Université Paris-Dauphine – PSL
Centre de Recherche en Mathématiques de la Décision

DROUPOUT REDUCES UNDERFITTING
EXPERIMENTAL VALIDATION OF THE ORIGINAL CLAIMS

Arthur Danjou - Alexis Mathieu - Axelle Meric
Philippine Quellec - Moritz Von Siemens

Course Supervisor:
Théo LOPES-QUINTAS
BPCE Payment Services

Deep learning
ACADEMIC YEAR 2025/2026

Contents

Introduction

This project is based on the research paper *"Dropout Reduces Underfitting"* by Liu et al. [1]. As more and more data are available and many advances have been made in sectors such as data augmentation, this suggests that underfitting may become a more common issue than overfitting in future training settings. This paper presents an alternative use of dropout to mitigate underfitting when used at the start of the training. We begin our analysis by briefly presenting the key concepts introduced in the paper. We then conduct our own experiments to evaluate the validity and generality of the author's claims. To do so, we reproduce their ideas on a smaller scale using several datasets of different sizes (such as `MNIST`, `FashionMNIST`, `CIFAR-10` and `CIFAR-100`), and we study how the effect of dropout varies with dataset complexity, model capacity, and training regime. This approach allows us to analyze to what extent the trends observed in the original paper hold in a reduced experimental setting. The complete implementation of our experiments can be accessed at:

`https://github.com/ArthurDanjou/DropoutReducesUnderfitting/`.

1 Definitions

Before doing any experiment, we briefly define the concepts we will be using based on the definition given in the paper [1] and in the course of Deep learning given by Théo Lopes-Quintas for the second year of master's degree ISF at Paris Dauphine-PSL [2].

Overfitting Overfitting happens when a model learns the training data too precisely, including noise or irrelevant details. It performs very well on the samples it has seen but struggles with new data because it has not learned general patterns. This typically occurs when the model is too complex or trained for too long without sufficient regularization.

Underfitting On the contrary, underfitting occurs when a model is too limited to capture the essential structure of the data. It performs poorly on both training and test sets because it has not learned enough from the available information. This can result from an overly simple model, insufficient training, a dataset so large that the model cannot fully adapt to it or a dataset so small that the model can't adequately recognize all the available information.

Dropout Dropout is a method introduced in 2014 by five computer scientists with a single objective: preventing neural networks from overfitting [3]. During each training iteration, a random subset of neurons is dropped from selected layers with a given probability. As a result, the model is effectively trained on a different sub-network at each step, which helps reduce overfitting since the data cannot consistently rely on the same computational pathways.

Drop rate The drop rate is the probability that a neuron is deactivated during training when applying dropout. It determines how much of the network is randomly ignored at each iteration. A higher drop rate increases the regularization effect by removing more units, while a lower rate keeps most neurons active and weakens this effect. In practice, the drop rate controls the intensity of dropout and influences how strongly the model is encouraged to learn more robust representations. It will be a hyper-parameter of our study.

Epoch An epoch corresponds to one complete pass over the entire training dataset. The number of updates performed during an epoch depends on the batch size: smaller batches result in more gradient updates per epoch, while larger batches reduce the number of updates but provide smoother gradient estimates. Choosing the right combination of epochs and batch size is important, as it influences both the speed of training and the model's ability to generalize.

Switch_epoch Switch_epochs refers to the moment during training when the dropout schedule changes, typically from an active dropout phase to a phase without dropout, or vice versa. This parameter controls when the model transitions between different regularization regimes.

2 Implementation and Software Architecture

To accurately reproduce the findings of Liu et al. [1] and overcome the static graph compilation limitations of standard deep learning frameworks, we develop a custom modular pipeline using `TensorFlow/Keras`. This section details the software design patterns and the neural network architectures employed.

2.1 Software Design Pattern

Standard `Keras` callbacks typically modify hyperparameters between epochs using Python variables. However, in `TensorFlow`'s graph execution mode, dropout rates are often frozen at compilation time. To enable real-time modulation of the dropout rate on the GPU without costly model recompilation, we implement a dynamic architecture based on three interconnected components:

1. **The `DynamicDropout` Layer:** A custom layer inheriting from `keras.layers.Layer`. Instead of a static float value, this layer references a shared `TensorFlow` variable (`tf.Variable`). During the forward pass, it reads the current value of this variable, allowing the dropout rate to be updated instantly across the entire network.

2. **The `DropoutScheduler` Callback:** This component acts as the logic controller. At the beginning of each epoch, it updates the shared variable according to the selected strategy:
   - *Early Dropout:* Active initially, then switched to 0.
   - *Late Dropout:* Inactive initially, then activated.
   - *Standard Dropout:* Constant rate throughout training.
   - *No Dropout:* Control experiment without Dropout.

3. **The `ExperimentPipeline` Orchestrator:** A high-level class that encapsulates data loading, preprocessing, model factory, and training loops. It ensures that all comparisons are performed under identical initialization and data split conditions.

2.2 Neural Network Architectures

We evaluate the strategies on two distinct architectures to assess robustness across different data modalities.

2.2.1 Dense Network (MLP)

For simple data structures (e.g., MNIST), we use a Multi-Layer Perceptron. As detailed in Table 1, it consists of three dense blocks with decreasing capacity.

Layer	Configuration	Activation
Input	Flatten ($28 \times 28 \rightarrow 784$)	-
Dense 1	256 units	ReLU
Dropout	`DynamicDropout` (variable p)	-
Dense 2	128 units	ReLU
Dropout	`DynamicDropout` (variable p)	-
Dense 3	64 units	ReLU
Dropout	`DynamicDropout` (variable p)	-
Output	$N_{classes}$ units	Softmax

Table 1: Architecture of the Dense Network (MLP).

2.2.2 Convolutional Neural Network (CNN)

For complex image datasets (`FashionMNIST`, `CIFAR-10/100`), we utilize a CNN to capture spatial hierarchies. Crucially, dynamic dropout is applied after feature extraction blocks to regularize the learning of convolutional filters, as outlined in Table 2.

Layer	Configuration	Activation
Input	Shape ($Height \times Width \times Channels$)	-
Block 1	Conv2D (32, 3×3)	ReLU
	`DynamicDropout` (variable p)	-
Block 2	Conv2D (32, 3×3) + MaxPool (2×2)	ReLU
	`DynamicDropout` (variable p)	-
Block 3	Conv2D (64, 3×3)	ReLU
	`DynamicDropout` (variable p)	-
Head	Flatten + Dense (64 units)	ReLU
	`DynamicDropout` (variable p)	-
Output	Dense ($N_{classes}$ units)	Softmax

Table 2: Architecture of the Convolutional Neural Network (CNN).

3 Experimental Methodology

Our experimental suite is designed to systematically isolate and analyze the specific effects of Early Dropout compared to Standard and Late strategies.

3.1 Dual-Metric Visualization Strategy

A key methodological choice in this study is the systematic side-by-side visualization of **accuracy** and **loss** on both `train` and `validation` sub-datasets. Analyzing accuracy alone is insufficient for understanding the regularization dynamics. Our preliminary results showed that Early Dropout often yields higher accuracy but higher loss compared to Standard Dropout. This paradox indicates that while the model classifies more samples correctly, it may do so with lower confidence (probability calibration), a direct consequence of the noise injection during the early optimization phase. Observing both metrics allows us to distinguish between genuine generalization improvements and mere probability scaling.

3.2 Experimental Protocols

The pipeline supports four automated experiments, each targeting a specific hypothesis from the original paper:

1. **Temporal Dynamics Analysis (`compare_learning_curves`):**
   - *Objective:* Observe the immediate impact of switching off dropout.
   - *Method:* Training Standard, Early, and Late or No Dropout models in parallel.
   - *Hypothesis:* We expect to see a sharp drop in training loss immediately after the switch epoch for Early Dropout, indicating a "release" of the optimization constraints.
2. **Hyperparameter Robustness (`compare_drop_rates`):**
   - *Objective:* Test sensitivity to the dropout rate intensity (p).
   - *Method:* Varying $p \in \{0.2, 0.4, 0.6\}$.
   - *Hypothesis:* Early Dropout should remain effective even at high rates (e.g., $p = 0.6$) where Standard Dropout typically causes model collapse due to over-regularization.
3. **Phase Duration Sensitivity (`compare_switch_epochs`):**
   - *Objective:* Determine the optimal duration for the initial regularization phase.
   - *Method:* Varying the switch epoch (e.g., 5, 10, 15 epochs).
   - *Hypothesis:* Extending the early phase should monotonically improve final convergence up to a saturation point.
4. **Data Regime Analysis (`run_dataset_size_comparison`):**
   - *Objective:* Validate that Early Dropout specifically targets underfitting.
   - *Method:* Training on dataset fractions (e.g., 10%, 50%, 100%).
   - *Hypothesis:* Early Dropout should outperform Standard Dropout in data-scarce regimes (simulating underfitting), while Standard Dropout may regain the advantage when data is abundant (overfitting regime).

4 Experiments on datasets

In this section, we apply our experimental pipeline to four diffrent datasets available in the `Tensorflow` package: `FashionMNIST`, `MNIST`, `CIFAR-10`, and `CIFAR-100`. We present the dataset characteristics, justify our modeling choices, and analyze the results of the four distinct experiments to assess the validity of the claims made by Liu et al. [1].

4.1 FashionMNIST

We begin with the `FashionMNIST` dataset.

4.1.1 Dataset Presentation

`FashionMNIST` is a dataset comprising 28×28 grayscale images of clothing items, intended as a more challenging replacement for the original MNIST digit dataset.

- **Dimensions:** Inputs are tensors of shape $(28, 28, 1)$.
- **Content:** 10 classes (T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot).
- **Input/Output:** The network accepts flattened or convolutional feature maps derived from the pixel intensities (normalized to $[0, 1]$) and outputs a probability vector of size 10 via a Softmax layer.

4.1.2 Model Justification: Convolutional Neural Network (CNN)

While a Dense network (MLP) is sufficient for simple digit recognition, clothing items contain complex spatial dependencies (textures, shapes, edges) that are invariant to translation. Therefore, we select the **CNN architecture** for this task. Using a CNN allows us to test the Early Dropout strategy in a context where feature extraction is critical. Furthermore, CNNs on `FashionMNIST` can be prone to overfitting if not properly regularized, making them an ideal candidate for comparing Standard and Early Dropout strategies.

4.1.3 Analysis of Experiments

We conduct four targeted experiments to evaluate the performance and robustness of the dropout strategies.

1. Temporal Dynamics (Learning Curves)

Parameters:

- $Switch = 10$: the switch occurs at 33% of the training process
- $Rate = 0.4$: moderate dropout rate
- $Epochs = 30$

Analysis: As observed on Figure 1, the Standard Dropout (purple) converges smoothly. The Early Dropout (red) shows a distinct behavior: accuracy is comparable to the standard approach ($\approx$ 92.5%), but the validation loss remains significantly higher ($\approx$ 0.55 vs $\approx$ 0.20). This paradox—high accuracy with high loss—suggests that while Early Dropout classifies correctly, the aggressive initial noise results in a model that is less "confident" (higher entropy in probability distributions) or settled in a different basis of the loss landscape compared to standard regularization.

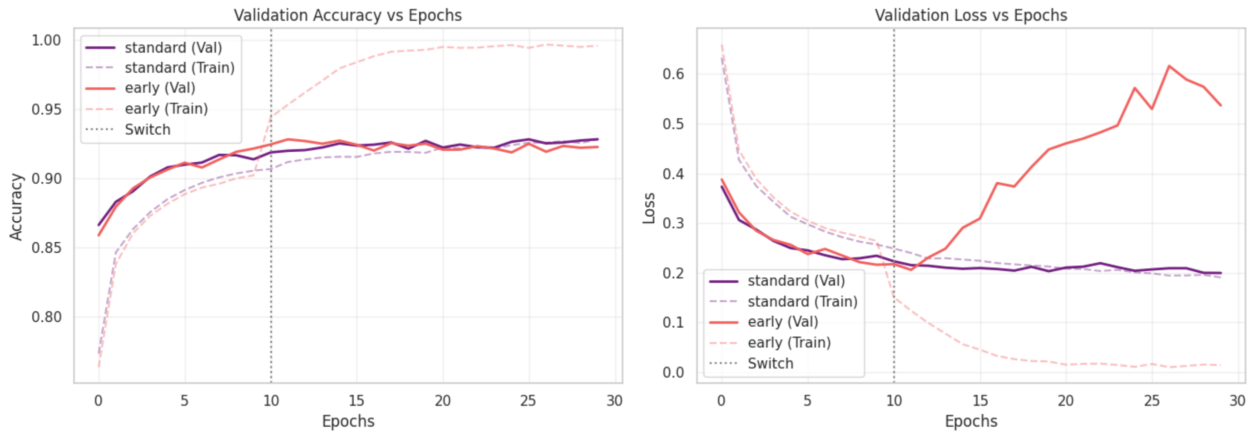

Figure 1: Learning Curves for Standard (Purple) vs. Early (Red) Dropout on FashionMNIST.

2. Hyperparameter Robustness (Drop Rate Ablation)

Parameters:

- $Switch = 10$
- $Rates \in \{0.2, 0.4, 0.6\}$: Testing extreme rates (0.6) is crucial to verify the paper's claim that Early Dropout prevents model collapse under heavy noise.
- $Epochs = 30$

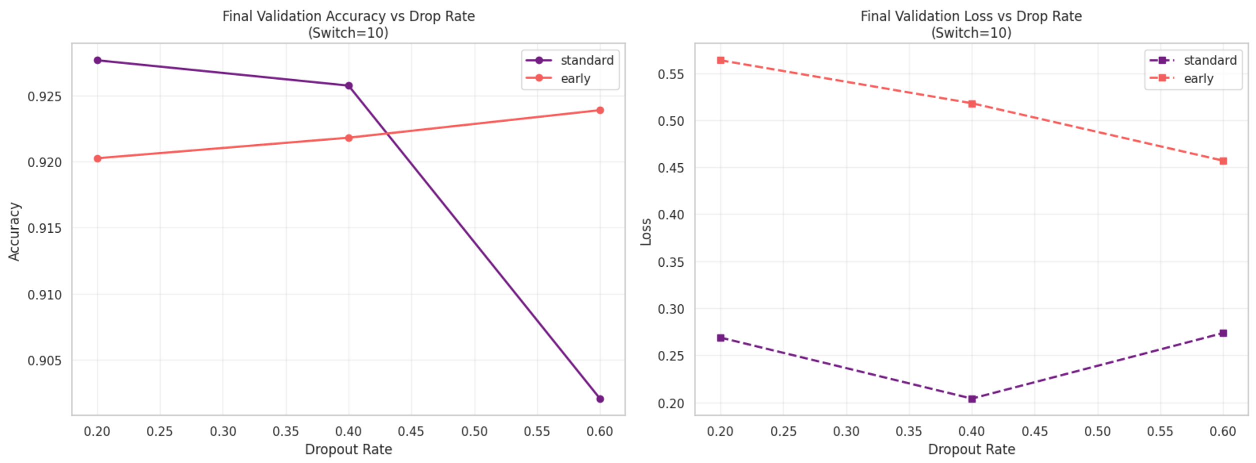

Figure 2: Impact of dropout Rate on Final Accuracy (left) and Loss (right).

Analysis: Figure 2 presents the strongest evidence validating the paper.

- **Standard Dropout (Purple):** Performance collapses at $p = 0.6$, dropping from $\approx 92.7\%$ to $\approx 90.2\%$. The model is over-regularized and fails to learn effectively.
- **Early Dropout (Red):** Conversely, performance *improves* as the rate increases, peaking at $p = 0.6$ ($\approx 92.4\%$).

This confirms that Early Dropout acts as a robust optimization aid: the heavy initial noise forces the learning of robust features, but turning it off allows the model to recover, whereas Standard Dropout permanently handicaps the model at high rates.

3. Phase Duration Sensitivity (Switch Epoch)

Parameters:

- $Switch \in \{5, 10, 15\}$: We vary the duration of the "Early" phase to find the optimal trade-off between noise injection and fine-tuning.
- $Rate = 0.4$
- $Epochs = 30$

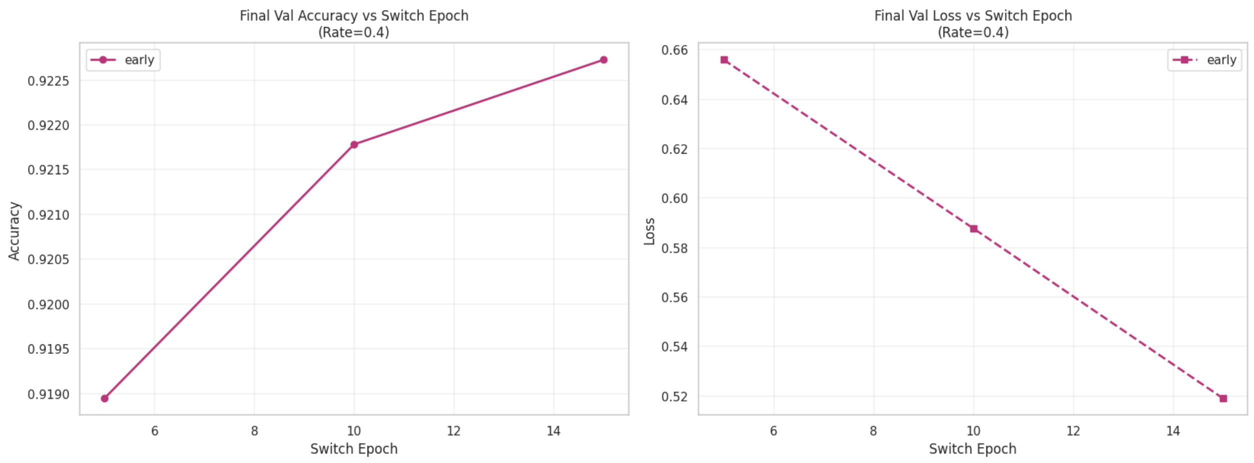

Figure 3: Impact of Switch Epoch on Accuracy and Loss.

Analysis: Figure 3 shows a linear improvement. Extending the dropout phase from 5 to 15 epochs increases accuracy (91.90% $\rightarrow$ 92.27%) and, crucially, significantly decreases the final loss (0.66 $\rightarrow$ 0.52). This indicates that a longer initial phase of gradient variance reduction leads to a better final optimization minimum.

4. Data Regime Analysis (Dataset Size)

Parameters:

- $Switch = 10$
- $Rates = 0.1$
- $Epochs = 20$
- Fractions $\in \{10\%, 30\%, 50\%, 100\%\}$: To verify the "Underfitting" hypothesis, we simulate data scarcity where models typically struggle to optimize.

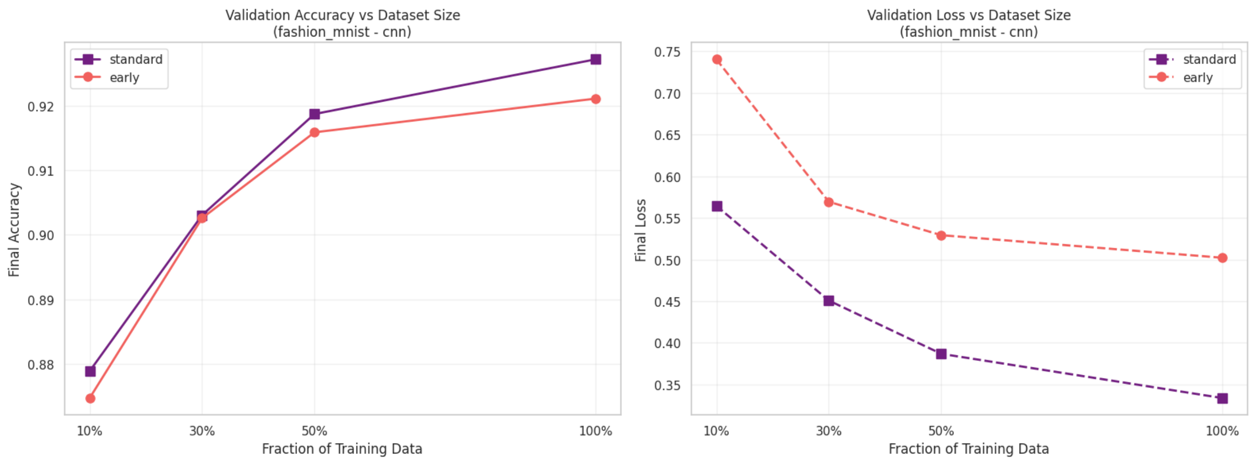

Figure 4: Impact of Training Set Size on Accuracy and Loss.

Analysis: Figure 4 presents the results of the data scarcity experiment on the Convolutional Neural Network, revealing that **Standard Dropout (Purple)** consistently outperforms Early Dropout (Red) across all data regimes. Specifically, even in the highly constrained setting of 10% training data, Standard Dropout achieves both higher accuracy ($\approx$ 87.9% vs $\approx$ 87.5%) and significantly lower validation loss ($\approx$ 0.56 vs $\approx$ 0.74). This trend persists as the dataset size increases to 100%, indicating that the CNN architecture benefits more from the sustained regularization of Standard Dropout than from the initial optimization boost provided by Early Dropout in this specific context.

4.1.4 Conclusion and Validation

Our investigation on the FashionMNIST dataset yields a nuanced verdict regarding the efficacy of Early Dropout. On the one hand, we successfully validated the **structural claims** of Liu et al. [1]. The method demonstrates remarkable robustness to high dropout rates ($p = 0.6$) and exhibits the predicted optimization dynamics, where the cessation of early noise triggers a sharp reduction in training loss.
However, regarding the primary performance metric, our results indicate that **Early Dropout is not the optimal strategy** for a Convolutional Neural Network on this specific dataset. The data scarcity analysis revealed that Standard Dropout maintains a performance edge even with limited training samples. This suggests that the CNN architecture naturally avoids the severe underfitting regime that Early Dropout is designed to correct. Consequently, while the mechanism functions as described, its practical utility is limited in contexts where the model architecture already possesses strong inductive biases and adequate capacity.

4.2 CIFAR-10

The `CIFAR-10` dataset serves as a rigorous benchmark for image classification, consisting of 60,000 color images (32×32 pixels) evenly distributed across 10 mutually exclusive classes (including airplanes, automobiles, and animals). Unlike the previous datasets, the high intra-class variability and low resolution of `CIFAR-10` require a model capable of capturing complex spatial hierarchies. Consequently, we employed a Convolutional Neural Network (CNN) rather than a dense architecture, leveraging convolutional kernels to efficiently extract features while minimizing parameter count and computational cost. All detailed results are provided in Appendix A.1. Initial observations of the learning curves reveal significant overfitting tendencies when training without constraints, prompting us to limit the number of epochs to ensure valid comparisons. Regarding hyperparameter sensitivity, our experiments align with the findings on `FashionMNIST`: Early Dropout demonstrates superior robustness, maintaining efficiency as the dropout rate increases, whereas standard methods often degrade under high noise. Furthermore, the analysis of the switch epoch indicates a positive correlation between the duration of the regularization phase and final performance; maintaining the dropout active for a longer period yields better validation accuracy and loss. Finally, to explicitly test the paper's core hypothesis regarding underfitting, we conducted a granular analysis of dataset size. By visualising the final validation and training accuracy and loss, we can confirm that reducing the dataset size leads to much lower training performances as the model underfits, especially for our Standard Dropout method. The training accuracy and loss remain very good for our Early Dropout, yet the model is not able to generalize correctly. Initial tests on standard fractions $(10\%, 30\%, 50\%, 100\%)$ showed correlated improvements for both methods without a clear advantage for Early Dropout. To further induce structural underfitting, we extend the analysis to extreme scarcity regimes $(2.5\%, 5\%, 10\%, 15\%)$. However, even in these severe conditions, the results remain inconclusive. While performance is closest at the 10% mark, we can not empirically deduce that stopping dropout provides a significant benefit for underfitted models on this specific CNN architecture, suggesting that the method's efficacy is highly dependent on the model capacity relative to the task complexity.

4.3 CIFAR-100

We then apply our pipeline to `CIFAR-100`. This dataset represents a significant jump in complexity compared to `CIFAR-10` and serves as the definitive "stress test" for the Early Dropout hypothesis. It is closest to the paper's conditions, that's why we hope to get closer results to prove or disprove the hypothesis. While `CIFAR-10` serves as a general-purpose benchmark, `CIFAR-100` introduces a crucial challenge: **data sparsity per class**. It contains the same number of total images (60,000) but distributed across 100 fine-grained classes instead of 10. Consequently, the model has only 600 training images per class to learn complex features (e.g., distinguishing a "beaver" from an "otter"). This setup was chosen to replicate the experimental conditions of Liu et al. [1], who utilized a ViT-Tiny on ImageNet. In our context, a standard CNN trained on `CIFAR-100` faces **structural underfitting**: the model capacity is barely sufficient for the task complexity, and it struggles to converge. While our MNIST experiment (Dense at 10%) theoretically isolated the "optimization aid" effect, `CIFAR-100` represents the "real-world" scenario where a model lacks the capacity or data to fully capture the distribution.
The experimental results (detailed in Appendix A.2) reveal the practical limits of the Early Dropout method. The data scarcity analysis highlights a sharp contrast in performance regimes:

- **Extreme Scarcity (10%):** We observed a marginal validation of the paper's hypothesis. In this regime of very strong data rarity, Early Dropout provides a slight advantage, suggesting it offers some aid to optimization when the signal-to-noise ratio is critical.
- **Sufficient Data (>30%):** As soon as data availability increases beyond this threshold, the trend reverses. Both **Standard Dropout** and **Late Dropout** consistently outperform Early Dropout.

These findings indicate that injecting noise early in the training process (Early Dropout) is detrimental when the task is highly complex and the model is structurally weak. We attribute this to

the interference with the acquisition of **fundamental features**. For a small CNN on `CIFAR-100`, the early epochs are critical for learning basic filters (edges, textures). Heavy regularization during this phase prevents these features from stabilizing.
Consequently, the optimal strategy on `CIFAR-100` appears to be the inverse of the paper's proposal: it is more effective to let the model learn without hindrance initially (Late Dropout) or to apply gentle, continuous regularization (Standard Dropout).
This experiment serves as a critical boundary condition for the *"Dropout Reduces Underfitting"* theory. While Early Dropout acts as a successful "starter" for optimization in theoretical setups, it struggles in complex practical scenarios (`CIFAR-100`) where the model's capacity bottleneck outweighs optimization difficulties. In such high-entropy tasks, stability during the early learning phase proves more valuable than the gradient variance reduction offered by Early Dropout.

4.4 MNIST

The `MNIST` dataset serves as a foundational benchmark in image classification, consisting of 28×28 grayscale images of handwritten digits. It shares identical structural characteristics with `Fashion-MNIST` (same dimensions and I/O format), differing only in semantic content (digits 0 to 9 rather than clothing items). Given the simpler spatial features of this dataset, we employed a Dense Neural Network (MLP), requiring the input to be flattened into a 784-dimensional vector, while maintaining the same experimental parameters used in the previous section. All corresponding results are detailed in Appendix A.3. Analyzing the learning dynamics (Figure 14, Appendix A.3), we observe a distinct drop in training loss immediately following the `switch_epoch`, validating the optimization mechanism described in the paper, while validation loss remains stable and all strategies converge to an accuracy of $\approx$ 97.5%. Regarding robustness, Early Dropout again proves superior to Standard Dropout under heavy noise conditions ($p = 0.6$), whereas the standard approach retains the advantage at lower rates ($p < 0.4$). Furthermore, our temporal analysis indicates that an early switch (epoch=5) is optimal; delaying this transition significantly degrades performance, causing a sharp drop in accuracy and a marked increase in loss. Finally, the data regime analysis highlights the specific utility of Early Dropout in constrained settings: at 30% of the dataset size, it outperforms the standard approach, achieving higher accuracy and marginally lower loss.
While we have not been able to reproduce or disprove the utility of the Early Dropout method in a global setting, we can confirm that in an easy prediction setting with a powerful and well structured neural network, Early Dropout can reduce underfitting. This has been visualised on the image classification dataset `MNIST` using an MLP.

5 Conclusion

In this study, we undertook a rigorous reproduction and analysis of the paper *"Dropout Reduces Underfitting"* (Liu et al., 2023 [1]). Our experimental framework was designed to align with the pedagogical progression of our Master's curriculum, utilizing standard benchmarks (`FashionMNIST`, `MNIST`, `CIFAR-10`, `CIFAR-100`) and fundamental architectures (Dense Networks and CNNs) to isolate the effects of the proposed regularization strategies.

5.1 Summary of Findings

Our results paint a more nuanced picture than the generalized claims found in the original literature. While we successfully validated the **mechanical robustness** of Early Dropout (its ability to withstand high noise rates) and its positive impact on training loss dynamics, the method failed to consistently outperform Standard or Late Dropout in terms of final generalization accuracy, particularly on **Convolutional Neural Networks**.

- **Validation Success:** The method proved effective in the specific context of **Dense Networks on MNIST** under data scarcity (10%). Here, the lack of inductive bias created an optimization hurdle that Early Dropout successfully bridged.
- **Contextual Limitations:** On CNN architectures applied to `FashionMNIST`, `CIFAR-10` and `CIFAR-100`, Standard and Late Dropout strategies yielded superior results.

5.2 The Role of Experimental Constraints

To interpret these divergences, it is essential to consider the specific constraints of our experimental setup:

1. **Model Architecture and Course Alignment:** We deliberately employed shallow CNNs and MLPs as studied in our coursework. Unlike the massive Vision Transformers (ViT) used in the paper [1], these "classic" architectures possess strong inductive biases (especially CNNs) or limited capacity. This shifted the problem from the *optimization underfitting* described by Liu et al [1]. (where a large model gets stuck) to *structural underfitting* (where a small model lacks capacity).
2. **Computational Constraints:** Limited computational resources restricted us from training large-scale overparameterized models or using extensive data augmentation. In this "low-compute" regime, regularizing a model that is already struggling for capacity (as seen with `CIFAR-100`) proved counter-productive during the early phase.

5.3 Final Verdict

Our investigation leads to a critical refinement of the paper's conclusion: **Early Dropout is not a universal remedy for underfitting**. It is specifically an *optimization aid* designed for high-capacity models that struggle to converge. For lower-capacity models or architectures with strong priors (like standard CNNs), the traditional wisdom holds: allowing the model to learn initially (Late Dropout) or regularizing gently throughout (Standard Dropout) remains the optimal strategy.
Ultimately, this project highlights the importance of context in Deep Learning: a method that represents a breakthrough for State-of-the-Art models on ImageNet may behave very differently on fundamental architectures used in academic settings.

A Appendix

A.1 CIFA-10

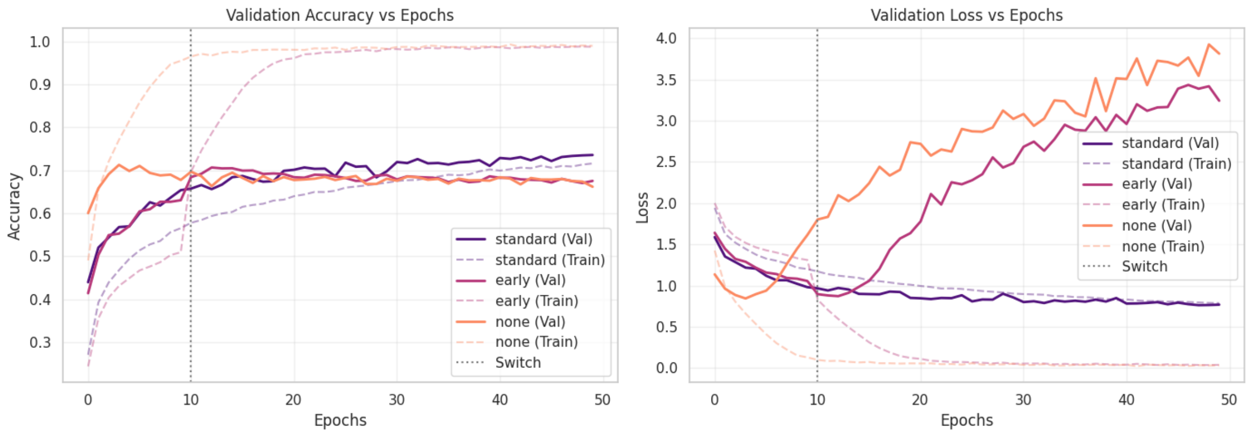

Figure 5: Learning Curves for Standard (Purple) vs. Early (Red) Dropout on CIFAR-10.

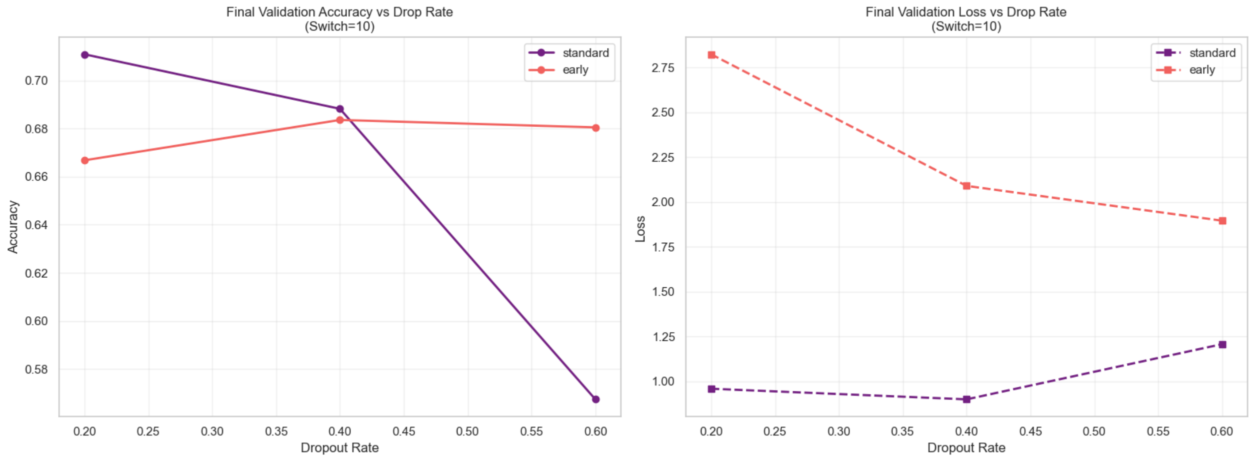

Figure 6: Impact of dropout Rate on Final Accuracy (left) and Loss (right).

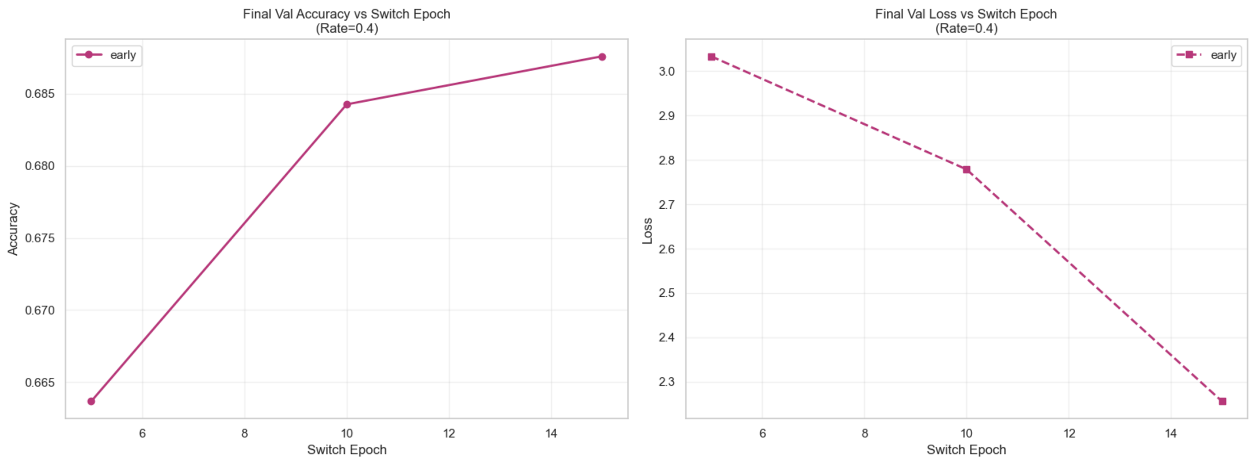

Figure 7: Impact of Switch Epoch on Accuracy and Loss.

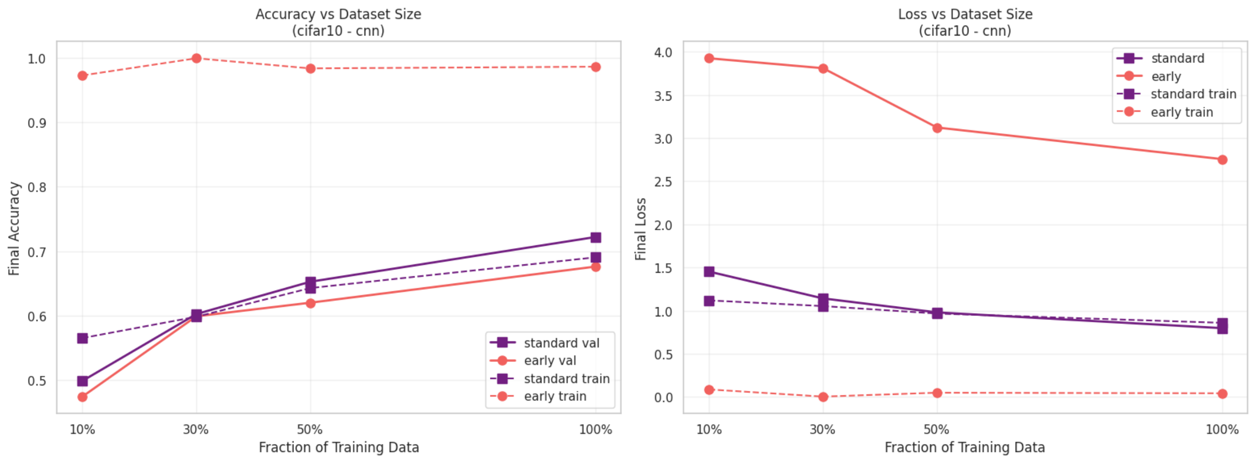

Figure 8: Confirmation of the impact of smaller datasets on underfitting

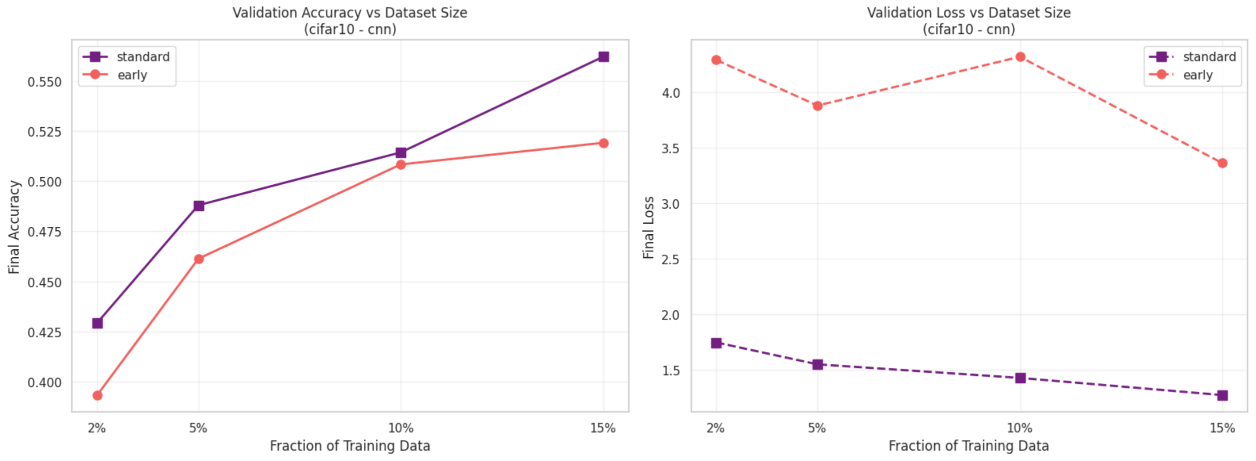

Figure 9: Impact of Training Set Size on Accuracy and Loss.

A.2 CIFAR-100

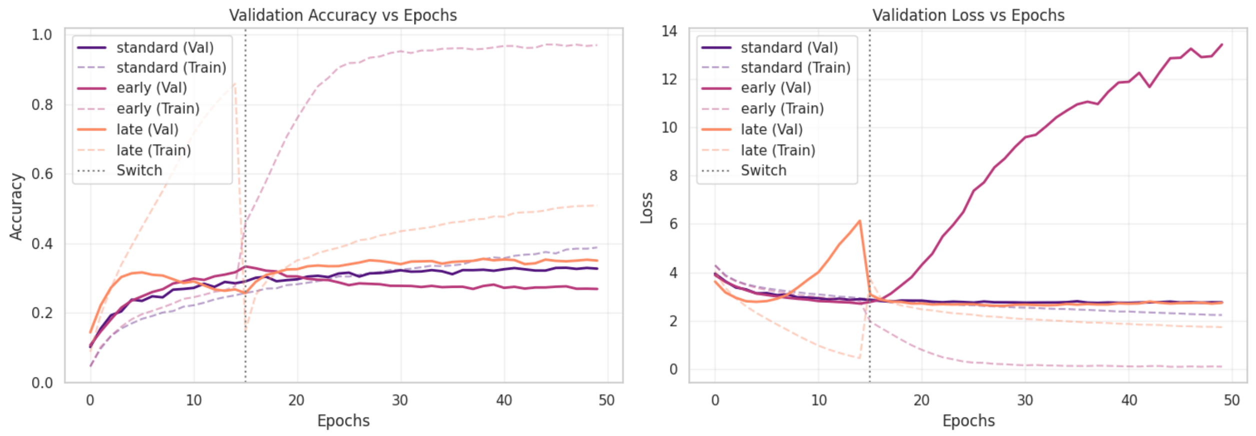

Figure 10: Learning Curves for Standard (Purple) vs. Early (Red) Dropout on CIFAR-100.

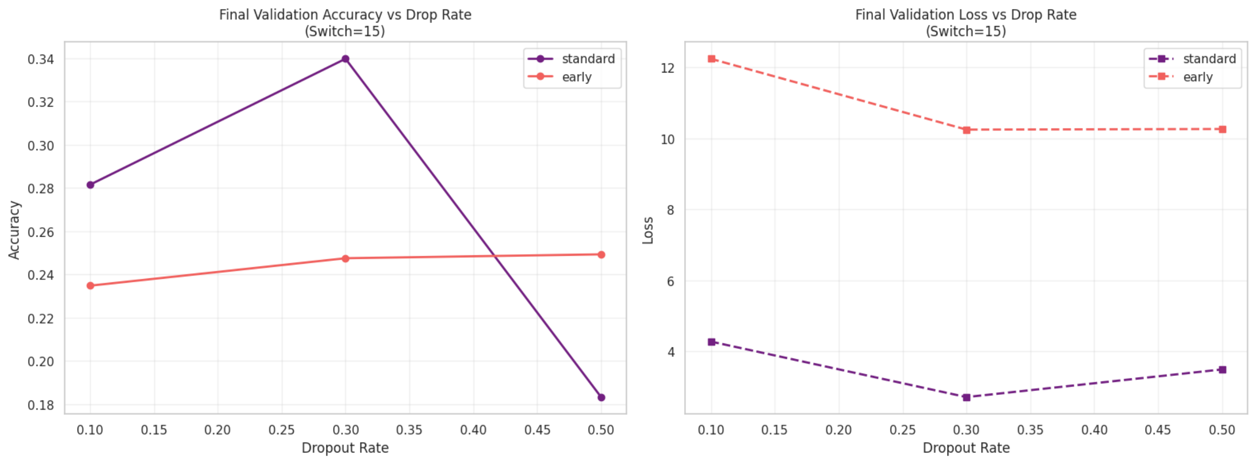

Figure 11: Impact of dropout Rate on Final Accuracy (left) and Loss (right).

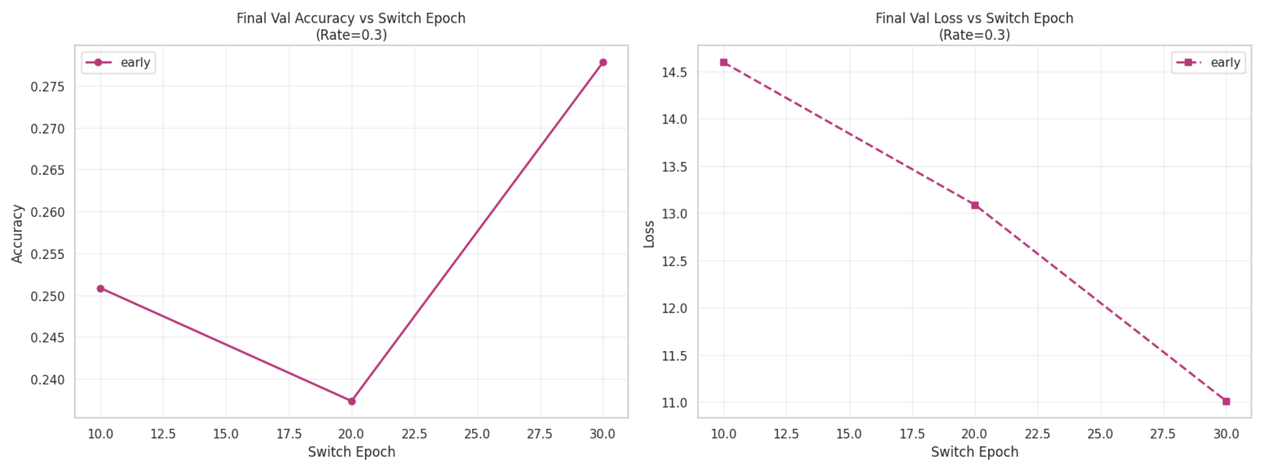

Figure 12: Impact of Switch Epoch on Accuracy and Loss.

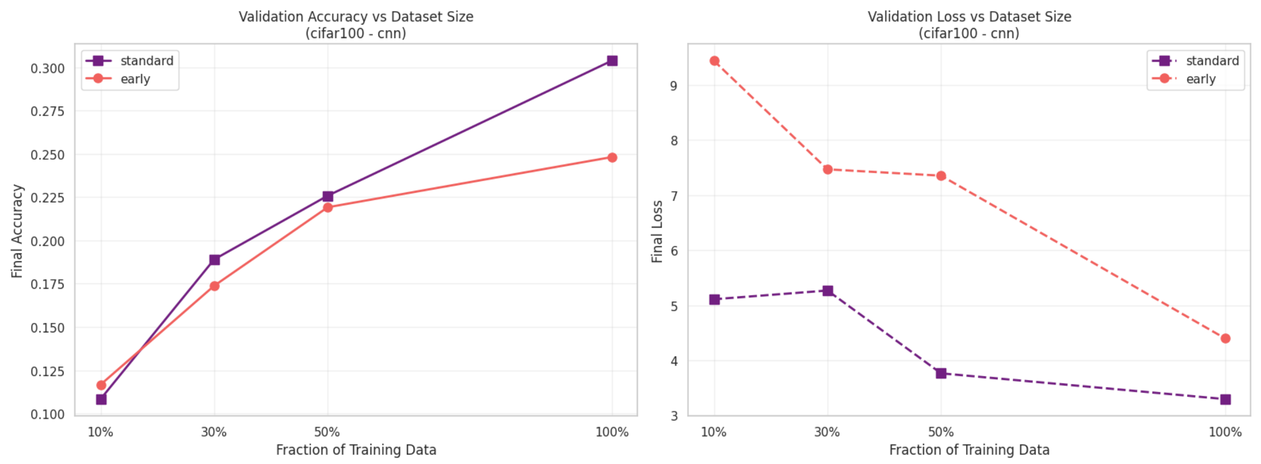

Figure 13: Impact of Training Set Size on Accuracy and Loss.

A.3 MNIST

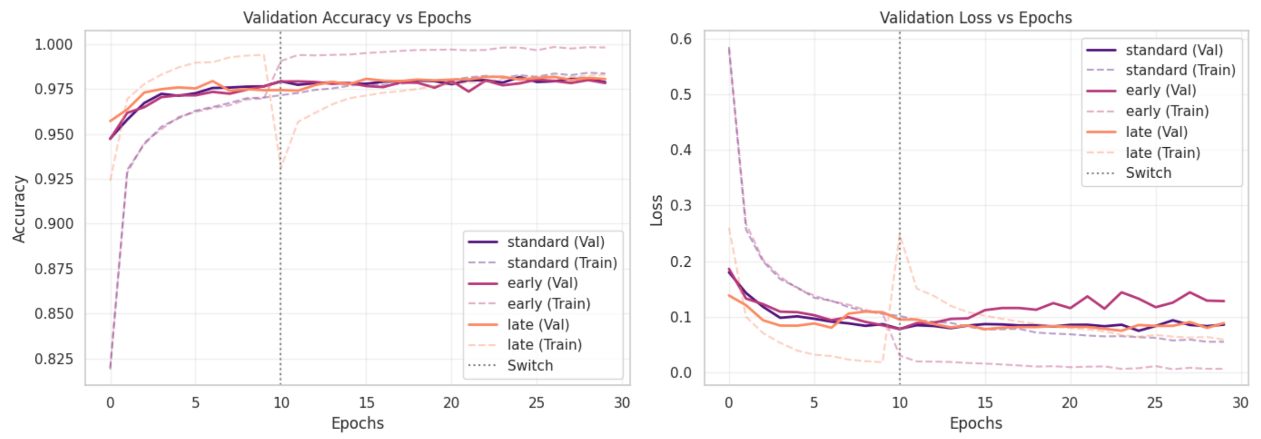

Figure 14: Learning Curves for Standard (Purple) vs. Early (Red) Dropout on MNIST.

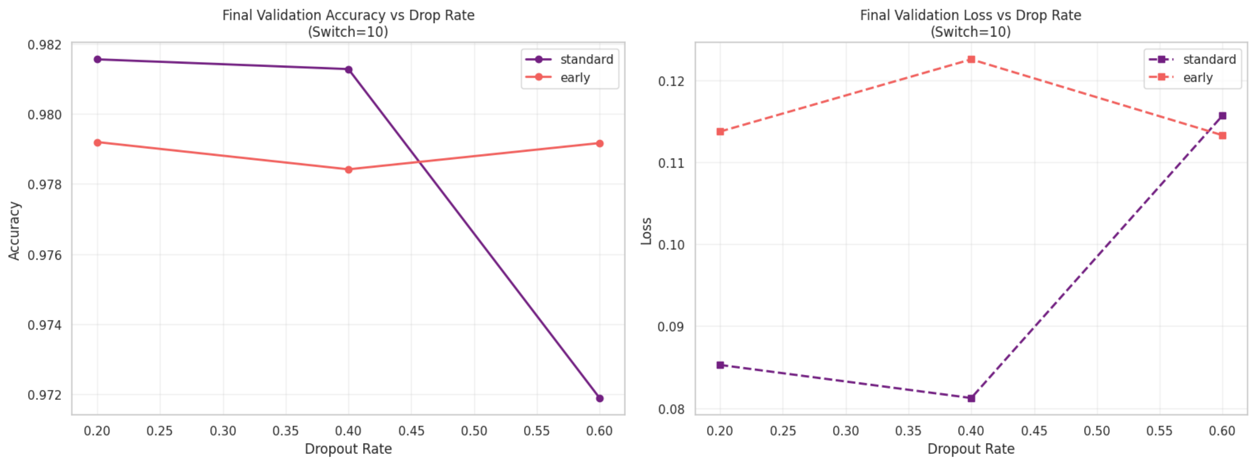

Figure 15: Impact of dropout Rate on Final Accuracy (left) and Loss (right).

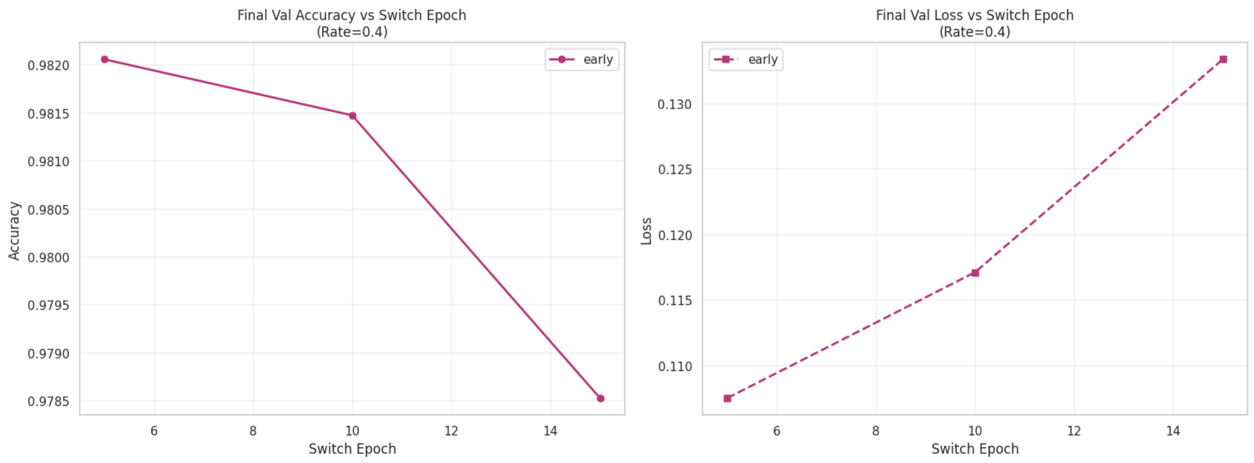

Figure 16: Impact of Switch Epoch on Accuracy and Loss.

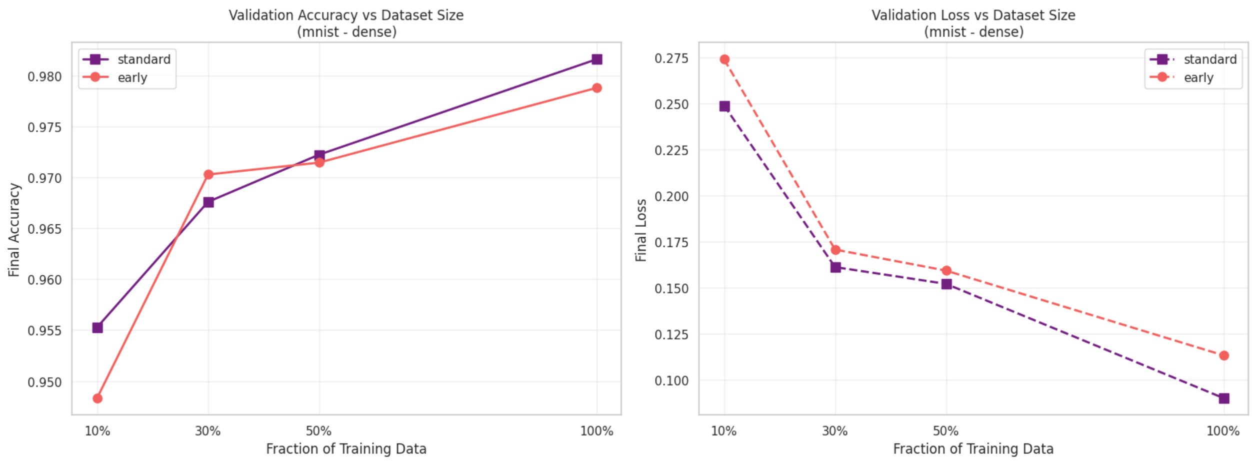

Figure 17: Impact of Training Set Size on Accuracy and Loss.

References

[1] Zhuang Liu et al. “Dropout Reduces Underfitting”. en. In: ().

[2] Théo Lopes-Quintas. *Deep Learning Course Notes*. Course material from Université Dauphine-PSL / M2 ISF. Lecture notes used in the Deep Learning course. 2025.

[3] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. en. In: ().