A. Proofs of Remarks

We took Theorems 1 and 2 from Theorem 9.44 and 9.46 in the reference [43]. Their proofs require many definitions and propositions that have not been directly used in this manuscript. Hence, interested readers are referred to the reference [43]. We provide the proofs of remarks as follows:

Proof of Remark 1. A method that assumes linear attribute arithmetic edits an attribute k by adding an attribute vector a_k, which is independent of the position z, scaled by a change amount t, that is, $z + t\,a_k$. We can define a vector field $Z_k(z) \equiv a_k$ and the flow $\phi_k^t(z) = z + \int_0^t Z_k \mathrm{d}\tau = z + t\,a_k$. Therefore, it is a special case of a method that assumes attribute vector fields. Moreover, because it holds that $(\phi_l^s \circ \phi_k^t)(z) = (z + t\,a_k) + s\,a_l = (z + s\,a_l) + t\,a_k = (\phi_k^t \circ \phi_l^s)(z)$, its edits are commutative. □

Proof of Remark 2. According to Theorem 1, the flows of two vector fields do not commute in general. Edits by a method that assumes attribute vector fields follow their flows, which are not commuting in general. □

Proof of Remark 3. While the flow ψ_k on the Cartesianized latent space $\mathcal{V}$ is linear, the mapping f can be nonlinear. As a result, the flow ϕ_k on the latent space $\mathcal{Z}$ can be nonlinear. For commutativity,

$$\begin{aligned}\phi_l^s \circ \phi_k^t &= f^{-1} \circ \psi_l^s \circ f \circ f^{-1} \circ \psi_k^t \circ f \\ &= f^{-1} \circ \psi_l^s \circ \psi_k^t \circ f \\ &= f^{-1} \circ \psi_k^t \circ \psi_l^s \circ f \\ &= f^{-1} \circ \psi_k^t \circ f \circ f^{-1} \circ \psi_l^s \circ f \\ &= \phi_k^t \circ \phi_l^s.\end{aligned} \tag{A1}$$

□

Proof of Remark 4. Given DeCurvEd, we can always define an attribute vector filed Z_k on the latent space $\mathcal{Z}$ by pushforwarding the coordinate vector field $\tilde{Z}_k$ on the Cartesianized latent space $\mathcal{V}$; in particular,

$$Z_k(z) = (f^{-1})_*(\tilde{Z}_k) = \tfrac{\partial f^{-1}(v)}{\partial v} e_k \tag{A2}$$

at point z for $v = f(z)$. Hence, DeCurvEd always assumes a set of N vector fields. □

Proof of Remark 5. Suppose the mapping f of DeCurvEd is linear and non-degenerate (i.e., $f(z) = Mz$ for a non-degenerate matrix M) and that the attribute vector a_k on the latent space $\mathcal{Z}$ is defined as $a_k = M^{-1}e_k$. Then, it holds that

$$\begin{aligned}\phi_k^t(z) &= (f^{-1} \circ \psi_k^t \circ f)(x) \\ &= M^{-1}\psi_k^t(Mz) \\ &= M^{-1}(t\,e_k + Mz) \\ &= z + t\,a_k,\end{aligned} \tag{A3}$$

implying that an edit by a method that assumes linear attribute arithmetic is a special case of an edit by DeCurvEd. □

B. Algorithms

We summarize the edit by DeCurvEd in Algorithm 1. We adopted the unsupervised training framework for GANs proposed by Voynov and Babenko [62]; we summarize the framework in Algorithm 2. The only difference from the original implementation is the latent variable manipulation and loss function at lines 4 and 8, respectively.

For the change amount distribution $\mathcal{P}_\epsilon$, we first sampled the change amount ϵ' from a continuous uniform distribution $\mathcal{U}[-6, 6]$. Because the regression of very small changes does not contribute to proper learning, we rounded up small change amounts ϵ; in particular, we considered $\epsilon = \mathrm{sign}(\epsilon') \cdot \max(|\epsilon'|, 0.1)$.

Algorithm 1 Edit Attribute k

Input: latent variable z, attribute index k, change amount ϵ
Output: edited latent variable z'

Obtain a mapped latent variable $v = f(z)$.
Obtain the edited mapped latent variable $v' = v + \epsilon e_k$.
Obtain the edited latent variable $z' = f^{-1}(v')$.

Algorithm 2 Training CurvilinearGANSpace

1: Sample a latent variable z from its prior $p(z)$.
2: Sample an attribute index k to be changed from the discrete uniform distribution $\mathcal{U}\{1, N'\}$.
3: Sample a change amount ϵ from a continuous probability distribution $\mathcal{P}_\epsilon$.
4: Edit latent variable z using Algorithm 1.
5: Generate a pair of images $x = G(z)$ and $x' = G(z')$ using the generator G.
6: Feed the pair (x, x') to the reconstructor R, and get two outputs $(\hat{k}, \hat{\epsilon})$.
7: Obtain the loss function that compares the outputs $(\hat{k}, \hat{\epsilon})$ and the actual edit (k, ϵ).
8: Train the mapping f and the reconstructor R jointly to minimize the loss function in Eq. (7).

C. Details of Experiments

C.1. Datasets and Backbones

In the experiments, we used the same combinations of the datasets, GANs, and reconstructors used in previous studies [60, 62]. GANs were pre-trained before being combined with the proposed method. SNGANs were trained by us, and the other GANs were obtained from external repositories. Reconstructors were trained jointly with the mapping f from scratch. We summarize them below.

1. MNIST [42] + Spectral Norm GAN (SNGAN) [48] + LeNet [41]. MNIST is a dataset of 32×32 monochrome images of hand-written digits. SNGAN had ResNet-like architecture composed of three residual blocks. The dimension number N of the latent space $\mathcal{Z}$ is $N = 128$.
2. AnimeFaces dataset [31] + SNGAN + LeNet. AnimeFaces dataset contains 64×64 RGB images of cartoon characters' faces. SNGAN had ResNet-like architecture composed of four residual blocks with $N = 128$.
3. ILSVRC dataset [14] + BigGAN [7] + ResNet-18 [23]. ILSVRC dataset contains 128×128 RGB natural images. We obtained a pre-trained BigGAN with $N = 120$.
4. CelebA-HQ dataset [47] + ProgGAN [33] + ResNet-18. CelebA-HQ dataset contains 1024×1024 RGB images of celebrities' faces. We obtained a pre-trained ProgGAN with $N = 512$.
5. CelebA-HQ dataset + StyleGAN2 [36] + ResNet-18. We obtained a pre-trained StyleGAN2 with $N = 512$.
6. LSUN Car dataset [40] + StyleGAN2 [36] + ResNet-18. We obtained a pre-trained StyleGAN2 with $N = 512$.

C.2. Normalizing Flow

For a smooth bijective mapping f, we employ a flow-based model [39], namely a continuous normalizing flow (CNF) [9, 21]. The CNF assumes an ordinary differential equation (ODE) $\frac{\mathrm{d}u}{\mathrm{d}t} = g(u(t), t; \theta)$ on the space equivalent to the latent space $\mathcal{Z}$, where u denotes a state variable, t denotes the time, and the function g parameterized by θ maps the state u to its time derivative. Given an initial condition $u(0) = u_0$, the solution $u(t)$ is given by $u(t) = u_0 + \int_0^t g(u(\tau), \tau; \theta)\mathrm{d}\tau$. The function g is modeled by a neural network. We define mapping f as the integration of the above ODE from 0 to T, namely $f : z = u(0) \mapsto v = u(T)$. One can regard the mapping f to be parameterized by θ. Additionally, its inverse mapping f^{-1} is defined by the integration from T to 0. Owing to the characteristics of ODE, the mapping f is differentiable and bijective. In practice, a numerical integration (such as a Runge-Kutta method) is required to solve the above ODE; numerical errors are introduced in the mapping f, but they are negligible. We used the Dormand-Prince method to integrate the ODE for $T = 0.1$. For CurvilinearGANSpace, the log-determinant $\log\det \frac{\partial f}{\partial z}$ of the Jacobian of the mapping f is stochastically obtainable using Hutchinson's estimator [9, 21].

The CNF is guaranteed to be bijective, and serves as a universal approximator for smooth bijections, as proven in [57]. Hence, DeCurvEd's editing is guaranteed to be commutative at the design stage, not trained to be commutative. In practice, numerical errors during numerical integration cause a slight increase in the commutative error, but it remains negligible, as shown in Tables 3, A1, and A2. Note that other normalizing flows are available [39].

C.3. Evaluation Metrics

Index Identification This process adopts the official implementation of WarpedGANSpace [60]. We considered prepared latent variables z, edited the index k by $t \in [-\tau, \tau]$ in increments of δ, measured the attribute scores of generated images by the attribute predictors, and obtained the covariances between the change amount t and the measured attribute scores. τ and δ were set to $\tau = 3$ and $\delta = 0.15$ for StyleGAN2 and $\tau = 4.5$ and $\delta = 0.15$ for ProgGAN. We selected index k with the largest covariance as the one corresponding to that attribute. Note that the original manuscript [60] suggests using correlation; however, the implementation actually uses covariance.

Normalization We sampled 100 latent variables z, edited attribute k by t, and obtained the edited latent variables $z' = \phi_k^t(z)$. We generated the original $x = G(z)$ and edited $x' = G(z')$ images. Using a separate attribute predictor A_k, we obtained the change in the attribute score in the image space $\mathcal{X}$, that is, $A_k(x') - A_k(x)$. We obtained the average change $\mathbb{E}_z[A_k(G(\phi_k^t(z))) - A_k(G(z))]$ of the measured attribute score. We identified the change amount t in the latent space $\mathcal{Z}$ with which the average change was 5 degrees for the pitch and yaw attributes, and 0.1 for other attributes. We normalized the change amount t as $\tilde{t} = 0.1$ for each attribute and method separately.

Commutativity Error Intuitively, *commutativity error* is the error when edits of two attributes k and l are applied in reversed orders. We defined it as follows: Obtain a latent variable z, and edit attributes k and l by amounts t and s of latent variable z in both orders; namely, obtain two latent variables $z_1 = \phi_l^s(\phi_k^t(z))$ and $z_2 = \phi_k^t(\phi_l^s(z))$. Then, generate images $x_1 = G(z_1)$ and $x_2 = G(z_2)$, and evaluate the attributes scores of the generated images x_1 and x_2 by separate attribute predictors A_k and A_l. The commutativity error for attribute k is the absolute difference $|A_k(x_1) - A_k(x_2)|$ in the attribute scores $A_k(x_1)$ and $A_k(x_2)$. We obtained the errors for attributes k and l; namely

$$\begin{aligned} &|A_k(G(\phi_l^s(\phi_k^t(z)))) - A_k(G(\phi_k^t(\phi_l^s(z))))|, \\ &|A_l(G(\phi_l^s(\phi_k^t(z)))) - A_l(G(\phi_k^t(\phi_l^s(z))))|. \end{aligned} \tag{A4}$$

Table A1. Commutativity Errors [%] of ProgGAN.

	ProgGAN		
	S+Y	**B+P**	**S+B+Y+P**
LinearGANSpace [62]	**0.09 / 0.12**	**0.09/ 0.13**	**0.08 / 0.07 / 0.12 / 0.20**
WarpedGANSpace [60]	5.86 / 1.97	5.87 / 2.49	1.51 / 7.80 / 3.00 / 2.08
CurvilinearGANSpace (ours)	0.32 / 0.44	0.24 / 0.59	0.22 / 0.25 / 0.64 / 0.51

S: “Smile”, B: “bangs”, P: “pitch”, Y: “yaw”.

Table A2. Commutativity Errors [%] of StyleGAN2.

	StyleGAN2		
	G+B+Y	**A+R+P**	**A+B+G+R+Y+P**
LinearGANSpace [62]	**0.04 / 0.02 / 0.21**	**0.01 / 0.01 / 0.16**	**0.02 / 0.02 / 0.06 / 0.02 / 0.12 / 0.45**
WarpedGANSpace [60]	3.58 / 1.05 / 8.54	3.77 / 3.28 / 3.33	9.48 / 1.71 / 7.43 / 1.19 / 6.90 / 6.52
CurvilinearGANSpace (ours)	0.23 / 0.07 / 0.51	0.09 / 0.07 / 0.90	0.06 / 0.03 / 0.27/ 0.10 / 0.89 / 0.60

A: “age”, G: “gender”, R: “race”, B: “bangs”, P: “pitch”, Y: “yaw”.

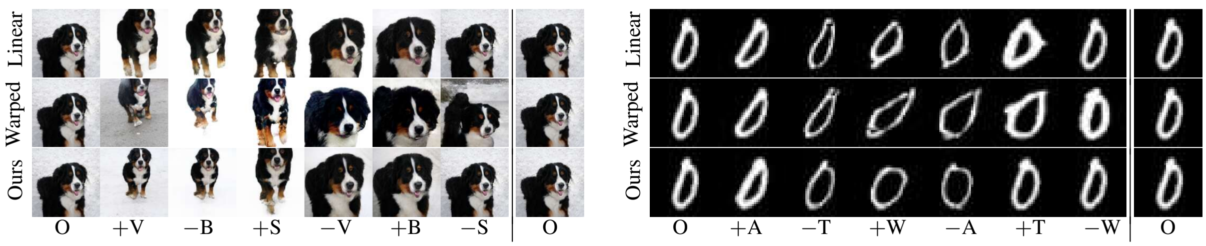

Figure A1. Results of sequential editing of attributes from left to right. (left) ILSVRC+BigGAN. (right) MNIST+SNGAN. Each row shows the results of LinearGANSpace, WarpedGANSpace, and CurvilinearGANSpace, from top to bottom. The signs + and − denote the addition and the subtraction of the corresponding attributes, respectively. O: original, V: “vertical position”, B: “background”, S: “object size”. A: “angle”, T: “thickness”, W: “width”.

We set the change amount to $\tilde{t} = \tilde{s} = 0.1$ in the normalized scale. This error vanishes if edits of attributes k and l are commutative. For over two attributes, we obtained the difference in attribute score between edited results in the given order and in the reverse order.

Side Effect Error We defined the *side effect error* as follows: Obtain a latent variable z, and edit attribute k by t, obtaining $z' = \phi_k^t(z)$. Then, measure the difference $|A_l(x) - A_l(x')|$ in the score of other attribute l between generated images $x = G(z)$ and $x' = G(z')$, and normalize it by that for the target attribute k; namely

$$\frac{|A_l(G(z)) - A_l(G(\phi_k^t(z)))|}{|A_k(G(z)) - A_k(G(\phi_k^t(z)))|} \tag{A5}$$

We set the change amount to $\tilde{t} = 0.1$ in the normalized scale. This error vanishes if the edit of attribute k has no side effect on attribute l.

Identity Error We defined the *identity error* as follows: Obtain a latent variable z, and edit attribute k by t, obtaining $z' = \phi_k^t(z)$. Then, evaluate the identity score $I(x, x')$ between the generated images $x = G(z)$ and $x' = G(z')$. The identity error is defined as 1.0 minus the identity score; namely,

$$1 - I(G(z), G(\phi_k^t(z))). \tag{A6}$$

We set the change amount to $\tilde{t} = 0.1$ in the normalized scale. For more than two attributes, we also obtained 1.0 minus the identity score between the original and edited images.

D. Additional Results

D.1. Commutativity

In this section, we provide additional results for demonstrating the commutativity of image editing methods. In a way similar to Table 3, Tables A1 and A2 show the commutativity errors. For any combination of attributes, the errors of LinearGANSpace and CurvilinearGANSpace were always less than 0.9 %, whereas those of WarpedGANSpace varied between 1.0 % and 9.5 %.

Following Fig. 2, we edited image attributes sequentially

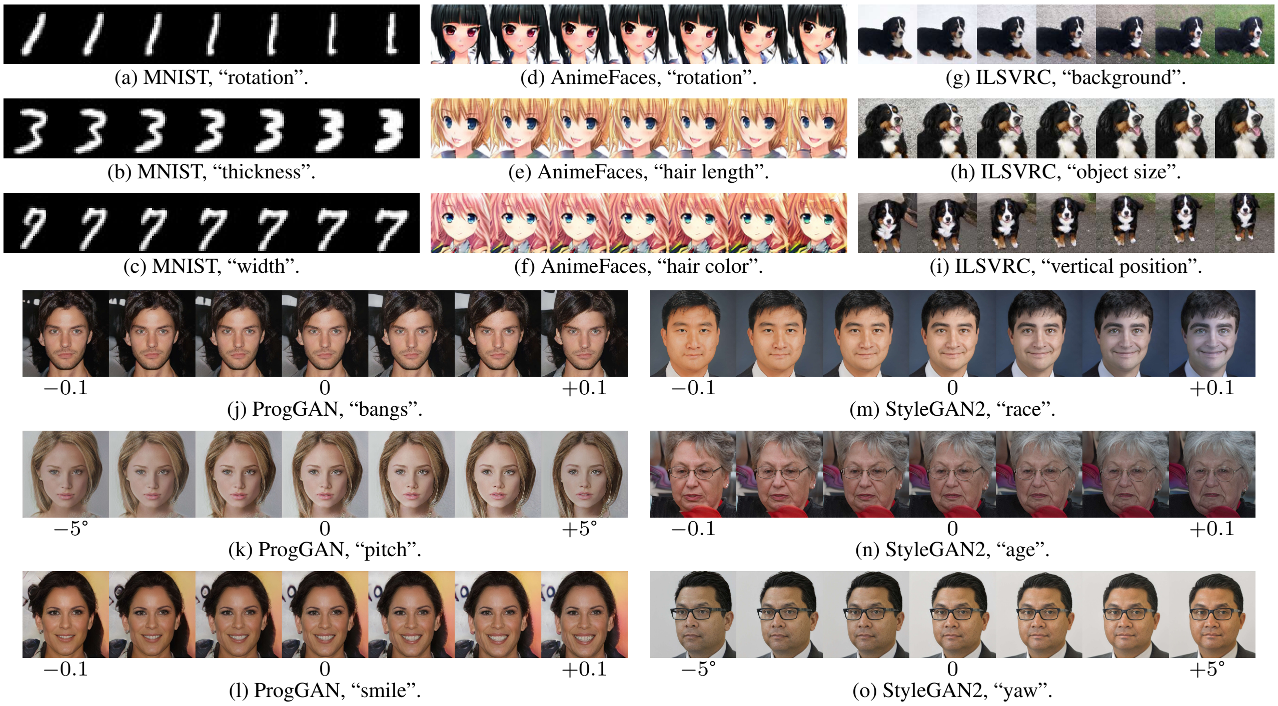

Figure A2. Visualization results of CurvilinearGANSpace. The models and edited attributes k are shown below the panels. The image in the center is the original, the images on the right have attributes added, and the images on the left have attributes subtracted, in the same way as Figs. 3 and 4.

so that the total amount of change is zero and summarized the results in Fig. A1. When using LinearGANSpace or CurvilinearGANSpace, the images returned to their original states. WarpedGANSpace did not restore the original images; the position and background of the dog were not restored, and the digit was thickened.

These results also demonstrate that the image editing by LinearGANSpace and CurvilinearGANSpace is commutative and that by WarpedGANSpace is non-commutative.

D.2. More Visualization

We provide further visualization results of CurvilinearGANSpace in Fig. A2, demonstrating that CurvilinearGANSpace identified and edited various attributes without severe side effects.