

# 从生成对抗网络隐空间中生成新桥型的尝试

张洪俊

(万世先数智交通科技有限公司, 南京 210016)

**摘要:** 尝试利用生成式人工智能技术生成新桥型。采用对称结构的三跨梁式桥、拱式桥、斜拉桥、悬索桥图像数据集, 基于 Python 编程语言、TensorFlow 及 Keras 深度学习平台框架, 以及 Wasserstein 损失函数、Lipschitz 约束, 构建和训练生成对抗网络。从得到的低维桥型隐空间采样, 能够生成不对称结构的新桥型。生成对抗网络能够在人类原创桥型的基础上, 将不同结构部件进行有机搭配, 创造生成新桥型, 一定程度上具有类似人类的原创能力。生成式人工智能技术能够打开想象空间, 给予人类启发。

**关键词:** 生成式人工智能; 桥型创新; 生成对抗网络; 隐空间; 深度学习

**中图分类号:** U448.2; TP181

**文献标志码:** A

## An attempt to generate new bridge types from latent space of generative adversarial network

ZHANG Hong-jun

Wanshi Antecedence Digital Intelligence Traffic Technology Co., Ltd, Nanjing, 210016, China

**Abstract:** Try to generate new bridge types using generative artificial intelligence technology. Using symmetric structured image dataset of three-span beam bridge, arch bridge, cable-stayed bridge and suspension bridge, based on Python programming language, TensorFlow and Keras deep learning platform framework, as well as Wasserstein loss function and Lipschitz constraints, generative adversarial network is constructed and trained. From the obtained low dimensional bridge-type latent space sampling, new bridge types with asymmetric structures can be generated. Generative adversarial network can organically combine different structural components on the basis of human original bridge types, creating new bridge types that have a certain degree of human original ability. Generative artificial intelligence technology can open up imagination space and inspire humanity.

**Keywords:** generative artificial intelligence; bridge-type innovation; generative adversarial network; latent space; deep learning

### 0 引言

桥梁设计师无法像艺术家一样, 仅凭想象肆意绘制天马行空的图案。桥梁设计首先要满足结构受力要求。不太合理的结构设计, 会极大地增加桥梁建造成本, 而力学上不可行的方案则无法落地实施。虽然桥型创新受结构力学严格制约, 但是当今桥型创新并非全不可能, 因为人类大规模建造桥梁的历史也就 100 多年, 桥梁领域尚有许多未知空间待去探索。近 10 年来生成式人工智能技术进展惊人, 这为桥型创新提供了新的手段。

作者之前论文<sup>[1]</sup>采用变分自编码器 (VAE), 生成了几种技术可行的新组合桥型, 但它们仅是数据集中两种桥型的简单叠合, 创新思路太单一。这是因为变分自编码器将标签相同的样本分布在同一区域, 不同区域之间存在搭接处, 为了满足输入样本与生成图像的交叉熵最低要求, 于是搭接处的采样总是具有周边不同桥型的特征, 这个特性让其能够生成组合桥型, 但也限制了其自由创造能力。(注: 变分自编码器并非没有原创能力, 这里仅能简单叠合的原因, 是与数据集、模型参数等前置条件相关。)

生成对抗网络 (GAN) 在计算机视觉、自然语言处理等领域取得了巨大成功, 它不要求输入样本与生成图像之间像素一一匹配, 而是判断图像视觉统计特征是否吻合, 这让它比变分自编码器更有创造力。本文采用生成对抗网络, 基于之前同样的数据集, 再次尝试桥型创新 (本文数据集与源代码开源地址 <https://github.com/QQ583304953/Bridge-GAN>)。

### 1 生成对抗网络简介

#### 1.1 概述

生成式人工智能技术可以粗略分为六类: 变分自编码器 (Variational Autoencoders)、生成对抗网络 (Generative Adversarial Networks)、自回归模型 (Autoregressive Models) 和 Transformers、标准化流模型 (Normalizing Flow Models)、基于能量模型 (Energy-Based Models) 和扩散模型 (Diffusion Models)、多模态模

型 (Multimodel Models) [2]。许多类别之间并没有绝对的边界，互相之间往往融合渗透。当代生成式人工智能发展时间线见下图（图 1）：

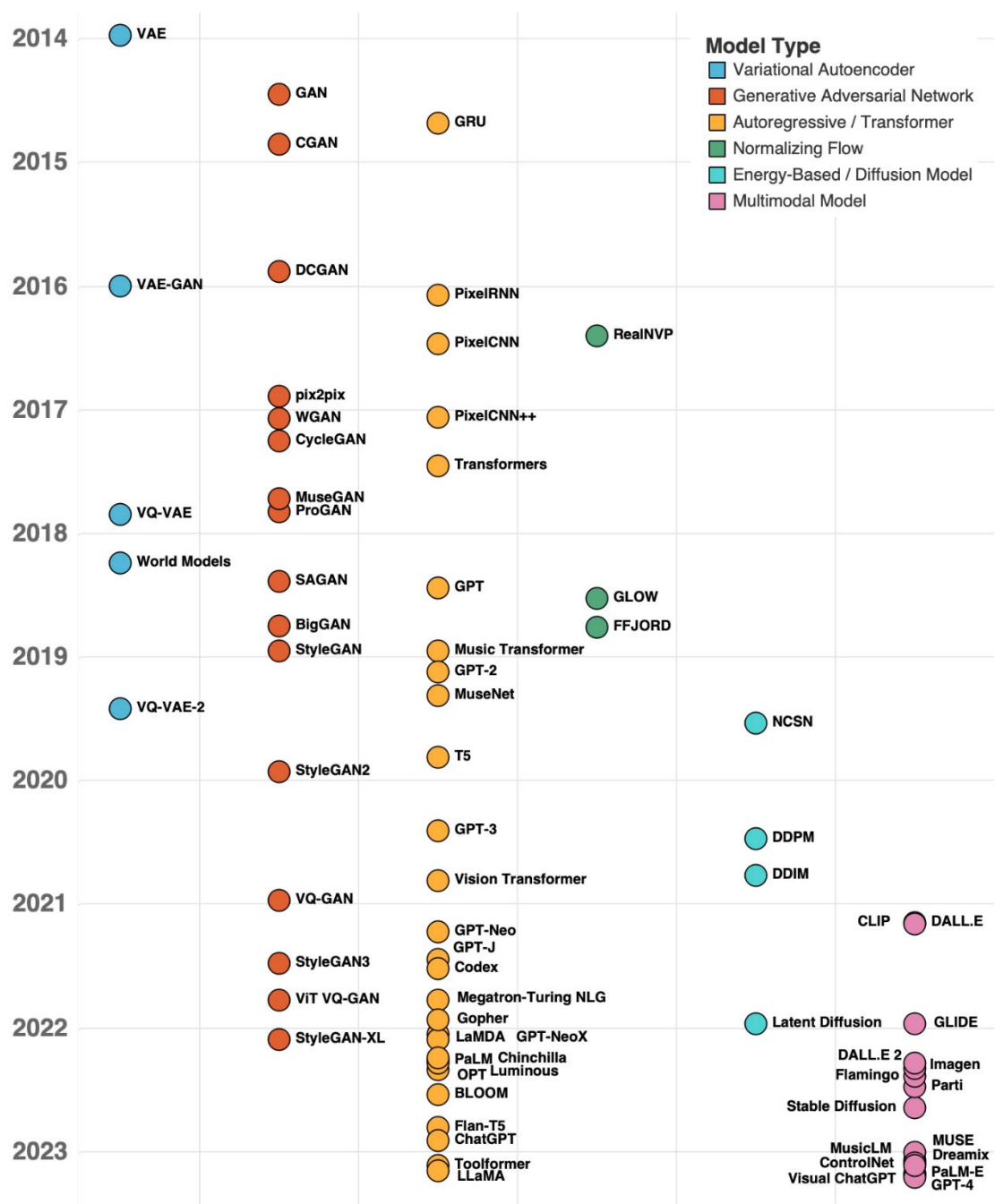


图 1 生成式人工智能发展时间线

Fig.1 The timeline for the development of generative artificial intelligence

资料来源: [https://github.com/davidADSP/Generative\\_Deep\\_Learning\\_2nd\\_Edition/tree/main/docs/timeline.png](https://github.com/davidADSP/Generative_Deep_Learning_2nd_Edition/tree/main/docs/timeline.png)

1.2 生成对抗网络

生成对抗网络可以代替变分自编码器来学习图像的隐空间，它能够让生成图像与训练图像在统计上几乎无法区分，从而生成逼真的全新图像。

它包含一个判别器（Discriminator）和一个生成器（Generator）[3]。判别器作用是：通过学习输入图像（训练集的和生成器合成的）的特征，区分输入的图像是训练集的（真实的），还是生成器合成的（虚假的）。生成器作用是：在判别器的指导下，通过学习，将一个随机向量（隐空间中的点）解码为具有训练集特征的合成图像。两个网络交替训练，能力同步提高，直到生成器生成的图像能够以假乱真，以至于判别器无法区分，具体见下图（图 2）：

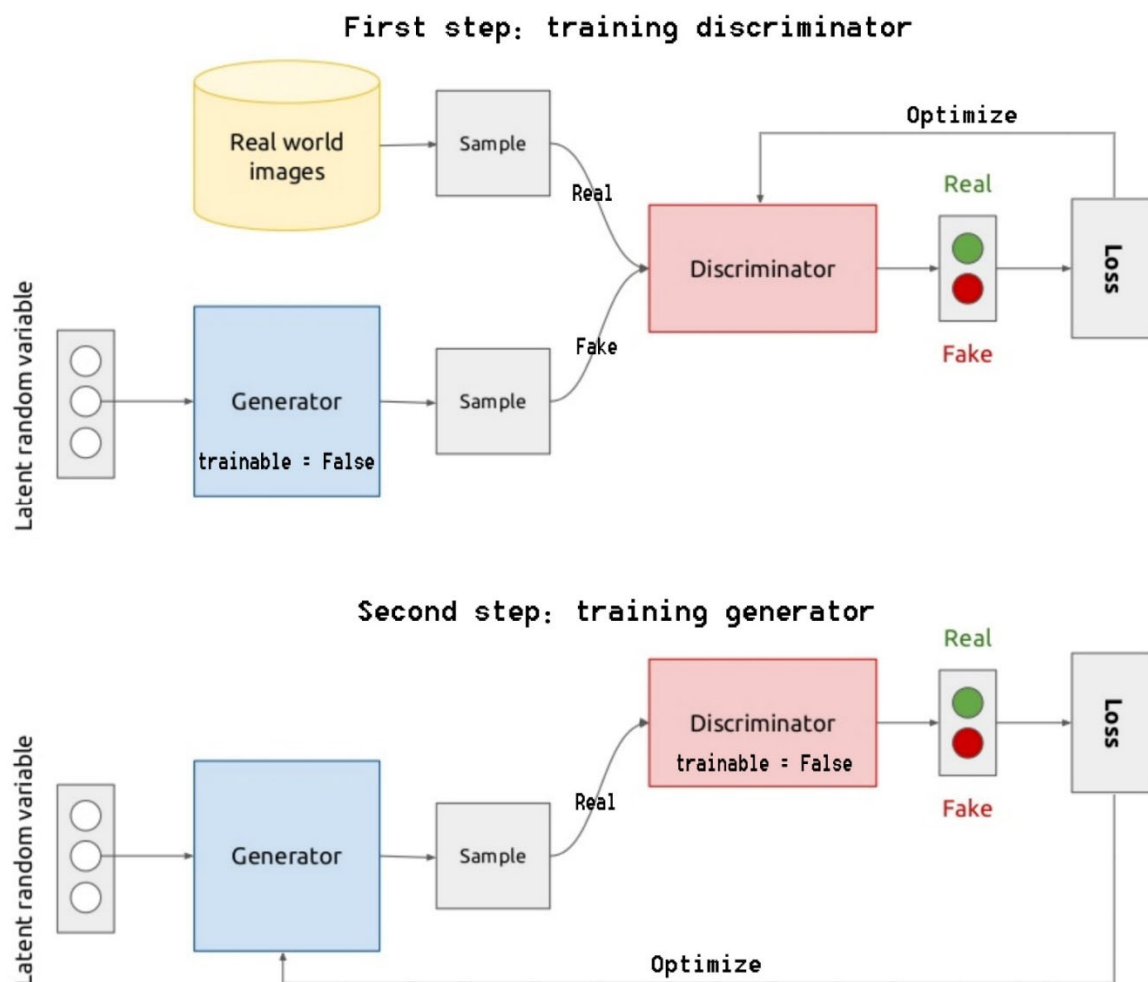


图2 生成对抗网络架构图

Fig.2 Architecture diagram of generative adversarial network

### 1.3 WGAN-GP

本次实践开始时尝试采用标准 GAN 架构，但遭遇模式崩溃（Mode collapse）难题，生成器只能输出一种桥型（如 V 形墩刚构梁式桥），于是改用 WGAN-GP 架构<sup>[4]</sup>。

（1）WGAN-GP 架构采用 Wasserstein 损失函数代替二元交叉熵。

Wasserstein 损失函数<sup>[5]</sup>为： $-1/n * \sum (y * p)$ 。

式中： $n$  为样本数目；标签  $y$  取值为 1（真实的，或者人为设定为真实的）、-1（虚假的）； $p$  为判别器输出值（实数标量）。

训练判别器时损失为： $-1/n * \sum \{D(x) - D[G(z)]\}$ 。

训练生成器时损失为： $-1/n * \sum \{D[G(z)]\}$ 。

式中： $D(x)$  为真实图像的判别器输出值； $z$  为隐空间采样点； $G(z)$  为生成器输出值（图像）； $D[G(z)]$  为生成图像的判别器输出值。

为了让损失变小，梯度下降算法迫使判别器通过修正权重参数，达到以下结果： $D(x)$  尽可能大、 $D[G(z)]$  尽可能小。同理，生成器优化的结果是： $D[G(z)]$  尽可能大。

这里 Wasserstein 损失函数从数学角度看，既没有上界也没有下界，损失不存在最小值（而 GAN 的二元交叉熵损失是有最小值的）。

可以通过想象理解 Wasserstein 损失函数的作用：①判别器训练时，一名造假者（生成器）仿制名师画作，鉴定商（判别器）把赝品与名师真迹进行对照评估，努力提升鉴定技能，将真迹评分为正值、赝品评分为负值，并且让评分尽可能两极化。②生成器训练时，为了让鉴定商打出尽可能高的正值评分，造假者唯有竭尽所能改进仿制技艺。可见，逻辑上 WGAN 与 GAN 基本相同。

理想状态下，博弈达到均衡时：判别器难以识别真假， $D(x)$  约等于  $D[G(z)]$ ，判别器损失由较大负值向 0 趋近；生成器能力也提升到优秀状态，生成器损失也向 0 趋近。

（2）训练过程中，对于真假不同类别的图像输入，要求判别器评分两极化，判别器的输出可以是  $(-\infty, +\infty)$  任何值。在梯度下降算法作用下，判别器函数（把判别器假象成图像映射为标量的  $y=f(x)$  高维函数）的梯度会变得

很陡峭，从而实现损失很小（负值）的目标。但陡峭的梯度导致模型稳定性差。

故需要施加约束，让判别器函数是 K-Lipschitz 函数，方法是：在判别器的损失函数中加入一个梯度惩罚项（Gradient penalty）<sup>[6]</sup>，该项会在判别器函数的梯度范数偏离 K 值时施加惩罚。训练结束时，判别器函数的梯度将接近于设定的 K 值（这里 K 值取 1）。

计算示例：判别器函数的梯度=[3,4]，梯度惩罚项=[sqrt(3<sup>2</sup>+4<sup>2</sup>)-1]<sup>2</sup>=16，相对系数为 10、判别器 Wasserstein 损失为-260，则判别器总损失=Wasserstein 损失+相对系数\*梯度惩罚项=-260+10\*16=-100。

2 从生成对抗网络隐空间中生成新桥型的尝试

2.1 数据集

采用作者之前论文<sup>[1]</sup>的数据集，即每种桥型两种子类（分别为等截面梁式桥、V 形墩刚构梁式桥、上承式拱式桥、下承式拱式桥、竖琴式斜拉桥、扇式斜拉桥、竖吊杆悬索桥、斜吊杆悬索桥），且均为三跨（梁式桥为 80+140+80m，其它桥型均为 67+166+67m），结构对称。

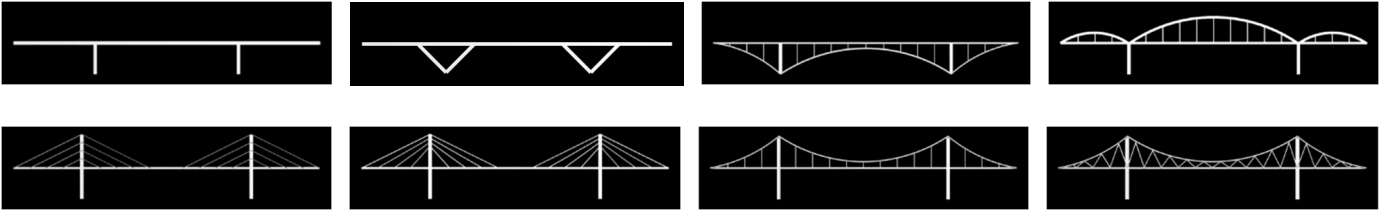


图 3 各桥型立面灰度图  
Fig.3 Grayscale image of each bridge facade

每个子类桥型 1200 张各不相同的图片，整个数据集共 9600 张图片。

2.2 生成对抗网络构建

基于 Python3.8 编程语言、TensorFlow2.6 及 Keras2.6 深度学习平台框架，构建和训练生成对抗网络。经过测试，2 维隐空间能够取得较为理想的结果。

（1）评论者（判别器）的架构：

它接收训练集的和生成器合成的图像作为输入，学习正确区分两者。  
先为图像创建一个输入层，然后依次传递给 6 个 Conv2D 层（激活函数 LeakyReLU），最后展平连接到 Dense 层（线性变换），输出结果是评分标量值。模型摘要见下表：

表 1 评论者模型摘要  
Tab.1 model summary of critic

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 512, 1)]	0
conv2d (Conv2D)	(None, 64, 256, 64)	1088
leaky_re_lu (LeakyReLU)	(None, 64, 256, 64)	0
dropout (Dropout)	(None, 64, 256, 64)	0
conv2d_1 (Conv2D)	(None, 32, 128, 128)	131200
leaky_re_lu_1 (LeakyReLU)	(None, 32, 128, 128)	0
dropout_1 (Dropout)	(None, 32, 128, 128)	0
conv2d_2 (Conv2D)	(None, 16, 64, 128)	262272
leaky_re_lu_2 (LeakyReLU)	(None, 16, 64, 128)	0
dropout_2 (Dropout)	(None, 16, 64, 128)	0
conv2d_3 (Conv2D)	(None, 8, 32, 128)	262272
leaky_re_lu_3 (LeakyReLU)	(None, 8, 32, 128)	0
dropout_3 (Dropout)	(None, 8, 32, 128)	0
conv2d_4 (Conv2D)	(None, 4, 16, 128)	262272

leaky_re_lu_4 (LeakyReLU)	(None, 4, 16, 128)	0
dropout_4 (Dropout)	(None, 4, 16, 128)	0
conv2d_5 (Conv2D)	(None, 2, 8, 128)	262272
leaky_re_lu_5 (LeakyReLU)	(None, 2, 8, 128)	0
dropout_5 (Dropout)	(None, 2, 8, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1)	2049
Total params: 1,183,425		
Trainable params: 1,183,425		
Non-trainable params: 0		

## （2）生成器的架构：

它将隐空间向量解码为具有训练集特征的合成图像。

先创建一个坐标点输入层和 Dense 层，然后依次传递给 6 个 Conv2D Transpose 层（前 5 层激活函数 LeakyReLU、最后一层激活函数 tanh），输出结果是该坐标点对应的合成图像。模型摘要见下表：

表 2 生成器模型摘要

Tab. 2 model summary of generator

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 2)]	0
dense_1 (Dense)	(None, 2048)	6144
reshape (Reshape)	(None, 2, 8, 128)	0
conv2d_transpose (Conv2DTran	(None, 4, 16, 128)	262272
batch_normalization (BatchNo	(None, 4, 16, 128)	512
leaky_re_lu_6 (LeakyReLU)	(None, 4, 16, 128)	0
dropout_6 (Dropout)	(None, 4, 16, 128)	0
conv2d_transpose_1 (Conv2DTr	(None, 8, 32, 128)	262272
batch_normalization_1 (Batch	(None, 8, 32, 128)	512
leaky_re_lu_7 (LeakyReLU)	(None, 8, 32, 128)	0
dropout_7 (Dropout)	(None, 8, 32, 128)	0
conv2d_transpose_2 (Conv2DTr	(None, 16, 64, 128)	262272
batch_normalization_2 (Batch	(None, 16, 64, 128)	512
leaky_re_lu_8 (LeakyReLU)	(None, 16, 64, 128)	0
dropout_8 (Dropout)	(None, 16, 64, 128)	0
conv2d_transpose_3 (Conv2DTr	(None, 32, 128, 128)	262272
batch_normalization_3 (Batch	(None, 32, 128, 128)	512
leaky_re_lu_9 (LeakyReLU)	(None, 32, 128, 128)	0
dropout_9 (Dropout)	(None, 32, 128, 128)	0
conv2d_transpose_4 (Conv2DTr	(None, 64, 256, 64)	131136
batch_normalization_4 (Batch	(None, 64, 256, 64)	256
leaky_re_lu_10 (LeakyReLU)	(None, 64, 256, 64)	0
dropout_10 (Dropout)	(None, 64, 256, 64)	0
conv2d_transpose_5 (Conv2DTr	(None, 128, 512, 1)	1025
Total params: 1,189,697		
Trainable params: 1,188,545		
Non-trainable params: 1,152		

## （3）Keras 模型子类化构建 WGAN-GP

仅改写 Model 类的 `__init__()`、`compile()`、`metrics()`、`train_step()` 四个方法<sup>[3]</sup>，而其它方法继承。

`__init__()` 方法导入网络架构部件及参数，`compile()` 方法提供优化器、屏显损失求均值算法，`metrics()` 方法让屏显损失值每轮重置。

`train_step()` 方法：先设置梯度带，在梯度带中，调用模型 `call()` 方法求得其输出，接着计算损失；然后求出梯度，采用模型 `compile()` 设定的优化器进行权重更新；最后损失经过累计求均值，转换为屏显损失，供 `fit()` 方法屏显。

在类中增加 `gradient_penalty()` 方法计算梯度惩罚项（也可以用自定义层的方法导入类中）。

## 2.3 训练

一开始，评论者统计出真实图像的各种特征，从而逼迫生成器生成具有部分这些特征的虚假图像，这些虚假图像有时只是将特征集中的部分特征进行组合，此时生成的图像往往具有较强的创造性。

为了避免以假乱真，评论者进一步学习，掌握真实图像不同特征之间的搭配关系，于是生成器进一步完善，让虚假图像的构图与真实图像所差无几，此时生成的图像会显得中规中矩。

因此，适度的模型容量与训练次数，能够在图像质量与创造性之间取得平衡。

因为需要调节的超参数数量过多，受硬件条件制约（仅拥有 4G 显存的 NVIDIA GTX 1650 显卡），不可能充分优化，所以只能边测试参数边采样与筛选，生成桥型图像来自多个事先保存的生成器模型。

## 2.4 隐空间采样探索新桥型

按照设定的密集空间网格从多个生成器模型的隐空间中采样，每个维度取 50 个点，坐标范围为  $[-10, +10]$ ，每个隐空间采样数目为 2500 张图像（50 的 2 次方）。

人工基于工程结构思维筛选，得到了与训练集完全不同的 5 种技术可行的新桥型（图 8）。

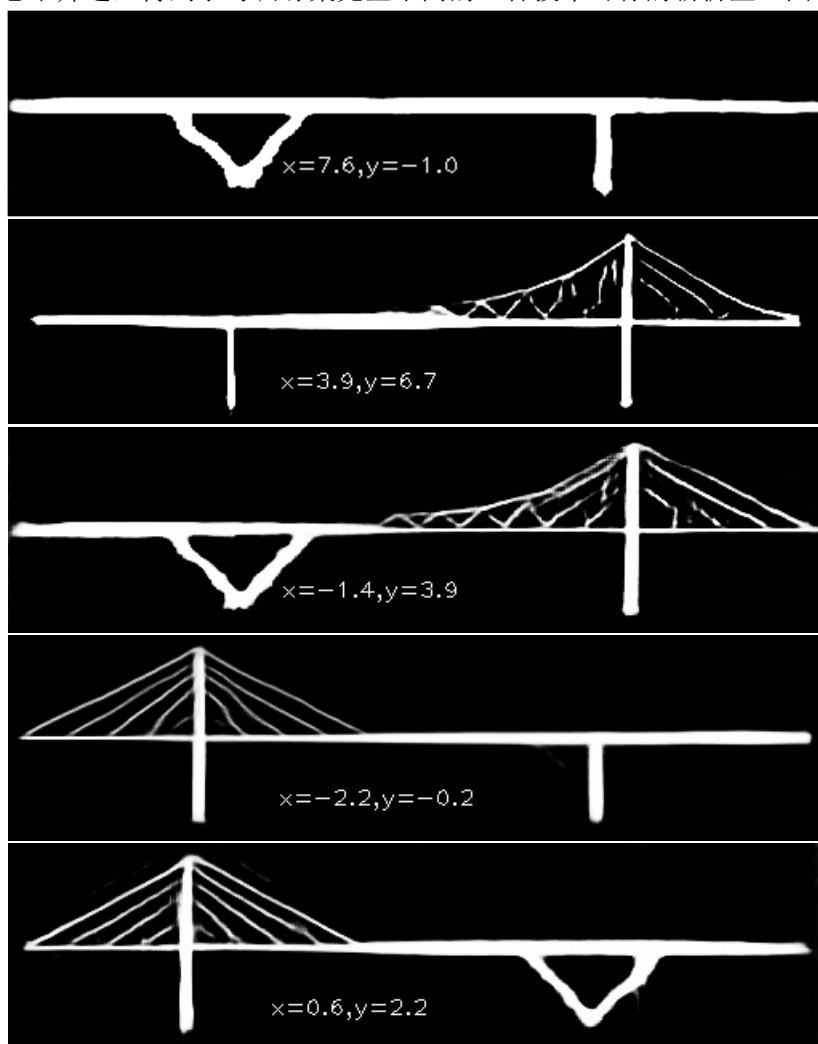


图 4 5 种技术可行的新桥型

Fig. 4 Five new bridge types with feasible technology

这里新桥型是指数据集中未曾出现过，而神经网络根据算法创作出的，这代表了模型具有创新能力。某些桥型如独塔斜拉桥，现实中很常见。

而 V 形墩刚构与独塔悬索桥、独塔斜拉桥的组合，据我所知未有工程师实践过。

## 2.5 结果分析

数据集的桥型全是对称结构，而生成对抗网络能够生成不对称的桥型，且并不是简单的叠合，而是将不同结构部件进行有机搭配，可以认为它具有类似人类的原创能力。（注：人类的桥梁设计原创能力，就是把梁、柱、索、拱等基本结构构件，按照力学原理组合，形成结实的结构物。）

## 2.6 待改进之处

（1）后期采样生成数万张桥型图像后，需要人工基于工程结构思维筛选。能否通过程序算法判断结构力学是否可行，从而代替人工筛选图像呢？

甚至在生成对抗网络的判别器中，添加结构力学程序算法，以结构是否成立为判断标准，从而让生成器的隐空间只生成结构成立的桥型。

（2）本次训练集的子桥型仅有 8 种，这严重制约了生成对抗网络的创新能力。如果把人类已建桥型收集齐全，甚至采用三维桥型，相信低维隐空间将会生成更多的新桥型。

遗憾的是，以上两点，本人作为一个普通桥梁设计师、民间业余人工智能爱好者，没有资源开展深入研究。

## 3 结语

（1）生成对抗网络比变分自编码器更有创造力，能够在人类原创桥型的基础上，将不同结构部件进行有机搭配，创造生成新桥型，一定程度上具有类似人类的原创能力，它能够打开想象空间，给予人类启发。

（2）目前阶段，生成式人工智能并非能够代替人类结构工程师的工作，而仅能作为虚拟助手。毕竟结构设计是非常高级的智力行为，即使人类，也需要经过多年严格的培养，才能成为合格的结构工程师。

人其实是大自然环境孕育的生物机器人。既然自然能够生成智能的人，那么只要找到其原理，人类也能模仿自然，创造出人工智能。人工智能的追求是：使机器“像人一样思考和行动”，在工程领域就是完全代替人类工程师。通过持续不断的研究，预期不久的将来，“强人工智能”将会实现，机器会像人一样思考和推理，拥有自主意识，通过学习知识、工程实践，也能够成为合格的工程师，进行工程设计、施工。

（3）本次探索的成果，可以复制到其它设计行业（如工业设计、建筑设计、景观设计等）。

## 参考文献

- [1] Hongjun Zhang. An attempt to generate new bridge types from latent space of variational autoencoder[J].arXiv preprint, 2023, arXiv:2311.03380.
- [2] David Foster. Generative Deep Learning[M]. 2nd Edition. America: O' Reilly, 2023.
- [3] 弗朗索瓦·肖莱. Python 深度学习[M]. 第二版. 北京:人民邮电出版社, 2023.
- [4] 大卫·福斯特. 生成式深度学习[M]. 北京:中国电力出版社, 2021.
- [5] Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein GAN [J].arXiv preprint, 2017, arXiv: 1701.07875.
- [6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky 等. Improved Training of Wasserstein GANs [J].arXiv preprint, 2017, arXiv: 1704.00028.