ICLR
International Conference On Learning Representations

KLAS

Using Similarity to Stitch Neural Networks for Improved Accuracy-Efficiency Tradeoffs

Debopam Sanyal, Anantharaman Iyer, Alind Khare, Trisha Jain, Akshay Jajoo, Myungjin Lee, Clayton Kerce, Alexey Tumanov

Georgia Institute of Technology

The Challenge: Flexible Model Deployment

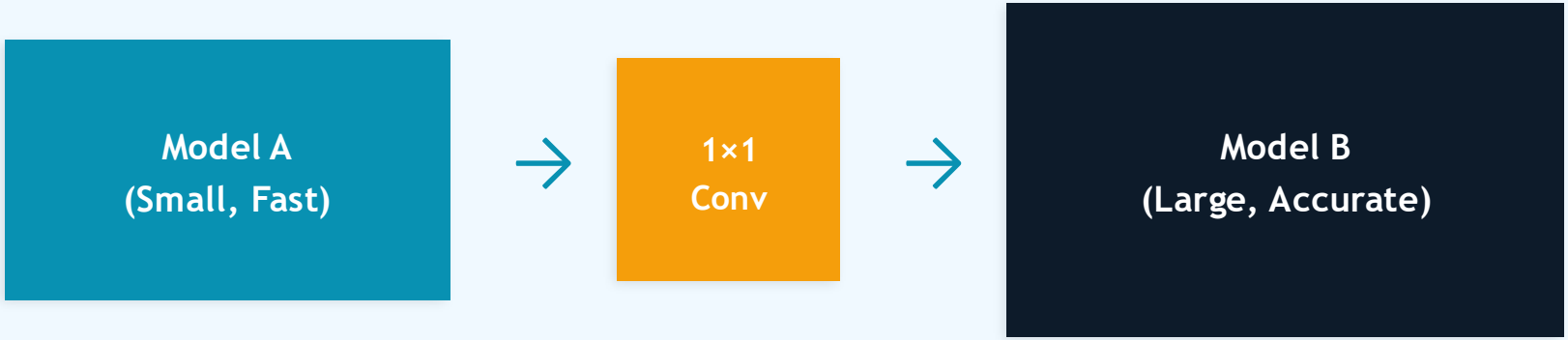
Problem

- Diverse deployment targets demand flexible model selection
- Training separate models for each budget is expensive
- Model stitching creates interpolated networks, but existing methods use heuristics
- $O(k^2n^2)$ possible stitch configs, which ones to pick?

Our Answer: KLAS

- Principled stitch selection using KL divergence
- Automatically identifies promising stitches
- No additional finetuning cost vs. baselines
- Generalizes across model families

Background: Model Stitching



Stitchable Neural Networks (SN-Net)
take pretrained models (anchors) and connect them via simple 1×1 conv stitch layers.

Key Limitation: SN-Net uses heuristic "nearest and paired/unpaired stitching" → limits to adjacent anchor pairs and predefined block mappings → misses optimal configurations.

The Search Space Problem

How many stitches are possible?

3	12	~1,296
Anchors (e.g. DeiT Ti/S/B)	Blocks per anchor	Possible binary stitch configs

Brute-force is impractical. Training all configs wastes compute. SN-Net's heuristic picks ~3%, but often a suboptimal 3%. KLAS identifies the top configurations before any finetuning.

KLAS: Method Overview

Core Insight: Stitching success depends on activation similarity between pretrained models.

1 Train Linear Probes

Attach linear probes to every block of each anchor to get softmax distributions.

2 Anchor Selection

Compute KL divergence between final-block probes. Low KL → compatible models.

3 Block Selection

Compute stitch score (Γ) for all block pairs. Select top-scoring stitches.

Step 1-2: Probes & Anchor Selection

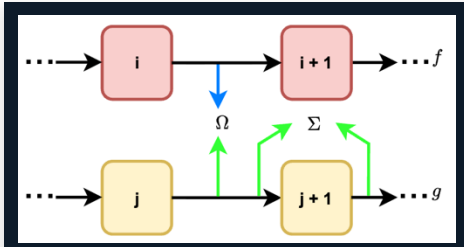
Training Linear Probes

- One linear layer attached after each block
- Maps block activations → softmax over distribution classes
- Trained on the validation split only
- **Eg: DeiT: 36 probes | Swin: 60 probes**
- Lightweight → adds negligible overhead

Selecting Compatible Anchors

- Compare last-block probe outputs across all anchor pairs
- KL divergence measures agreement on class-confidence distributions
- **Low KL → similar decision boundaries → compatible for stitching**
- Replaces SN-Net's restriction to nearest-complexity pairs
- Enables non-adjacent and cross-architecture anchor pairing

Step 3: Block Selection via Stitch Score



$$\Theta(P_i^f, P_j^g) = \frac{\sum_{x \in \mathcal{D}_v} D_{KL}\left(P_i^f(x) \,||\, P_j^g(x)\right)}{|\mathcal{D}_v|}$$

$$\Gamma(i,j) = \frac{\overbrace{\Theta(P_i^f, P_j^g)}^{\text{cross-anchor activation distance } (\Omega)}}{\underbrace{\Theta(P_j^g, P_{j+1}^g)}_{\text{intra-anchor block capacity } (\Sigma)}}$$

Low Γ → blocks produce similar class predictions → good stitch point

How It Works

Compute Γ for every (block_i, block_j) pair across selected anchors. Rank all pairs by score. The lowest-Γ pairs become the stitch configurations to finetune.

Negligible Selection Cost

No stitch layer is instantiated or trained during selection. Probes training is negligible and a one-time cost. The entire scoring uses only the softmax distributions form the probes.

Why KL Divergence?

Similarity Metric	Config Overlap
MSE	Low
Cross-Entropy	Low
CKA	Low
Dist. Matching	Low
KL Divergence	High ✓

Key Finding

KL divergence between softmax probe outputs captures whether two models share compatible decision boundaries, critical for stitching success.

CKA (a popular metric) recovers only ~5% of optimal stitch configurations

Experimental Setup

Models & Data

- DeiT-Ti/S/B (plain ViTs)
- Swin-Ti/S/B (hierarchical ViTs)
- LeViT (lightweight ViTs)
- ResNet-18/34/50 (CNNs)
- Swin ↔ ResNet (cross-architecture)
- **ImageNet-1K (1.28M train images)**

Training Details

- **50 epochs finetuning (same as SN-Net)**
- FLOPs as efficiency proxy
- KL values from validation splits
- Accuracy reported on test splits
- AUC of accuracy-efficiency curve as primary metric
- **Similar budget for KLAS and all baselines**

Experimental Results

+1.21%

Top-1 Accuracy
gain at same FLOPs

1.33×

FLOPs Reduction
at same accuracy

Evaluated across multiple model families:

DeiT Ti / S / B	Swin Ti / S / B	LeViT	ResNet (CNN)	Cross-Arch Swin ↔ ResNet
Best AUC	Best AUC	Best AUC	Best AUC	Best AUC

Cross-Architecture Stitching

KLAS generalizes beyond same-family stitching

Swin ↔ ResNet (CNN ↔ Transformer)

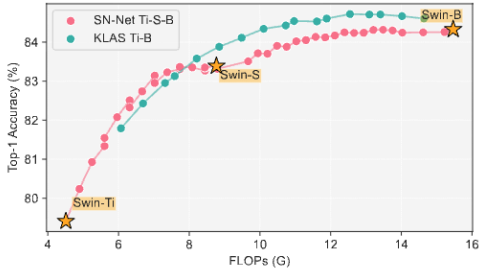
- Stitching across fundamentally different architectures
- KLAS achieves best AUC on the accuracy-efficiency frontier
- SN-Net's heuristic mapping fails, different block structures

Why KLAS Succeeds Here

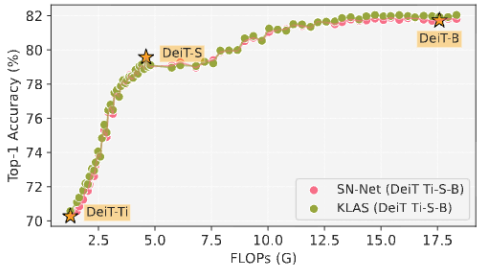
- KL measures functional compatibility and structural similarity
- Probes normalize different architectures into shared softmax space
- No fixed block mapping, adapts to actual anchor weights

AUC: Accuracy-Efficiency Frontier

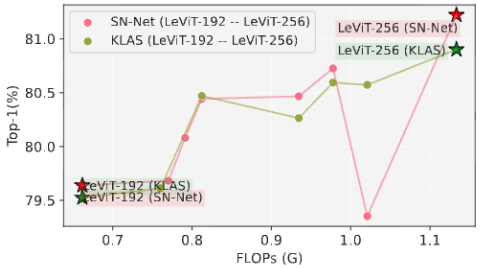
Area Under the accuracy-efficiency Curve (higher = better)



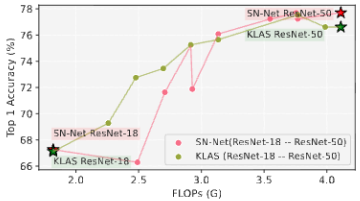
(a): Swin, ΔAUC : +0.06



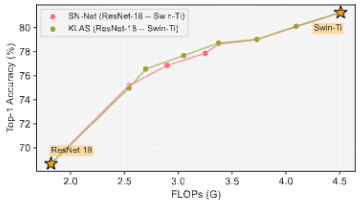
(b): DeiT, ΔAUC : +0.012



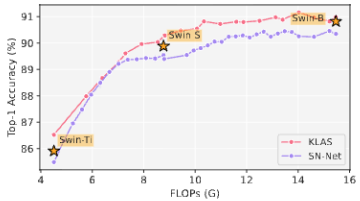
(c): LeViT, ΔAUC : +0.006



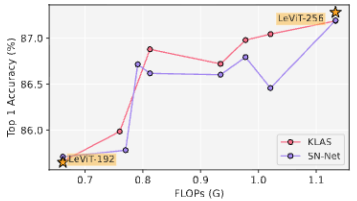
(d): ResNet, ΔAUC : +0.014



(e): ResNet-Swin, ΔAUC : +0.002



(f): Swin CIFAR-100, ΔAUC : +0.014



(g): LeViT CIFAR-100, ΔAUC : +0.008

Ablation Studies & Analysis

Anchor Selection Matters

Removing anchor selection (using all pairs) degrades the accuracy-efficiency curve. KL-based anchor filtering eliminates incompatible model pairs early.

Stitch Score Predicts Accuracy

Block pairs with lower Γ consistently achieve higher post-finetuning accuracy. The correlation validates KL as a reliable selection criterion.

Sparse but Superior

KLAS produces fewer stitches at some FLOPs ranges (sparser set), but each selected stitch performs better: quality over quantity.

Conclusion & Takeaways

- KLAS is a principled stitch selection framework using KL divergence
- Up to +1.21% accuracy or 1.33× FLOPs reduction at no extra cost
- Generalizes across transformers, CNNs, and cross-architecture setups
- KL divergence outperforms MSE, CKA, CE, and distribution matching

Thank You!