# Part 2-KD Project

# Abstract


We explore a series of Knowledge Distillation (KD) strategies for improving student model performance:

1. **Baseline:** Training with cross-entropy loss only
2. **KD:** Adding KL divergence to transfer soft targets from the teacher
3. **KD + Alignment:** Aligning student and teacher attention on causally relevant regions
4. **KD + Attention Transfer:** Enhancing knowledge transfer with spatial attention maps
5. **KD + Attention + Alignment:** Combining both mechanisms for optimal knowledge transfer

# Abstract


We evaluate whether this step-by-step or leave one out enhancement leads to:

- Higher classification accuracy
- Better robustness to corrupted inputs (e.g., CIFAR-10)
- Greater model efficiency (fewer parameters, faster inference)

# Why Is This Problem Challenging?

- The teacher and student networks differ in depth, width, and representation capacity
- Aligning their attention maps is non-trivial due to structural differences
- Not all teacher attention is useful , Teacher sometimes focus on irrelevant regions. requires causal filtering
- Balancing multiple losses (CE, KL, Attention, Alignment) requires careful tuning , Choosing the right weights ($\lambda_1$–$\lambda_4$) is delicate — too much of one signal can dominate the learning

# How We Structured the Solution

**Step-wise comparison:**

- Baseline → KD → KD +Alignment → KD+Attention → KD + Attention + Alignment

**Robust evaluation:**

- Accuracy, efficiency

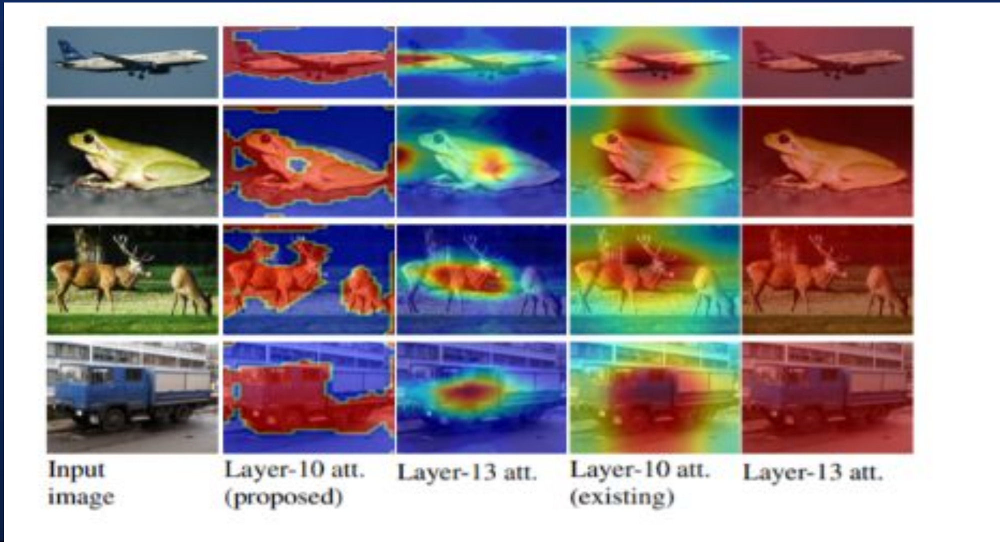
**Loss integration:**

- Combine CE, KL, masked MSE, and alignment losses into unified training

# Our Experiment

# Research Question:

Can the incorporation of Alignment and Attention Transfer techniques enhance the effectiveness of Knowledge Distillation in improving student model accuracy for image classification?



# Innovation Through Selective and Aligned Attention Transfer

- Instead of transferring all attention maps from the teacher to the student, we transfer only the causally influential attention.

- These regions are identified using gradient-based causal masks.

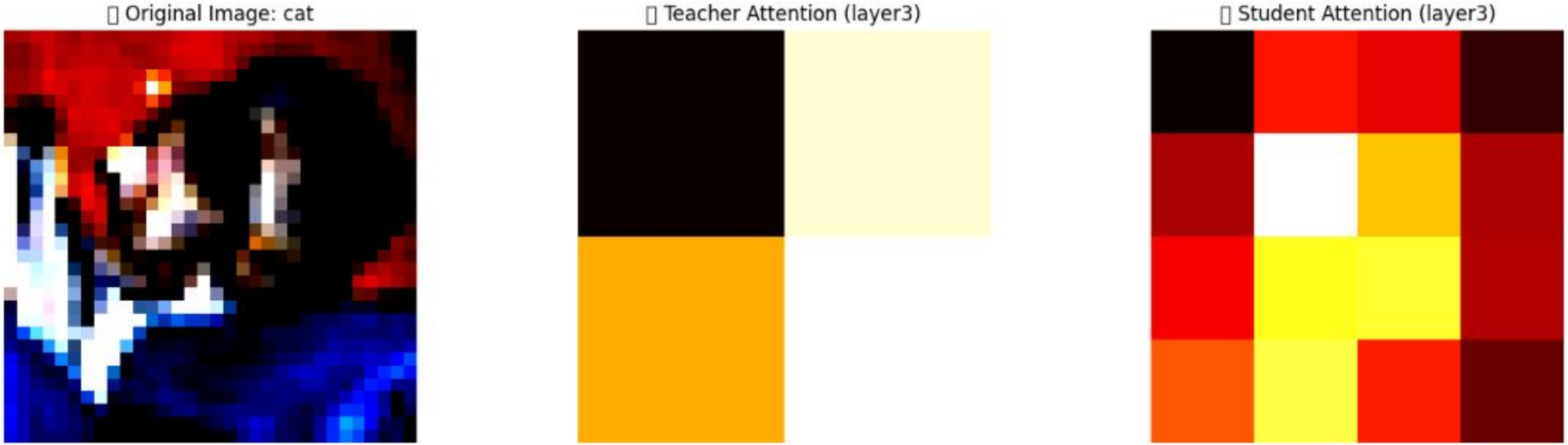
- A masked alignment loss is applied so the student learns to focus where it truly matters.

- This results in cleaner, more focused knowledge transfer, which improves generalization and reduces overfitting.

# Attention Maps in KD – Key Ideas

- What is attention? A spatial map indicating which parts of the input the network focuses on.
- How it's used: Extracted by computing the sum of squared feature activations across channels.
- Why it matters: Transferring attention maps helps the student learn where to focus, mimicking the teacher’s internal reasoning.

# Attention Maps in KD – Example




MSE בין המורה לתלמיד (layer3): 0.614280

# Alignment in KD-Key Ideas

- What is alignment? A contrastive learning mechanism that encourages the student’s attention to match the teacher’s only for the same image.
- How it's used: Apply contrastive loss: pull positive pairs (same image) together, push negative pairs (different images) apart in the attention space.
- Why it matters: It avoids blind copying — the student learns to focus like the teacher, but only when appropriate.

# What Are We Changing and Exploring?

what are we changing?

1. We replace traditional training or ensembling approaches with a Knowledge Distillation framework, where a compact student model learns from both:
   - The true labels (hard supervision)

   - The soft output distributions of a larger teacher (soft supervision)

# What Are We Changing and Exploring?

what are we exploring?

1. Beyond Standard KD:
   - Can we achieve significant gains by enhancing KD with attention and alignment mechanisms?
2. Student Efficiency vs. Teacher Performance:
   - How close can a smaller, cheaper student get to a large teacher's accuracy?
3. Role of Attention & Alignment:
   - Do attention transfer and masked alignment lead to better generalization and robustness?

# Knowledge Distillation in image classification task

- Goal: Enhance the performance of a lightweight neural network by distilling knowledge from a larger, well-trained Teacher model.

- Method: Knowledge Distillation (KD) – Apply KD to transfer soft targets and internal representations (e.g., attention maps) from the Teacher to the Student.

- Focus: Accuracy, efficiency, and generalization.

# Overview-

1. Baseline (Student Only – No KD)

- Model: Lightweight ResNet18 trained from scratch
- Supervision: Cross-entropy loss with ground-truth labels only
- Goal: Serve as a lower-bound reference for student performance

# Overview-

2. KD (Standard Knowledge Distillation)

- Teacher: Pretrained ResNet50 with CBAM
- Student: ResNet18

- Supervision:
  - Cross-entropy loss (CE)
  - KL Divergence loss (KD) on soft logits from the teacher

- Goal: Teach student via soft labels and improve accuracy

# Overview-

3. KD + Alignment

- Additional Component: Masked Attention Alignment Loss
- Steps:
    - Extract attention maps from both teacher and student
    - Generate causal mask from teacher (via gradients)
    - Align student's attention only in masked regions
- Objective: Encourage spatial agreement without requiring negative samples
  Loss: MSE between masked attention maps

# Overview-

4. KD + Attention Transfer

- Additional Component: Masked Attention Transfer
- Steps:
  - Compute squared attention activations
  - Mask irrelevant regions using the causal mask
  - Minimize MSE on masked areas
- Objective: Guide student to focus like the teacher, but only on useful areas

# Overview -

5. KD + Attention + Alignment (Full Method)

- Full Loss Components

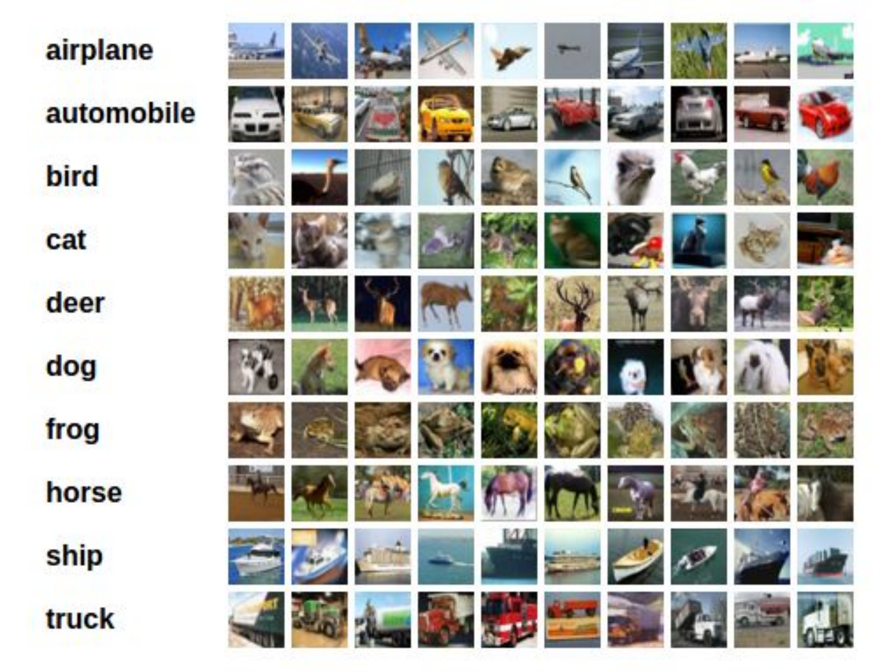
- Combines hard labels, soft logits, masked attention transfer, and masked attention alignment

- Goal: Achieve best accuracy, generalization, and robustness

# Dataset and Preprocessing

**Dataset:** CIFAR-10 – 60,000 color images (32x32), 10 balanced classes.

- **Split:** 50k training/10k validation
- **Preprocessing:**
    - Random horizontal flip
    - Random crop with padding
    - Normalization using dataset mean and std



# Training Strategy

- **Step 1:** Forward input through teacher → extract attention & causal mask
- **Step 2:** Forward same batch through student
- **Step 3:** Compute relevant loss terms (CE, KD, masked MSE, alignment)
- **Step 4:** Backpropagate & update student only

# Evaluation Metrics

- Top-1 Accuracy on validation set
- Model size (parameter count )
- Inference time per sample

# Methodology

## **Methodology Overview:** Comparative Knowledge Distillation Approaches

- Objective: Compare multiple distillation strategies using CIFAR-10
- Teacher: ResNet-50
- Student: ResNet-18

**Methods Compared:**

1. Baseline
2. KD
3. KD + Alignment
4. KD + Attention Transfer
5. KD + Both

**Method 1 – Baseline:** Training with cross-entropy loss only

- Cross-entropy loss with ground-truth labels only

- Used between the student model's output and the true class labels (i.e., hard targets).

$$\mathcal{L}_{CE} = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i)$$

## Method 2: KD (Standard Knowledge Distillation)

- Cross-entropy loss (CE)

- KL Divergence loss (KD) on soft logits from the teacher
  a. Goal: Mimic the teacher's soft probability distribution
  b. Captures teacher's knowledge of class similarities, helping student generalize better

$$\mathcal{L}_{KD} = \sum \hat{p}^T_{\text{teacher}} \cdot \log\left(\frac{\hat{p}^T_{\text{teacher}}}{\hat{p}^T_{\text{student}}}\right)$$

## Method 3: KD + Alignment

- Additional Component: Masked Attention Alignment Loss

- **Loss:** MSE between masked attention maps
    a. Only compute MSE where the teacher's mask indicates high importance.

$$\mathcal{L}_{\text{Align}} = \frac{1}{N}\sum_{i=1}^{N} M_i \cdot \left(A_i^{\text{student}} - A_i^{\text{teacher}}\right)^2$$

## Method 4: KD + Attention Transfer

- **Additional Component:** Masked Attention Transfer
- **Loss:** Minimize MSE on masked areas
- **Objective:** Guide student to focus like the teacher, but not only on useful areas

$$\mathcal{L}_{\text{attn}} = \text{MSE}(A_{\text{student}} \odot M, \ A_{\text{teacher}} \odot M)$$

## Method 5: KD + Attention + Alignment (Full Method)

- **Full Loss Components**

- Combines hard labels, soft logits, masked attention transfer, and masked attention alignment

- **Goal**: Achieve best accuracy, generalization, and robustness

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{KD}} + \beta \cdot \mathcal{L}_{\text{Align}} + \gamma \cdot \mathcal{L}_{\text{Attn}}$$

# Method Comparison Summary

| Model | Architecture | Supervision Type | Additional Components | Loss Functions | Objective / Goal |
|---|---|---|---|---|---|
| 🔹 **1. Baseline** (Student Only – No KD) | ResNet18 (lightweight, from scratch) | Ground-truth only | None | **Cross-Entropy (CE)** | Serve as lower-bound for student model performance |
| 🔹 **2. Standard KD** (Knowledge Distillation) | Teacher: ResNet50 + CBAM<br>Student: ResNet18 | Hard & Soft labels | None | **Cross-Entropy (CE)<br>KL Divergence (KD)** | Improve student by mimicking soft logits of teacher |
| 🔹 **3. KD + Alignment** | Same as above | Hard & Soft labels + masked attention regions | Masked Attention Alignment | **CE + KD + MSE (masked attn alignment)** | Align spatial attention between teacher and student using causal masking |
| 🔹 **4. KD + Attention Transfer** | Same as above | Hard & Soft labels + masked attention regions | Masked Attention Transfer | **CE + KD + MSE (masked attn transfer)** | Transfer squared attention maps in causal regions to guide focus |
| 🔹 **5. Full Method** (KD + Attention + Alignment) | Same as above | Hard & Soft labels + attention alignment and transfer | Masked Attention Transfer + Masked Attention Alignment | **CE + KD + $\lambda_1 \cdot$ MaskedAttn + $\lambda_2 \cdot$ Alignment** | Combine all techniques for best generalization, robustness, and accuracy |

# Hyperparameters Training Configuration

| Model | Learning Rate | Optimizer | Momentum | Weight Decay | Epochs | Early Stopping | Batch Size |
|---|---|---|---|---|---|---|---|
| **1. Baseline** (Student Only) | 0.01 | SGD | 0.9 | 5e-4 | 30 | Yes | 64 |
| **2. KD (Standard)** | 0.01 | SGD | 0.9 | 5e-4 | 30 | Yes | 64 |
| **3. KD + Alignment** | 0.01 | SGD | 0.9 | 5e-4 | 30 | Yes | 64 |
| **4. KD + Attention Transfer** | 0.01 | SGD | 0.9 | 5e-4 | 30 | Yes | 64 |
| **5. Full Method** (KD + Attention + Alignment) | 0.01 | SGD | 0.9 | 5e-4 | 30 | Yes | 64 |

# Summary & Fair Comparison

- Same teacher/student architecture across all methods
- Same dataset, preprocessing, and training setup
- Controlled evaluation with identical hyperparameters
- Early stopping ensures unbiased performance comparison
- Enables fair ablation of Alignment & Attention impact

# Training Strategy

1. Forward input batch through the teacher
   a. Extract intermediate attention maps
   b. Compute causal masks based on teacher gradients
2. Forward same input batch through the student
   a. Get student predictions
   b. Extract corresponding attention maps
3. Compute total loss
   a. KD loss (e.g., KL-divergence with soft targets)
   b. classification loss (e.g., CrossEntropy)
   c. Attention alignment loss (e.g., MSE between student and teacher maps over causal regions)
4. Backpropagate total loss and update student weights
   a. Use optimizer to update student parameters

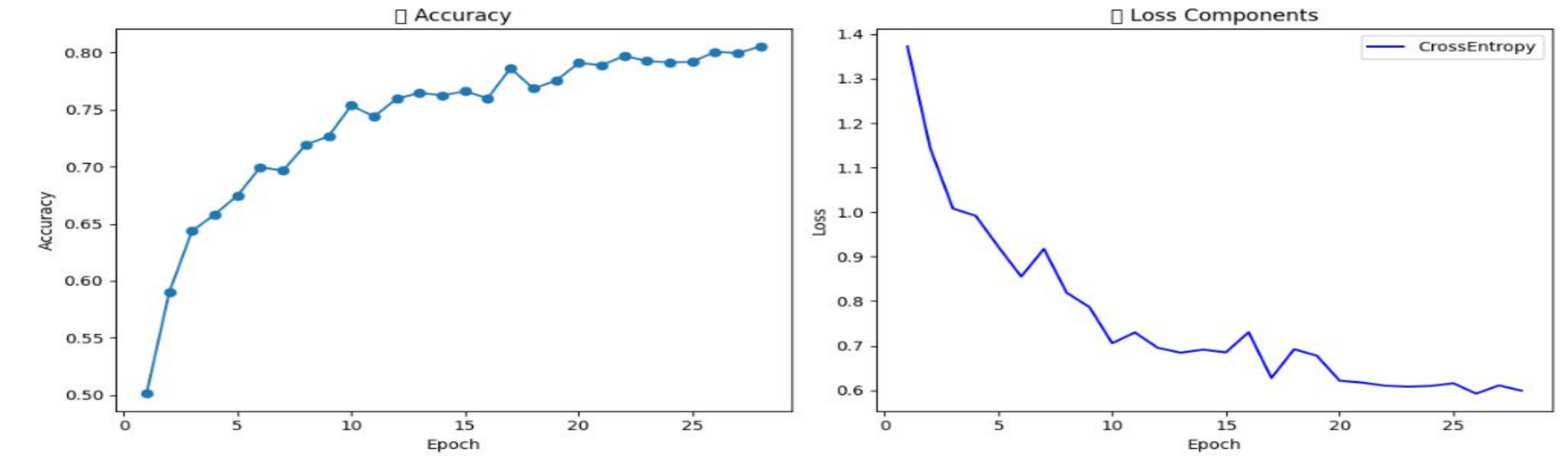
```
🌱 Epoch 20/30
🟢 Train Loss: 0.3433 | Accuracy: 0.8791
🔵 Test  Loss: 0.4560 | Accuracy: 0.8513
💾 נשמר המודל הטוב ביותר ✅

🌱 Epoch 21/30
🟢 Train Loss: 0.3318 | Accuracy: 0.8830
🔵 Test  Loss: 0.4567 | Accuracy: 0.8494

🌱 Epoch 22/30
🟢 Train Loss: 0.3338 | Accuracy: 0.8817
🔵 Test  Loss: 0.5091 | Accuracy: 0.8356

🌱 Epoch 23/30
🟢 Train Loss: 0.3219 | Accuracy: 0.8885
🔵 Test  Loss: 0.4796 | Accuracy: 0.8441

🌱 Epoch 24/30
🟢 Train Loss: 0.3102 | Accuracy: 0.8907
🔵 Test  Loss: 0.4659 | Accuracy: 0.8488

🌱 Epoch 25/30
🟢 Train Loss: 0.3109 | Accuracy: 0.8900
🔵 Test  Loss: 0.4646 | Accuracy: 0.8497

🌱 Epoch 26/30
🟢 Train Loss: 0.2980 | Accuracy: 0.8959
🔵 Test  Loss: 0.4431 | Accuracy: 0.8558
💾 נשמר המודל הטוב ביותר ✅

🌱 Epoch 27/30
🟢 Train Loss: 0.2889 | Accuracy: 0.8981
🔵 Test  Loss: 0.5032 | Accuracy: 0.8379

🌱 Epoch 28/30
🟢 Train Loss: 0.2830 | Accuracy: 0.9010
🔵 Test  Loss: 0.4659 | Accuracy: 0.8529

🌱 Epoch 29/30
🟢 Train Loss: 0.2812 | Accuracy: 0.9011
🔵 Test  Loss: 0.4796 | Accuracy: 0.8492

🌱 Epoch 30/30
🟢 Train Loss: 0.2787 | Accuracy: 0.9032
🔵 Test  Loss: 0.5252 | Accuracy: 0.8310

🎉 האימון הסתיים! דיוק מירבי: 0.8558
```

# Teacher training for the task

| Metric | Best Epoch (26/30) | Final Epoch (30/30) | Comments |
|---|---|---|---|
| Train Loss | 0.2980 | 0.2787 | Continued to decrease slightly |
| Train Accuracy | 0.8959 | 0.9032 | Slight improvement |
| **Test Loss** | 0.4431 | 0.5252 | *Lowest at Epoch 26* – then worsened |
| **Test Accuracy** | 0.8558 | 0.8310 | *Highest at Epoch 26* – slight decline |
| Early Stopping | Not Used | Not Used | Model continued past optimum |
| Optimization Outcome | Local Optimum Reached | Slight Overfitting | Training beyond Epoch 26 not beneficial |

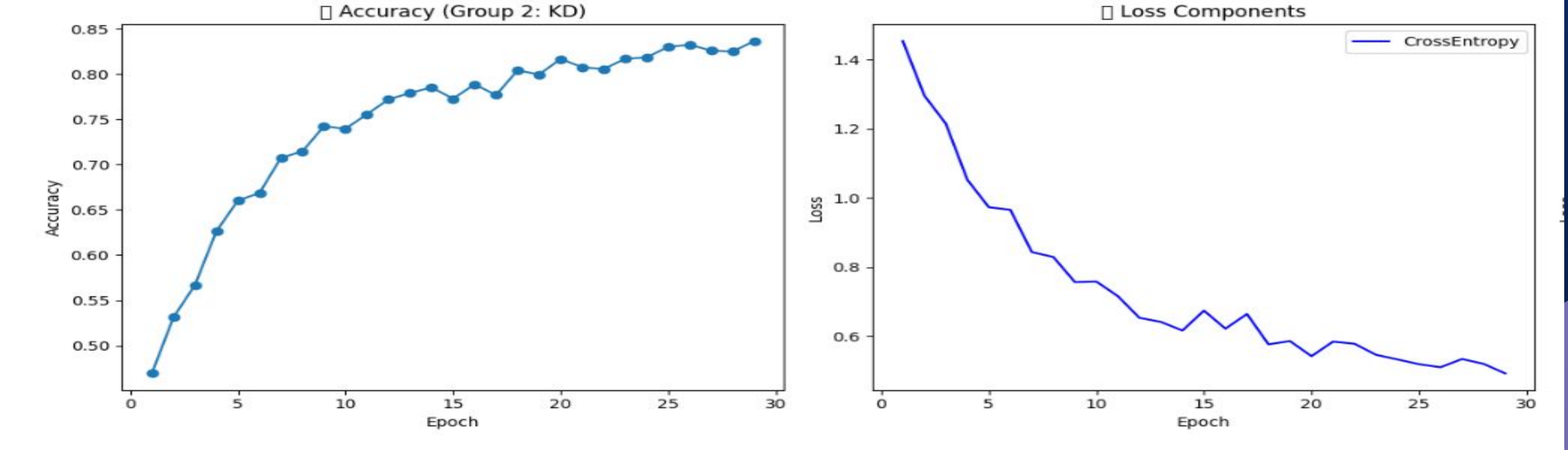
# Method 1 - baseline

```
Train Acc: 0.8287 | 🔹 Val Acc: 0.7920

Baseline Epoch 26/30
Train Acc: 0.8347 | 🔹 Val Acc: 0.8008
✅ שמירה של המודל עם הביצועים הטובים ביותר

Baseline Epoch 27/30
Train Acc: 0.8358 | 🔹 Val Acc: 0.7996

Baseline Epoch 28/30
Train Acc: 0.8400 | 🔹 Val Acc: 0.8056
✅ שמירה של המודל עם הביצועים הטובים ביותר

Baseline Epoch 29/30
Train Acc: 0.8428 | 🔹 Val Acc: 0.8054

Baseline Epoch 30/30
Train Acc: 0.8452 | 🔹 Val Acc: 0.7999
```

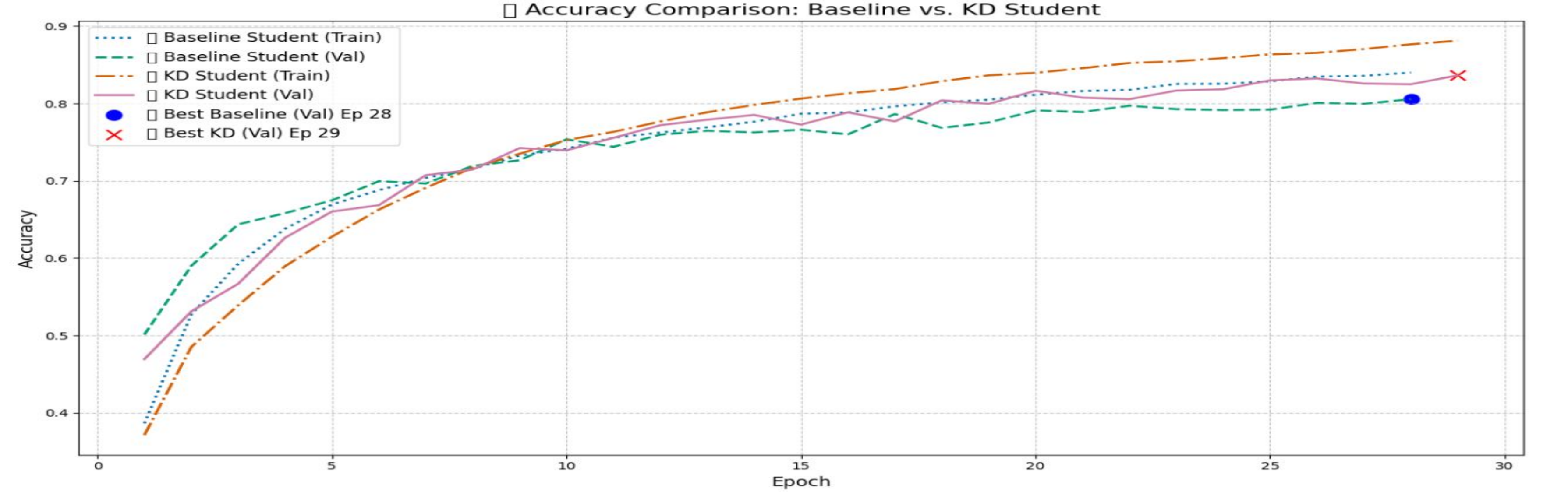


# Method 2 - KD

```
🎓 KD Epoch 25/30
📊 Train Acc: 0.8636 | Test Acc: 0.8300
💾 ✅ שמירה של המודל הטוב ביותר

🎓 KD Epoch 26/30
📊 Train Acc: 0.8655 | Test Acc: 0.8324
💾 ✅ שמירה של המודל הטוב ביותר

🎓 KD Epoch 27/30
📊 Train Acc: 0.8704 | Test Acc: 0.8259

🎓 KD Epoch 28/30
📊 Train Acc: 0.8766 | Test Acc: 0.8249

🎓 KD Epoch 29/30
📊 Train Acc: 0.8812 | Test Acc: 0.8364
💾 ✅ שמירה של המודל הטוב ביותר

🎓 KD Epoch 30/30
📊 Train Acc: 0.8844 | Test Acc: 0.8316
```



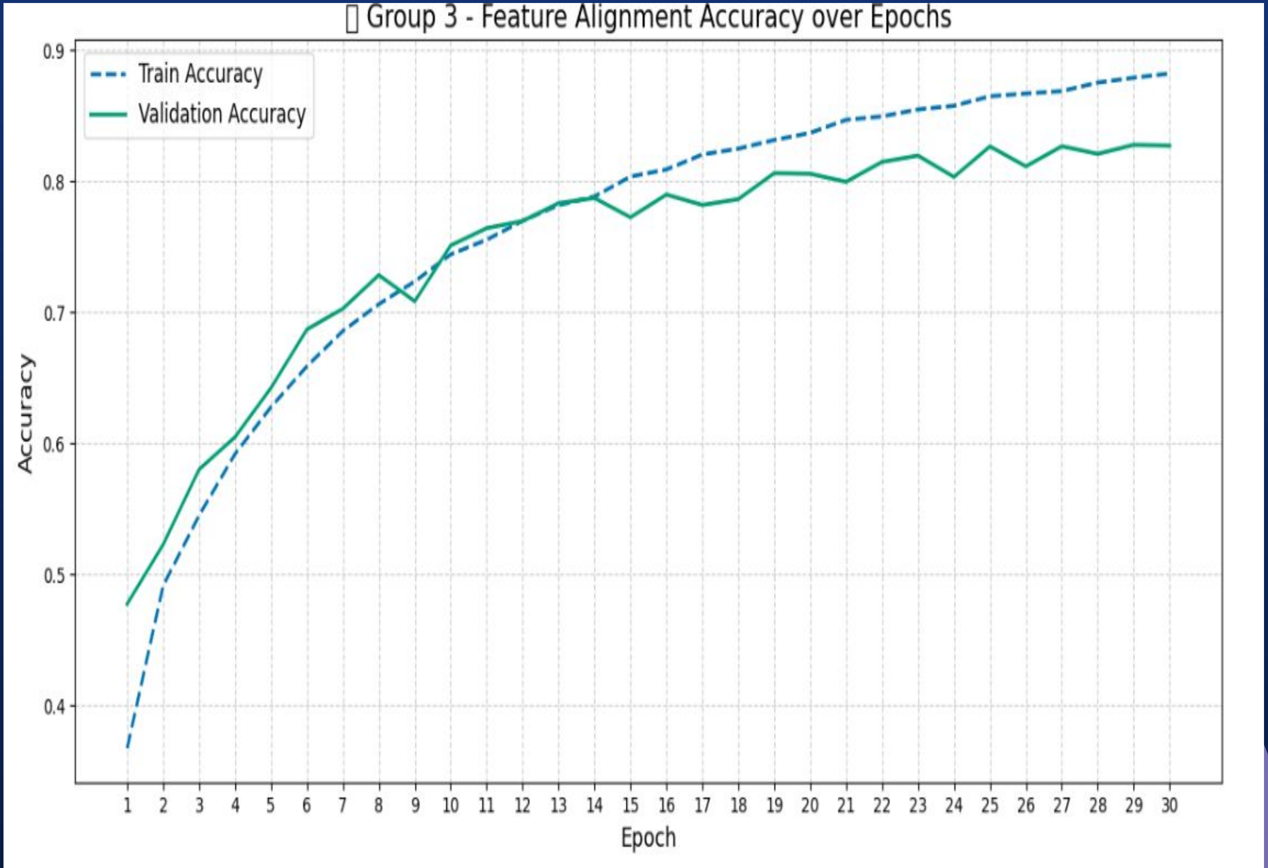
# Accuray comparison on training and val - baseline vs KD




| | Model | Best Train Accuracy | Best Validation Accuracy |
|---|---|---|---|
| 0 | Baseline Student | 0.8400 | 0.8056 |
| 1 | KD Student | 0.8812 | 0.8364 |

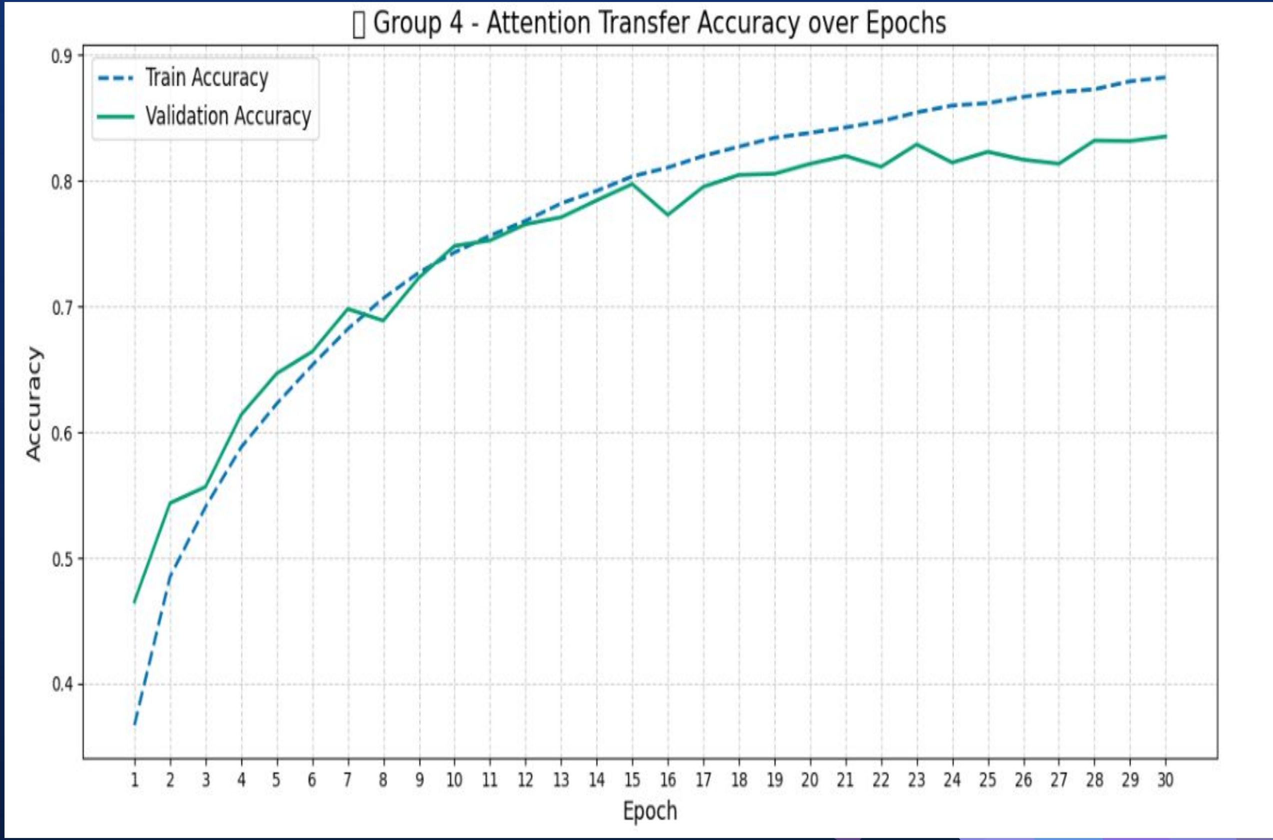
# Method 3 - KD + Alignment

```
Group 3 Epoch 23/30
Train Acc: 0.8549 | Val Acc: 0.8196
Loss CE: 0.4168 | Align: 0.0041 | Total: 0.2104
New best model saved! Val Acc: 0.8196

Group 3 Epoch 24/30
Train Acc: 0.8576 | Val Acc: 0.8033
Loss CE: 0.4044 | Align: 0.0041 | Total: 0.2043
No improvement. Patience: 1/7

Group 3 Epoch 25/30
Train Acc: 0.8649 | Val Acc: 0.8266
Loss CE: 0.3851 | Align: 0.0041 | Total: 0.1946
New best model saved! Val Acc: 0.8266

Group 3 Epoch 26/30
Train Acc: 0.8670 | Val Acc: 0.8114
Loss CE: 0.3772 | Align: 0.0041 | Total: 0.1907
No improvement. Patience: 1/7

Group 3 Epoch 27/30
Train Acc: 0.8689 | Val Acc: 0.8267
Loss CE: 0.3715 | Align: 0.0041 | Total: 0.1878
New best model saved! Val Acc: 0.8267

Group 3 Epoch 28/30
Train Acc: 0.8753 | Val Acc: 0.8209
Loss CE: 0.3572 | Align: 0.0041 | Total: 0.1807
No improvement. Patience: 1/7

Group 3 Epoch 29/30
Train Acc: 0.8790 | Val Acc: 0.8278
Loss CE: 0.3442 | Align: 0.0041 | Total: 0.1742
New best model saved! Val Acc: 0.8278

Group 3 Epoch 30/30
Train Acc: 0.8822 | Val Acc: 0.8272
Loss CE: 0.3349 | Align: 0.0041 | Total: 0.1695
No improvement. Patience: 1/7

Best Val Acc: 0.8278 at Epoch 29
```

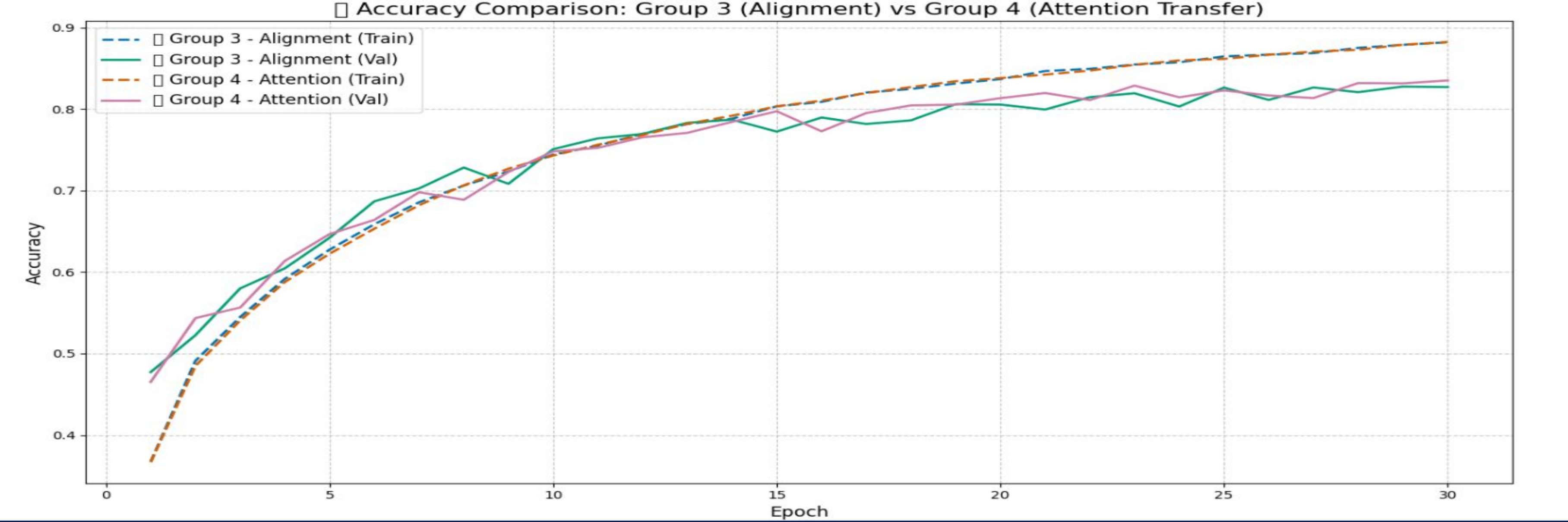


# Method 4 - KD + Attention Transfer

```
🌐 Group 4 Epoch 23/30
📊 Train Acc: 0.8545 | Val Acc: 0.8290
🧪 Losses => CE: 0.4144 | Attention: 0.0009 | Total: 0.2077
💾 New best model saved! Val Acc: 0.8290

🌐 Group 4 Epoch 24/30
📊 Train Acc: 0.8599 | Val Acc: 0.8146
🧪 Losses => CE: 0.3987 | Attention: 0.0009 | Total: 0.1998
⏳ No improvement. Patience: 1/7

🌐 Group 4 Epoch 25/30
📊 Train Acc: 0.8618 | Val Acc: 0.8231
🧪 Losses => CE: 0.3916 | Attention: 0.0009 | Total: 0.1963
⏳ No improvement. Patience: 2/7

🌐 Group 4 Epoch 26/30
📊 Train Acc: 0.8669 | Val Acc: 0.8169
🧪 Losses => CE: 0.3767 | Attention: 0.0009 | Total: 0.1888
⏳ No improvement. Patience: 3/7

🌐 Group 4 Epoch 27/30
📊 Train Acc: 0.8708 | Val Acc: 0.8137
🧪 Losses => CE: 0.3704 | Attention: 0.0009 | Total: 0.1857
⏳ No improvement. Patience: 4/7

🌐 Group 4 Epoch 28/30
📊 Train Acc: 0.8729 | Val Acc: 0.8321
🧪 Losses => CE: 0.3604 | Attention: 0.0009 | Total: 0.1807
💾 New best model saved! Val Acc: 0.8321

🌐 Group 4 Epoch 29/30
📊 Train Acc: 0.8792 | Val Acc: 0.8317
🧪 Losses => CE: 0.3463 | Attention: 0.0009 | Total: 0.1736
⏳ No improvement. Patience: 1/7

🌐 Group 4 Epoch 30/30
📊 Train Acc: 0.8824 | Val Acc: 0.8353
🧪 Losses => CE: 0.3353 | Attention: 0.0009 | Total: 0.1681
💾 New best model saved! Val Acc: 0.8353

🏁 Best Val Acc: 0.8353 at Epoch 30
```



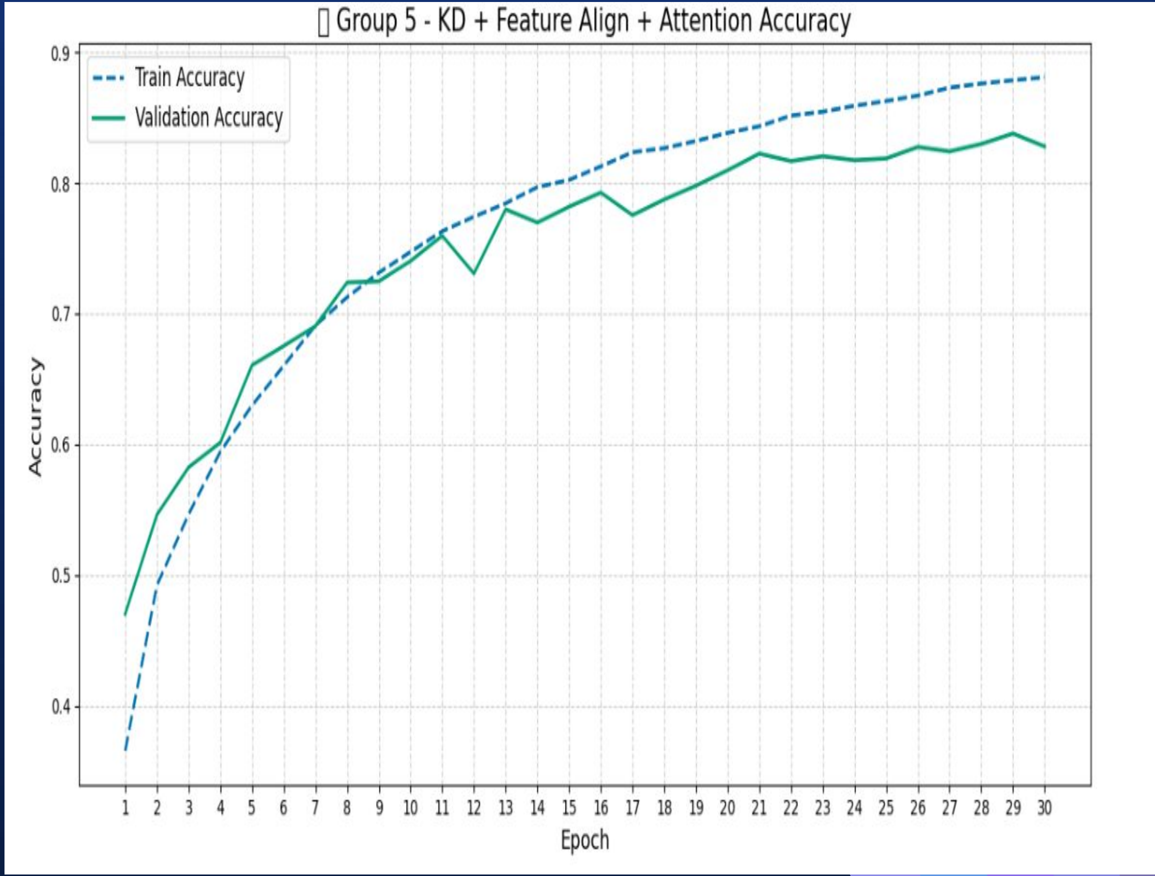
# Method 3 Vs Method 4




| | Model | Epoch | Train Accuracy | Val Accuracy |
|---|---|---|---|---|
| 0 | Group 3 (Align) | 29 | 0.87902 | 0.8278 |
| 1 | Group 4 (Attention) | 30 | 0.88236 | 0.8353 |

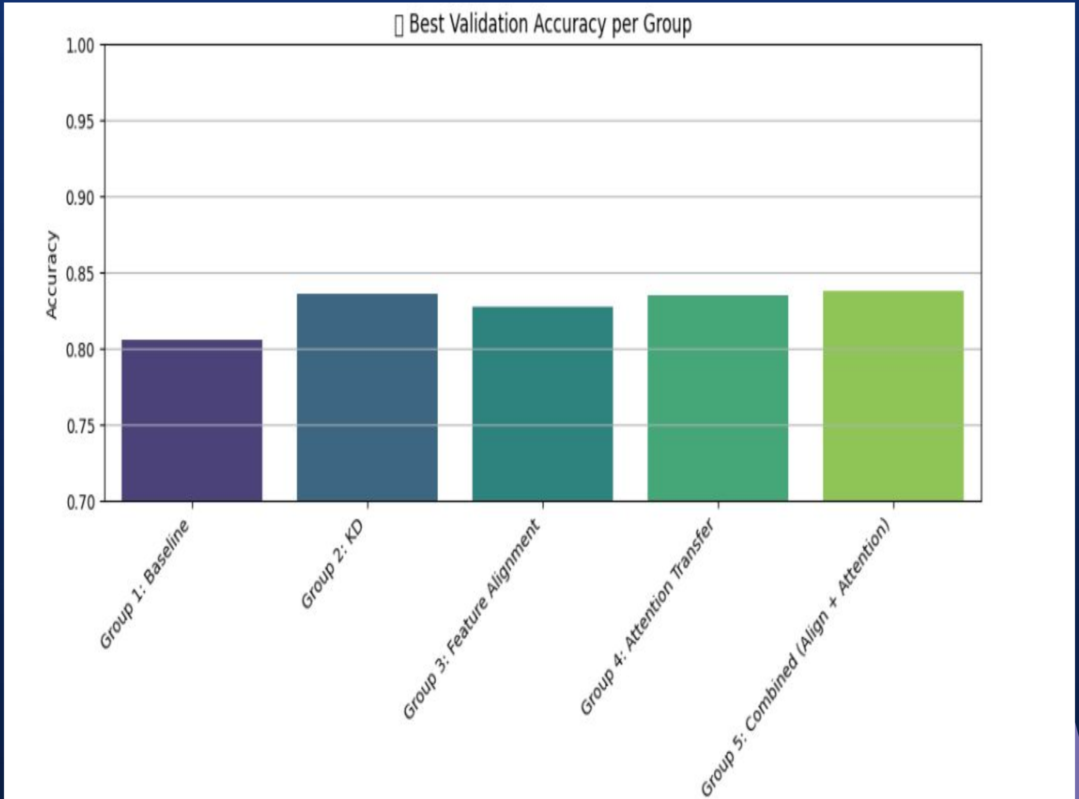
# Method 5 KD + Attention Transfer + Alignment

```
Group 5 Epoch 23/30
Train Acc: 0.8543 | Val Acc: 0.8203
Losses | CE: 0.5088, Feat: 0.0041, Att: 0.0009, Total: 0.2085
No improvement. Patience: 2/7

Group 5 Epoch 24/30
Train Acc: 0.8589 | Val Acc: 0.8173
Losses | CE: 0.4539, Feat: 0.0041, Att: 0.0009, Total: 0.2019
No improvement. Patience: 3/7

Group 5 Epoch 25/30
Train Acc: 0.8625 | Val Acc: 0.8186
Losses | CE: 0.6118, Feat: 0.0041, Att: 0.0009, Total: 0.1960
No improvement. Patience: 4/7

Group 5 Epoch 26/30
Train Acc: 0.8667 | Val Acc: 0.8274
Losses | CE: 0.3167, Feat: 0.0040, Att: 0.0009, Total: 0.1919
New best model saved! Val Acc: 0.8274

Group 5 Epoch 27/30
Train Acc: 0.8728 | Val Acc: 0.8240
Losses | CE: 0.3407, Feat: 0.0041, Att: 0.0009, Total: 0.1839
No improvement. Patience: 1/7

Group 5 Epoch 28/30
Train Acc: 0.8759 | Val Acc: 0.8297
Losses | CE: 0.4371, Feat: 0.0040, Att: 0.0009, Total: 0.1785
New best model saved! Val Acc: 0.8297

Group 5 Epoch 29/30
Train Acc: 0.8784 | Val Acc: 0.8377
Losses | CE: 0.5948, Feat: 0.0041, Att: 0.0009, Total: 0.1758
New best model saved! Val Acc: 0.8377

Group 5 Epoch 30/30
Train Acc: 0.8808 | Val Acc: 0.8278
Losses | CE: 0.3342, Feat: 0.0041, Att: 0.0009, Total: 0.1707
No improvement. Patience: 1/7

Best Val Acc: 0.8377 at Epoch 29
```



# Results of all of our 5 models

| | Group | Best Validation Accuracy |
|---|---|---|
| 0 | Group 1: Baseline | 0.8056 |
| 1 | Group 2: KD | 0.8364 |
| 2 | Group 3: Feature Alignment | 0.8278 |
| 3 | Group 4: Attention Transfer | 0.8353 |
| 4 | Group 5: Combined (Align + Attention) | 0.8377 |



# Questions ?