EE5102/CS6302 - Advanced Topics in Machine Learning – Fall 2025

Assignment 1 — *Release Date: 11 September 2025*

Instructions: Please read the following instructions carefully and abide by while preparing your submissions:

- This is the first graded assignment of the course which counts towards 7% of your final assignment aggregate.
- We need only one submission per group.
- This assignment is due by **Thursday, 25 September 2025 on LMS**.
- As a submission, you need to prepare and submit you deliverables according to the instructions in **Task 4 of this assignment**.
- **AI Usage Policy:** You are not allowed to use any generative AI model—including LLMs—to write any part of your PDF report. The language/text and analysis must be entirely your own, in the report.

  For the coding part, you ***may*** use public libraries, built-in functions, or even get help from an LLM if needed. That's acceptable.

  However, once you submit your assignment, you are fully responsible for everything in it—text and code both. If your submission overlaps significantly with another group's work, you cannot later claim that some part was LLM-generated. That will not be accepted as an excuse.

Overview & Motivation: Modern deep learning models in vision – from Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) to generative models (VAEs, GANs) and contrastive models (like CLIP) – exhibit different inductive biases by design. Inductive bias refers to a model's built-in assumptions that guide how it learns and generalizes [1]. For example, CNNs assume locality and translational invariance (convolution filters scan across the image), whereas ViTs rely on global self-attention with minimal built-in spatial bias [2]. These architectural choices influence what features models prioritize and how they perform on new data outside the training distribution.

One striking illustration is the semantic bias toward texture vs. shape. CNNs trained on ImageNet have been shown to classify images by texture rather than global shape – e.g. a picture of a cat drawn with elephant-skin texture can fool a CNN into labeling it "elephant" [3]. Humans, in contrast, rely mainly on shape. Vision Transformers, with their global receptive field and weaker priors, tend to be more shape-biased (more human-like in perception) given sufficient data [4] [1]. These differences in bias have practical consequences: models with a shape bias have better robustness to distribution shifts like style changes or sketch-like images.

Beyond semantic cues, architectural biases (like CNNs' local filtering vs. ViTs' global attention) and property biases (e.g. CNNs' translation equivariance vs. Transformers' sensitivity to absolute patch positions) can affect generalization. A CNN's learned features are inherently shift-invariant to some degree, while a ViT must learn position dependencies (otherwise a shifted image might confuse it). Likewise, Transformers can handle permutations or jumbled patches more robustly than CNNs in some cases, due to global context integration. CLIP, a vision-language model trained on 400 million image-text pairs, has shown impressive zero-shot classification ability and robustness to unusual inputs like sketches. CLIP's features cluster by high-level semantic concepts rather than surface details, indicating a bias toward conceptual/shape features likely imparted by its multimodal training.

Understanding these biases is crucial for out-of-distribution (OOD) generalization: when models face data that differ from training (different styles, domains, corruptions, etc.), the inductive biases will often determine their resilience or failure.

In this assignment, you will conduct a series of hypothesis-driven experiments to investigate inductive biases in discriminative, generative, and contrastive vision models. You will analyze how biases manifest as semantic preferences (e.g. shape vs. texture or color), architectural behaviors (locality vs. global context), and other properties (like invariances), and how these biases affect each model's learned representations and ability to generalize OOD. The goal is to encourage critical thinking and scientific reasoning: treat each experiment like a small research study. Emphasis is on analysis and insight – if not stated, use relatively lightweight models/datasets (suitable for a few Google Colab runs) and focus on evaluating hypotheses and interpreting results rather than exhaustive training or hyper-parameter tuning.

Research Questions: Before diving into the tasks, consider the key questions guiding this assignment. These hypotheses frame our objectives and should guide you in your analysis and report write-up (more on the write-up organization in Task 4):

1. **Shape vs. Texture Bias** – *Do CNNs and ViTs rely on different visual cues (e.g. texture patterns vs. object shapes) when recognizing images?* We hypothesize that CNNs will show a stronger texture bias, whereas ViTs (with global self-attention) may focus more on shape/structure, potentially improving OOD recognition of shape-preserved images (e.g. sketches).

2. **Architectural Biases** – *How do built-in architectural assumptions (convolutional locality vs. transformer global context) affect model properties like translation invariance and permutation sensitivity?* For instance, a CNN should be relatively robust to small image translations due to translational weight sharing, while a ViT without appropriate positional encoding might be more sensitive to absolute position changes. *Conversely, can ViTs handle shuffled patches or occlusions better, given their ability to incorporate global context?*

3. **Generative Modeling Biases** – *In what ways do a VAE and a GAN differ in the representations they learn and the images they generate?* We expect a VAE (which optimizes reconstruction likelihood with a latent prior) to learn a smooth, continuous latent space (small latent changes $\Rightarrow$ smooth output changes) and cover a broad range of the data distribution, at the cost of blurrier outputs. A GAN, trained adversarially to produce realistic samples, should generate sharper, high-fidelity images but may concentrate on fewer modes (less diversity). *How do these differences reflect inductive biases, and how do VAEs vs. GANs handle inputs they haven't seen e.g. OOD samples?*

4. **Multimodal & Contrastive Biases** – *What biases emerge in a contrastively trained vision-language model like CLIP compared to a standard vision model? Does CLIP exhibit a shape-bias more akin to humans or otherwise modulate the biases of its visual encoder? How well can CLIP generalize zero-shot to new tasks or domains compared to supervised models?*

5. **Inductive Bias & OOD Generalization** – *Overall, how do different inductive biases impact generalization to out-of-distribution data? For example, do models with a stronger human-like bias (shape-oriented, semantic focus) achieve better OOD performance (e.g. classifying images with altered textures or from new domains) than models relying on low-level cues? What trade-offs are observed between in-distribution accuracy and OOD robustness?*

Task 1: Discriminative Models – CNN vs. ViT Inductive Biases: In this first task, you will compare a CNN and a ViT on image classification, to expose differences in their inductive biases. We will use a pair of standard models – ResNet-50 and ViT-S/16 – and evaluate their behavior on various image manipulations. The focus is on semantic biases (texture vs. shape, color cues), architectural biases (local vs. global receptive fields), and certain invariances. Ultimately, you'll assess which model generalizes better under distribution shifts and why.

Dataset: Use a lightweight dataset like STL-10 or CIFAR-10 as the primary training set for this task. *STL-10 will give you better visualizations, while CIFAR will require less compute for training and fine-tuning.* Prepare a few modified versions of the dataset to test biases, as follows:

1. **Semantic Biases (Shape vs. Texture vs. Color):** To test whether models rely more on shape, texture, or color cues, experiment with the following ideas:
   - **Grayscale Dataset** Convert the test set to grayscale to remove color information and test for color bias.
   - **Cue Conflict** Apply style transfer (using any pretrained model of your choice) to CIFAR or STL images to decouple shape and texture (e.g., preserving object outline while altering textures). Create images where shape and texture cues intentionally conflict (e.g., a cat with elephant skin texture). These reveal whether the model prioritizes shape or texture when cues disagree. *(Refer to Example 1)*
2. **Locality Biases (Translation Invariance, Patch Structure):** To examine how models captures spatial locality and global relationships:
   - **Translation Tests:** Shift images by a few pixels in different directions to evaluate translation invariance.
   - **Patch Shuffling:** Split images into patches and shuffle them to disrupt global structure but preserve local content. This tests the extent to which models rely on local features vs. holistic layout.
   - **Patch Occlusion:** Mask out or blurr square regions of the images to see how the model performs when some parts of the image are unavailable.
3. **Generalizability Across Domains** To evaluate robustness to domain shifts, we will use the PACS dataset, which spans four diverse domains: *Photo, Art Painting, Cartoon,* and *Sketch*. We will use this to test whether CNNs or ViTs generalize better when trained on one domain and evaluated on another.

Steps:

1. **Model Fine-tuning:** Fine-tune a pre-trained ResNet-50 and ViT-S/16 on CIFAR-10 or STL-10 training data. Train until they reach a reasonable accuracy e.g. ∼90%. Use identical train/val splits to ensure a fair comparison and record the final accuracy and training curves for each model.
2. **In-Distribution Performance:** Evaluate both models on a clean test set. Note their baseline accuracy. This checks if both models learned the task similarly in-distribution.
3. **Color Bias Test:** Evaluate the models on the Grayscale version of the test images. Keep labels the same. Measure the drop in accuracy or change in predictions relative to color images. This reveals if models were relying on color cues. A large drop would indicate a color bias: the model depends on color information. Compare CNN vs. ViT: Which is more robust to losing color? Discuss possible reasons, e.g. did one model maybe learn more shape features that survive grayscale conversion?.
4. **Shape vs. Texture Bias – Stylized Images:** Use your style-transfer dataset to evaluate whether models rely more on shape or texture. You may remove texture information by replacing it with different textures, leaving shape as the dominant signal *(Refer to Fig. 1)*. You may also create cue-conflict images by transferring textures from another class within the dataset (e.g., a "cat shape with elephant skin"), and both "cat" and "elephant" classes exist. Analyze whether the model's prediction is driven by shape or texture, and reflect on what this reveals about its inductive biases. Then compute the shape bias as:

$$\text{Shape Bias (\%)} = \frac{\#\text{images classified by shape}}{\#\text{images classified by either shape or texture label}} \times 100$$

   Compare ResNet-50 and your ViT: does the CNN tend to misclassify by focusing on texture, while the ViT shows higher shape bias? Include examples and corresponding predictions in your report to illustrate the difference.
5. **Translation Invariance Test:** Using your translated images, evaluate the models. Does the CNN maintain its predictions under small shifts? Why CNNs are approximately translation-invariant to small shifts, if they

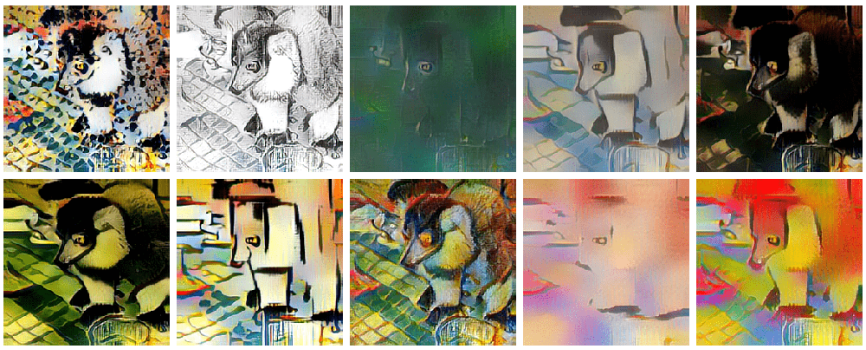

Figure 1: Example of Stylized-ImageNet images for shape-vs-texture analysis: Left is an original photo (a ring-tailed lemur), and right are stylized versions with various artistic textures applied. A model relying on shape should still recognize the lemur in all images, whereas a texture-biased model may confuse them with the content of the textures.

are? Check if the ViT's predictions change more significantly when an object moves in the frame. Quantify this by measuring accuracy on the shifted set or even the consistency: e.g. what fraction of images get the same top-1 prediction after shifting? We expect the CNN to be more stable to shifts (higher consistency), whereas the ViT might sometimes drop confidence or predict a different class if the object's position is unfamiliar. If this happens, why does this happen? This tests the equivariance/invariance bias.

6. **Permutation / Occlusion Test:** Evaluate robustness to disrupted spatial structure. For your patch permuted dataset, see if either model can still make a reasonable prediction. Intuitively, a CNN might still fire on local texture patches and attempt a prediction often incorrectly, since global arrangement is lost, whereas a ViT might be confused by the missing global coherence but could leverage any learned positional info. Similarly, ViTs have been reported to be robust to patch dropout and occlusion. Track any change in accuracy. Document a few examples: e.g. show an occluded image and note how each model's prediction and confidence differ from the original. This will illustrate the role of global context vs. local features in each model.

7. **Feature Representation Analysis:** To delve into how each model represents the data, perform a visualization of the feature space. Take a subset of images (maybe include some stylized or OOD examples as well) and extract the penultimate layer features (the embedding before the final classification layer) from the ResNet and ViT. Apply a dimensionality reduction (e.g. t-SNE or PCA) to each model's embeddings separately and plot them. Color-code the points by class (and perhaps by whether the image was normal or stylized, etc.). Compare the plots: does one model separate the classes more distinctly in feature space? Are stylized images embedding closer to their original class in one model vs the other? For example, you might find the ViT's representation clusters images by their true object category more clearly even across style changes, whereas the CNN's clusters might mix up classes that share textural similarities. Such observations can provide evidence of the different semantic focus of their features. Include these plots in your report and interpret them.

8. **Domain Generalization Test on PACS:** Test the models on a simple domain shift using the PACS dataset. For example, fine-tune both models on three domains of PACS and evaluate on the remaining domain (take Sketch as test domain as this one is challenging). Measure which model's accuracy drops more. We anticipate that the ViT (with its shape bias and global context) may handle the domain shift better than the CNN. For instance, CNNs often struggle with sketches (line drawings with no texture), whereas a shape-biased model should fare better. Report any findings – even if both fail, it's insightful to note the failure modes: did the CNN predict based on background or texture-like cues that weren't there? Did the ViT manage to pick the correct or a semantically close class?.

Expected Outcomes & Analysis: By the end of Task 1, you should have a thorough comparison of ResNet vs ViT. In your report, summarize the key differences you found; some examples:

- Which model is more texture-biased vs shape-biased? Provide quantitative evidence (accuracy on stylized images or shape-bias percentages) and qualitative examples. Relate to the hypothesis and cite any relevant literature (e.g., "our CNN results mirror the prior finding that ImageNet-trained CNNs rely on texture, as it misclassified a cat with checkerboard texture as 'chessboard'"). Did the ViT indeed act more shape-centric?
- How do the models differ in invariance properties? For instance, note if ResNet-50's predictions were un-

changed for 90% of images under small shifts, while ViT-S/16 only 70% unchanged (hypothetical numbers). What does this say about their inductive biases (CNN's built-in shift invariance vs ViT needing to learn position)? If the ViT used learned positional embeddings, did it generalize to shifts it hadn't seen? Use your results to discuss this.

- Comment on permutation/occlusion robustness: perhaps you saw that shuffling patches destroys CNN performance (since it scrambles meaningful compositions), but the ViT might retain some capability or at least fails more gracefully when partial information is present. Or maybe both failed – discuss why that might be.
- Include the feature visualization and interpret it. For example: "In the ResNet feature PCA, images cluster by background texture (all green-ish images grouped together, regardless of class), whereas the ViT's features cluster more by object identity, indicating it captured more global object information." Relate this to semantic bias.
- Tie these observations back to architecture: Why do we see these differences? Connect to the idea that CNNs, with limited receptive fields and local convolutions, naturally pick up on high-frequency/textural details, whereas ViTs, with global attention (and no bias forcing locality), can learn to use shape/global structure – but only if data or training encourages it. If your ViT was trained on CIFAR-10 only (a relatively small dataset), you might note it didn't exhibit as strong shape bias as expected – possibly due to not having enough data (ViTs often need large data or augmentation to reach their potential). This is an insight on data vs inductive bias trade-off.

Task 2: Generative Models – Investigating VAE vs. GAN Biases In this task, you will explore the inductive biases of two generative modeling approaches – VAEs and GANs. Both can generate images, but they do so with different assumptions and results. We'll examine how these differences manifest in terms of output quality, diversity of samples, structure of the learned latent space, and ability to interpolate or handle OOD inputs. The goal is to gain intuition for how a model's design and training objective bias it toward certain behaviors, e.g. VAEs' bias toward capturing the entire data distribution vs. GANs' bias toward realism at the expense of some modes. We also look at representation aspects like how each model encodes information in the latent space.

Dataset: Use CIFAR-10 again for training the generative models, so that results are comparable. CIFAR-10's 32×32 images are small enough to train a VAE and a GAN relatively quickly. If training from scratch is too slow, you may use a pre-trained small GAN and VAE on CIFAR-10 (or train on fewer epochs just to get a rough model). The focus is not achieving SOTA image quality, but comparing the two methods. Optionally, you could use an even simpler dataset like MNIST or a single CIFAR class if needed for speed, but multi-class CIFAR-10 is preferred to also see how class features are represented.

Model Setup: Use a basic VAE (with convolutional encoder and decoder) and a basic GAN (e.g. DCGAN architecture) for comparability. The VAE should output the mean and variance of a latent Gaussian (use a latent dimension of e.g. 64 or 128 or any standard), and be trained with a reconstruction + KL loss. The GAN can be a DCGAN with a similar conv generator and discriminator. Aim for both models to reach a point where they produce recognizable (even if imperfect) images of CIFAR-10 classes (or classes on your chosen datasets).

Steps:

1. **Train/Obtain Models:** Train the VAE on CIFAR-10 training set. Monitor its reconstruction loss. After training, it should reconstruct images approximately (they may appear blurry but class content visible). Separately, train the GAN on CIFAR-10. Monitor the GAN's training (e.g. look at generator and discriminator loss, watch for mode collapse or instability). Save checkpoints of both models for evaluation.
2. **Reconstruction vs Generation:** Take a random sample of images from the test set and reconstruct them with the VAE (pass them through encoder to latent, then decoder). Also sample an equal number of images from the GAN (generate from random latent vectors). Compare these outputs:
   - **Visual Quality:** Are GAN samples sharper or more detailed than VAE reconstructions? Typically, yes – GANs excel at high-frequency details (e.g. textures) because the adversarial loss pushes for realism, whereas VAEs tend to blur details as they optimize pixel-wise loss (which averages over uncertain details). Document examples: include a figure in your report showing e.g. an original image vs VAE reconstruction vs a random GAN sample. You'll likely see the VAE outputs look like smoothed versions of the input (with some loss of fine detail), while GAN outputs (not tied to an input) might look more photo-real but sometimes have odd artifacts or missing diversity.
   - **Diversity:** Do the GAN samples cover various classes and styles, or do they tend to look similar (mode collapse)? A common GAN bias is to mode-collapse on a subset of data modes, whereas VAEs by design try to model the full distribution (they maximize likelihood). You can quantify diversity by, say, generating 1000 samples from each and checking how many unique classes are represented (if you have a classifier to predict class labels for them) or compute an inception score / FID if possible. If not, a qualitative judgment: Does your GAN mostly generate say planes and cars and rarely animals (indicating mode bias)? Does your VAE reconstruction tend to make images look like an "average" of various things? Note these tendencies.
   - **Metrics:** If feasible, compute the FID (Fréchet Inception Distance) – read about it – for both models' outputs against the real dataset (using a pre-trained classifier's embeddings). VAEs often have higher FID (worse) because outputs are blurrier (lower fidelity), although they cover data diversity. GANs often achieve lower FID (sharper, more realistic) but might have lower likelihood if measured (since they miss modes). Alternatively, compute the reconstruction error for VAE on test images (e.g. average MSE per pixel) – that measures how well it captures data even if blurry. For the GAN, since it doesn't do reconstruction, you could measure something like inception score for samples to gauge quality vs variety. Use these metrics to back your claims about fidelity vs diversity trade-offs.
3. **Latent Space Structure:** Explore the latent space properties of each model:
   - **Interpolation:** Pick two test images from different classes. For the VAE: encode each to get latent vectors z_1 and z_2. Linearly interpolate between z_1 and z_2 (e.g. 10 intermediate points) and run them through the VAE decoder. Do the same for the GAN: take two random latent vectors that produce distinct outputs, interpolate between them, and generate images. Visualize the interpolation sequences.

VAEs, due to their smooth latent space (enforced by the Gaussian prior and continuous latent distributions), should yield a smooth morphing from one image to the other. You'll likely see intermediate images that gradually change features – this shows the VAE has learned a continuous manifold of data where each point is a plausible image. GANs, if well-trained, often also show fairly smooth interpolations (the generator learns a somewhat continuous space of realistic images), but sometimes there might be abrupt changes or some weird artifacts in between if the GAN's latent space has discontinuities. Compare: which model's interpolation was more coherent? Include a few example interpolated images in your report (perhaps as a figure with a row of images from one class to another). This demonstrates the inductive bias towards continuity in VAEs versus the GAN's ability to generate sharp transitions.

- **Latent Representation Analysis:** For the VAE, since it has an encoder, you can examine how it represents different inputs. Take a sample of test images (covering different classes) and get their latent vectors (e.g. mean of $q(z|x)$). Use 2D PCA or t-SNE on these latent vectors to see if images cluster by class in latent space. Often VAEs (especially without labels) do not perfectly cluster classes, but similar images might end up near each other. You might find, for example, that all images of trucks are in one region of latent space, whereas GAN's latent has no such explicit meaning (since GAN latent is unstructured and not inferred from images directly). If you have labels for images, you can color the VAE latent points by class to see any structure.
- **Semantic Factors:** Another test: see if specific latent dimensions have semantic meaning. Sometimes VAEs learn some interpretable factors (especially if you used a β-VAE or small dataset). For instance, vary one latent dimension while fixing others and observe the effect on the decoded image (does it consistently change something like object size, rotation, or color?). Document any interesting factor if found. GANs typically don't have as straightforward interpretation per dimension, though techniques like PCA on GAN latent or GAN inversion can be explored (optional).

4. **Out-of-Distribution (OOD) Inputs:** Evaluate how each model handles inputs outside its training distribution:
   - **VAE OOD reconstruction:** Feed the VAE some images it was not trained on (e.g. if trained on CIFAR-10, try an image from CIFAR-100 or a completely different dataset like an SVHN digit or a simple geometric shape). Observe the reconstruction. VAEs tend to "reconstruct" OOD inputs as something within the training distribution, due to the inductive bias of the latent prior. For example, if you give a VAE (trained on animals and vehicles) a picture of a house, it might reconstruct it as a blend of nearest familiar shapes (maybe it turns it into a truck, etc.). Measure the reconstruction error for OOD images vs. normal images – likely the error is higher, which could be used for anomaly detection. Discuss this behavior: it shows the VAE is biased to assume inputs come from the training data manifold, and it struggles or projects anomalies onto that manifold.
   - **GAN OOD:** A GAN doesn't directly take input images, so instead you could do a conceptually related test: extrapolation in latent space. Generate images by sampling latent vectors that are far from the training latent distribution (e.g. very large values, since GANs usually expect a standard normal input). See if the GAN outputs degrade or produce nonsense when fed latents outside the distribution it was trained on. This is more of a stress test for the GAN's generalization in latent space.
   - **Anomaly detection:** Another angle: Anomaly detection – use the VAE's reconstruction error to identify if an image is OOD. For example, take a batch of CIFAR-10 images and some not-in-CIFAR images, run them through VAE and see if you can distinguish by reconstruction MSE. This tests the model's ability to signal it's seen something unfamiliar, which is a desirable bias for safety. Comment on the results.

5. **Training Dynamics and Stability:** Reflect on the differences in training the two models:
   - VAEs have a clear training objective (maximize ELBO) that usually converges without too much fuss (maybe tuning the weight of KL term). Did your VAE training progress steadily?
   - GANs often are trickier to train (due to the adversarial game, mode collapse, oscillating losses). Note if you encountered any difficulties (e.g. did you have to balance generator/discriminator learning rates or use tricks like feature matching, etc.?). This highlights an inductive bias vs. flexibility trade-off: GANs have no explicit density estimation (which is why they can focus on realistic samples), but that makes training less stable. Mention any such observations.
   - If your GAN collapsed or had limited diversity, that itself is a key result: it demonstrates how the model might bias towards a subset of data (e.g. generating only one class it finds easiest to mimic) – a kind of overfitting to modes.

Expected Outcomes & Analysis: In your report, compare the representational and output differences between the VAE and GAN:

- **Fidelity vs Diversity:** State which model produced sharper images and which covered the data distribution better. Connect to the known result: "GANs are known for high-fidelity samples but can suffer from low diversity, while VAEs cover more diversity but with lower fidelity (blurriness)." Did your experiments align with this? Provide evidence (e.g. "Our GAN samples looked more detailed – see Fig X – but when we generated 1000 images, only 6 out of 10 classes appeared, indicating mode bias. The VAE reconstructions were blurrier (average MSE=... higher), but when sampling, it produced examples of every class.")
- **Latent Space Continuity:** Discuss the interpolation results. Perhaps you observed the VAE transition was smooth and every intermediate image was plausible (showing it learned a smooth manifold), whereas the GAN interpolation might have had some discontinuities or odd intermediate images. Or if the GAN was well-behaved, highlight that both can have smooth latent traversals, but VAEs have the advantage of an explicit latent variable model (which encourages linearity in latent space). This leads to the concept of VAEs having a more semantic or structured latent space (sometimes even learning features like "this dimension roughly corresponds to object rotation", etc.), whereas GAN latents are not as constrained to align with semantic changes (though empirical evidence shows many GANs do learn some interpretable directions). If you found any interpretable latent factor in the VAE, mention it.
- **Reconstruction Ability:** Note that the VAE can reconstruct inputs, whereas the GAN cannot (without an encoder). This is an inductive bias difference: VAEs explicitly learn to invert the data to latent (good for representation learning), while GANs focus only on generation. If you plotted some reconstructions, comment on what the VAE tends to get wrong (usually fine details or edges). This ties to how the VAE objective incentivizes averaging over minor details to minimize overall error (hence blur).
- **OOD Generalization:** Summarize how each model deals with unseen data. Perhaps: "When given a completely different image (not from CIFAR-10), our VAE's reconstruction looked like a distorted training-class image, indicating it was 'imagining' the input as one of the known classes. This shows the VAE's bias: it assumes inputs are from the training distribution. The GAN, on the other hand, cannot even process an input image – illustrating a limitation if we wanted to, say, use it for anomaly detection." If you did the anomaly detection test, report how well reconstruction error separated known vs unknown images.
- **Implications:** Reflect on what these biases mean for generalization and use-cases. For example, VAEs might be better for tasks requiring understanding of the data manifold or detecting anomalies, due to their inclusive latent space (but their generated samples might not fool humans due to blur). GANs produce very realistic samples (great for data augmentation or media generation), but one must be careful of what they might miss (they might not represent all variations, which is a generalization concern). Also, note the training bias – VAE's log-likelihood training ensures every training sample is accounted for, whereas a GAN could theoretically ignore some training samples as long as it can fool the discriminator with others. This difference in objective is itself an inductive bias: VAEs explicitly optimize for data coverage.

By the end of Task 2, you should have explored how the design of generative models influences their behavior and what they learn. These observations will enrich your perspective when considering representation learning and even how these generative approaches might complement discriminative models, e.g. in semi-supervised learning or as data augmenters. You will include sample images and quantitative results in your report to substantiate each point.

Task 3: Contrastive Models – Analyzing CLIP (ViT-B/32) and Multimodal Biases In this task, you will study a contrastively trained vision-language model, specifically CLIP. CLIP consists of an image encoder (we'll use the ViT-B/32 variant) and a text encoder that are trained together to align images with their textual descriptions. This model has very different training inductive biases: instead of single-label supervision on one dataset, it learned from natural language supervision across 400 million image-text pairs. The hypothesis is that CLIP's training endows it with more semantic, high-level inductive biases – it should focus on object identity and conceptual features since it needed to match captions, rather than surface statistics. CLIP also has demonstrated strong zero-shot generalization to new tasks and robustness to distribution shifts (like cartoons, sketches). We will investigate these properties by comparing CLIP's performance and representations to a standard supervised model on a variety of tasks.

Setup: We'll use a pre-trained CLIP model: you can load OpenAI's CLIP ViT-B/32 through libraries like clip or HuggingFace. No training is required here – we will use CLIP as is. For comparison, have one of your Task 1 models on hand: the ResNet-50 or ViT-S/16 trained on CIFAR-10, to serve as a *supervised baseline*.

Steps:

1. **Zero-Shot Classification:** First, test CLIP on a standard classification task without any fine-tuning. For example, use CIFAR-10 (which CLIP was never explicitly trained on). CLIP can do zero-shot by comparing image features to text features of candidate labels. Implement this: for each CIFAR-10 class (e.g. "airplane, automobile, bird, etc."), create a prompt like"a photo of a {class}" (or several prompts, CLIP allows multiple prompts per class for ensembling). Compute the CLIP text embeddings for each prompt and the CLIP image embedding for a given test image, and pick the class whose text embedding has highest cosine similarity with the image embedding. Measure the accuracy on CIFAR-10 test set. It might not be as high as a model trained on CIFAR (since CIFAR images are small and CLIP wasn't specifically trained on those classes), but it will likely be far above random chance, demonstrating zero-shot learning. Record this accuracy. For example, you might find CLIP gets 80% on CIFAR-10 without training – quite impressive compared to a random guess 10%. This shows the power of its semantic knowledge. Compare with your ResNet that was fine-tuned on CIFAR-10 – the supervised model might still be better in-domain (maybe it got 90% because it was specialized), but CLIP's result is notable given no training on CIFAR. Also try a domain-shifted classification: e.g. use PACS Sketch images (or ImageNet-Sketch if available) and see if CLIP can classify them zero-shot. For instance, if you have some sketch images of animals or objects, see if CLIP can correctly identify them with prompts ("a drawing of a horse", "a sketch of a horse" – you can try different wording). CLIP is known to handle sketches and art better than standard models. If you did Task 1 experiment, compare: how did CLIP (without fine-tuning) do on sketch images vs. your CNN that was trained on photos? Often CLIP will outperform a photo-trained model on sketches even without being trained on sketches, due to its broad training. Note those results.
2. **Few-Shot or Prompt Engineering:** CLIP's performance can often improve with prompt engineering or a few-shot context. If you have time, you can see if phrasing the text prompts differently affects accuracy (e.g. "a photo of a {class}, {class}." vs. simple "{class}"). This isn't required but can demonstrate the importance of language priors – e.g. including the article "a photo of" often helped CLIP as per the paper.
3. **Image-Text Retrieval:** To see CLIP's multimodal alignment in action, do a small retrieval test. For a given set of images, say 20 images from different classes or domains, and a set of text queries (descriptions of some of those images), use CLIP to find the best match. For example:
   - Calculate CLIP image embeddings for your 20 images.
   - Calculate CLIP text embeddings for a few queries like "a photo of a cat", "a photo of a car", etc. (or even more descriptive phrases if you want).
   - Compute similarity and retrieve the top matching image for each text. Does CLIP retrieve the correct image for the query?
   - You can also do reverse: for each image, find which text description among a set is most similar.

   Document a couple of examples in your report (maybe a small table or figure: query vs retrieved image). This will support the claim that CLIP has learned a rich joint vision-language representation.
4. **Representation Analysis – Image-Only:** Similar to Task 1, visualize the image feature space of CLIP vs a supervised model:
   - Take a variety of images (you can mix CIFAR-10 test images and some OOD images like a few from different domains – e.g. throw in a sketch or a grayscale image). Ensure you label these images with their true class or type (and domain).

- Compute CLIP image embeddings (from the ViT-B/32 encoder) for each, and likewise get embeddings from your baseline (ResNet or ViT from Task 1) for each.
- Use t-SNE or PCA to reduce to 2D, and plot points, marking them by class and maybe shape vs texture or domain.
- Compare clustering: We expect CLIP's embeddings to cluster primarily by semantic class because it learned to group images that "mean" the same thing (regardless of appearance) together. In contrast, a supervised CNN might cluster images by dataset or superficial features if they weren't all seen in training.

Include the plots and a brief interpretation. For example: "In CLIP's feature t-SNE, images cluster by their object category across different styles – the cat photo and cat sketch are neighbors – indicating CLIP's representation encodes the high-level concept 'cat'. The ResNet's t-SNE shows the sketch cat embedding far from real cats, likely because the ResNet did not recognize it without texture/color."

5. **Shape vs Texture Bias in CLIP:** Do a quick shape-texture bias test on CLIP (similar to Task 1). You can use a couple of the cue conflict images (e.g. the famous "cat shape with elephant texture"). If those are available or you can generate one, feed it to CLIP. Ask CLIP via text queries what it sees: e.g. compute similarity with "an elephant" vs "a cat". See which is higher. There is evidence that CLIP (and vision-language models) are more shape-biased than the vision-only CNNs. If CLIP indeed picks "cat" for the cat-elephant image whereas a supervised CNN picked "elephant", that's a powerful demonstration. Even if you can't run this exact test, you can cite known results: "According to recent research, vision-language models like CLIP prefer shape over texture more than their pure vision counterparts. Our observations align with this: CLIP correctly identified a stylized cat image as a cat, whereas the ResNet misclassified it." This underscores how multimodal training modulated the visual biases.
6. **Robustness tests:** You can also test CLIP on a range of perturbations (similar to ImageNet-C corruptions or ImageNet-R renditions if you have them). CLIP was found to be robust largely due to the diversity of training data. If you have, say, images with noise, blur, or different artistic renditions, see if CLIP still recognizes them (zero-shot). A qualitative check: maybe take one image and apply a filter (make it cartoonish or add heavy noise) and run CLIP vs your baseline to classify. Likely CLIP will be more stable. Note any such anecdotal results.

Expected Outcomes & Analysis: In your report, detail CLIP's capabilities and biases:

- **Zero-Shot Performance:** Report the zero-shot accuracy on CIFAR-10 (or whichever dataset you tried). Emphasize that CLIP achieved this without any task-specific training, highlighting the power of its learned representations. Compare it to the supervised model's performance – perhaps the supervised model is higher on its own domain, but CLIP is not far off and much more flexible. If CLIP struggled on some classes, mention which (maybe small, less distinctive CIFAR classes might confuse it – e.g. "frog" vs "bird"). This might be due to resolution or the domain gap (CIFAR images are tiny). In any case, the takeaway is CLIP has a strong inductive bias of broad semantic knowledge.
- **Generalization to New Domains:** Summarize how CLIP handled sketches or other domain shifts. For example: "CLIP correctly recognized 8/10 sketch images in our test (using prompts 'drawing of X'), whereas the supervised ResNet-50 (trained on normal images) only got 2/10 right, often mistaking the sketches. This indicates CLIP's features are less dependent on texture/color and more on the abstract shape or concept – a sign of improved OOD generalization." This addresses Q4 and ties back to Q1 about shape bias (CLIP behaving more shape-biased).
- **Representation Quality:** From the t-SNE/PCA results and retrieval experiments, conclude that CLIP's image embeddings form a space where semantic similarity correlates with feature similarity. Mention the example: CLIP's deeper features group images by conceptual categories rather than superficial visual similarity. In your own smaller-scale test, you might say: "We observed CLIP embedding images of dogs from photos, cartoons, and sketches all near the text embedding for 'dog', demonstrating a high-level grouping of concepts. The supervised model's embeddings, in contrast, were more scattered and tended to cluster images by style (all sketches together, separate from photos) – indicating it didn't bridge the domain gap." This highlights CLIP's semantic/shape bias in representations.
- **CLIP vs Inductive Biases of Prior Models:** Reflect on how CLIP, despite using a ViT architecture, behaves differently than the ViT in Task 1 that was purely supervised. The difference is the training: CLIP's training on diverse data with a language objective acted as a powerful regularizer and guided it to pick up on meaningful features. Essentially, CLIP's inductive bias comes from data and objective more than architecture – it learned an almost shape-based, concept-based approach because to match text it had to focus on what

the object is rather than low-level cues. This can segue into a broader discussion in the report: sometimes learned biases from huge data can overshadow architectural biases. For instance, your ViT in Task 1 needed explicit encouragement (like SIN training or heavy augmentation) to be shape-biased, but CLIP's training achieved it automatically.

- **Robustness and Limitations:** Note that CLIP's robustness is largely attributed to its training distribution – it likely saw many styles and contexts in the 400M data, so it knows sketches and cartoons. This is an important point: inductive bias can be injected via training data variety (not just architecture). Mention if you saw any failure cases for CLIP: e.g., maybe it had trouble with very fine-grained details or some classes where text prompts were ambiguous. Also, CLIP might have its own biases (like societal or linguistic biases, which are outside our scope but worth acknowledging: e.g. it might associate certain objects with certain contexts from the web data).

Task 4: Synthesis of results & Report Writing Guidelines Now that you have conducted the experiments, the final task is to synthesize your findings into a coherent analysis and present it in a professional manner.

Deliverables:

- GitHub Repo: containing code (with documentation), any saved models or data subsets, and a README explaining how to run your analysis.
- PDF Report: ICML style, 6-10 pages including figures, addressing all points above. Treat it as a professional paper – clarity, structure, and correctness are key. Guidelines are given below.

Instructions:

- **Organize Your Code and Results:** Ensure your scripts for each task are clean, well-documented, and placed in the GitHub repo. Include instructions or scripts to install requirements and run the experiments. If some experiments are computationally heavy, provide saved outputs (e.g. learned model weights, logged metrics, or sample images) so the TAs/instructor can verify results without rerunning everything. The repository should be structured logically (perhaps a folder for Task1, Task2, etc., each with code and maybe a short README of its own).
- **Prepare Figures and Tables:** From Tasks 1–3, you likely have several plots, images, and metrics. Select the most meaningful ones to include in your report:
  - A figure or table comparing CNN vs ViT performance on various tests (e.g. a table of accuracies: CIFAR clean vs grayscale vs stylized, CNN vs ViT).
  - A visualization figure: possibly the t-SNE plots of features for CNN vs ViT (could be combined subplots).
  - A figure illustrating shape vs texture bias: maybe an example image with predictions, or a bar chart of shape-bias percentage for each model.
  - For generative models: a figure showing sample outputs (reconstructions and generations side by side), and maybe a plot of interpolation results.
  - Possibly a small table of VAE vs GAN metrics (FID, reconstruction error, etc).
  - For CLIP: a figure of the embedding space comparison or a retrieval example. Also possibly a table of zero-shot accuracy vs a baseline.

  Use clear captions and refer to them in the text. Ensure every figure is legible (use sufficient resolution as needed) and every table is properly labeled.
- **Writing the Report:** The report should roughly include:
  - **Abstract:** A short summary of what you did and key findings such as: *(e.g. "We investigate how inductive biases in CNNs, ViTs, VAEs, GANs, and CLIP affect their generalization. We find that CNNs heavily rely on textures, whereas ViTs and CLIP are more shape/semantic oriented, leading to better robustness on distribution shifts. VAEs vs GANs trade off diversity and fidelity due to their training biases. Our results highlight that incorporating human-aligned inductive biases (either via architecture or training data) can improve OOD generalization.")*
  - **Introduction:** Introduce the problem of inductive bias and OOD generalization. State the objectives and research questions. Motivate why this study is important (e.g. understanding model failures, guiding future model design). You can briefly preview your approach (tasks on different model types) and findings. Cite relevant background in proper academic citation format.
  - **Methodology/Experiments:** This can be structured by your tasks.

    Explain for each experiment the setup: models used, datasets, what was measured. You don't need to enumerate "Task 1, 2, 3" in the report, you can combine logically (e.g. one section on Discriminative models: CNN vs ViT, another on Generative models: VAE vs GAN, etc.). Ensure to define metrics *(e.g. "we measure shape bias as the fraction of shape-consistent decisions")*. Keep this section concise on what you did, as the detailed results come next.
  - **Results:** Present the findings for each experiment, ideally intertwining the quantitative results with analysis. This is where you include those figures and tables. Each subsection should tie back to the hypotheses. For example, a subsection "Texture vs Shape Bias in CNN vs ViT" where you describe the stylized image test outcome, referencing a figure and stating that CNN had X% accuracy vs ViT

Y%, indicating CNN relied on texture. Another subsection "Effect of Architectural Bias on Invariance" discussing the translation test and patch shuffle test. For generative models, discuss reconstruction vs generation results. For CLIP, discuss zero-shot and representation results.

- **Discussion:** This is a crucial part to demonstrate deep reflection. Here you synthesize across experiments:
  - Q1 (CNN vs ViT biases): Summarize whether CNNs showed texture bias and whether ViTs exhibited more shape bias, tying back to evidence and prior literature.
  - Q2 (Architectural biases): Summarize observations of CNN's translation invariance vs ViT's sensitivity to shifts.
  - Q3 (VAE vs GAN): Summarize the fidelity-diversity trade-off between VAEs and GANs.
  - Q4 (CLIP and contrastive biases): Summarize CLIP's performance and how it reflects semantic inductive bias.
  - Q5 (Generalization): Tie together which biases most helped OOD generalization, noting limitations (e.g. ViTs need large data, CLIP needs diverse data).

  Also discuss the interplay of architecture and data-driven biases and potential future designs.
- **Conclusion:** A short paragraph wrapping up. Reiterate the most important takeaways: *e.g. "In summary, our experiments highlight that inductive biases significantly impact model generalization. Models with human-aligned biases (global shape focus, semantic understanding) – whether via architecture (ViTs) or training (CLIP, stylized data) – generalize better to out-of-distribution scenarios than those relying on low-level cues. Conversely, models without strong biases (GANs or standard ViTs on small data) can excel in-domain but may suffer under shifts or require massive data to learn the right features. These findings encourage incorporating appropriate biases in model design and training to improve robustness."* End on a forward-looking note.
- **Citations:** Throughout the report, if needed, cite sources in academic style. Make sure to cite any claim that is not your own result. A References section should list all cited works. Compare your findings with known literature to show deep understanding.
- Finally, reflect on the process in a brief note (could be in the report discussion or a separate markdown in the repo): What surprised you? Did any results conflict with your expectations or published results? For instance, if your ViT didn't show shape bias due to limited data, note that and understand why inductive bias isn't absolute – data and training matter. This kind of insight will show the depth of your engagement.

References

[1] CNNs vs Vision Transformers — Biological Computer Vision:
https://medium.com/bits-and-neurons/cnns-vs-vision-transformers-biological-computer-vision-3-3-56ff955ba4

[2] A Deep Dive into Vision Transformers (ViT): Concepts, Fundamentals, Methods, and Applications:
https://medium.com/@hexiangnan/a-deep-dive-into-vision-transformers-vit-concepts-fundamentals-methods-and

[3] Geirhos, Robert, et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." International conference on learning representations.
2018. https://arxiv.org/abs/1811.12231

[4] Understanding Vision Transformers (ViTs): Hidden properties, insights, and robustness of their representations:
https://theaisummer.com/vit-properties/