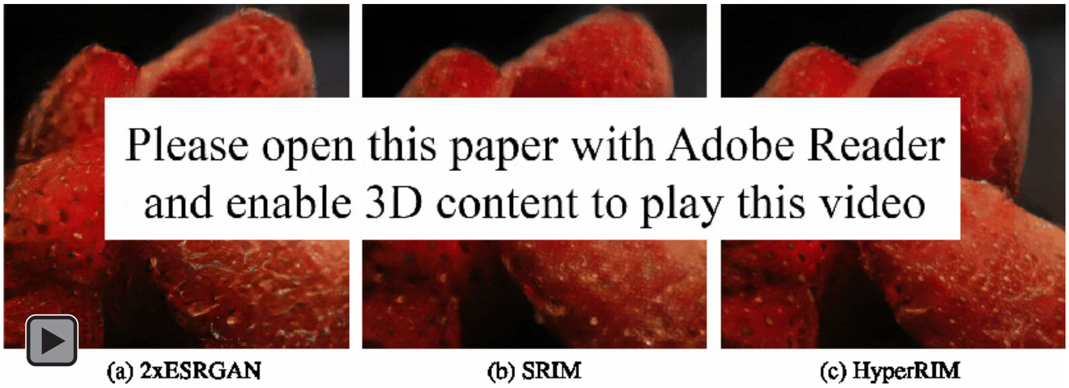


Figure 9: Visualization of output images from each method while traversing the space of random codes using gradient descent to reach the observed output image. As shown, (a) fails to reach the observed output, (b) comes close to the observed output but does not quite reach it, and (c) reaches the observed output and only encounters images with plausible content and texture. This reveals both the (i) precision and (ii) recall of each method, i.e.: the ability of each method to generate (i) *only* plausible images and (ii) *all* plausible images, which include the observed output. Since (a) cannot generate the observed output, its recall is low. Since (b) cannot reach the observed output, its recall is also unsatisfactory. Since (c) can reach the observed output and does so smoothly without generating an implausible image, the recall and precision of (c) are high.

A Precision and Recall

In Figure 9, we evaluate the precision and recall of each method, i.e.: whether the trained model can generate (a) *only* valid outputs, and (b) *all* valid outputs. Since only images that have a corresponding latent code $\mathbf{z}$ can be generated, we can explore the space of latent codes, which should be equivalent to the space of images that can be generated. We perform the following experiment: for a test image, we optimize over the latent code to try to find an image that is as close as possible to the original high-resolution image as measured by LPIPS and visualize the images we encounter along the way. For an ideal model, traversing the space of latent codes should (a) *only* pass through valid outputs (i.e. achieves high precision), and (b) be able to reach *any* valid image, including the original image (i.e.: achieves high recall). We find that HyperRIM is able to achieve better precision and recall than the baselines.

B Conditional IMLE Pseudocode

Algorithm 1 Conditional IMLE Training Procedure

Require: The set of inputs $\{\mathbf{x}_i\}_{i=1}^n$ and the set of corresponding observed outputs $\{\mathbf{y}_i\}_{i=1}^n$
  Initialize the parameters θ of the generator T_θ
  for $p = 1$ **to** N **do**
    Pick a random batch $S \subseteq \{1, \ldots, n\}$
    for $i \in S$ **do**
      Randomly generate i.i.d. m latent codes $\mathbf{z}_1, \ldots, \mathbf{z}_m$
      $\tilde{\mathbf{y}}_{i,j} \leftarrow T_\theta(\mathbf{x}_i, \mathbf{z}_j)\ \forall j \in [m]$
      $\sigma(i) \leftarrow \arg\min_j d(\mathbf{y}_i, \tilde{\mathbf{y}}_{i,j})\ \forall j \in [m]$
    end for
    for $q = 1$ **to** M **do**
      Pick a random mini-batch $\tilde{S} \subseteq S$
      $\theta \leftarrow \theta - \eta \nabla_\theta \left(\sum_{i \in \tilde{S}} d(\mathbf{y}_i, \tilde{\mathbf{y}}_{i,\sigma(i)}) \right) / |\tilde{S}|$
    end for
  end for
  return θ

(a) Input (b) Bicubic (c) 2xESRGAN

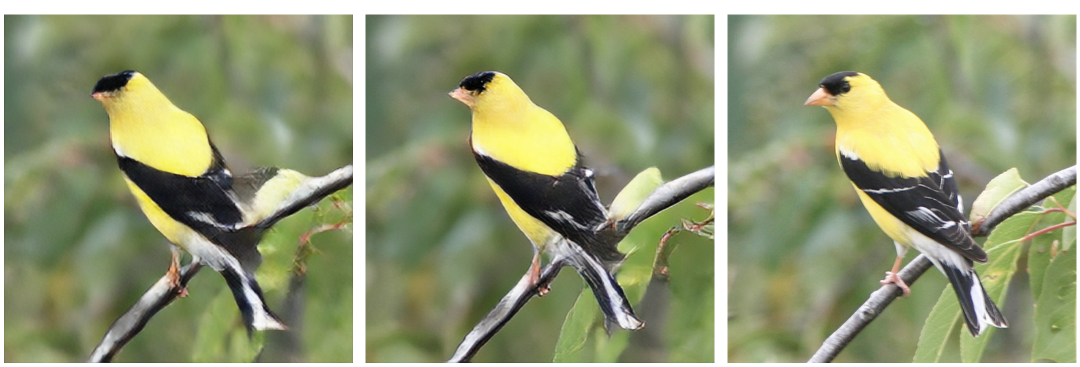
(d) SRIM (e) HyperRIM (f) Observed Output

C More Samples

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN



(d) SRIM (e) HyperRIM (f) Observed Output


(a) Input


(b) Bicubic


(c) 2xESRGAN

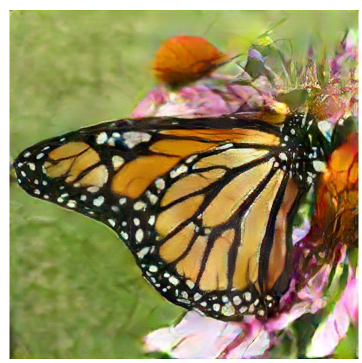

(d) SRIM

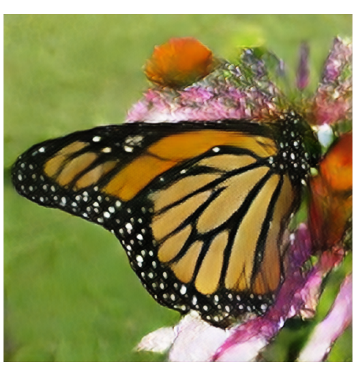

(e) HyperRIM

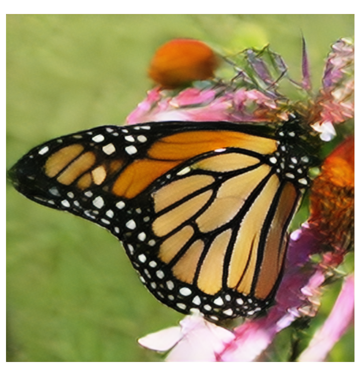

(f) Observed Output

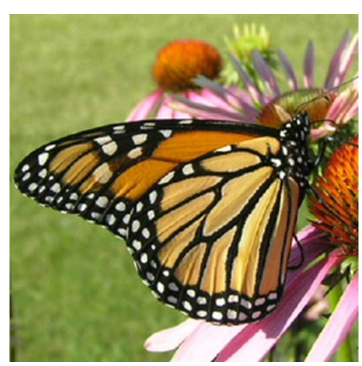

(a) Input


(b) Bicubic


(c) 2xESRGAN


(d) SRIM

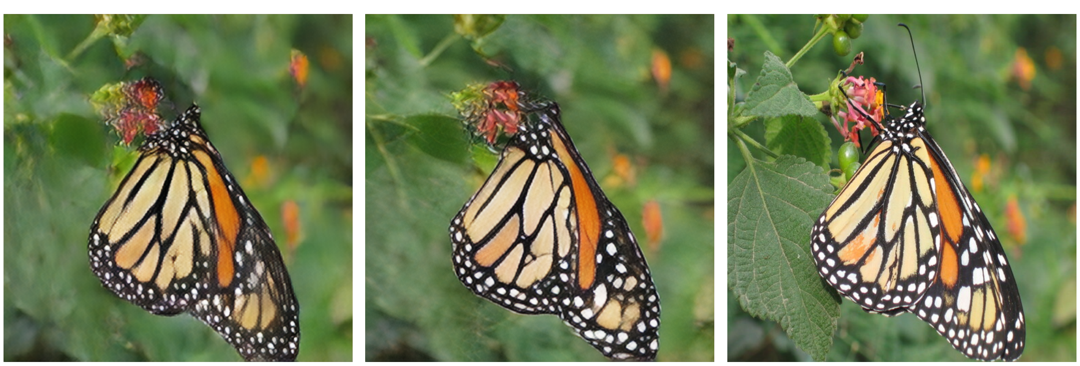

(e) HyperRIM


(f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN



(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

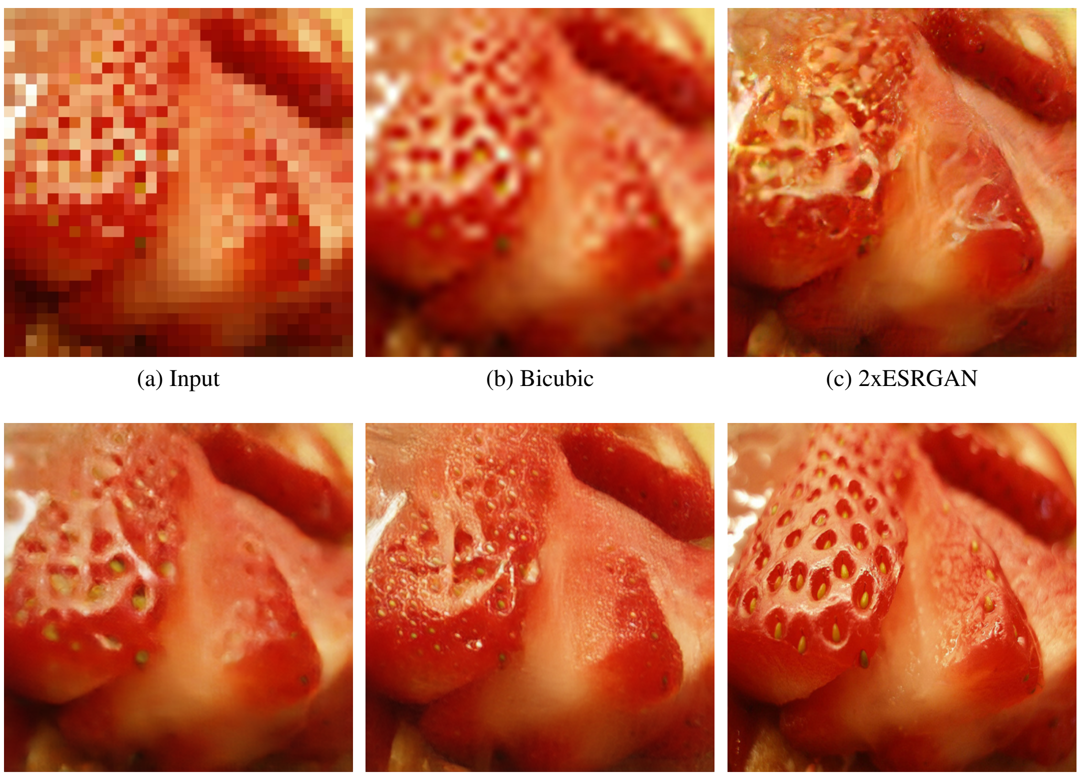
(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output

(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output



(a) Input (b) Bicubic (c) 2xESRGAN

(d) SRIM (e) HyperRIM (f) Observed Output