Style GAN

StyleGANTerminology. In the StyleGAN2 architecture, a 512 di-mensional latent code $z \in N(0, I)$ is mapped to an intermediate latent code $w \in R^{512}$ via a mapping network. The space of w is com-monly referred to as W. w is then sent to the generator G to produce the output image $I = G(w)$. In this process, w is copied several times and sent to different layers of the generator G to control different levels of attributes. Alternatively, one can also use different w for different layers, in which case the input would be $w \in R^{l \times 512} = W^+$,where l is the number of layers. This less constrained W^+ space is shown to be more expressive.

Interactive Point-based Manipulation

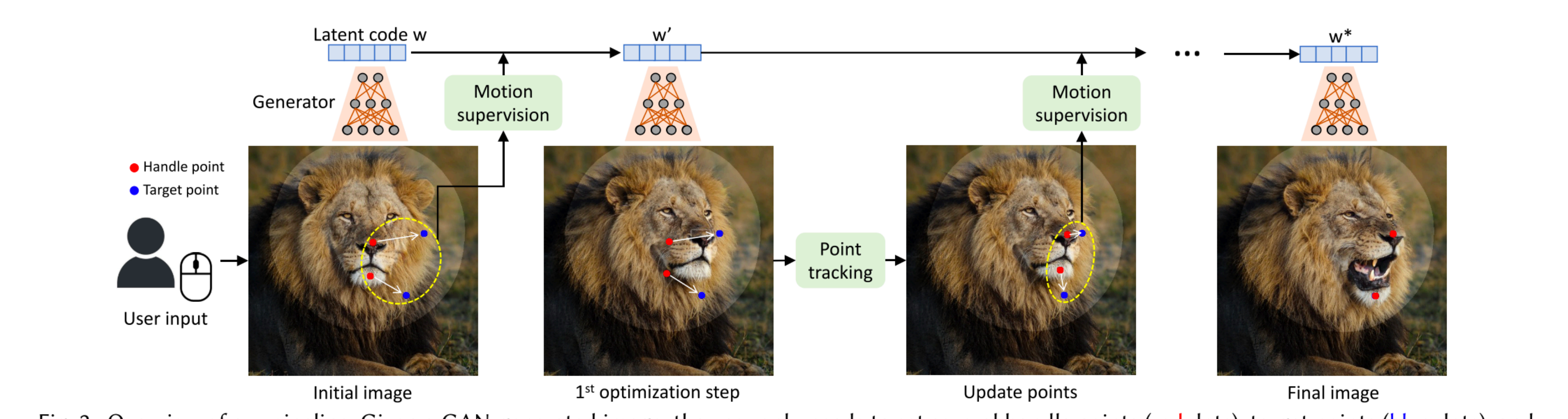

Fig. 2. Overview of our pipeline. Given a GAN-generated image, the user only needs to set several handle points (red dots), target points (blue dots), and optionally a mask denoting the movable region during editing (brighter area). Our approach iteratively performs *motion supervision* (Sec. 3.2) and *point tracking* (Sec. 3.3). The motion supervision step drives the handle points (red dots) to move towards the target points (blue dots) and the point tracking step updates the handle points to track the object in the image. This process continues until the handle points reach their corresponding target points.

As shown in Fig. 2, each optimization step consists of two sub-steps, including 1) motion supervision and 2) point tracking.

1. In motion supervision, a loss that enforces handle points to move towards target points is used to optimize the latent code w. After one optimization step, we get a new latent code w′ and a new image I′.
2. Thus, we then update the positions of the handle points to track the corresponding points on the object
3. Motion supervision step only moves each handle point towards its target by a small step but the exact length of the step is unclear.
4. This optimization process continues until the handle point reach the position of the target points, which usually takes **30-200 iterations** in our experiments

Motion Supervision

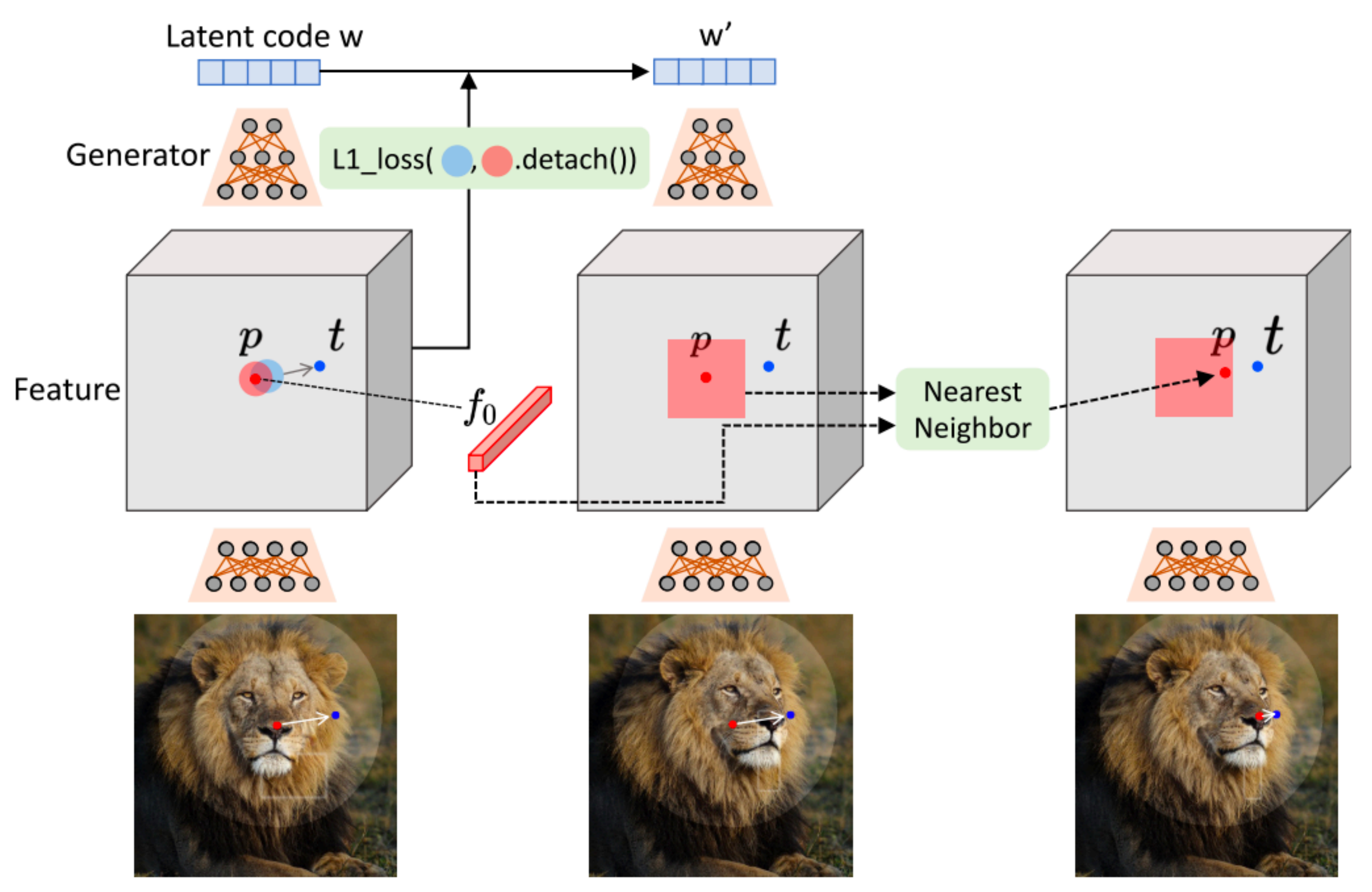

Fig. 3. Method. Our motion supervision is achieved via a shifted patch loss on the feature maps of the generator. We perform point tracking on the same feature space via the nearest neighbor search.

1. We consider the feature maps F after the 6th block ofStyleGAN2, which performs the best among all features due to a good trade-off between resolution and discriminativeness.
2. We resize F to have the same resolution as the final image via bilinear interpolation.
3. As shown in Fig. 3, to move a handle point p_i to the target point t_i , our idea is to supervise a small patch around p_i (red circle) to move towards t_i by a small step (blue circle).

Motion supervision loss:

$$\mathcal{L} = \sum_{i=0}^{n} \sum_{\boldsymbol{q}_i \in \Omega_1(\boldsymbol{p}_i, r_1)} \|\mathbf{F}(\boldsymbol{q}_i) - \mathbf{F}(\boldsymbol{q}_i + \boldsymbol{d}_i)\|_1 + \lambda \|(\mathbf{F} - \mathbf{F}_0) \cdot (1 - \mathbf{M})\|_1, \tag{1}$$

where $F(q)$ denotes the feature values of F at pixel q,

$$d_i = \frac{t_i - p_i}{\| t_i - p_i \|_2}$$

is a normalized vector pointing from p_i to t_i ($d_i = 0$ if $t_i = p_i$), and F_0 is the feature maps corresponding to the initial image. Note that the first term is summed up over all handle points $\{p_i\}$. As the components of $q_i + d_i$ are not integers, we obtain $F(q_i + d_i)$ via bilinear interpolation. In case the binary mask M is given, we keep the unmasked region fixed with a reconstruction loss shown as the second term.

4. At each motion supervision step, this loss is used to **optimize the latent code w for one step.** w can be optimized either in the W space or in the W^+. W^+ space is easier to achieve out-of-distribution manipulations. We observe that the spatial attributes of the image are mainly affected by the w for **the first 6 layers** while the remaining ones only affect appearance.

Point Tracking

The discriminative features of GANs well capture dense correspondence and thus tracking can be effectively performed via nearest neighbor search in a feature patch. Specifically, we denote the feature of the initial handle point as $f_i = F_0(p_i)$. We denote the patch around p_i as

$$\Omega_2(p_i, r_2) = \{(x, y) \mid |x - x_{p,i}| < r_2, |y - y_{p,i}| < r_2\}$$

Then the tracked point is obtained by searching for the nearest neighbor of f_i in $\Omega_2(p_i, r_2)$:

$$\boldsymbol{p}_i := \underset{\boldsymbol{q}_i \in \Omega_2(\boldsymbol{p}_i, r_2)}{\arg\min} \|\mathbf{F}'(\boldsymbol{q}_i) - f_i\|_1. \tag{2}$$

we are also **considering the feature maps F′ after the 6th block** of StyleGAN2

Implementation Details