Understanding Feature Transferability in Neural Networks

- This presentation explores concept of feature transferability in neural networks, inspired from the work of Yosinski et al.
- Our work simplifies the original methodology, demonstrating that key findings remain consistent even with smaller datasets and models.
- We'll cover experimental design, implementation details, results, and implications for practical transfer learning.

by Yair Reichman & Kai

Experimental Design: Splitting Tasks & Training Networks

- We divided the CIFAR-10 dataset into two distinct tasks (A and B), each has 5 classes.
- We trained base networks on each task independently to establish performance baselines.
- Then, we created transfer networks by copying the first ***n*** layers from a "source" network and randomly initializing the remaining layers.
- These transfer networks were then trained on the "target" task, and their performance was compared against the baseline.
- This allowed us to isolate the impact of transferring different layers and assess their generality.

Network Types: Base, Frozen, and Fine-Tuned

We employed several network configurations.

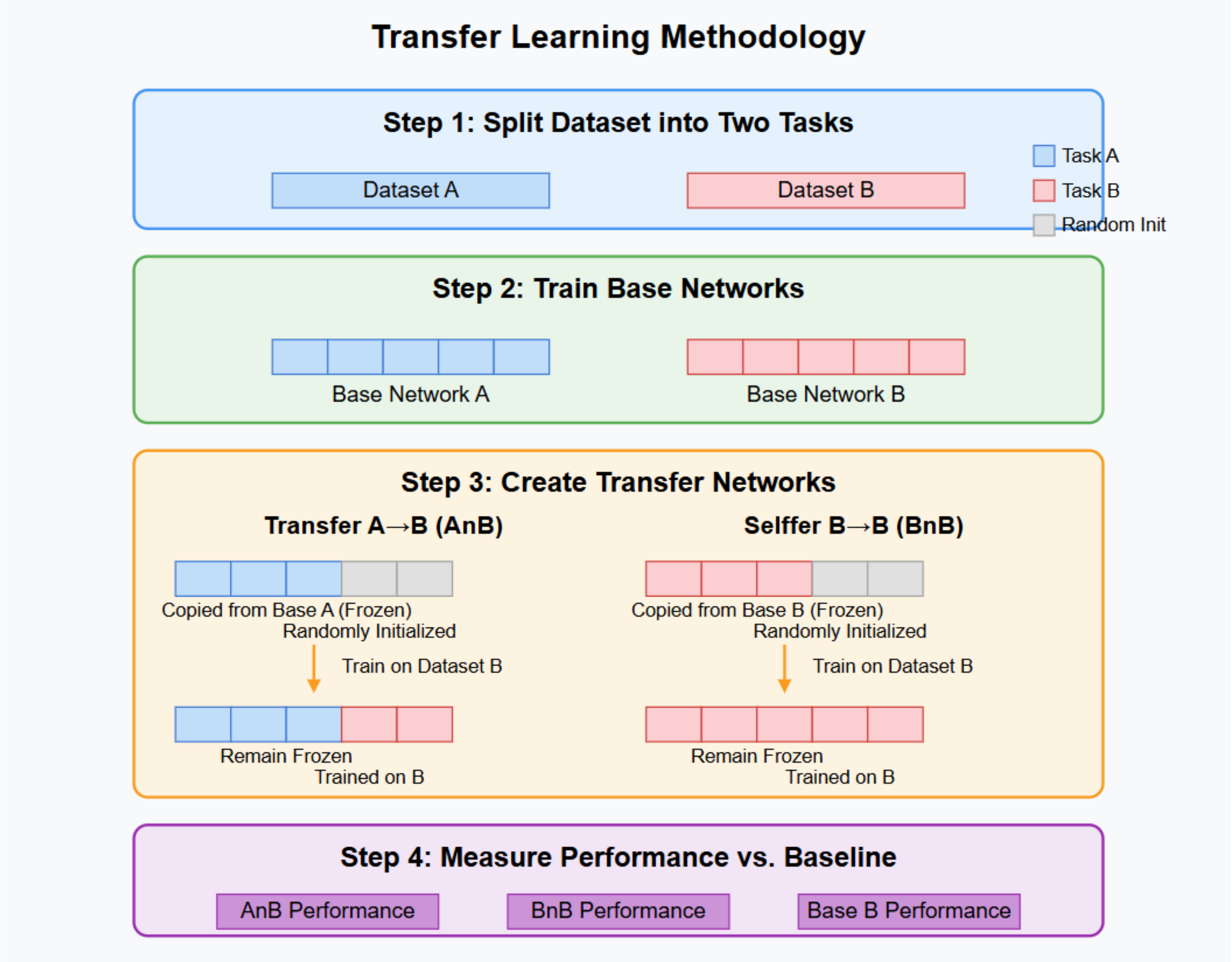
baseB is a baseline network trained solely on task B.

BnB networks have their initial ***n*** layers copied from ***baseB*** and then frozen during training on task B.

BnB+ is similar to **BnB**, but the copied layers are fine-tuned.

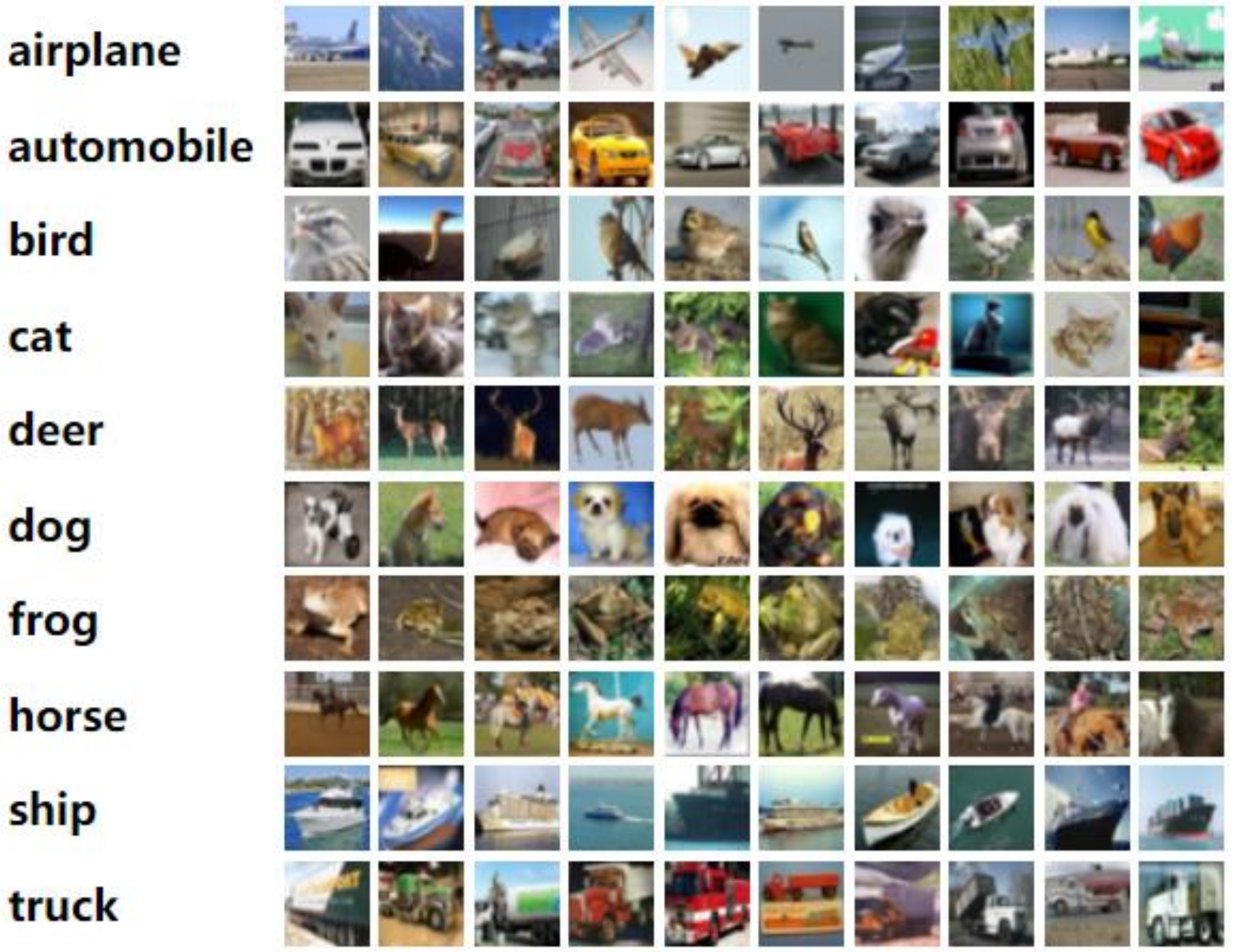
AnB and **AnB+** are analogous to **BnB** and **BnB+** but transfer layers from the network trained on task A.

These different network types allowed us to measure the benefits of transfer learning and the impact of fine-tuning.

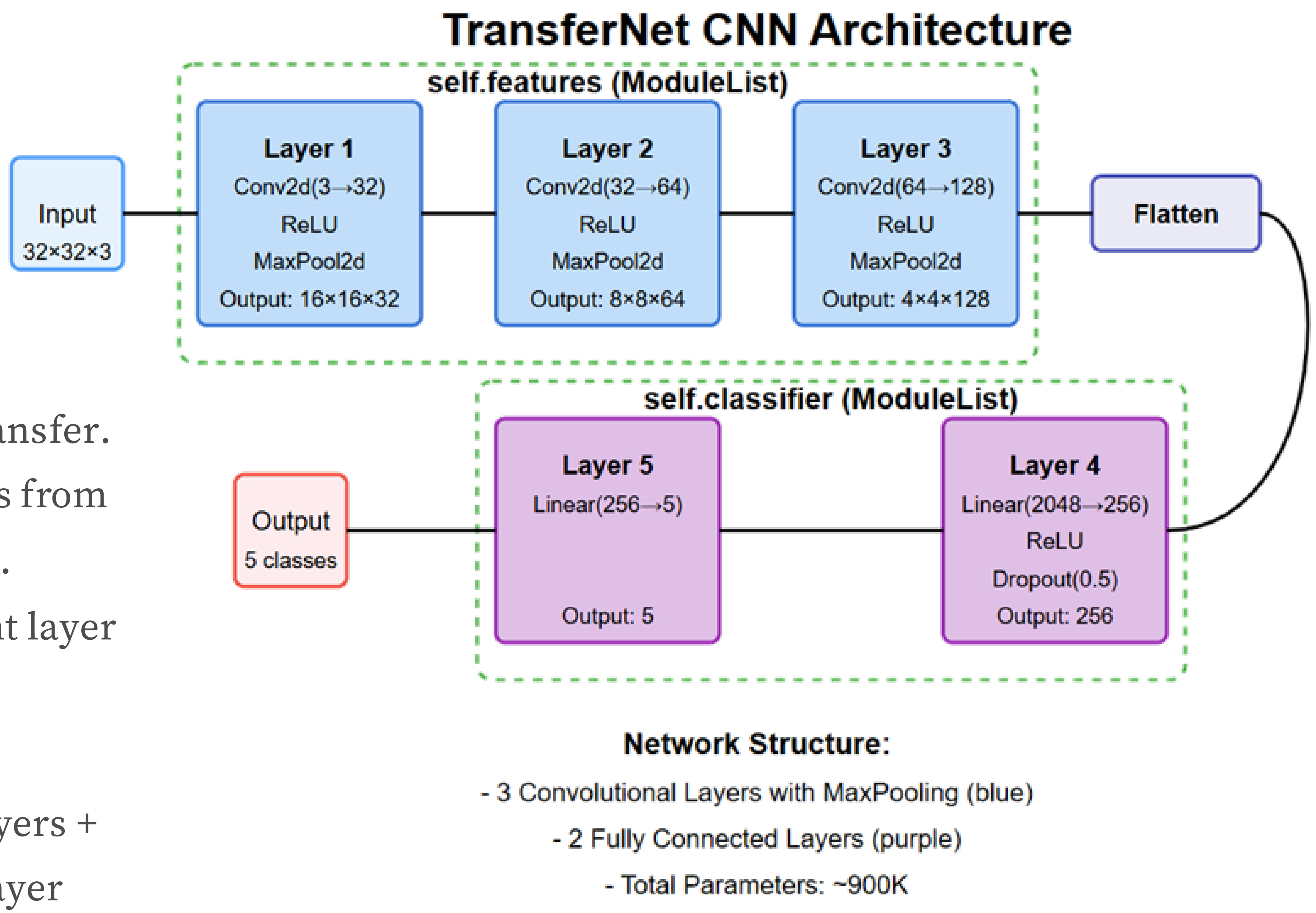


Implementation Details: Lightweight CNN and CIFAR-10

- Our implementation diverges from the original paper in a few key areas.
- We utilized the **CIFAR-10 dataset** instead of ImageNet, significantly reducing computational demands.
- We also designed a **custom lightweight CNN architecture with only five layers**, whereas the original paper used AlexNet.
- Training was limited to **20 epochs** due to resource constraints. These simplifications enabled us to run the experiments on a laptop.



Transfer Learning Implementation: Layer Copying and Freezing



- Our **TransferNet** class utilizes **nn.ModuleList** for easy layer transfer.
- The **create_transfer_network** function copies the first ***n*** layers from the source model to the target model, optionally freezing them.
- This modular approach allowed us to experiment with different layer combinations and assess their impact on transfer learning performance.
- About 900K parameters vs. 60M in AlexNet, 3 convolutional layers + 2 fully connected layers, and a **ModuleList** structure for easy layer transfer.

Our Experiment Results: Accuracy by Layer

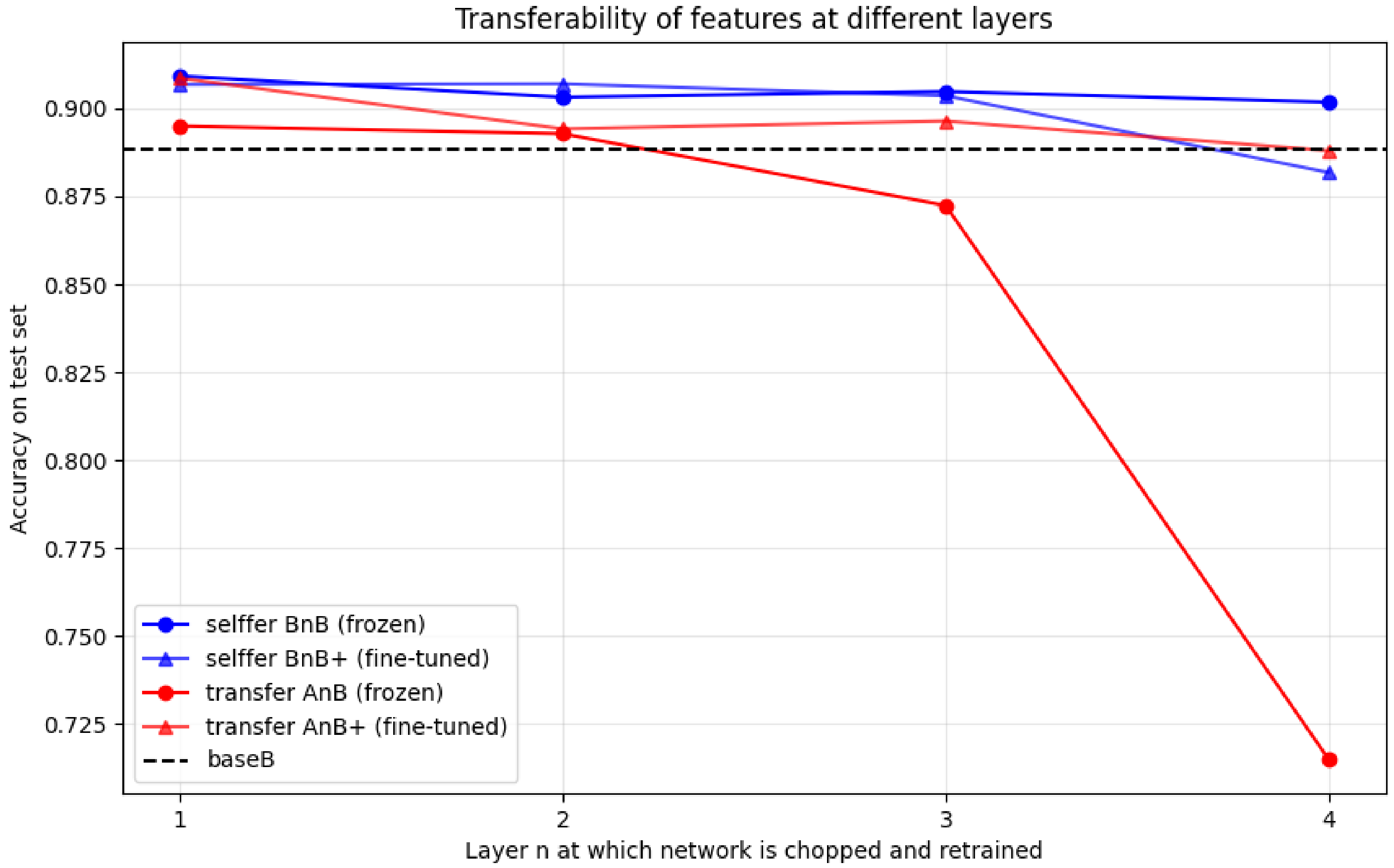
- **Early layers consistently transferred better** than later layers, indicating that the initial layers learn more general features.

- **Fine-tuning** the transferred layers **improved performance**.

  Both fine-tuned networks (blue triangles and red triangles) maintain high performance throughout most layers.

  However, even with fine-tuning, there's still a slight performance drop at layer 4, indicating some irreversible task-specific specialization.

- Moreover, a performance drop observed in the **AnB.** This is more extreme than drops seen in earlier layers and suggests that the first fully connected layer contains highly task-specific representations.



Interpreting the Results: Co-adaptation, Specificity, and Fine-tuning

Co-adaptation Gap: Neurons that work together can't be easily separated
Specificity Gap: How specialized features are to their original task
Fine-tuning Benefit: How much improvement comes from adapting features

- The **BnB+** vs. **baseB** difference reveals the **co-adaptation gap**.
- The **AnB** vs. **BnB** difference quantifies **feature specificity**.
- The improvement from **AnB+** vs. **AnB** demonstrates the **fine-tuning benefit**. These metrics help understand the underlying mechanisms of transfer learning.

<u>Key findings:</u>
Feature Generality in early layers.
Performance drops due to co-adaptation and the consistent advantage of fine-tuning.
Initializing with transferred features and fine-tuning (AnB+) can give similar or better performance as training from scratch (baseB).

Conclusion: Applications, Limitations, and Extensions

Our findings have practical implications for transfer learning:

- Utilize early layers from pre-trained models and always fine-tune if possible.

We also acknowledge limitations, such as the smaller dataset and model. Future work could explore more diverse task pairs and compare with modern architectures.