A. Training and Evaluation Details

A.1. Model training.

Our training configurations primarily followed the guidelines established by He et al. [27]. In the ImageNet-1K experiment, our model was trained for 800 epochs, utilizing the AdamW [43] optimizer with a constant weight decay of 5e-2 for a batch size of 1024. We set the maximum learning rate to 6e-4. Initially, the learning rate started at 0 and linearly increased to its maximum over the first 40 epochs, after which it followed a cosine schedule to gradually decrease to zero by the end of the training period. It is worth noting that the learning rate per sample, or effective learning rate, in our setup matched that of He et al. [27], although our maximum learning rate was set lower due to our batch size being a quarter of theirs. We applied random resizing, cropping, and horizontal flipping during training as part of our augmentation scheme. To enhance the quality of the learned representations in most experiments, we employed the normalized pixel loss, as proposed by [27]. In the ImageNet-100 experiment, we employed the identical training configuration used in the ImageNet-1K experiments. We train our model with 4 NVIDIA A40 GPUs and a completed trianing usually takes 20 hours on ImageNet-100 and 200 hours on ImageNet-1k.

A.2. Evaluation with Linear Probing.

For the ImageNet-1k dataset, we use the exact same evaluation protocols employed in He et al. [27], which includes random data augmentation.

For the ImageNet-100 dataset, we employed a simpler evaluation protocol: We train the linear classifier with a batch size of 1024 for 200 epochs, where the learning rate starts at 1e-2 and then decays towards 0 using a cosine scheduler. During this evaluation, we do not apply any data augmentation.

A.3. Modified Architecture

We present a visualization of our UNet transformer design, as outlined in Section 3.2, in Fig. 5. It's important to note that decayed identity shortcuts are exclusively implemented within the encoder block. Additionally, we establish skip connections from alternating blocks in the encoder to the decoder, following the UNet [50] architecture's design principles.

A.4. Learnable α_l over network layers

In the ablation study of learnable α_l, we apply no additional regularization beyond standard weight decay to the model parameters. Table 6 presents the α_l values for each layer. The results do not reveal any meaningful pattern across network depth.

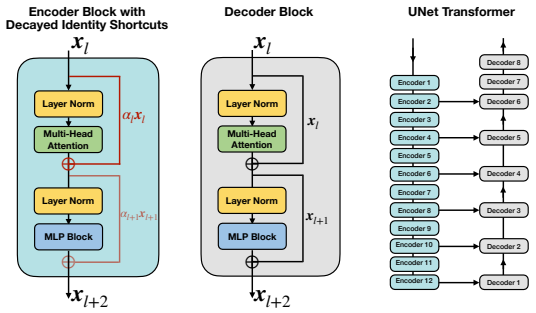


Figure 5. We present our enhanced UNet Transformer architecture for Masked Auto-encoder. (1) ***Left***: Our customized encoder blocks, equipped with our proposed decay identity shortcuts. (2) ***Middle***: Standard transformer blocks as the decoder blocks. (3) ***Right***: We incorporate the decay identity shortcuts exclusively within the encoder blocks of our UNet transformer and employ standard transformer blocks for the decoder. To support abstract representation learning at the bottleneck, *i.e.,* the last layer of the Encoder 12, we adopt the UNet [50] architecture and create skip connections that transmit every other encoder feature directly to the decoder.

B. Further analysis

B.1. Reconstruction quality.

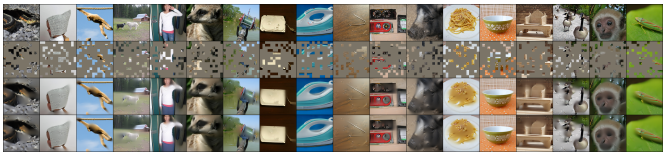


Figure 6. **Qualitative comparison of images reconstructed by MAE with and without our method.** We observe our method learns features with higher linear probing accuracy without compromising reconstruction quality. Row 1: ground truth test image. Row 2: images masked at 75%. Row 3: reconstructions with our method. Row 4: reconstructions with baseline MAE.

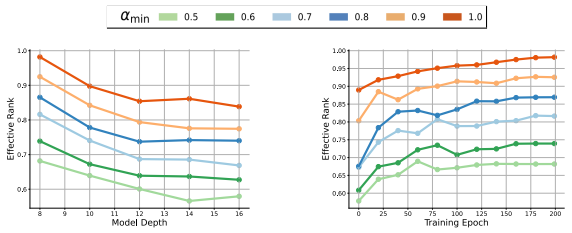
We qualitatively evaluate test images reconstructed by an MAE using our framework and images reconstructed by the original MAE. We show the reconstructed images in Figure 6. While the focus of our work is entirely to improve the representations learned by an encoder, we observe that our framework does not harm the reconstructions. Hence, there is no qualitative tradeoff for our increase in linear probing accuracy.

B.2. Abstraction and Low-rank in the Supervised Setting

In this experiment, we modify the standard ResNet-18 model to experiment with different depth models. By default, the ResNet-18 has a total of 8 residual blocks that are equally distributed into 4 layers. To increase model depth,

Layer Index	1	2	3	4	5	6	7	8	9	10	11	12
Attention	0.993	0.947	0.982	0.766	0.992	0.795	0.988	0.849	0.998	0.723	0.811	0.488
FFN	0.989	0.926	0.961	0.620	0.961	0.442	0.711	0.322	0.810	0.475	0.637	0.353

Table 6. Learnable α_l values for different model layers. In contrast to our proposed linear scheduler, learnable α_l does not exhibit a consistent decay pattern across network depth.



(a) Effective rank of ResNet for different depths at the convergence of the training.

(b) Effective rank of ResNet over training epoch.

Figure 7. **Dynamics of the feature rank in the supervised setup.** We train ResNet models for a supervised classification task on a small subset of ImageNet. And visualize (a) effective rank across different depths at convergence and (b) training dynamics of effective rank over time for various $\alpha_{\min}$. In (a) we see that at convergence, our method consistently decreases the feature rank with various depth and, in (b), this pattern is also shown for standard ResNet model at every stage of training.

we repeat residual blocks in the 3rd layer to obtain models varying between 8 and 16 total layers. At convergence, we observe that the models of different depths achieve a similar test accuracy. However, despite similar accuracies, in Figure 7a, which visualizes the effective rank over depth for different values of $\alpha_{\min}$, we see that the effective rank decreases over depth. Furthermore, smaller values of α_{min} consistently lead to features with lower effective rank.

Next, in Figure 7b, we try to verify our conjecture by visualizing the evolution of effective rank during training when choosing different $\alpha_{\min}$ in our method. For this experiment, we choose to train the standard ResNet-18 using our decayed identity shortcuts. In this setup, we observe that the optimal choice of $\alpha_{\min}$ slightly improves the test accuracy of the classification network: 94.4% with $\alpha_{\min} = 0.7$ *vs.* 93.6% with $\alpha_{\min} = 1.0$. We observe that the effective rank of the final features decreases with decreasing $\alpha_{\min}$. This supports our hypothesis that (1) decayed identity shortcuts substantially decrease the rank of bottleneck features and (2) decreasing feature rank may help improve learned features.

B.3. Further analysis on low-rank property

Consider a deep linear encoder with residual (skip) connections, $h_{l+1} = \alpha_l\, h_l + W_l\, h_l = (\alpha_l I + W_l)\, h_l$, where $\alpha_l \in (0, 1]$ is a depth-dependent decay factor. The overall encoder mapping is $H = (\alpha_{L-1} I + W_{L-1}) \cdots (\alpha_0 I + W_0)$, a product of perturbed identity transformations. In the standard $\alpha_l = 1$ case (ResNet), unweighted identity shortcuts cause the output to remain largely a copy of the input, reducing the need to learn complex transformations and yielding high-rank feature representations that preserve fine-grained input details. By contrast, decaying $\alpha_l < 1$ gradually suppresses this direct copy effect. Expanding the recursion shows that the raw input contribution to h_L is scaled by the product $\prod_{i=0}^{L-1} \alpha_i$, whereas contributions involving deeper transformations W_i omit some of these factors. Thus, for a monotonically decreasing schedule $\{\alpha_l\}$, the direct "identity" component $\prod_i \alpha_i\, x$ vanishes exponentially with depth. Gradient-based training of deep linear networks is known to converge to minimal-norm solutions, which correspond to mappings with reduced effective rank. Decayed identity shortcuts recover this low-rank inductive bias by progressively weakening the high-rank "copy" component of residual connections, while still preserving the optimization benefits of skip connections for stable training.

Moreover, the denoising (or masked) autoencoder objective accentuates the effect: since the input has noised (masked) entries, the encoder–decoder must rely on the most salient shared variations in the data (analogous to principal components) to infer missing content, rather than trivially forwarding local details. Thus, the encoder's output covariance concentrates in a subspace spanned by a few significant singular vectors, and the normalized singular-value entropy is correspondingly low. In other words, the representation has *low effective rank*, which aligns with empirical observations that networks with decayed shortcuts learn features of lower rank that are associated with improved downstream performance.

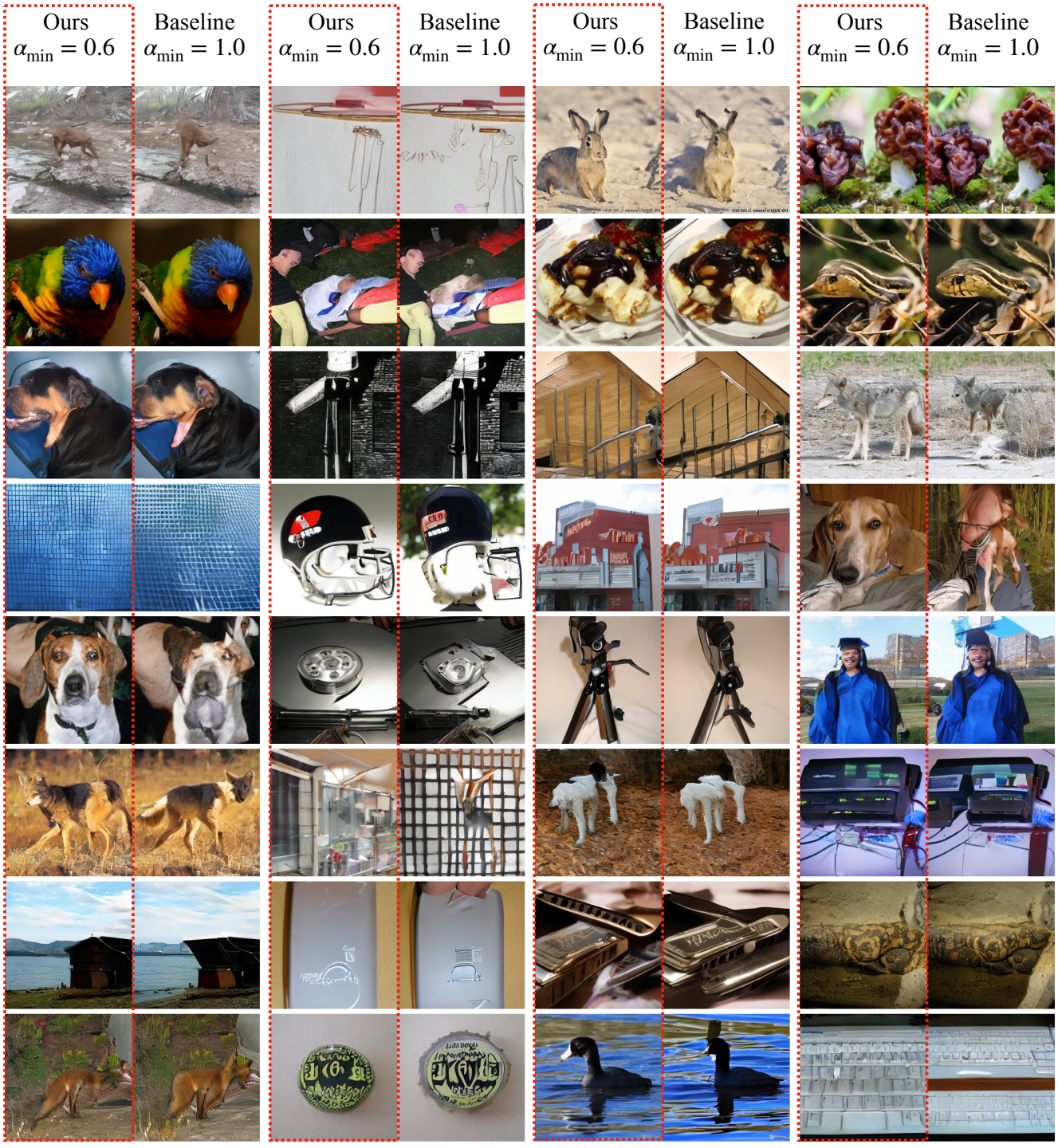


Figure 8. **Qualitative comparison of images generated by diffusion models.** Our method, decayed identity shortcuts with $\alpha_{\text{min}} = 0.6$, shows improved representation learning and produces higher-quality generated images compared to the baseline, which employs full residual connections ($\alpha_{\text{min}} = 1.0$).