# IISER THIRUVANANTHAPURAM INTERNSHIP – COMPUTATIONAL IMAGING AND DATA SCIENCE LAB

**NAME**: ASHWIN CHANDAR S
**INSTITUTION:** CHENNAI INSTITUTE OF TECHNOLOGY
**INTERNSHIP START DATE:** 19TH MAY 2025
**MENTOR/GUIDE:** DR. RAJI SUSAN MATHEW

# TABLE OF CONTENT

# WORK DONE

- Implemented and the "*ResLT: Residual Learning for Long-tailed Recognition*" paper from scratch.
- Designed and experimented with hybrid architectures combining "*ELF: An Early-Exiting Framework for Long-Tailed Classification*" and "*ResLT: Residual Learning for Long-tailed Recognition*" .
- Iteratively debugged, analyzed, and refined the models to achieve state-of-the-art performance.

# FOUNDATIONAL CONCEPTS

**ResLT's Contribution: Class-Level Re-balancing**

- Introduced re-balancing in the parameter space.
- Its specialized three-branch head is explicitly designed to improve performance on under-represented medium and tail classes.

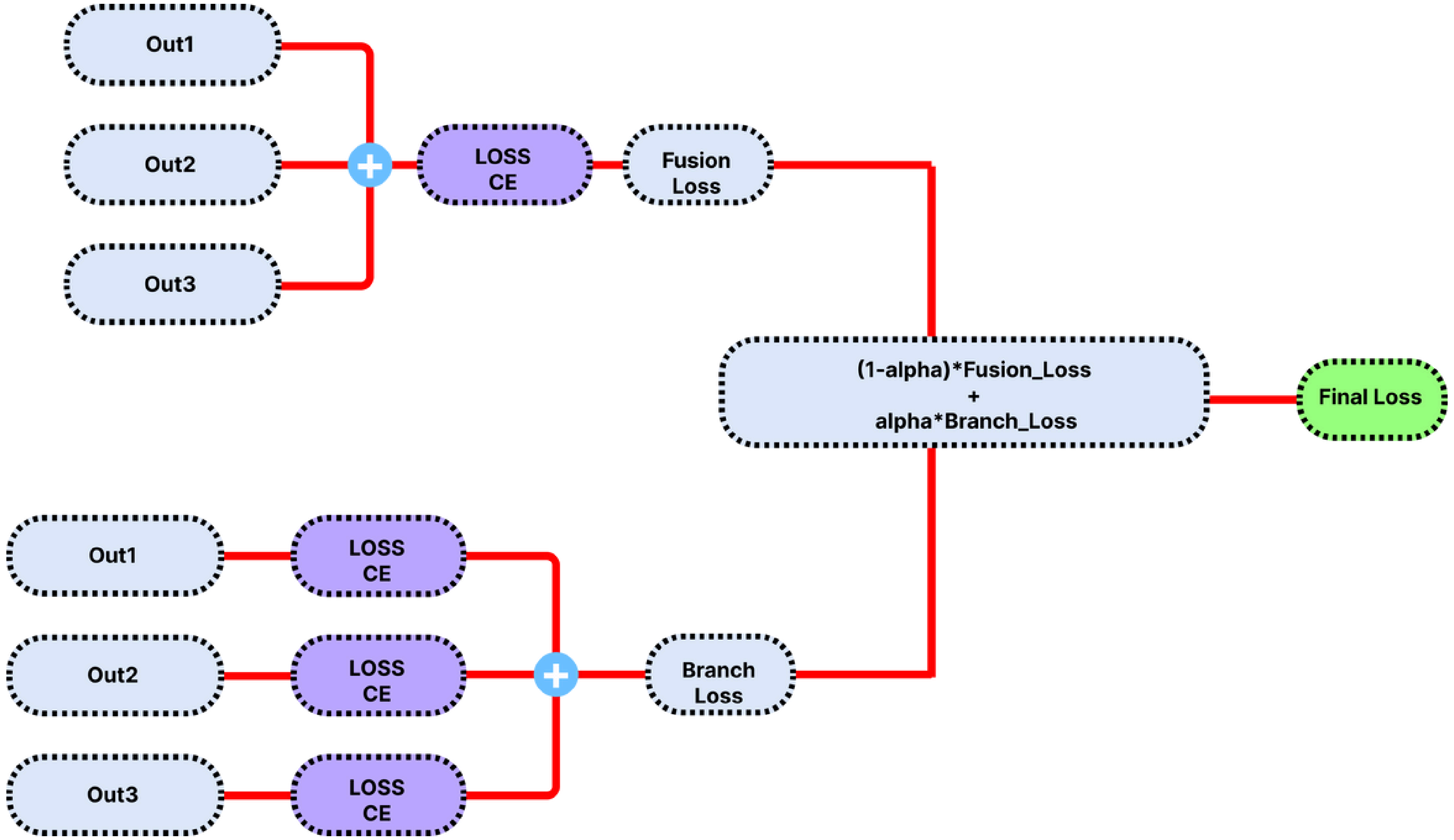
**ELF's Contribution: Example-Level Hardness**

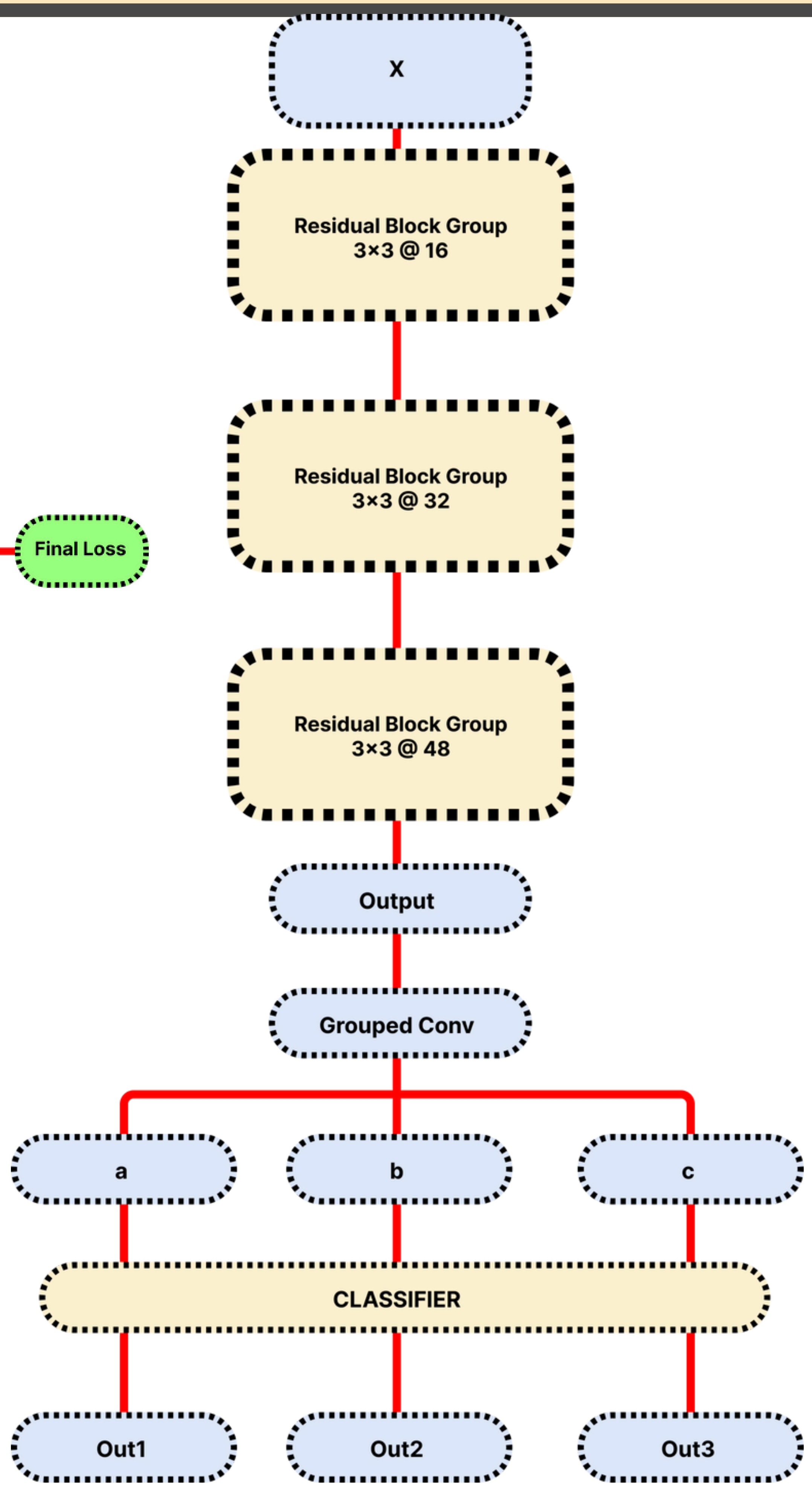
- Introduced the concept of example hardness.
- Its early-exiting framework filters out "easy" images to focus the model's capacity on "hard" ones.

# PHASE 1.1 - IMPLEMENTING RESLT

My first step was to create a strong baseline by implementing the ResLT paper.

- **Architecture:** I implemented a ResNet-32 backbone with the specialized ResLT head (a grouped convolution with 3 groups).
- **Practical Challenge:** I encountered a ValueError as the standard ResNet-32's 64 output channels are not divisible by 3. I resolved this by modifying the final block to output 48 channels.
- **Fusion Loss:** A standard Cross-Entropy loss on the summed output of the three branches.
- **Branch-Independent Loss:** A specialized loss calculated on individual branches using filtered data subsets to focus on medium and tail classes.



# PHASE 1.2 - TUNING THE RESLT MODEL

The model's performance was critically dependent on the alpha hyperparameter. I performed extensive tuning on the 100x imbalanced dataset with different alpha vlaues to get the optimal value.

| **100x Models** | Overall | Many | Medium | Few | |
|---|---|---|---|---|---|
| Alpha= 0.995 | 53.9 | 27.8 | 46.93 | 78.7 | Tail-Specialist |
| Alpha= 0.9 | 72 | 83.37 | 66.27 | 67.95 | Head-Specialist |
| Alpha= 0.99 | 61.77 | 43.9 | 52.5 | 82.12 | Tail-Specialist |
| **Alpha= 0.95** | **70.79** | **74.73** | **64.73** | **72.45** | **Most Balanced:** |

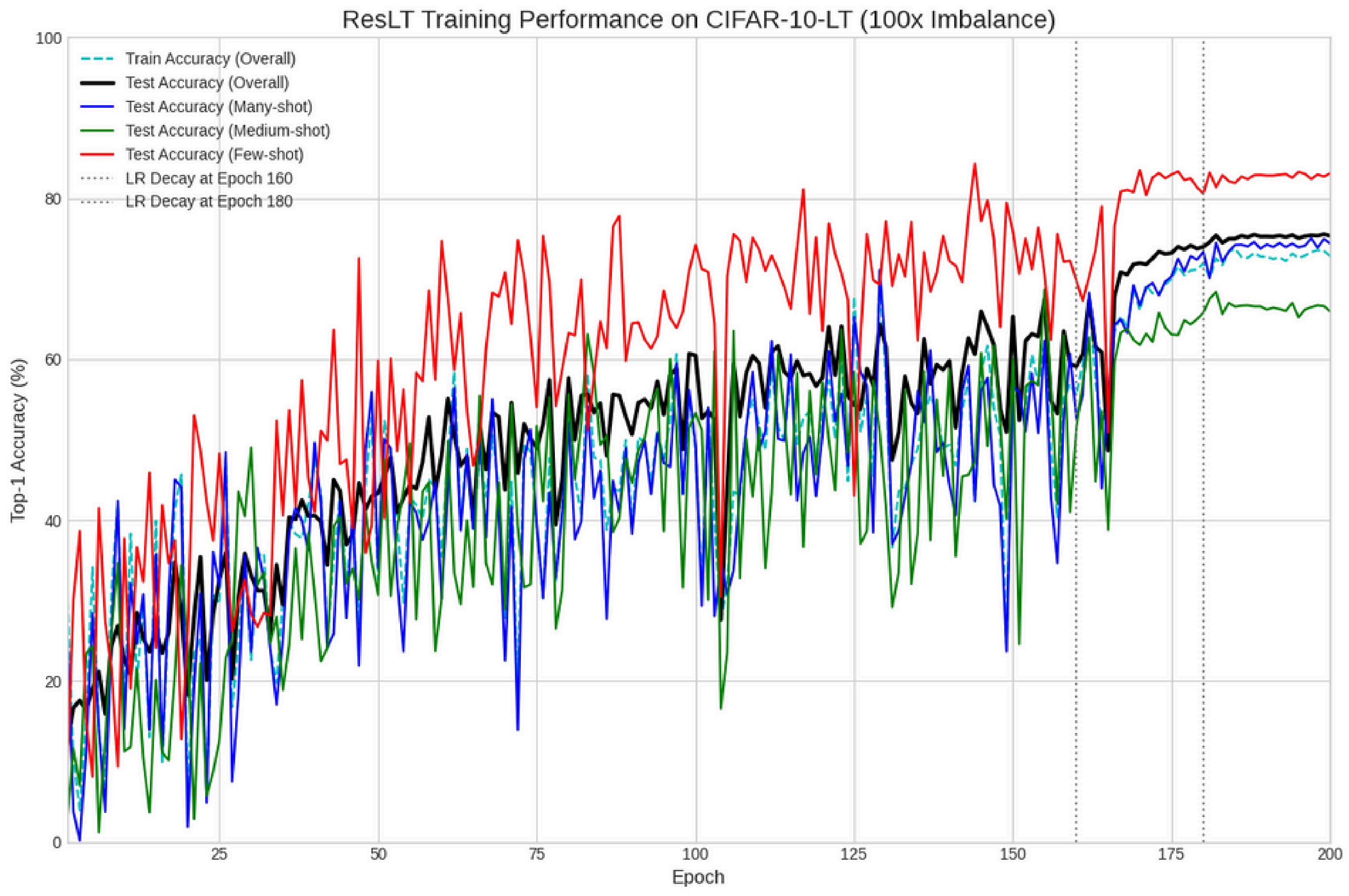
- The run with alpha = 0.95 was the most successful. It achieved a final accuracy of ~71% and, most importantly, produced a balanced model that performed well across all three class splits (Many: ~75%, Medium: ~65%, Few: ~73%). the final phase focused on closing the remaining ~9% gap to the paper's reported ~80% accuracy.

| **100x Models** | Overall | Many | Medium | Few |
|---|---|---|---|---|
| Alpha= 0.95 | 70.79 | 74.73 | 64.73 | 72.45 |
| Alpha= 0.95 with AutoAugmentation | 70.22 | 67.87 | 62.3 | 77.92 |
| **Alpha= 0.95 with AutoAugmentation and Label Smoothing** | **75.33** | **74.37** | **65.93** | **83.1** |

# PHASE 1.3 - FINAL RESLT MODEL

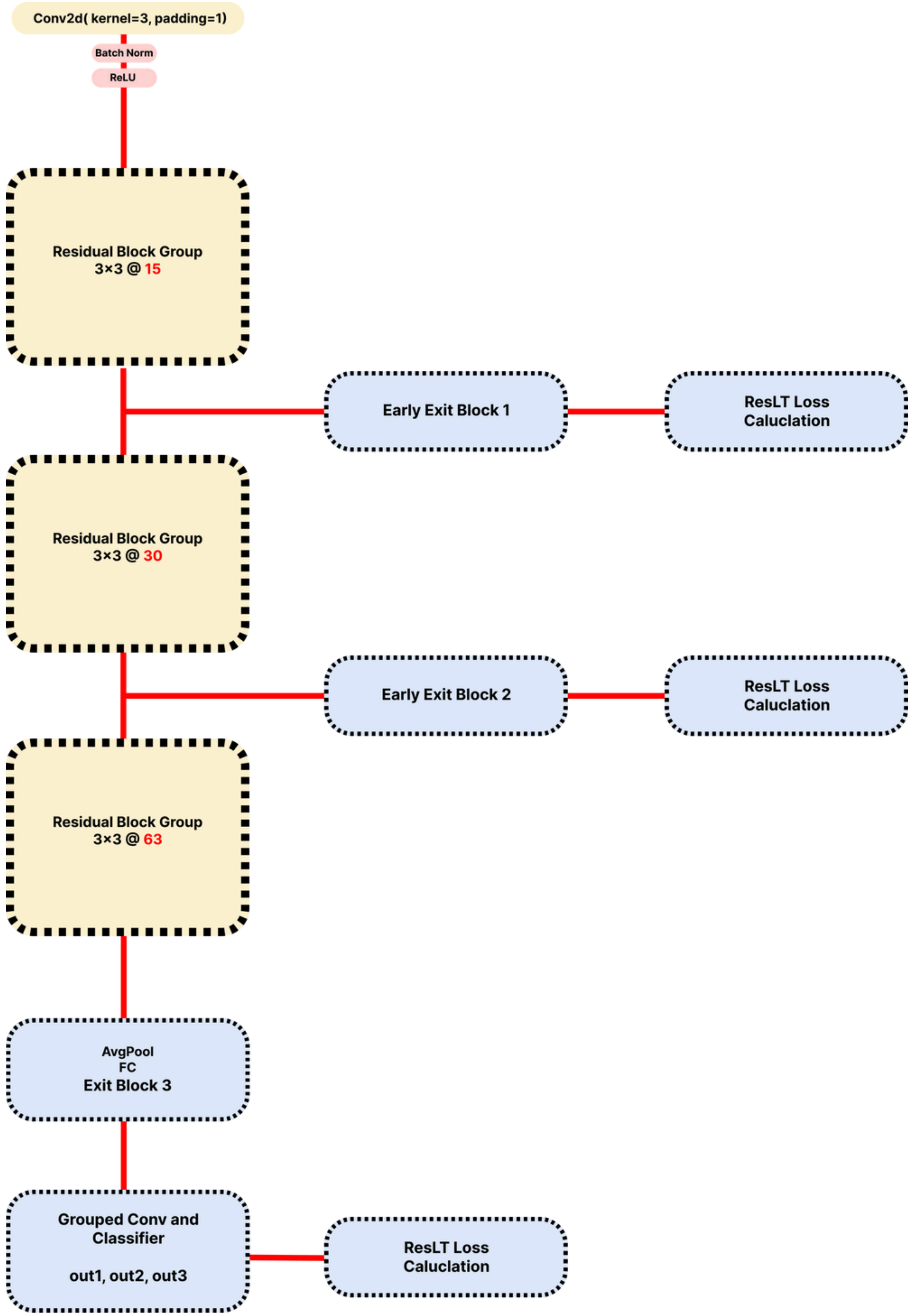
| **Alpha= 0.95 with AutoAugmentation and Label Smoothing** | **Paper's Accuracy** | Overall | Many | Medium | Few | **Difference** |
|---|---|---|---|---|---|---|
| 100x | 80.44 | **75.33** | 74.37 | 65.93 | 83.1 | **~5.11** |
| 50x | 83.46 | **80.14** | 80.5 | 70.1 | 87.4 | **~3.32** |
| 10x | 89.06 | **86.78** | 84.63 | 79.23 | 94.05 | **~2.28** |

- The model with alpha value as 0.95 along with AutoAugmentation and Label Smoothing yielded me the best accuracy with a difference of -5.11%, -.3.32%, -2.28% at 100x, 50x and 10x respectively.
- I moved on to the implementation of the hybrid model combining the ELF and ResLT implementation.

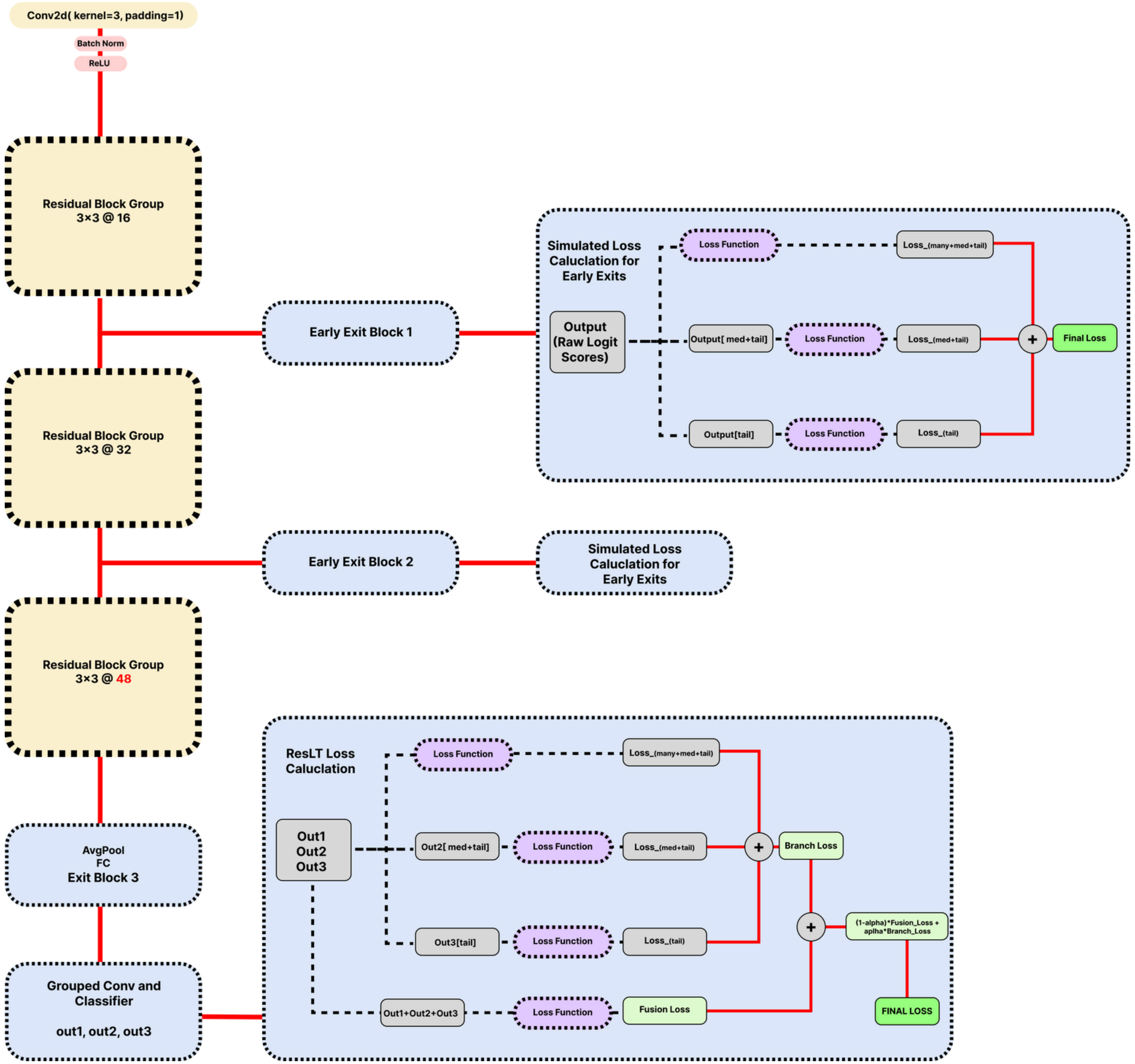


# PHASE 2 - THE HYBRID MODEL ARCHITECTURES

**Full Power Architecture:**
Include ResLT Loss calculation at every exit of the model, with ELF backbone.




**Simulated Loss Architecture:**
Include ResLT Loss calculation at every exit of the model, with ELF backbone.

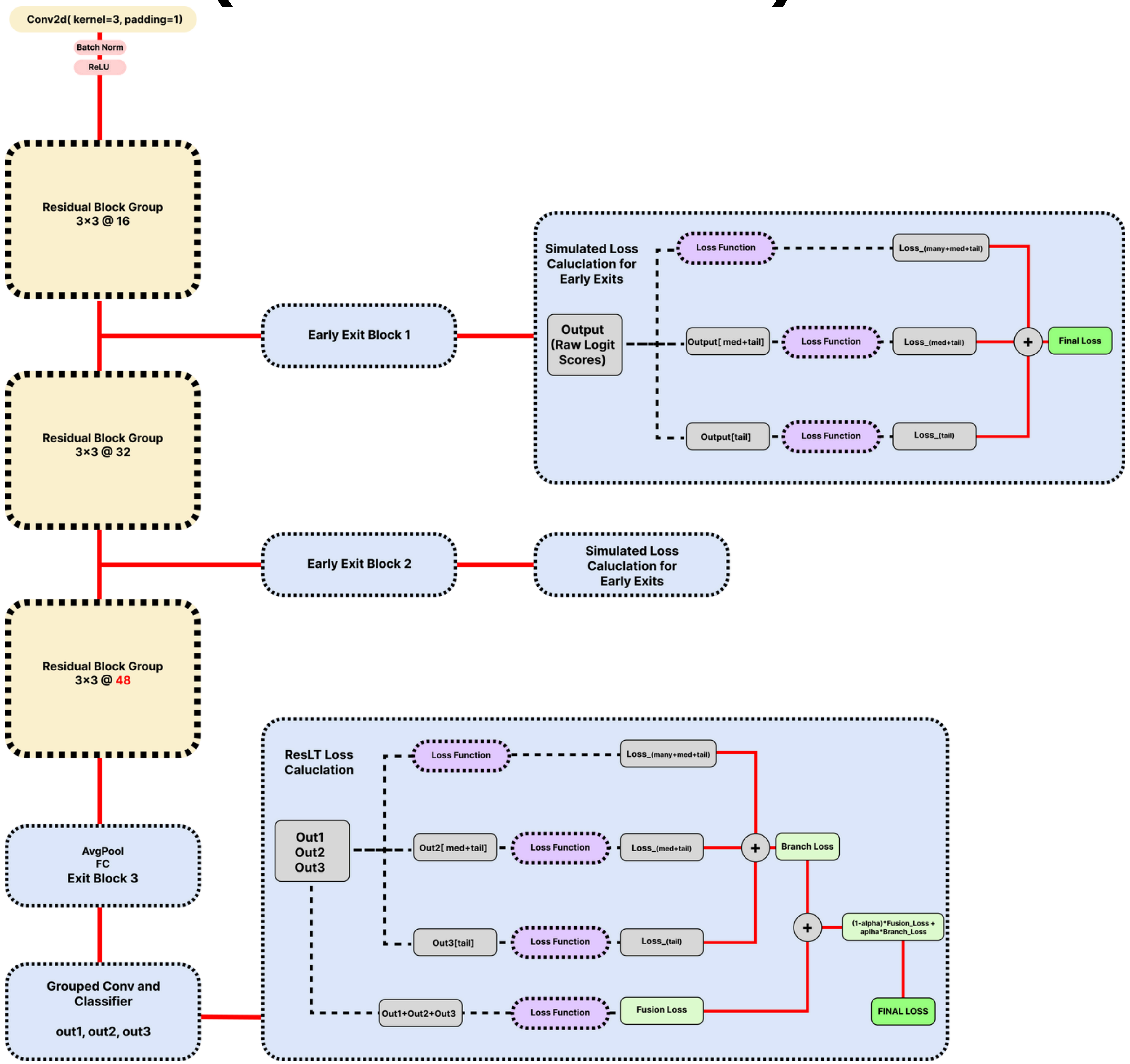


# PHASE 3.1 - THE "ACCUMULATIVE HYBRID" (SIMULATED LOSS) (MODELS 1-3)

- My initial idea was a direct combination of the two frameworks, using ELF's loss accumulation principle. Used Simulated Loss for these models for early exits alone and included a full ResLT Loss at the Final exit

| 100x Models | Accuracy |
|---|---|
| Model 1: Accumulative Hybrid w/ Cross-Entropy | 56.9 |
| Model 2:Accumulative Hybrid w/ LDAM, threshold=0.2 | 64.94 |
| **Model 3: Accumulative Hybrid w/ LDAM, threshold=0.8** | **68.09** |

- The performance was underwhelming. The accumulative loss created a noisy and difficult optimization problem, leading to a suboptimal result.
- Model 3 seems to be a further improvement, but this design seemed to have a performance ceiling well below our target. The accumulative loss was likely too complex.



# PHASE 3.2 - THE "ROUTED HYBRID" (W/O LOSS ACCUMULATION) (MODELS 4 & 7)

- Based on the instability of the first models, I did the same ***but did not accumulate the losses for each sample from the previous exits***. I tried this to address the complexity of the model.

| **100x Models** | Overall | Many | Medium | Few |
|---|---|---|---|---|
| Model 4: Routed Hybrid w/ LDAM | CRASHED | | | |
| **Model 7: Routed Hybrid w/ CE and Label Smoothing** | **78.99** | **92.63** | **72.43** | **73.67** |

| **100x Models** | Acc |
|---|---|
| Model 1: Accum | 56.9 |
| Model 2: Accum | 64.94 |
| Model 3: Accum | 68.09 |

**Model 4 (Routed Hybrid w/ LDAM)**

- Design: Instead of accumulating loss, I would route each sample to a single exit for its loss calculation. This "single-point loss" was designed to be much more stable.
- Result: **CRASHED**.

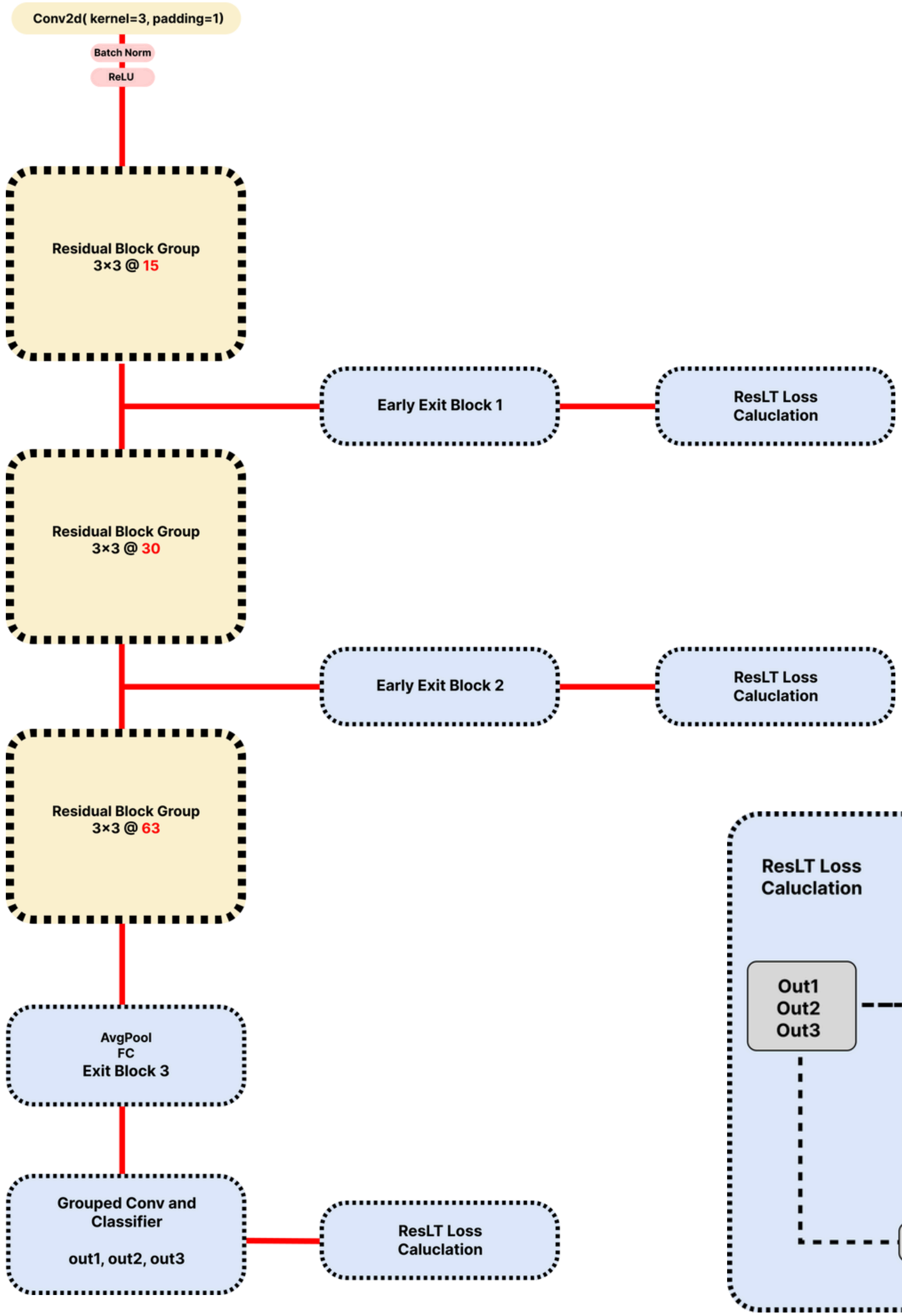
**Model 7 (Routed Hybrid w/ CE)**

- **Design:** I retried the "Routed" design, but with the more stable CrossEntropyLoss (with label smoothing) to prevent collapse.
- **Result: A major success!** This model was stable and achieved a final accuracy of **78.99%**, outperforming our individual models and getting very close to the state-of-the-art.

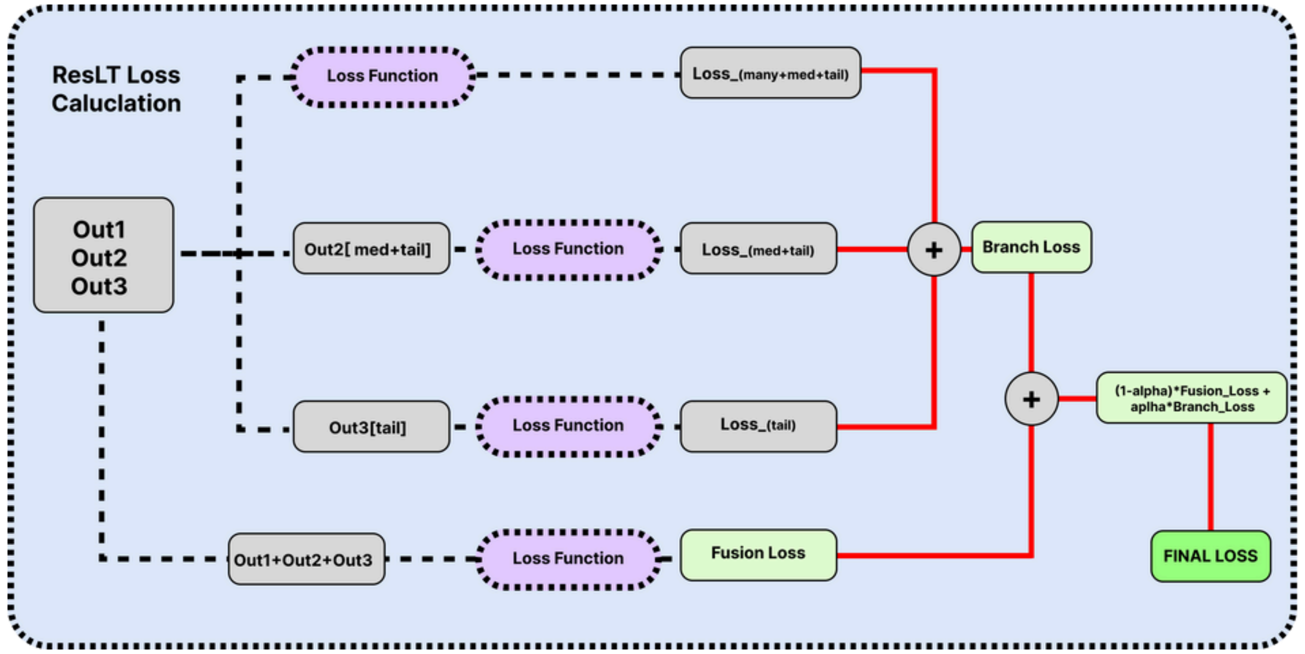
# PHASE 4.1 - THE "FULL POWER HYBRID" WITH LOSS ACCUMULATION (MODELS 5,6)

- I built a model where every single exit was a full, computationally expensive ResLT head, trained with accumulative Loss at first

| 100x Models | Overall | Many | Medium | Few |
|---|---|---|---|---|
| Model 5: **LDAM** with Loss accumulation | 60.26 | 29.33 | 57.1 | 85.83 |
| Model 6: **CE** with Loss accumulation | 62.71 | 79.8 | 56 | 54.8 |



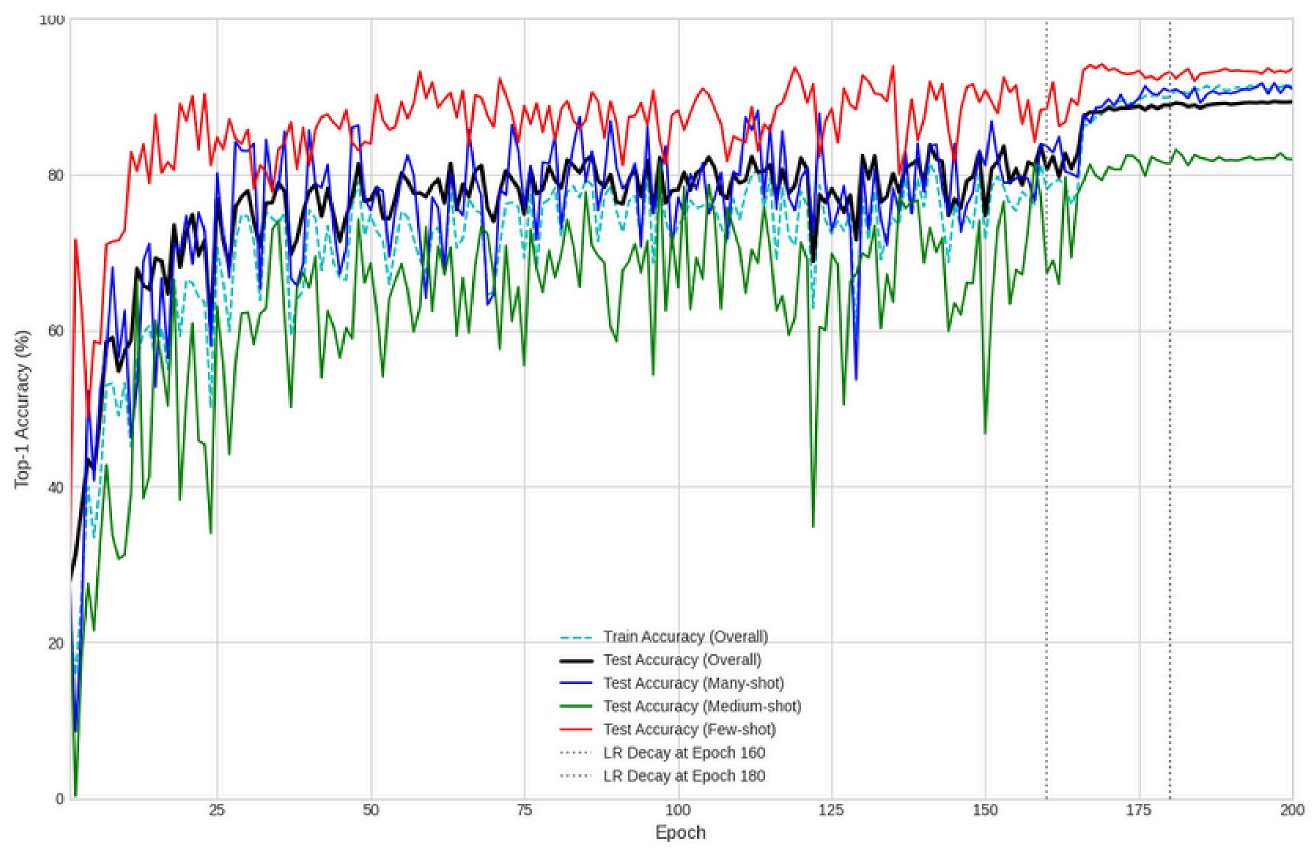

| 100x Models | Acc |
|---|---|
| Model 1: Accum | 56.9 |
| Model 2: Accum | 64.94 |
| Model 3: Accum | 68.09 |
| Model 4: Routed | Crashed |
| Model 7: Routed | 78.99 |
| Model 5: Accum | 60.26 |
| Model 6: Accum | 62.71 |



# PHASE 4.2 - THE "FULL POWER HYBRID" W/O LOSS ACCUMULATION (MODELS 14-16)

- I built a model where every single exit was a full, computationally expensive ResLT head, trained ***without acummulation of loss from earlier exits***

| **Full Power Hybrid: Cross Entropy, AutoAugment, w/o loss accumulation** | Overall Accuracy | Many | Medium | Few |
|---|---|---|---|---|
| Model 14: **100x** | 78.94 | 93 | 72.3 | 73.38 |
| Model 15: **50x** | 83.36 | 92.3 | 75.83 | 82.3 |
| Model 16: **10x** | 89.32 | 90.27 | 81.93 | 93.62 |




| **Models** | | Acc |
|---|---|---|
| Model 1:Accum | | 56.9 |
| Model 2:Accum | | 64.94 |
| Model 3:Accum | | 68.09 |
| Model 4: Routed | | Crashed |
| Model 7: Routed | | 78.99 |
| Model 5:Accum | | 60.26 |
| Model 6:Accum | | 62.71 |
| Full Power: Routed | Model 14: 100x | 78.94 |
| | Model 15: 50x | 83.36 |
| | Model 16: 10x | 89.32 |

# PHASE 5 - FINAL OPTIMIZATION OF THE ROUTED MODELS (SIMULATED LOSS) (MODELS 8, 12, 13)

- Design: I took my best architecture (Model 7) and replaced AutoAugment with the more powerful Mixup augmentation on 100x.
- **Result: 79.59%.**
- A new best, this confirmed that stronger regularization was a key path to improvement.

| **Routed Hybrid: CE, w/o loss accumulation, Mixup Augmentation** | Overall Accuracy | Many | Medium | Few |
|---|---|---|---|---|
| Model 8, **100x** | 79.59 | 90.4 | 69.93 | 78.72 |
| Model 12, **50x** | 82.89 | 89.9 | 75.97 | 82.83 |
| Model 13, **10x** | 85.77 | 83.63 | 76.43 | 94.38 |

| **Models** | | Acc |
|---|---|---|
| Model 1:Accum | | 56.9 |
| Model 2:Accum | | 64.94 |
| Model 3:Accum | | 68.09 |
| Model 4: Routed | | Crashed |
| Model 7: Routed | | 78.99 |
| Model 5:Accum | | 60.26 |
| Model 6:Accum | | 62.71 |
| Full Power: Routed | Model 14: 100x | 78.94 |
| | Model 15: 50x | 83.36 |
| | Model 16: 10x | 89.32 |
| Model 7 with Mixup | Model 8, **100x** | 79.59 |
| | Model 12, **50x** | 82.89 |
| | Model 13, **10x** | 85.77 |

# PHASE 6 - FINAL COMPARISON & RESULTS

- My final hybrid models successfully achieved state-of-the-art performance, outperforming my own strong baselines and the original papers.
- The Full Pwer Model w/o Loss Accumulation [Models 14,15,16] and Simulated Loss with Mixup Augmentation [Models 8,12,13] are the best ever model ,yielding higher accuracy

| **Models** | 100x | 50x | 10x |
|---|---|---|---|
| ResLT Paper's Accuracy | 80.44 | 83.46 | 89.06 |
| ELF Paper's Accuracy | 78.1 | 82.4 | 88 |
| | | | |
| My Implementation of ResLT | 75.33 | 80.14 | 86.78 |
| My Implementation of ELF | 72.29 | 76.74 | 82.67 |
| | | | |
| **Routed Hybrid: CE, w/o loss accumulation, Mixup Augmentation. Models[8,12,13]** | 79.59 | 82.89 | 85.77 |
| **Full Power Hybrid: Cross Entropy, AutoAugment, w/o loss accumulation. Models[14,15,16]** | **78.94** | **83.36** | **89.32** |

| **Models** | | Acc |
|---|---|---|
| Model 1:Accum | | 56.9 |
| Model 2:Accum | | 64.94 |
| Model 3:Accum | | 68.09 |
| Model 4: Routed | | Crashed |
| Model 7: Routed | | 78.99 |
| Model 5:Accum | | 60.26 |
| Model 6:Accum | | 62.71 |
| Full Power: Routed | Model 14: 100x | 78.94 |
| | Model 15: 50x | 83.36 |
| | Model 16: 10x | 89.32 |
| Model 7: Routed, with Mixup | Model 8, **100x** | 79.59 |
| | Model 12, **50x** | 82.89 |
| | Model 13, **10x** | 85.77 |

# CONCLUSION

Key Findings:

- "Routing" Loss is More Stable than Accumulating Loss and boosts accuracy as well
- The Full Power Architecture, despite being computationally expensive, yields higher accuracy than all the other models
- The Simulated Loss struggles to achieve paper's level and is boosted by Mixup Augmentation only. However Full Power achieved higher accuracy with AutoAugment
- The Simulated Model performs better at 100x model than Full Power Model.
- The Full Power Model performs better at 10x dataset than Simulated Model

| | Models | | Acc | Difference |
|---|---|---|---|---|
| Loss Accumulated | Model 1:Accumulative Hybrid w/ Cross-Entropy | | 56.9 | -23.54 |
| | Model 2:Accumulative Hybrid w/ LDAM, threshold=0.2 | | 64.94 | -15.5 |
| | Model 3:Accumulative Hybrid w/ LDAM, threshold=0.8 | | 68.09 | -12.35 |
| Loss not Accumulated | Model 4: Routed Hybrid w/ LDAM | | Crashed | - |
| | Model 7: Routed Hybrid w/ CE and Label Smoothing | | 78.99 | -1.45 |
| Loss Accumulated | Model 5: Full Power: LDAM with Loss accumulation | | 60.26 | -20.18 |
| | Model 6: Full Power: CE with Loss accumulation | | 62.71 | -17.73 |
| Loss not Accumulated | Full Power Hybrid: Cross Entropy, AutoAugment, w/o loss accumulation | Model 14: 100x | 78.94 | -1.5 |
| | | Model 15: 50x | 83.36 | -0.1 |
| | | Model 16: 10x | 89.32 | **/+0.26** |
| Loss not Accumulated | Model 7: Routed Hybrid w/ CE and Label Smoothing, **with Mixup** | Model 8, **100x** | 79.59 | -0.85 |
| | | Model 12, **50x** | 82.89 | -0.57 |
| | | Model 13, **10x** | 85.77 | -3.29 |