Project Proposal Template

EE 641: Deep Learning Systems

Project Title: Attention Head Specialization in Vision Transformers Under Distribution Shift

Group Members: [Student Name 1, Student Name 2]

Abstract

We propose to investigate whether attention heads in vision transformers specialize for different functions and how this relates to robustness under distribution shift. Attention heads in language transformers learn distinct roles (syntax vs. semantics), but similar analysis for vision models remains limited. We will train vision transformers on ImageNet, systematically ablate individual attention heads, and measure performance on clean and corrupted test sets (Figure 1). This requires implementing head ablation mechanisms, analyzing attention patterns, and evaluating across multiple distribution shifts to understand which components contribute to robustness.

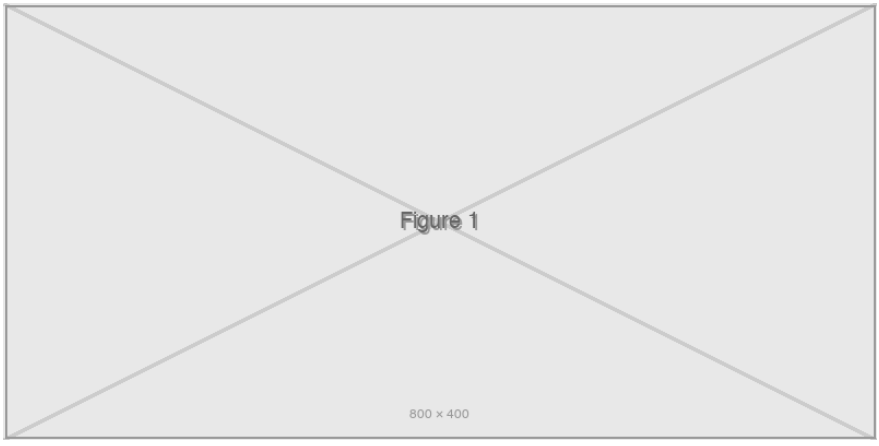

Figure 1: We will train vision transformers with different head configurations, ablate individual attention heads, and measure performance on clean ImageNet (top) versus corrupted variants (bottom). Color intensity indicates head importance—red heads cause large accuracy drops when removed, blue heads are redundant. This reveals whether heads specialize for robustness.

Your abstract should concisely describe what problem you will address, what you will build or investigate, and what technical requirements it involves. Including a figure or diagram early (with a substantive caption explaining what is shown) helps communicate your approach visually.

Introduction

Vision transformers achieve strong performance on image classification, but how they work internally is less clear than for CNNs. CNNs learn hierarchical features (edges → textures → objects), but the role of individual attention heads in ViTs is not well understood. This matters because understanding which heads are important could improve model efficiency and robustness.

Research on language transformers shows that attention heads specialize—some focus on syntax, others on semantics. Voita et al. (2019) and Michel et al. (2019) demonstrated that many heads can be pruned without hurting performance, suggesting redundancy. Whether vision transformers exhibit similar patterns is unclear. Images have different structure than text (spatial rather than sequential), so heads might [...]

A related question is how transformers handle distribution shift. Models trained on ImageNet often fail when tested on corrupted images (blur, noise, weather effects). Understanding which attention heads contribute to robustness could [...]

Your introduction should establish problem context, explain why it matters, provide necessary technical background, and describe your investigation approach.

Related Work

Dosovitskiy et al. (2021) introduced Vision Transformer (ViT), showing that transformers can match CNN performance on image classification. ViT splits images into patches and processes them through standard transformer layers. Touvron et al. (2021) developed DeiT with training improvements for smaller datasets. These establish transformers as viable for vision tasks, but how they work internally is less clear than for CNNs.

In natural language processing, several papers have analyzed attention head function. Voita et al. (2019) showed many heads in translation models are redundant. Michel et al. (2019) found 40% of BERT heads can be removed without major accuracy loss. Clark et al. (2019) visualized attention patterns revealing syntactic specialization. Not all attention heads contribute equally in language models.

For vision transformers, most interpretability work examines aggregate patterns rather than individual heads. Raghu et al. (2021) found ViTs attend to global structure while CNNs process locally. Dosovitskiy et al. (2021) visualized attention maps showing semantically meaningful patterns. Systematic ablation studies of individual heads—common in NLP—are less developed for vision models.

Hendrycks & Dietterich (2019) introduced ImageNet-C to measure robustness to corruptions (blur, noise, weather). Bhojanapalli et al. (2021) found ViTs show different robustness properties than CNNs, relying more on shape than texture. Paul & Chen (2022) showed attention mechanisms contribute to [...]

Your related work should cover foundation papers, prior approaches to similar problems, recent relevant developments, and the specific gap your project addresses.

Technical Approach

We will train vision transformers on ImageNet, systematically ablate individual attention heads, and measure performance on clean and corrupted test sets.

Design Decisions

Several design decisions reduce project scope to fit within a term timeline and compute budget while preserving the core investigation:

Model Size - ViT-Tiny vs. ViT-Base: We use ViT-Tiny (depth=8, hidden=192) rather than ViT-Base (depth=12, hidden=768). ViT-Tiny trains in 15-20 GPU-hours per model versus 100+ hours for ViT-Base. This brings total compute from 200+ GPU-hours down to ~40 GPU-hours. The model still has sufficient complexity—24 to 48 attention heads across 8 layers—to study specialization patterns. The research question doesn't require maximum accuracy; it requires comparing head behavior across conditions.

Number of Variants - Two vs. Many: We train two head configurations (3 and 6 heads per layer) rather than exploring a wider range (3, 6, 9, 12). Two configurations let us see if head count affects specialization while keeping training time reasonable. More variants would strengthen the analysis but aren't essential for a term project.

Training from Scratch vs. Fine-tuning: We train from scratch rather than fine-tuning pre-trained models. This costs more compute but lets us control the training process and observe how heads specialize during learning. For this project, understanding the training dynamics matters more than achieving maximum accuracy.

Full ImageNet vs. Subset: We use full ImageNet-1K rather than a subset. While a subset would reduce training time further, ImageNet is standard for robustness benchmarks (ImageNet-C, ImageNet-R, ImageNet-Sketch). Using the full dataset ensures our results are comparable to published work.

Model Architecture: We will train ViT-Tiny variants with different head counts (3, 6 heads per layer) to see if head count affects specialization. Maintaining constant hidden dimension lets us isolate the effect of head count from overall model capacity. The architecture consists of patch embedding, transformer blocks with multi-head attention, and a classification head.

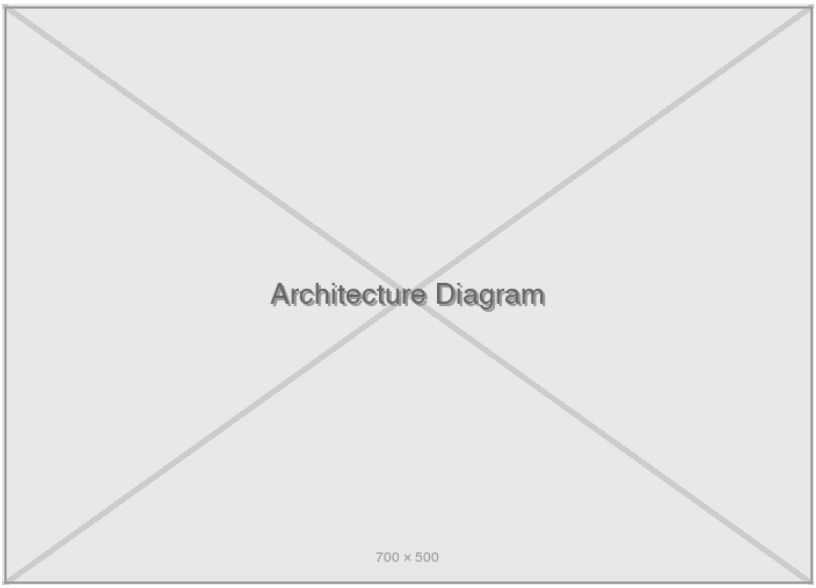

Figure 2: ViT-Tiny architecture showing patch embedding, transformer blocks with multi-head attention, and classification head. We will systematically ablate individual attention heads (shown in different colors) within each layer to measure their importance for clean and corrupted image classification.

Training: We will train on ImageNet-1K using the DeiT training setup: AdamW optimizer, cosine learning rate schedule, batch size 256, 300 epochs. This should achieve around 72% ImageNet accuracy for ViT-Tiny based on published results. Training requires approximately 15-20 GPU-hours per model on A100-class hardware.

Head Ablation: For each trained model, we will zero out individual attention heads and measure the accuracy drop. Heads that cause large drops when removed are important; heads with minimal impact are redundant. We will test this on both clean ImageNet validation and corrupted versions (ImageNet-C with blur, noise, weather effects).

Analysis: We will measure attention distance between heads using Jensen-Shannon divergence to quantify specialization. We will also compare which heads matter for clean versus corrupted data. If certain heads are critical only for corruptions, this suggests specialization for robustness.

Evaluation: Beyond ImageNet-C corruptions, we plan to test on ImageNet-R (artistic renditions) and ImageNet-Sketch (line drawings). These different shift types will show whether robustness generalizes or depends on specific heads.

Alternative Approaches: We considered using pre-trained models instead of training from scratch, which would be faster but wouldn't let us observe how heads specialize during training. We also considered fine-tuning on a smaller dataset (CIFAR-100), but ImageNet is necessary for [...]

Your technical approach should describe model architecture with key design decisions, training procedures, and evaluation strategy. Discuss alternative approaches and why you chose your method. For example: How would a human solve this problem? What features or patterns would they look for? Are there simpler baselines you considered?

Dataset Description

ImageNet-1K: We will train on the ILSVRC 2012 classification dataset containing 1.28M training images across 1000 classes. Images vary in resolution (resized to 224×224 for ViT input). Class distribution is balanced with approximately 1300 images per class. This dataset is standard for vision transformer research and allows comparison with published results.

ImageNet-C (Corruptions): 50,000 validation images with 15 corruption types applied at 5 severity levels. Corruptions include noise (Gaussian, shot, impulse), blur (defocus, motion, glass, zoom), weather (snow, frost, fog), and digital artifacts (JPEG compression, pixelation, contrast changes). Each corruption severity is calibrated based on human perception studies. This benchmark tests robustness to common real-world image degradations.

ImageNet-R (Renditions): 30,000 images of ImageNet classes in different artistic renditions including paintings, cartoons, graffiti, embroidery, graphics, and sketches. Classes overlap with ImageNet-1K but images differ substantially in style. This tests whether models rely on texture features versus shape and semantic features.

ImageNet-Sketch: 50,889 sketch images covering 1000 ImageNet classes, collected from search queries. Images are black-and-white line drawings representing objects. This extreme distribution shift tests [...]

Representative samples from each distribution shift type are shown below.

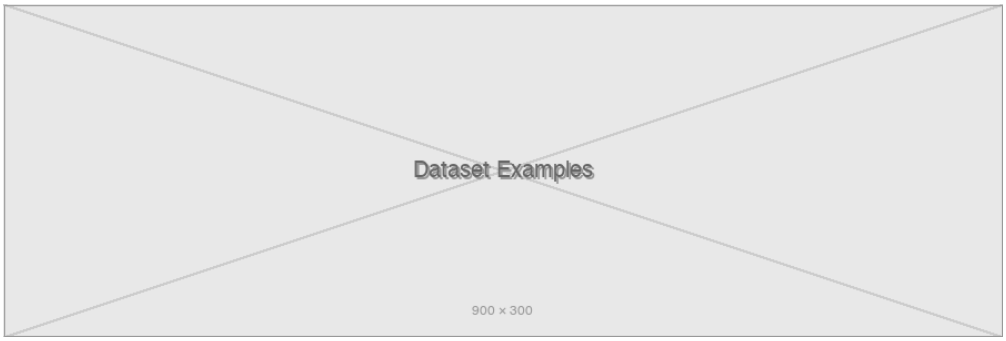

Figure 3: Dataset samples showing the same object class (e.g., "golden retriever") across different distribution shifts: clean ImageNet (left), ImageNet-C with corruption (center-left), ImageNet-R artistic rendition (center-right), and ImageNet-Sketch line drawing (right). These diverse test conditions reveal whether attention heads specialize for different types of robustness.

We will use standard data augmentation during training (random crop, horizontal flip, color jitter, RandAugment) following DeiT protocol. All datasets are publicly available for academic research. These benchmarks test multiple distribution shift types (corruptions, style changes, modality changes). Dataset sizes are large enough that per-head ablation measurements should have low variance.

Evaluation Metrics: We will measure top-1 accuracy on ImageNet validation and corruption robustness using mean Corruption Error (mCE):

$$\text{mCE} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{\text{Error}_c(\text{model})}{\text{Error}_c(\text{baseline})}$$

where $\mathcal{C}$ is the set of corruption types. Lower mCE indicates better robustness. We will also report per-corruption accuracy to identify which heads matter for specific shift types.

Your dataset description should characterize the data (size, structure, properties), explain appropriateness for your problem, and describe preprocessing and augmentation. Include evaluation metrics—show equations if they help clarify what you're measuring. Address whether datasets are representative, balanced, and sufficient. Note: Avoid using pre-packaged competition datasets (Kaggle, etc.) without substantial modification or novel analysis.

Architecture Investigation Plan

Baseline Model Training: We will start by training ViT-Tiny models with different head counts (3, 6 heads per layer) on ImageNet. The goal is to get models that achieve reasonable accuracy (~72% based on DeiT results) so we have a good baseline for ablation experiments. If training doesn't converge well, we'll need to debug (adjust learning rate, check data augmentation) before moving forward.

Ablation Experiments: Once we have trained models, we'll implement the head ablation mechanism—basically zeroing out individual heads and measuring how much accuracy drops. We'll systematically test each head to identify which ones are critical and which are redundant. We'll also compute metrics like attention distance between heads to quantify specialization.

Robustness Testing: After identifying important heads on clean ImageNet data, we'll test the same models on corrupted data (ImageNet-C, ImageNet-R, ImageNet-Sketch). The key question is whether the same heads that matter for clean data also matter for corrupted data, or if different heads handle robustness.

Understanding What Heads Do: For heads we identify as important, we'll visualize their attention patterns to understand what they're looking at. Are critical heads attending to edges? Textures? Global structure? This helps interpret why certain heads matter more than others.

Analysis and Writeup: We'll compile all results, look for patterns across the different head configurations we tested, and prepare the final analysis. If we have extra time, we might explore whether we can improve efficiency by pruning redundant heads.

Your architecture investigation plan should describe specific variants you will test, how you will isolate effects of design decisions, and the logical progression of experiments.

Estimated Compute Needs

Training Requirements: ViT-Tiny training requires approximately 15-20 GPU-hours per model variant on A100-class GPUs (40GB memory). We plan to train two variants (3 and 6 heads per layer), totaling around 40 GPU-hours. Batch size 256 should fit in memory. Training throughput gives roughly 1 day wall-clock time per model.

Compute Resources: We will use cloud GPU instances (e.g., Lambda Labs at ~$1/hour for A100, totaling $40-50 for training). Alternatively, institutional cluster access or Google Colab Pro could work, though Colab may require breaking training into sessions.

Evaluation Requirements: ImageNet validation evaluation takes approximately 10 minutes per run. Head ablation experiments (up to 48 per model: 8 layers $\times$ 6 heads) combined with distribution shift benchmarks require evaluation compute. Total estimate includes margin for additional

experiments.

Backup Plan: If compute budget is insufficient, we can train on a smaller dataset (CIFAR-100 or ImageNet subset) or reduce to a single model configuration. Evaluation workload is light and could run on consumer GPUs (RTX 3080/3090).

Software: PyTorch 2.0 with torchvision for data loading, timm library for ViT implementation (modified to support head ablation), Weights & Biases for experiment tracking.

Your compute needs should specify hardware resources with verified access, estimate requirements for training and evaluation, describe software stack, and provide backup plans.

Likely Outcome and Expected Results

Success Criteria: This project succeeds if we can answer: (1) Do attention heads in vision transformers develop interpretable specialization? (2) Does head specialization predict robustness to distribution shift? (3) How does head count affect both specialization and robustness?

Expected Results: Based on NLP findings, we expect 30-50% of heads may be redundant. Some heads will likely be critical—removing them causes large accuracy drops. We expect heads important for clean data may differ from heads important for corruptions. Visualizing attention patterns should show whether critical heads attend to specific features (edges, textures) or global context.

Likely Failure Modes: Training may not converge—ViT training is sensitive to hyperparameters, though following the DeiT recipe should help. Heads might all be equally important or equally redundant, which would still be interesting (suggesting vision differs from language). Specialization might exist but not relate to robustness. Compute might be insufficient—we can fall back to smaller models or fewer variants.

Learning Goals: This project will show whether vision transformers develop functional specialization like language models, what components contribute to robustness, and whether head-level analysis provides information beyond aggregate attention patterns.

Your outcomes section should define success criteria, describe expected results, identify likely failure modes with mitigation strategies, and articulate learning goals.

Project Timeline (Proposed)

Provide an anticipated timeline showing major milestones. Progress rarely follows a linear timeline, but planning major phases helps organize the work.

- **28 Jan - 05 Feb:** Finalize project scope, set up compute environment, download datasets, verify data pipeline works correctly
- **06 Feb - 12 Feb:** Implement training framework with configurable architectures, begin training first model, implement ablation mechanism, validate code on small-scale experiments
- **13 Feb - 19 Feb:** Complete training all model variants, verify models reach expected performance, implement specialization metrics, run preliminary ablation analysis
- **20 Feb - 26 Feb:** Complete ablation analysis for all models, evaluate on distribution shift benchmarks, identify critical versus redundant heads
- **27 Feb - 05 Mar:** Analyze head specialization patterns, visualize attention maps, perform mechanistic analysis of what features critical heads attend to, run ablation correlation analysis

- **06 Mar - 12 Mar:** Complete remaining experiments to test specific hypotheses, prepare presentation with key findings and visualizations
- **13 Mar - 18 Mar:** Write final report with complete analysis, create model card documenting trained models, record video demonstration, finalize code documentation

Critical Dependencies: Training must complete before detailed analysis. If experiments run smoothly, pursue stretch goals such as analyzing attention rollout across layers or testing whether pruning redundant heads during training improves efficiency.

Your timeline should show week-by-week milestones with specific deliverables and dependencies between tasks.

Primary References and Codebases

Vision Transformer Architectures:

- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … & Houlsby, N. (2021). "An image is worth 16x16 words: Transformers for image recognition at scale." *ICLR 2021*. [Foundation paper introducing ViT architecture]
- Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). "Training data-efficient image transformers & distillation through attention." *ICML 2021*. [DeiT training procedures we will follow]

Attention Analysis Methods:

- Voita, E., Talbot, D., Moiseev, F., Sennrich, R., & Titov, I. (2019). "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned." *ACL 2019*. [Head pruning methodology adapted to vision domain]
- Michel, P., Levy, O., & Neubig, G. (2019). "Are sixteen heads really better than one?" *NeurIPS 2019*. [Ablation analysis techniques we will apply]
- Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., & Dosovitskiy, A. (2021). "Do vision transformers see like convolutional neural networks?" *NeurIPS 2021*. [ViT representation analysis providing context]

Robustness Benchmarks:

- Hendrycks, D., & Dietterich, T. (2019). "Benchmarking neural network robustness to common corruptions and perturbations." *ICLR 2019*. [ImageNet-C benchmark definition]
- Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., … & Steinhardt, J. (2021). "The many faces of robustness: A critical analysis of out-of-distribution generalization." *ICCV 2021*. [ImageNet-R and robustness benchmarks]

Codebases:

- `timm` (PyTorch Image Models) - Repository: https://github.com/rwightman/pytorch-image-models

We will use this library's ViT implementation as a starting point, modifying it to support individual attention head ablation. **How we will use it:** The library provides ViT architecture definitions and training utilities. **What we will modify:** We will add hooks to the attention mechanism to enable per-head ablation (zeroing specific head outputs). **What we will add:** Specialization

metrics (attention distance, ablation correlation), systematic evaluation across distribution shifts, and visualization tools for attention patterns.

Your references should include foundation papers and codebases you will use. For each codebase, explicitly state: (1) How will you use it? (2) What will you modify? (3) What will you add? This clarifies your contribution versus existing work.