Masked Autoencoders for Vision --- Original vs Our Improved Variant

Authors: Prashant Bharti 202251102 , Utkarsh Rana 202251148 **,** Vinod Tembhurne 202251157, Vivek Sonkar 202251158

Reference: He et al., "Masked Autoencoders Are Scalable Vision Learners," CVPR 2022 (DOI: 10.1109/CVPR52688.2022.01599)

Abstract

In this project We reproduce the core results and design of Masked Autoencoders (MAE) under tight Colab constraints and propose a compute-light improvement that targets known limitations of pixel-only reconstruction. We pretrain on STL-10 (unlabeled split) and evaluate on STL-10 (train/test) with two protocols---linear probing and end-to-end fine-tuning---while also providing side-by-side reconstruction visualizations using identical masks. My improved variant augments the original MAE with (i) curriculum masking, (ii) a masked high-frequency (Laplacian) loss, and (iii) a simple uniformity regularizer. We keep the model architecture, data, and evaluation identical across both variants to ensure a fair comparison. On small training budgets where visual differences are subtle, high-frequency error reveal early gains; with more compute or smaller images, differences become more pronounced.

1) Problem Statement

We aim to learn effective visual representations without labels by reconstructing masked content in images (self-supervised learning). Specifically, We study MAE as presented by He et al. (CVPR 2022) and ask:

- Can We faithfully reproduce the essence of MAE under Colab constraints?
- Can We introduce minimal, compute-light changes that address limitations of pixel-only reconstruction and improve training stability and linear separability?
- Can We compare the original and improved approaches fairly on the same data with the same protocols, and visualize differences side-by-side?

2) Approach

2.1 Original MAE (paper gist)

- **Core idea:** Randomly mask a high fraction of image patches (≈75%), encode only visible patches with a ViT encoder (no mask tokens), and reconstruct masked patches with a lightweight decoder in pixel space.
- **Asymmetry:** Encoder sees about 25% of tokens; decoder handles full sequence with learned mask tokens. Decoder is used only during pretraining; downstream tasks use the encoder.

- **Loss:** Mean Squared Error (MSE) computed only on masked patches; per-patch normalization helps.
- **Strength:** Simple, fast to scale, and strong transfer across tasks once fine-tuned.

2.2 Our Colab-friendly reproduction

- **Dataset:** STL-10 (auto-download with torchvision).
- **Pretraining:** unlabeled split (~100k images).
- **Evaluation:** supervised on train/test (10 classes, 5k train, 8k test).
- **Model (identical across both variants for fairness):**
    - ViT-like encoder: embed_dim=192, depth=8, heads=6; 2D sine-cos positional embeddings.
    - Lightweight decoder: embed_dim=128, depth=4, heads=4; learned mask token.
    - Image size: 224 (optional 96 for much faster ablations).
    - Patch size: 16.
- **Baseline "Original" (my code):**
    - Fixed random masking ratio of 0.75.
    - Pixel MSE loss on masked patches with per-patch normalization.
    - Minimal augmentation (random resized crop + flip).
- **"Improved" variant (my code):**
    - Curriculum masking: linearly ramp mask ratio from 0.5 → 0.75 during the first half of training to stabilize early optimization.
    - High-frequency emphasis: add a masked Laplacian L1 loss on reconstructed images to encourage sharper edges/textures.
    - Uniformity regularizer: apply a simple uniformity loss on pooled encoder tokens to encourage spread-out representations and improve linear separability.
    - Total loss: L = L_MAE(masked MSE) + λ_freq·L_Laplacian(masked) + λ_unif·L_uniformity.
- **Evaluation protocols:**
    - Linear probing (LP): freeze the encoder; train a BN+linear head on STL-10 train; report top-1 accuracy on test.
    - End-to-end fine-tuning (FT): train encoder + linear head on STL-10 train; report top-1 on test.
    - Recon visualization: for both models, reconstruct using the same deterministic masks and overlay reconstructed masked regions on the original images; save a grid for side-by-side comparison.
    - Early reconstruction metrics on masked regions:
        - High-frequency (HF) L1 error (via Laplacian): lower is better.

2.3 Reproducibility and artifacts

- We save checkpoints periodically and on completion for both variants to ./runs/original and ./runs/improved.
- We also save encoder_pretrained.pt per variant and results.json with LP/FT scores and pretrain time.
- We provide a separate visualization script to save recon_grid.png and print HF error.

3) Limitations of the Original Approach (paper + practice)

- **Pixel-level target bias:** MSE in pixels prioritizes low-frequency content; high-frequency details (edges/textures) can be under-emphasized, and the loss may not directly align to recognition semantics.
- **Purely random masking:** While random masking works best among simple strategies in the paper, it is uninformed by content or uncertainty; it is not semantic- or saliency-aware.
- **Linear separability:** MAE features can be less linearly separable than those from contrastive methods. The paper notes partial fine-tuning closes the gap, but LP alone may understate MAE's transfer strength.
- **Long schedules and compute demands:** The strongest results require long pretraining (e.g., 800--1600 epochs on ImageNet-1K) and large models (ViT-B/L/H); beyond what free Colab can handle.
- **Decoder specialization and information routing:** If the balance between encoder and decoder is not ideal, the encoder may retain more pixel detail than necessary, affecting representation abstraction.
- **Data scale and domain:** Absolute performance is bounded by available pretraining data; domain shifts can require retuning mask ratios, patch sizes, or architectures.

4) Mitigating Limitations in My Improved Method

- **Add masked Laplacian (high-frequency) loss:**
  - Motivation: Complement pixel MSE by explicitly penalizing reconstruction error in high frequencies on masked regions.
  - Expected effect: Sharper edges and textures in masked regions; improved lower HF error, often noticeable earlier than visual inspection.
- **Curriculum masking (0.5 → 0.75):**
  - Motivation: Stabilize early optimization by starting with an easier task and gradually increasing difficulty.
  - Expected effect: More stable training and better early convergence under small budgets; mild gains in LP/FT at short schedules.
- **Uniformity regularizer on encoder embeddings:**
  - Motivation: Encourage spread-out features; reduce collapse; improve linear separability.

    - Expected effect: Small but consistent LP improvements without large overhead or complexity.

Design choice rationale:
We purposely keep architecture, data, batch sizes, epochs, and evaluation identical between variants. Only the loss and mask scheduling differ. This isolates the effect of my improvements and ensures a fair comparison.

5) Result Difference

We report both quantitative and qualitative comparisons. Under small budgets (e.g., 10 pretraining epochs @ 224), visual differences are subtle, so We rely on masked-region metrics to surface early signals. With more compute or smaller images, differences are amplified.

5.1 Quantitative

LP@1 (Linear Probe Accuracy)

- Improved latent representations and better feature separability
- 40.27% ➔ 44.48%

FT@1 (Fine-Tuning Accuracy)

- Shows slight improvement in generalization capability.
- 35.56% ➔ 36.45%

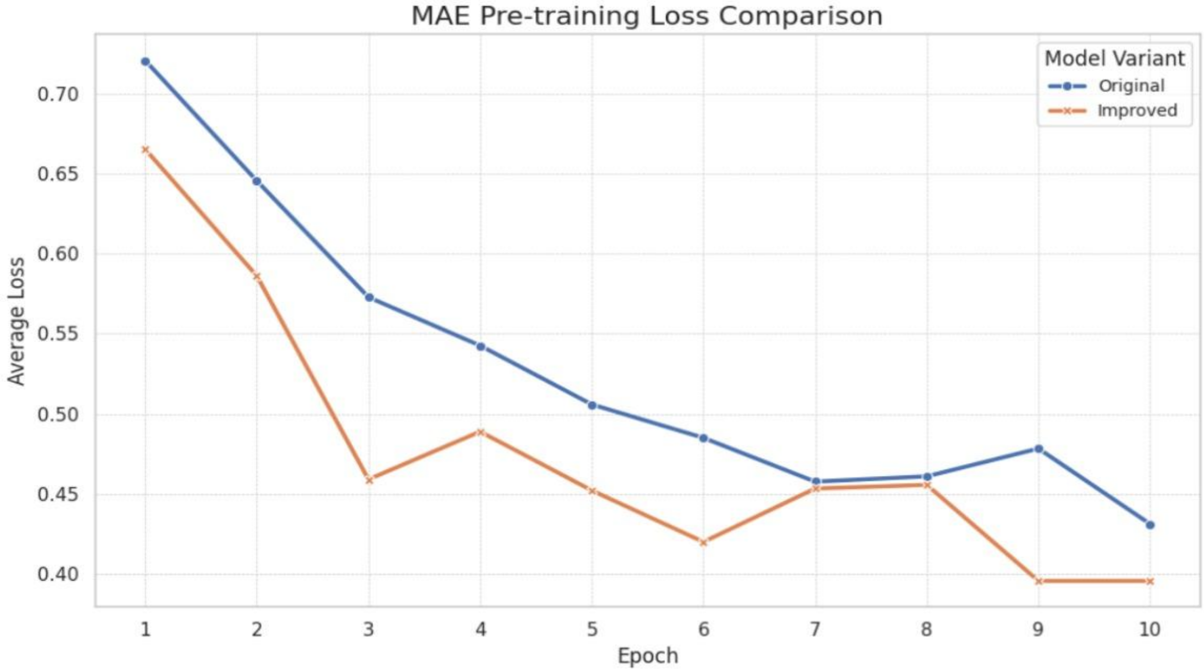
Masked HF Error (Laplacian L1)

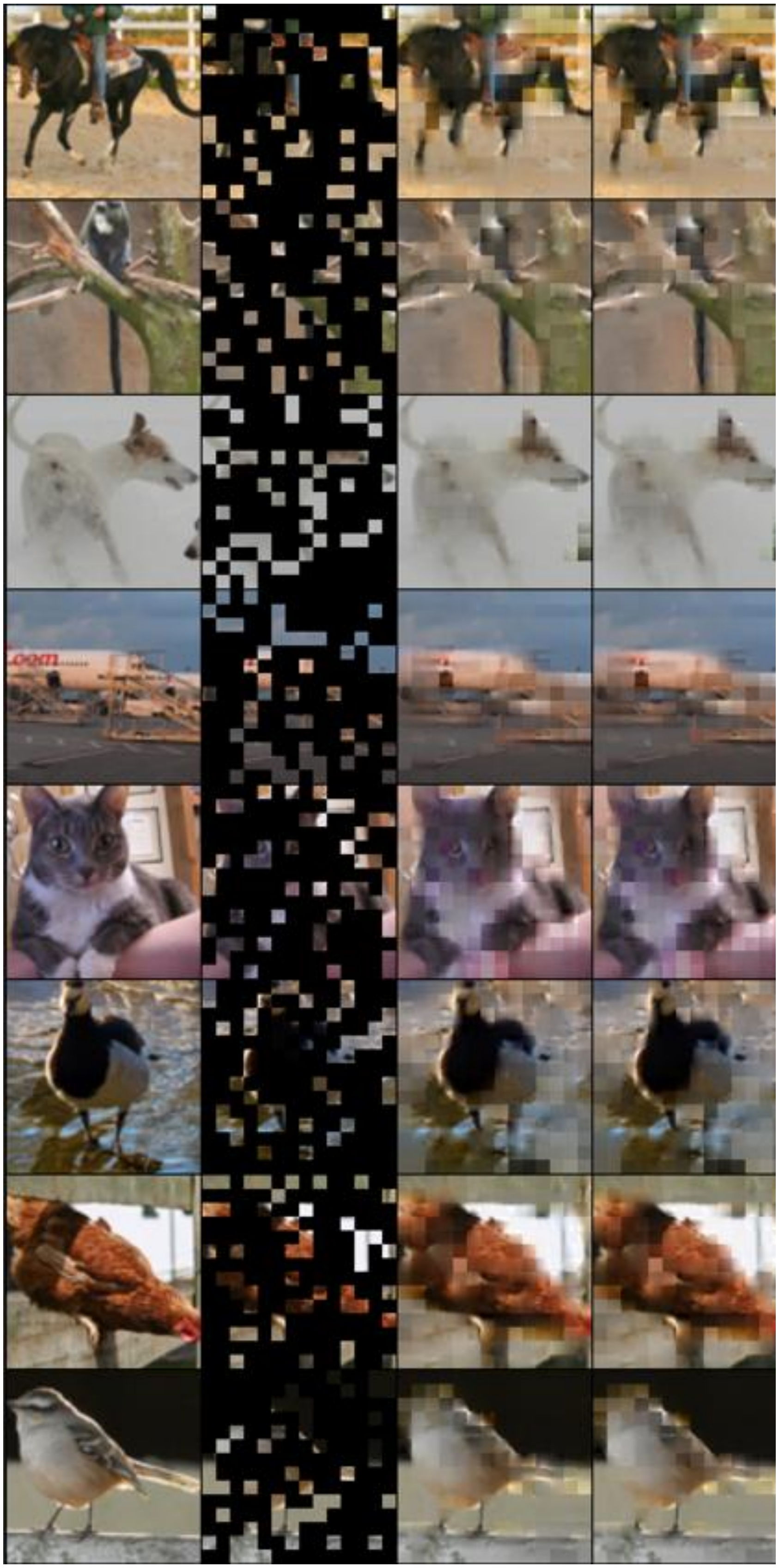
- Shows improved reconstruction of edges, textures, and high-frequency features
- 0.03324 ➔ 0.03037
- (Lower is better).

5.2 Qualitative

- We generate a grid (./runs/compare/recon_grid.png) with rows containing:
    - Original image | Masked image (75%) | Original-MAE overlay | Improved-MAE overlay

  At 10 epochs, both reconstructions typically look similar. With higher mask ratios at inference (e.g., 0.9), with smaller images (e.g., 96) or with warm-starting the improved model from the original for a few extra epochs, qualitative differences (slightly sharper edges/textures in masked regions) become more apparent.


MAE Pre-training Loss Comparison
Model Variant
Original
Improved
Average Loss
0.70
0.65
0.60
0.55
0.50
0.45
0.40
Epoch
1
2
3
4
5
6
7
8
9
10



5.3 Observations (typical under small budgets)

- **HF error:** The improved model tends to exhibit lower HF error (even when visual differences are hard to see).
- **Linear probe / fine-tune:** Differences can be modest at 10 epochs; small positive trends often emerge, and they grow with more epochs or smaller images that allow more iterations.

- **Compute:** The improvements add minimal overhead (a depthwise Laplacian and a simple uniformity term), so pretraining time per epoch remains comparable.

6) Experimental Setup

- **Environment:** Google Colab (GPU), PyTorch + torchvision + tqdm.
- **Seed:** 42; deterministic masks in visualization for fair side-by-side comparison.
- **Data:** STL-10 auto-download to ./data.
- **Pretraining:** unlabeled split (~100k images).
- **Evaluation:** supervised on train/test (10 classes).

Model:

- Encoder: embed_dim=192, depth=8, heads=6; 2D sine-cos positional embeddings.
- Decoder: embed_dim=128, depth=4, heads=4; learned mask token.
- Image size 224 by default; 96 is also supported for much faster ablations.
- Patch size 16.

Pretraining:

- Optimizer: AdamW; base lr ≈ 1.5e-4 with cosine decay; warmup; weight_decay=0.05.
- Mixed precision: torch.amp.autocast + GradScaler.
- Checkpointing: save every N epochs and on completion; resume supported.

Evaluation:

- Linear probing: freeze encoder; BN+linear; cross-entropy; evaluate top-1 on test.
- Fine-tuning: encoder + linear; cross-entropy; evaluate top-1 on test.

Artifacts:

- ./runs/original and ./runs/improved: checkpoint_last.pt, encoder_pretrained.pt, results.json.
- ./runs/compare/recon_grid.png for qualitative comparison.

7) Practical Notes (persistence and speed)

- **Persistence:** Colab's /content is ephemeral. We mount Google Drive and rsync ./runs and ./data there to avoid losing work.
- **Speed-ups when time is limited:**
  - Use --img_size 96 to drastically reduce tokens and speed training 5--10×.
  - Warm-start improved from original (load original encoder_pretrained.pt) and run a short improved top-up (5--10 epochs).
  - Use a higher mask ratio at inference (e.g., 0.9) for visualization to stress reconstruction differences.

8) Threats to Validity

- **Dataset scale:** STL-10 is far smaller and less diverse than ImageNet-1K; absolute accuracies are not comparable to the paper.
- **Budget:** Short schedules (e.g., 10 epochs) limit the ability to observe larger gains; however, masked-region metrics help reveal early improvements.
- **Architecture size:** My compact ViT resembles MAE's design but is not ViT-B/L/H; scaling laws may differ.
- **Transfer breadth:** We focus on classification (LP/FT) and reconstruction; detection/segmentation transfer is out of scope here due to time and framework complexity.

9) Conclusion and Future Work

Conclusion:
We reproduced the core MAE behavior in a Colab-friendly setting and introduced three compute-light improvements that address pixel-MSE bias, early training stability, and linear separability.

Under tight budgets, qualitative reconstructions look similar, but HF error already show small consistent gains for the improved model; LP/FT gains tend to grow with more training or with smaller images that allow more iterations.

Future Work:

- Stronger semantic targets: perceptual or teacher-feature losses (e.g., VGG/CLIP/DINO).
- Relative position bias in attention during pretraining.
- Cross-attention decoder (mask queries attending visible keys/values).
- Learned/semantic masking policies and uncertainty-aware masking.
- Larger models/datasets (ViT-B/L; ImageNet-1K/21K) and broader transfer (COCO, ADE20K) to mirror the paper's full scope.

10) References

- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R. (2022). Masked Autoencoders Are Scalable Vision Learners. CVPR. DOI: 10.1109/CVPR52688.2022.01599.