's uncertainty estimates are meaningful and that the model can use this confidence score to make relevant predictions.

3.2 Experiment 2: Model Uncertainty in regression tasks

3.2.1 Experiment overview

The purpose of this experiment is to reproduce the paper's regression results, and evaluate how well MC Dropout captures predictive uncertainty in extrapolation tasks. We use the Mauna Loa CO_2 dataset to train and compare three approaches:

- A standard neural network trained with dropout but evaluated in its usual mode as a baseline;
- A classic Gaussian Process, commonly used for its principled uncertainty estimates;
- Two MC Dropout networks, one with ReLu activations and one with Tanh, where dropout remains active at inference. We perform multiple stochastic forward passes to sample each network's predictive distribution.

Using a simpler experiment setup than in the paper, each neural network has two hidden layers of 100 units and uses a dropout rate of 0.1. Both were trained for 2,000 epochs with the Adam optimizer, and uncertainty was estimated through 100 stochastic forward passes (instead of 1000 in the paper). For comparison, we also trained a Gaussian Process with an RBF kernel as a reference for predictive uncertainty. Additionally, we applied a single split (75/25) of the Mauna Loa CO_2 data instead of multiple runs or repeated experiments as done in the paper.

By comparing MC Dropout's uncertainty bands and those from the Gaussian Process on unseen data, we can assess the quality of MC Dropout uncertainty estimate under different non-linearities and understand how it evolves as we move further from the training data.

3.3 Experiment results

Results are shown in figure 3: in figure (a), the standard dropout model simply extrapolates into regions without any training data and offers no uncertainty estimates, whereas the Gaussian Process in (b) produces smooth predictions with progressively bigger confidence bands beyond the observed range, which was expected: the uncertainty rises when we get far from the observed data.

Figure (c) display MC Dropout with ReLu activations. It shows that uncertainty rises in the extrapolation zone, and results are directly comparable with the Gaussian Process uncertainty estimate: the further we go from the training data, the bigger the uncertainty gets. This proves the quality of MC Dropout.

Finally, figure (d) shows how Tanh activation models behave when extrapolating. The figure shows the saturation nature of the model: since tanh saturate quickly around ± 1 and the model was not trained to extrapolate (but rather to interpolate), and the predictions are almost constant. In addition, uncertainty also remains bounded as we move further from the training data.

Overall, this experiment was successfully reproduced compared to the paper results, showing both how accurate MC Dropout is compared to existing but computationally intensive methods, and also how uncertainty estimate correctly rises when moving away from observed data.

3.4 Experiment 3: Model uncertainty in classification tasks

3.4.1 Experiment overview

The goal of this experiment is to evaluate how MC Dropout behaves on a classification task. To do this, we will use the MNIST dataset and progressively rotate the image of one of the digits to create ambiguous inputs.

For this task, we train a LeNet-Style convolutional neural network on the MNIST dataset, following the setup from the original paper. We apply dropout with a probability of 0.5, and then train the model over 1,000,000 iterations with a learning rate of 0.0001 and a decay factor of 0.75. To assess uncertainty, we selected a sample image of the digit "1" and applied varying degrees of rotation from 0° to 180°. We then performed 100 stochastic forward passes per angle to analyze the predictions. We then analyze how the predicted digit changes as the image is rotated (visualized through the scatter of softmax inputs, see figure 4, and the shape of the softmax output distribution (see figure 5)), which reflects the model's uncertainty across different rotation angles.

3.4.2 Results

As shown in both plots, the model confidently and correctly predicts the digit "1" at low rotation angles, but the uncertainty grows as the image is rotated. At higher angles, it starts misclassifying the digit (cf. figure 4) (we only represented classes '5' and '7', whose probabilities start to increase), and softmax outputs become more dispersed (cf. figure 5). Despite a simpler experiment setup and fewer stochastic passes, the behavior of our experiment matches the paper's results: MC Dropout effectively captures uncertainty in classification tasks, allowing us to visualize when the model is confident or not in its predictions.

3.5 Experiment 4: MC Dropout in Reinforcement Learning

3.5.1 Experiment overview

For our final experiment, we chose to reproduce the reinforcement learning (RL) results from the original paper. The aim is to demonstrate how MC Dropout's uncertainty estimates can improve the trade-off between exploration and exploitation in RL tasks. While the paper uses a more complex 2D environment involving a 9-eyed robot, we implement a simpler setup that is easier to reproduce: a 20-arm Bernoulli bandit.

In this environment, there are 20 arms, each associated with a fixed but unknown reward probability. The agent's objective is to identify and exploit the most rewarding arms over time. Figure 6 represents the 20-arm bandit environment, showing the reward probability associated with each arm. To estimate the success probabilities, we use a simple neural network with one hidden layer and apply dropout during training and inference.

We compare two exploration strategies:

- Epsilon-Greedy policy, which selects random actions with a fixed probability, regardless of uncertainty;
- Thompson sampling with MC Dropout, which uses uncertainty estimates from multiple stochastic forward passes to guide exploration more intelligently.exploration.

Results are averaged over three runs using the same environment, and we track the evolution of the average reward (computed every 500 steps) throughout the training process.

3.6 Results

The log plot in figure 7 in the appendix shows that Thompson Sampling with MC Dropout reaches higher rewards more quickly, converging around 900 steps compared to 2000 for the Epsilon-Greedy policy. By leveraging uncertainty estimates, MC Dropout guides exploration more intelligently, focusing on uncertain actions rather than exploring randomly, which results in a better balance between exploration and exploitation. Although both approaches eventually reach the same average reward, MC Dropout leads to significantly faster convergence.

4 Conclusion

Through this work, we explored both the theoretical and practical aspects of MC Dropout. From a theoretical perspective, we understood how it can be used as an approximate Bayesian inference method to estimate uncertainty. Practically, we saw how it can be applied across different tasks to estimate uncertainty and guide predictions.

Aiming to reproduce the paper's results in a simplified way, we were able to visualize how MC dropout can be used in regression and classification tasks, as well as offering better exploration in RL tasks, which demonstrates how powerful this method is. With more time, future work could involve applying it to larger models for a better assessment of the quality of the method. Personally, I would be particularly interested in expanding the RL experiment and comparing it with other exploration techniques.

References

[1] Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of the 33rd International Conference on Machine Learning*, 1050–1059.

5 Appendix

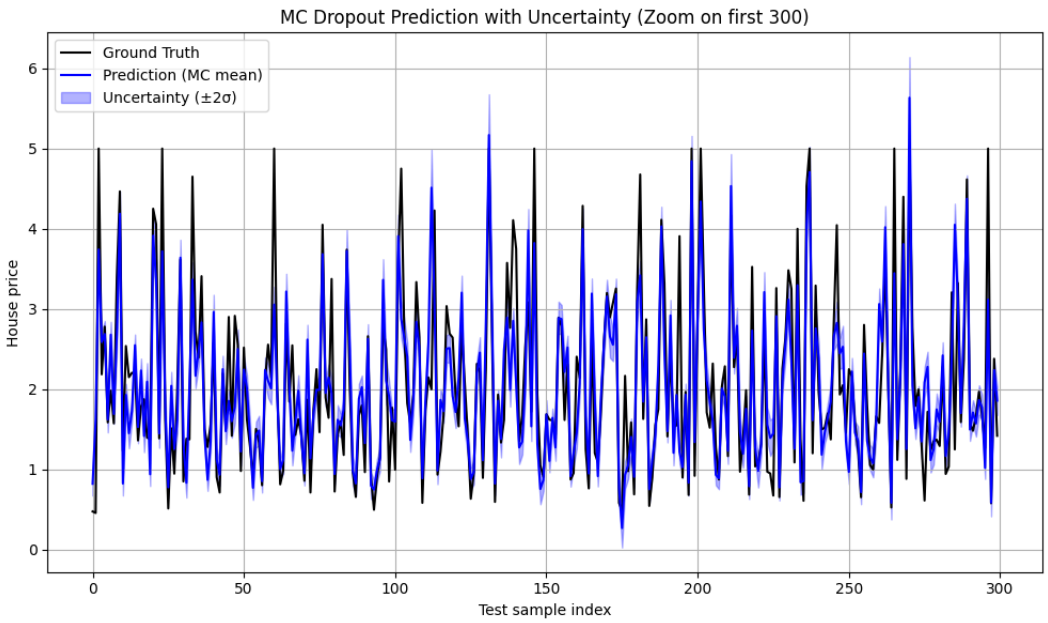

Figure 1: MC Dropout predictions (blue) vs. ground truth house prices (black), with $\pm 2\sigma$ uncertainty bands showing confidence levels across samples. This figure shows how local prediction confidence varies across samples.

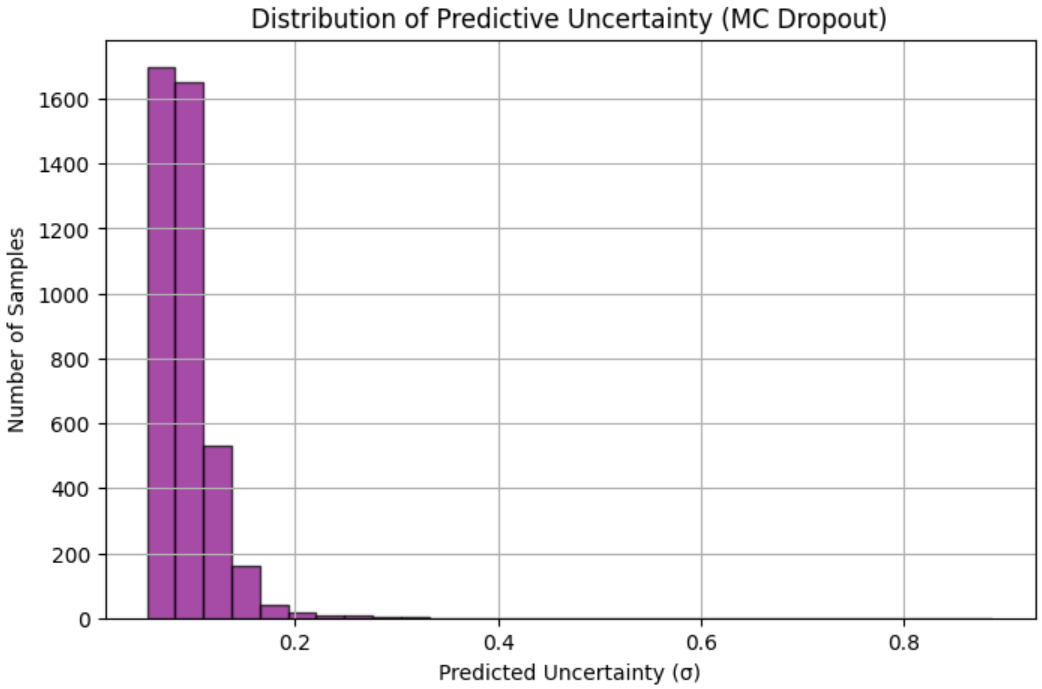

Figure 2: Distribution of predictive uncertainty across the test set, measured by standard deviation (σ). The majority of predictions have low uncertainty values, while a few samples show higher uncertainty, exposing the model's overall confidence.

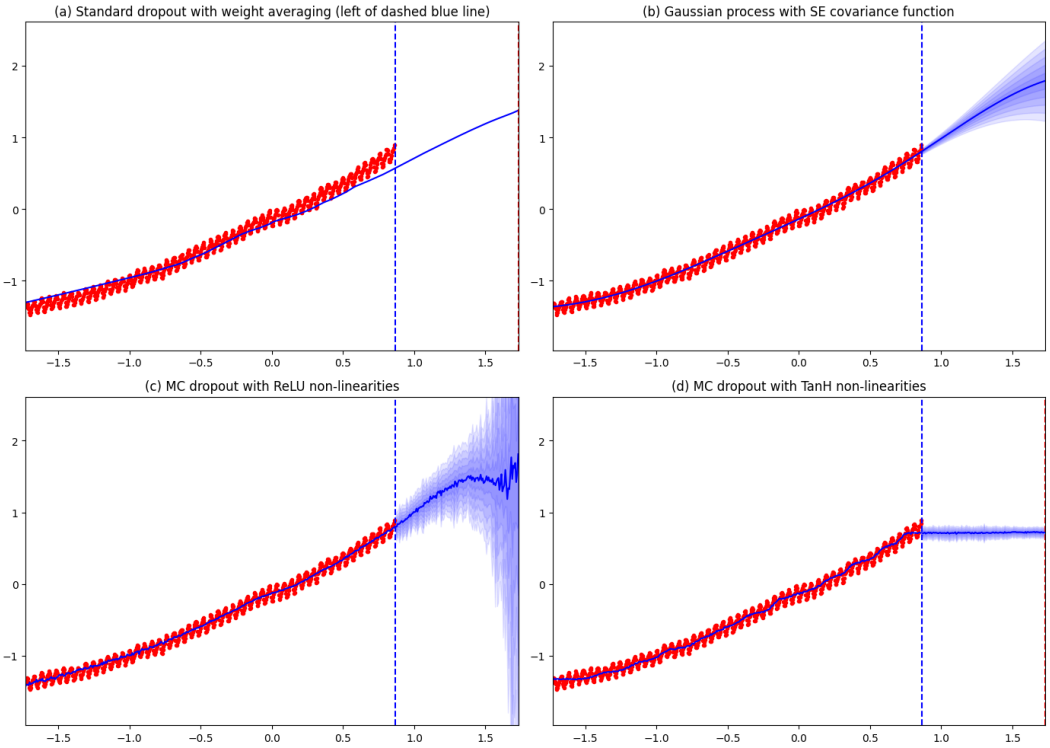

Figure 3: Visual comparison of model behavior beyond the training data. While the standard dropout model (a) does not provide uncertainty estimate (to serve as a reference), the Gaussian Process (b) and MC Dropout models with ReLU (c) and Tanh (d) activations show increasing uncertainty, each reflecting the model's ability to extrapolate differently.

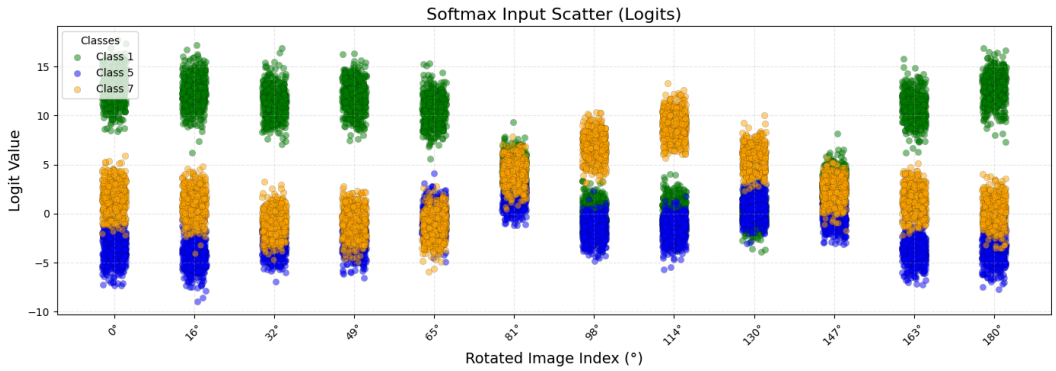

Figure 4: Softmax input scatter: raw class scores (logits) distribution for digits 1, 5, and 7 across increasing rotation of a digit image. For low and high rotation angles, the predicted class is clearly separated from other classes, indicating strong strong confidence. For intermediate rotation angles, classes predictions overlap, showing growing prediction uncertainty.

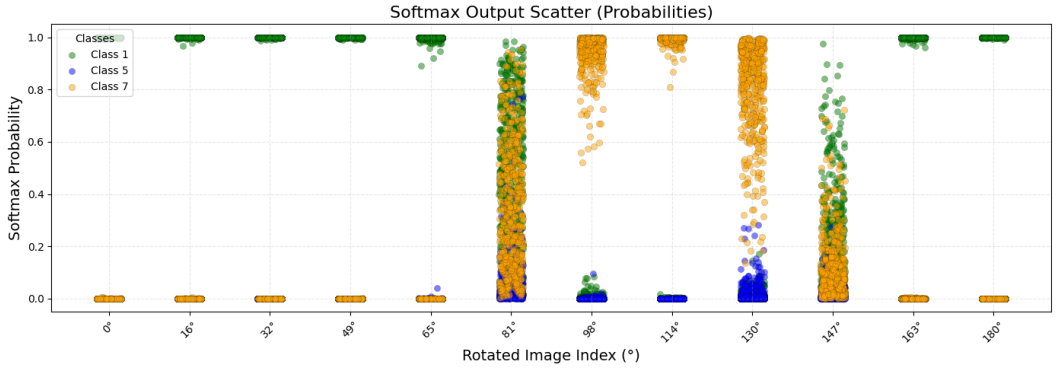

Figure 5: Softmax output scatter: probabilities for digits 1, 5, and 7 across increasing rotation of a digit image. For low and high rotation angles, probabilities for the predicted class are very high and clearly separated from other classes, indicating strong strong confidence. For intermediate rotation angles, classes probabilities are more scattered and overlap, showing growing prediction uncertainty.

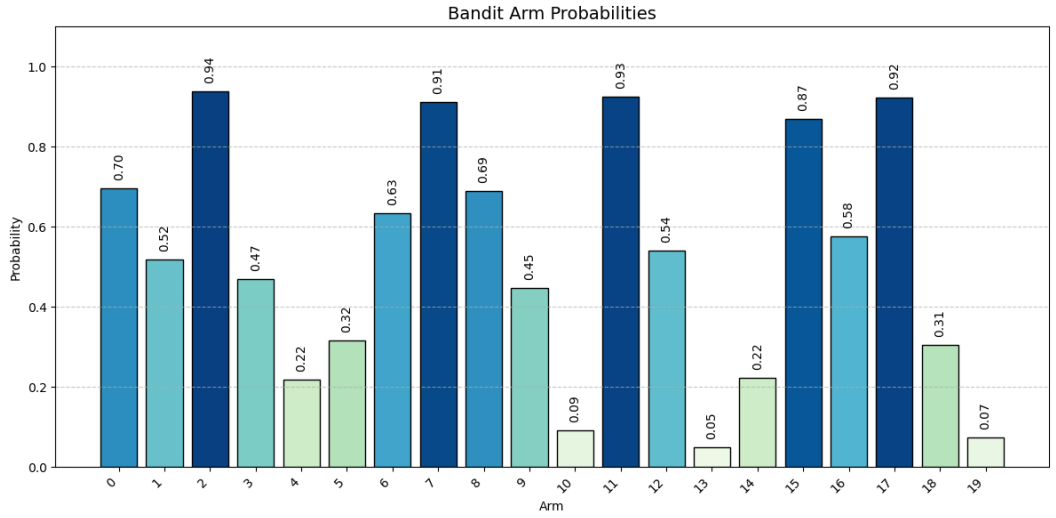

Figure 6: Reward probability distribution in the 20-arm bandit environment for the reinforcement learning experiment. Each bar represents the reward probability of one arm. The agent must learn to retrieve high probability arms to optimize reward.

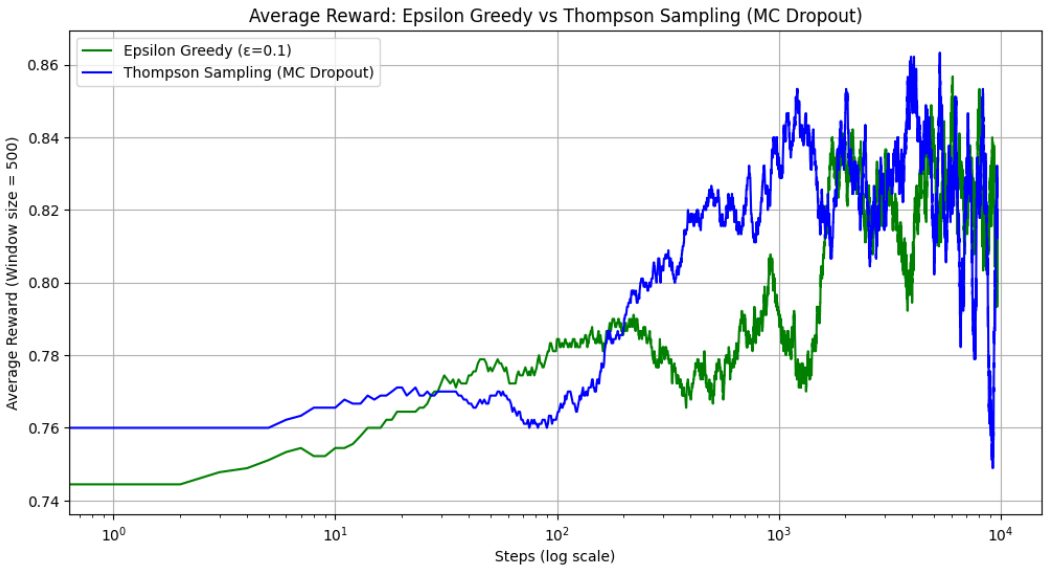

Figure 7: Evolution of average reward across training steps. The average is computed every 500 steps. Comparison between Epsilon-Greedy policy and Thompson Sampling with MC Dropout, which estimates uncertainty for each action. Thompson Sampling converges more quickly to higher rewards by identifying the best arms earlier, showing the power of MC Dropout for exploration in reinforcement learning.