半监督角色动画风格迁移

郑瑞坤, 刘耿欣, 胡瑞珍*

(深圳大学可视计算研究中心 深圳 518061)
(ruizhen.hu@szu.edu.cn)

摘 要: 针对有监督动画风格迁移方法对未知内容泛化性差, 以及无监督方法生成动画风格不一致且不自然的问题, 提出一种半监督动画风格迁移方法. 所提方法使用 2 个编码器分别从内容动画和风格动画中提取内容和风格特征, 并使用解码器进行特征融合和动画生成; 为提升泛化性, 使用大规模无标签动画扩充内容动画数据集, 并用内容保留损失与末端执行器损失使生成结果与输入内容动画相似; 为保持风格一致性, 将生成结果重新输入风格编码器提取风格特征, 并计算与原风格和另一其他风格特征之间的三元组损失. 在 2 个公开数据集 BFA 和 CMU 上与 Motion Puzzle 等方法对比, 所提方法生成动画的 MID 值与用户调研结果均得到明显提升.

关键词: 动画风格迁移; 图卷积神经网络; 角色动画; 深度学习
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023-00385

Semi-Supervised Character Motion Style Transfer

Zheng Ruikun, Liu Gengxin, and Hu Ruizhen*

(*Visual Computing Research Center, Shenzhen University, Shenzhen* 518061)

Abstract: To address the poor generalization of supervised style transfer methods to unseen content, and the style inconsistency and unnaturalness of the motion generated by unsupervised methods, a semi-supervised motion style transfer method is proposed. The method employs two encoders to extract content and style features from content and style motions, respectively, and utilizes a decoder to fuse features and generate motions. To enhance generalization, a large-scale unlabeled motion dataset is used to augment the content motion dataset, and content preservation loss along with end-effector loss is applied to ensure the generated result resembles the input content motion. To maintain style consistency, the generated result is re-input into the style encoder to extract style features, and a triplet loss is calculated between the original style, the generated style, and another different style feature. Compared with methods such as Motion Puzzle on two public datasets, BFA and CMU, the proposed method significantly improves the MID values and user study results.

Key words: motion style transfer; graph convolutional neural network; character animation; deep learning

动画风格是角色动画的重要组成部分, 反映了角色的各个方面, 如角色的情感、个性或者特征. 然而, 因为人类观察者对于这种微妙的风格差异十分敏感, 并且对风格的感知也比较主观, 所以获取这种风格化的动作是一项具有挑战性的任务. 为了制作高质量的风格动画, 通常需要使用动作捕捉设备, 或者让动画师手工制作动画, 但由于动作捕捉设备往往十分昂贵, 并且需要聘请专业的

收稿日期: 2023-06-29; 修回日期: 2024-02-02. **基金项目**: 国家自然科学基金(61872250); 广东省自然科学基金(2021B1515020085).
郑瑞坤(2000—), 男, 硕士研究生, 主要研究方向为计算机动画; **刘耿欣**(1999—), 男, 硕士研究生, 主要研究方向为点云分析; **胡瑞珍**(1988—), 女, 博士, 副教授, 博士生导师, CCF 高级会员, 论文通信作者, 主要研究方向为计算机图形学.

演员来表演, 而让动画师手工制作动画又是一项费时费力的工作, 阻碍了风格动画在实际中的应用. 一个更可行的方式是在现有的风格动画数据上利用机器学习训练风格迁移模型, 将输入的一段示例的风格动画(如年迈)迁移到另一段内容动画(如行走)中, 可以快速地生成风格化的动画(年迈地行走).

已有的大多数方法[1-6]依赖于现有的少量带风格标签的动画数据, 这些数据包含多个风格类别, 每个类别的动画内容彼此相似. 基于传统机器学习的方法[1-2,6]需要一一配对的风格动画, 以此设计特殊的风格迁移算法. 基于深度学习的方法[3-5]借鉴图像到图像翻译[7-9]的思想, 提出无需配对数据的风格迁移模型, 但需要借助风格标签作为域监督信号[10]. 然而, 为每种风格都收集大量不同内容的数据是困难的, 使得这类有标签数据集覆盖的动画内容有限且较为简单(如行走和奔跑), 导致这类方法只能迁移与训练集相似的动画, 难以泛化到数据集分布较少或分布之外的动画, 对一些未见过的内容(如拳击和舞蹈)生成质量较差.

为了增强风格迁移模型对未知内容动画的泛化性, Motion Puzzle[11]借鉴任意图像风格迁移[12-16]的思想, 在大规模无标签的动画数据上训练模型, 无需风格标签作为域监督信号, 这类数据没有明确的风格类别划分, 并且所含的动画内容也复杂多样; 为了利用这类数据训练风格迁移模型, 他们将每段动画视为具有独立的风格, 把任意一段动画的风格迁移到另一段动画上. 大量的训练数据使得这种方法具有很好的泛化性, 但由于其没有从同种风格类别的动画中学习共同的风格特征, 而是将每段动画视为具有独立的风格, 因此模型无法很好地将动画的内容与风格分离开, 使得风格化结果往往趋向于将内容动画与风格动画简单地融合; 此外, 这种方法生成的动画会违反风格一致性, 即对于同种类别不同内容的风格动画, 得到的风格化结果并不一致, 生成的动画会受到风格动画本身所含内容的影响, 风格化质量较差.

为了解决现有动画风格迁移方法泛化性差、风格不一致等问题, 本文将半监督学习的思想引入角色动画风格迁移研究中, 提出一种使用带风格标签和无标签的数据的半监督角色动画风格迁移方法. 以 Aberman 等[3]的工作为基础, 将有标签的内容动画从源风格域迁移到目标风格域, 并利用风格标签设置内容一致性损失和风格三元组损失[3], 生成高质量的动画, 保证风格的一致性. 为了增强对未知的内容动画的泛化能力, 使用大规模无标签动画数据参与训练, 由于缺乏风格标签, 因此已有的损失函数并不适用于这类数据. 针对这类无标签动画, 本文设计内容保留损失和末端执行器损失保留动画内容, 并设计了风格迁移损失将输出动画的风格迁移到目标风格. 在训练阶段, 根据输入的内容动画是否带有标签, 选择相应的损失函数. 实验结果表明, 与 Motion Puzzle[11]相比, 本文方法利用有标签的数据和监督网络, 可以学习到符合人类理解的风格特征.

本文还对 Aberman 等[3]的工作进行改进. 首先使用时空图卷积网络(spatial temporal graph convolutional networks, STGCN)[17]作为网络模型, 与传统的一维卷积网络相比, 该模型能够利用角色骨架的拓扑信息增强网络的表征能力; 其次使用风格网络模块 StyleNet[11]作为内容与风格的融合模块, 与传统的自适应实例归一化(adaptive instance normalization, AdaIN)[13]全局风格迁移模块相比, StyleNet 能够同时迁移局部的风格; 最后使用多层级的网络架构模型, 由粗到细地控制网络的风格化结果, 生成更精细的风格动画[14-16,18].

1 相关工作

1.1 基于传统机器学习的动画风格迁移

早期的工作使用传统机器学习, 通过手工设计特征的方式对动画风格进行建模. Hsu 等[6]利用线性时不变系统对 2 段内容相似、风格相异的动画进行建模, 可以对具有相似内容的动画进行风格迁移; Xia 等[1]对输入的内容动画当前帧使用 K 最近邻搜索, 在数据集上寻找内容相似但具有风格差异的其他帧, 利用这些帧建立混合自回归模型, 能实时地对输入动画进行风格迁移; Yumer 等[2]利用傅里叶变换提取风格特征, 将 2 段内容相似风格相异的动画在频域上的差异表示为风格, 并将此差异加到新的动画的频域上, 生成风格化的动画: 这些方法十分依赖能够逐帧对齐、配对好的风格动画, 即角色的同一个动作在不同风格下的动画, 但收集这种配对的风格动画十分困难, 限制了它们在实际中的应用; 此外, 这些方法往往泛化性较差, 无法处理数据集分布之外的动画.

1.2 基于深度学习的动画风格迁移

近年来, 深度学习方法在多模态文本到动画生成[19-20]、基于物理的动画[21-22]等多个领域得到广泛应用, 并取得了显著的成果. 在动画风格迁移领

域, 也借鉴了神经图像风格迁移[7-8,12-16]方法, 提高了生成动画的质量. Holden 等[23]提出一个用于动画编辑的通用深度学习框架, 对输入动画优化使其满足2个条件: 输入动画的潜码与内容动画潜码相似, 而输入动画潜码的Gram矩阵[12]与风格动画潜码的Gram矩阵相匹配, 以此使用该框架风格迁移. 由于通过优化的方式迁移风格十分耗时, 因此 Holden 等[24]将优化过程替换为前馈网络, 并使用相同的损失函数训练网络, 提高了风格迁移的效率.

上述方法只能处理单一的风格, 当需要迁移新的风格时, 需要重新优化或训练新的网络模型. 因此, Aberman 等[3]使用风格编码器和内容编码器, 将内容动画与风格动画编码为内容潜码与风格潜码, 通过重新组合这 2 个潜码生成风格化动画. 在此基础上, Park 等[4]使用额外的风格编码器, 将高斯噪声映射为风格潜码, 模型在推理时无需额外的风格动画作为示例输入. 为了能够在游戏等实时场景中应用风格迁移, Tao 等[5]使用循环神经网络作为网络模型, 能够实时、逐帧地对输入动画风格迁移. 上述的方法依赖于现有的少量带有风格标签的动画数据, 并且需要利用标签作为风格迁移的监督信号, 模型往往泛化性较差. Jang 等[11]提出使用大规模的无标签动画数据训练模型, 但由于缺乏对同风格动画的利用, 使得风格化结果较差, 并且导致风格不一致问题. 本文提出一种半监督的深度学习方法, 同时使用带风格标签和无标签的动画数据集, 能够对更多类型、更复杂的角色动画进行风格迁移, 并保持风格一致性.

1.3　半监督学习

作为计算机视觉领域中一种常用的学习方式, 与只用标签数据学习的模型相比, 半监督学习旨在通过同时利用有标签和无标签数据从无标签的数据中获得性能提升. 通常, 基于伪标签和自训练的方法[25-27]在无标签数据上使用模型预测的结果作为标签训练模型. 基于一致性正则化的方法[28-30]假设对一个无标签的数据进行扰动, 其预测结果不应该发生显著变化, 因此在无标签数据上使用一致性正则损失. 本文将半监督学习的思想引入角色动画风格迁移研究中, 同时利用有风格标签和无标签的动画数据训练模型, 提升模型在未知数据上的风格迁移能力. 与半监督学习方法不同, 本文并没有预测伪标签或使用一致性正则化方法, 而是针对数据是否具有标签来使用不同的损失函数训练模型.

2　本文方法

2.1　概　述

首先对动画数据的表示与动画数据集进行说明. 为了使输出的动画可以直接应用于游戏设计或者电影动画领域而无需额外的调整, 本文使用关节局部旋转而非关节位置来表示动画. 记长度为 T 的动画序列 $m=\left[v_{1i},v_{2i},\cdots,v_{Ti}\right]_{i=1}^{N_1}$, 其中, $v_{ti}\in\mathbb{R}^d$ 表示第 t 帧第 i 个顶点(关节)的特征; d 表示顶点的旋转自由度; $N_1=21$, 表示初始的顶点数量. 本文使用 6D 旋转[31]表示每个关节的局部旋转, 即 $d=6$, 与欧拉角和四元数可能存在的插值问题和不连续问题相比, 这种表达方式更容易让网络学习. 本文使用 2 类动画数据集进行训练: 带有风格标签的动画数据集 M_{s} 与无标签的动画数据集 M_{c}, 2 个数据集的并集表示为 $M=M_{\mathrm{s}}\bigcup M_{\mathrm{c}}$, 通过裁剪, 得到 M 中每段动画长度 $T=128$.

本文方法的流程图如图 1 所示, 包含内容编码器 F、风格编码器 E 和解码器 D 这 3 个网络模块. F 和 E 分别接受内容动画 $m_{\mathrm{c}}\in M$ 与风格动画 $m_{\mathrm{s}}^q\in M_{\mathrm{s}}$ 作为输入, 其中, q 表示风格类别, 分别经过图卷积运算和池化运算不断减少关节数与帧数, 最终得到一个内容潜码与多个层次的风格潜码. 内容动画经过实例归一化(instance normalization, IN)运算去除其潜在的风格; 然后, 内容潜码与风格潜码共同输入到 D, 在每一层级利用 StyleNet 模块融合相应层级的内容潜码与风格潜码, 输出得到风格化的动画 $\tilde{m}_{\mathrm{c}}^q$. 为了表示方便, 图 1 中只绘制了 2 个层级的架构, 实际的网络架构具有多个层级.

在训练过程中, 对来自有标签动画数据集的内容动画 $m_{\mathrm{c}}\in M_{\mathrm{s}}$, 由于 m_{c} 的风格标签已知, 因此可以利用内容一致性损失 L_{con} 和风格三元组损失 L_{tri} [3]训练网络. 其中, L_{con} 用其内容潜码与同标签的另一风格潜码作为解码器的输入, 重构动画; L_{tri} 则用来更好地聚合同标签的风格潜码. 由于只利用上述方式训练的网络对未知的内容动画泛化性差, 因此本文利用来自无标签动画数据集的内容动画 $m_{\mathrm{c}}\in M_{\mathrm{c}}$ 提高网络泛化性. 由于 m_{c} 的风格标签未知, 上述损失函数并不适用, 因此本文提出内容保留损失 L_{c} 与末端执行器损失 L_{ee}, 使得输出动画尽可能保留动画的内容; 同时, 由于输出动画

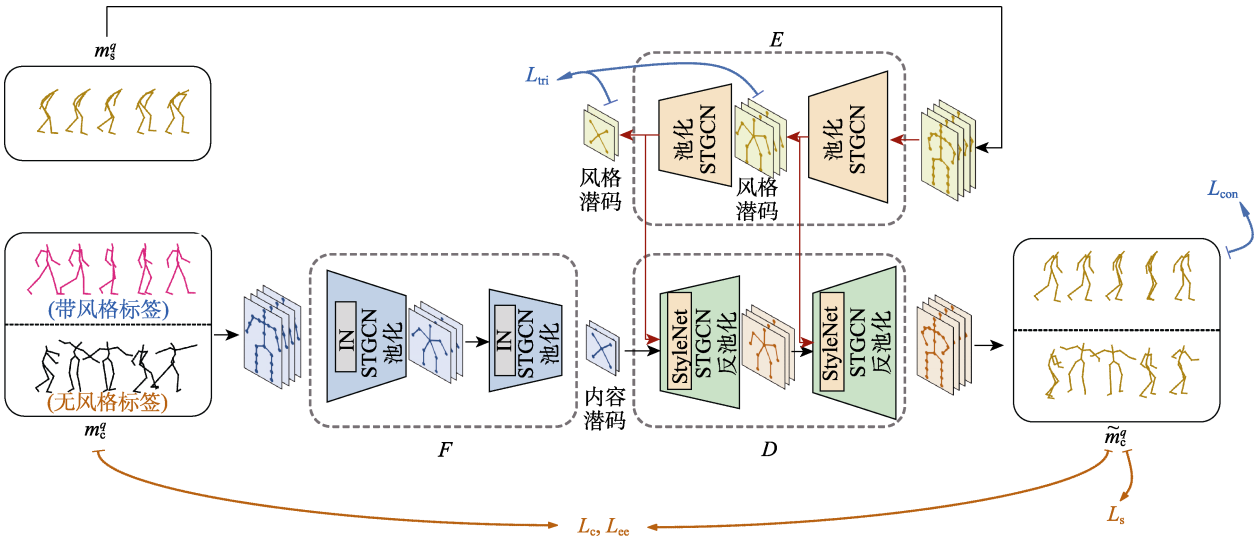


图 1 本文方法流程

$\tilde{m}_c^q$ 的风格标签 q 是已知的，因此根据此特点设计风格迁移损失 L_s，将 $\tilde{m}_c^q$ 的风格潜码向 m_s^q 的风格潜码拉近，并远离不同类别的风格潜码，使其迁移到目标风格.

2.2 图卷积网络

2.2.1 STGCN

角色动画骨架中，各相邻顶点在时间上和空间上都有着紧密的联系. 本文使用 STGCN[17]从动画骨架的时空图中学习更好的特征. 下面介绍 STGCN 在本文中的实现方式，更多关于 STGCN 的细节请参阅文献[17].

STGCN 包含空间图卷积和时域一维卷积 2 个步骤的运算. 空间图卷积定义为

$$\boldsymbol{f}_{\text{out}}(v_{ti}) = \sum_{v_{tj}\in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} \boldsymbol{f}_{\text{in}}(v_{tj}) \cdot w\left(l_{ti}(v_{tj})\right) \tag{1}$$

其中，$\boldsymbol{f}_{\text{in}}(v)$ 和 $\boldsymbol{f}_{\text{out}}(v)$ 分别表示顶点 v 输入和输出的特征向量；$B(v_{ti})$ 表示与顶点 v_{ti} 相邻边跳数为 D 的所有顶点 v_{tj}，即 $B(v_{ti}) = \{v_{ti} \mid d(v_{ti}, v_{tj}) \leqslant D\}$，$d(v_{ti}, v_{tj})$ 表示从顶点 v_{ti} 到顶点 v_{tj} 的最短边跳数，本文中只考虑一阶邻域的最短边跳数，即 $D=1$；标签 $l_{ti}(v_{tj}) = d(v_{ti}, v_{tj})$ 表示顶点 v_{tj} 的数值标签，值为顶点 v_{ti} 到顶点 v_{tj} 的最短边跳数，$l_{ti}(v_{tj})$ 决定了顶点 v_{tj} 所对应的可学习权重 $w\left(l_{ti}(v_{tj})\right)$，在顶点 v_{ti} 的 D 邻域内，具有相同数值标签的顶点共享相同的权重；归一化项 $Z_{ti}(v_{tj}) = \left|\{v_{tk} \mid l_{ti}(v_{tk}) = l_{ti}(v_{tj})\}\right|$ 表示具有同一数值标签的每个子类的顶点数量.

应用式(1)的空间图卷积之后，时域一维卷积被应用到输出的顶点特征向量 $\boldsymbol{f}_{\text{out}}(v_{ti})$ 上，完成 STGCN 的全部过程.

2.2.2 图池化与反池化

在时间维度上，角色动画的骨架图结构是均匀的，因此容易使用标准的池化操作在时间轴上减少输出的维度. 但由于在空间维度上图结构是非均匀的，因此传统的池化操作并不适用. 本文使用 Jang 等[11]提出的池化方法，通过平均骨架相邻顶点的特征向量逐步缩减顶点数量，最终得到更简化的拓扑结构，如图 2 所示.

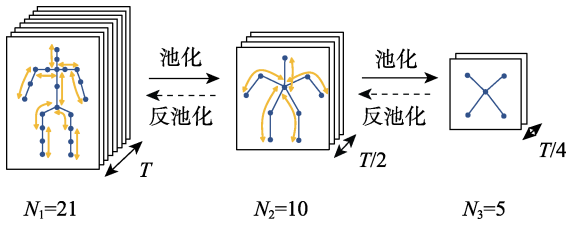


图 2 图池化与反池化

反池化操作与池化操作相反. 在时间维度上，通过最近邻插值的方式上采样，逐步增加动画的维度. 在空间维度上，通过将池化后的顶点映射回先前的骨骼结构，以进行空间维度上图的反池化，如图 2 所示.

2.3 网络架构

2.3.1 风格编码器

风格编码器 E 由线性层和多个 STGCN 模块与图池化层堆叠，组成一个多层级的网络 E，其结构如图 2 所示. E 接受风格动画 $m_s^q \in \mathbb{R}^{N_1 \times d \times T}$ 为输入，依次经过多个图卷积与图池化层的运算，得到

多个层级的输出. 本文中只使用 3 个层级的输出 $\left\{z_{si}^q\right\}_{i=1}^3=E\left(m_s^q\right)$. 其中, $z_{s1}^q\in\mathbb{R}^{N_1\times C\times T}$, $z_{s2}^q\in\mathbb{R}^{N_2\times 2C\times\frac{T}{2}}$, $z_{s3}^q\in\mathbb{R}^{N_3\times 4C\times\frac{T}{4}}$ 分别为第 1 层级、第 2 层级和第 3 层级的风格潜码输出; C 表示顶点特征向量的维度; T 表示动画序列的长度; N_1 表示初始顶点数量; N_2 和 N_3 分别表示 1 次图池化和 2 次图池化后的顶点数量. 顶点的映射方式如图 2 所示.

2.3.2　内容编码器

内容编码器 F 与风格编码器 E 相似, 唯一的区别在于 F 使用了 IN, 通过归一化操作移除内容动画可能携带的风格特征, 得到与风格无关的内容潜码. 输入 $m_c\in\mathbb{R}^{N_1\times d\times T}$ 和 F, 计算得到相应的内容潜码 $z_c=F\left(m_c\right)$. 其中, $z_c\in\mathbb{R}^{N_3\times 4C\times\frac{T}{4}}$ 表示多个层级输出的最后一层级.

2.3.3　解码器

解码器 D 接受内容潜码 z_c 与风格潜码 $\left\{z_{si}^q\right\}_{i=1}^3$ 作为输入, 将 z_c 不断上采样与反池化后, 每一层级输出与对应层级的风格潜码融合, 得到最终风格化的输出 $\tilde{m}_c^q$. 初始时有 $z_{c3}^q=z_c$, 每一层级的运算公式为 $z_{ci-1}^q=D_i\left(z_{ci}^q,z_{si}^q\right)$. 其中, D_i 表示 D 所对应层级 i 的网络模块; 最后一层级的输出为 $z_{c0}^q\in\mathbb{R}^{N_1\times C\times T}$, 其的每个顶点经过线性层的计算, 得到最终输出的动画 $\tilde{m}_c^q\in\mathbb{R}^{N_1\times d\times T}$. 因此, 解码过程表示为 $\tilde{m}_c^q=D\left(F\left(m_c\right),E\left(m_s^q\right)\right)=D\left(z_c,\left\{z_{si}^q\right\}_{i=1}^3\right)$.

上述计算中, z_{ci}^q 与 z_{si}^q 通过 StyleNet[11]进行融合, 得到风格化的潜码 $\tilde{z}_{ci}^q$; 再经过图卷积与图反池化计算, 得到下一层级的潜码 z_{ci-1}^q. 下面阐述 StyleNet 模块的实现细节.

图 3 所示为 StyleNet 模块的架构, 包含 2 个风格迁移模块: 全局风格迁移模块 AdaIN, 局部风格迁移模块 ATN(attention network). AdaIN 将输入的 z_{si}^q 注入到 z_{ci}^q 中, 对 z_{si}^q 归一化, 并施加一个时间不变的仿射变换

$$\mathrm{AdaIN}\left(z_{ci}^q,z_{si}^q\right)=\gamma_i\left(z_{si}^q\right)\left(\frac{z_{ci}^q-\mu\left(z_{ci}^q\right)}{\sigma\left(z_{ci}^q\right)}\right)+\beta_i\left(z_{si}^q\right).$$

其中, μ 和 σ 分别表示 z_{ci}^q 逐通道的均值和方差, γ_i 和 β_i 分别表示从 z_{si}^q 中学习得到的仿射变换的缩放与偏移. 最后, 将变换后的特征输入到 STGCN1 层中, 得到全局风格迁移的特征

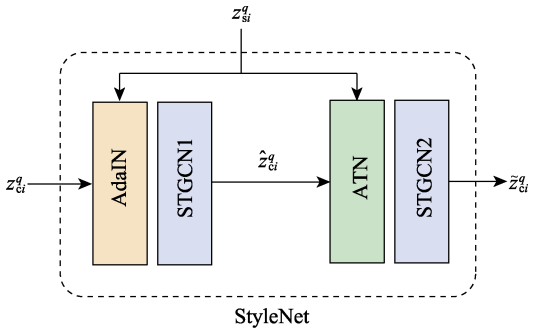


图 3　StyleNet 网络模块架构

$$\hat{z}_{ci}^q=\mathrm{STGCN1}\left(\mathrm{AdaIN}\left(z_{ci}^q,z_{si}^q\right)\right).$$

得到 AdaIN 的特征之后, 使用基于自注意力的局部风格迁移模块 ATN. 图 4 所示为 ATN 模块的结构, 更多关于 ATN 的实现细节详见文献[11,16]. ATN 以 z_{si}^q 和 $\hat{z}_{ci}^q$ 作为输入, 将其经过 AdaIN 后的 $\overline{z}_{si}^q$ 与 $\overline{z}_{ci}^q$ 分别输入卷积神经网络, 通过变形调整向量尺寸, 得到矩阵 $\boldsymbol{g}\left(\overline{z}_{si}^q\right)\in\mathbb{R}^{C_i\times|V_i|_s}$ 和矩阵 $\boldsymbol{h}\left(\overline{z}_{ci}^q\right)\in\mathbb{R}^{|V_i|_c\times C_i}$ (简记为矩阵 $\boldsymbol{g}$ 和矩阵 $\boldsymbol{h}$). 其中, $|V_i|_s$ 和 $|V_i|_c$ 表示相应的顶点数量; C_i 表示特征维度. 将矩阵 $\boldsymbol{g}$ 和 $\boldsymbol{h}$ 相乘, 使用 Softmax 计算注意力图

$$\boldsymbol{A}_{\beta,\alpha}=\exp\left(\boldsymbol{g}_\alpha\cdot\boldsymbol{h}_\beta\right)\Big/\sum_{\alpha=1}^{|V_i|_s}\exp\left(\boldsymbol{g}_\alpha\cdot\boldsymbol{h}_\beta\right).$$

其中, $\boldsymbol{A}_{\beta,\alpha}$ 度量了风格特征 $\boldsymbol{g}$ 的第 α 个顶点与输出特征 $\boldsymbol{h}$ 的第 β 个顶点之间的相似性. z_{si}^q 也经过与上述类似计算得到新的特征矩阵 $\boldsymbol{f}\left(z_{si}^q\right)\in\mathbb{R}^{|V_i|_s\times C_i}$ (简记为 $\boldsymbol{f}$). 将 $\boldsymbol{A}_{\beta,\alpha}$ 与 $\boldsymbol{f}$ 相乘, 利用注意力图计算逐顶点的风格特征(局部风格), 再将结果与全局风格迁移特征 $\hat{z}_{ci}^q$ 相加, 即 $\mathrm{ATN}\left(\hat{z}_{ci}^q,z_{si}^q\right)=\boldsymbol{A}^{\mathrm{T}}\otimes\boldsymbol{f}\oplus\hat{z}_{ci}^q$. 其中, $\otimes$ 表示矩阵乘法, $\oplus$ 表示逐元素相加. 将输出的特征输入到时空图卷积 STGCN2 层中, 得到最终风格化的特征 $\tilde{z}_{ci}^q=\mathrm{STGCN2}\left(\mathrm{ATN}\left(\hat{z}_{ci}^q,z_{si}^q\right)\right)$.

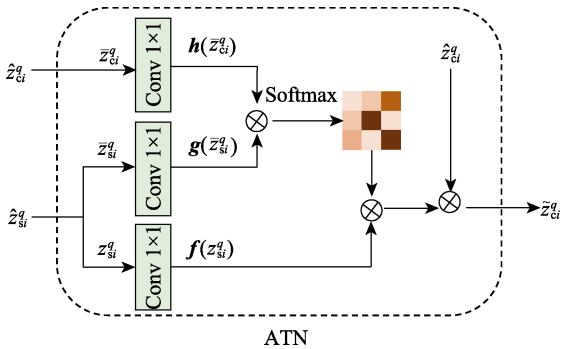


图 4　ATN 模块结构

2.4 训练与损失函数

2.4.1 有标签动画损失函数

对于来自有标签动画数据集 M_s 的内容动画，使用 Aberman 等[3]提出的内容一致性损失 L_{con} 与风格三元组损失 L_{tri} 作为损失函数训练网络. 取内容动画 $m_c^q \in M_s$ (即 $m_c = m_c^q$)，q 表示风格标签，并取另一段具有相同风格标签的风格动画 $m_s^q \in M_s$，则定义

$$L_{con} = \left\| m_c^q - D\left(F\left(m_c^q\right), E\left(m_s^q\right)\right)\right\| \tag{2}$$

其中，$\|\cdot\|$ 表示向量 1-范数. 特别地，当 $m_c^q = m_s^q$ 时，式(2)退化为标准重构误差.

L_{tri} 旨在将隐空间的风格潜码更好地聚合. 通过设置 L_{tri}，相同风格标签的风格潜码 z_{si}^q 被拉近、不同风格标签的风格潜码 z_{si}^q 被推远. 从数据集 M_s 中取另外 2 段风格动画 m_{s1}^q 和 m_{s2}^p，其中，q 和 p 为不同的风格标签，则定义

$$L_{tri} = \sum_{i=1}^{3} \max\left(\left\|E_i\left(m_s^q\right) - E_i\left(m_{s1}^q\right)\right\| - \left\|E_i\left(m_s^q\right) - E_i\left(m_{s2}^p\right)\right\| + \delta_1, 0\right) \tag{3}$$

其中，$E_i(\cdot)$ 表示第 i 层的风格潜码，本文方法使用 3 层层级的网络；δ_1 表示三元组损失的间隔.

2.4.2 无标签动画损失函数

本质上，Aberman 等[3]的动画风格迁移模型是依靠对动画的重构来实现的，当输入的内容动画与数据集 M_s 的动画相似时，模型能够有一定程度的泛化性，并在不同的风格域之间迁移风格. 然而，当输入的内容动画在数据集 M_s 中出现较少或者完全不相似时，模型会过拟合到与训练集相似的内容上. 本文使用大规模无标签的动画数据集 M_c 增强对未知内容动画的泛化性，对其中的动画进行风格迁移，使用内容保留损失 L_c 与末端执行器损失 L_{ee} 保留动画内容；同时，使用风格迁移损失 L_s 保留所迁移的风格. 下面阐述这 3 个损失函数的定义与作用.

L_c 旨在令输出动画保留与输入的内容动画一定的相似性. 取内容动画 $m_c \in M_c$，风格动画 $m_s^q \in M_s$，其中，q 为风格标签时，风格迁移后输出动画为

$$\tilde{m}_c^q = D\left(F\left(m_c\right), E\left(m_s^q\right)\right) \tag{4}$$

则定义

$$L_c = \left\| m_c - \tilde{m}_c^q \right\| \tag{5}$$

式(5)旨在让 $\tilde{m}_c^q$ 与 m_c 具有一定的相似性，而非让两者完全相等. 所以，在训练过程中，式(5)中定义的损失函数应当乘以一个较小的超参数 λ_c，防止 $\tilde{m}_c^q$ 与 m_c 完全相同. 但在实际训练的过程中，λ_c 的值难以确定. 如果 λ_c 的数值过小，则 $\tilde{m}_c^q$ 无法很好地保留内容；而如果增大 λ_c 的数值，$\tilde{m}_c^q$ 又会失去所迁移的风格. 为了解决这个问题，本文设置了较小的 λ_c 值，然后使用 L_{ee} 进一步保留内容，这是基于对现实的观察，即在绝大多数情况下，角色动画的内容都可以通过角色的末端执行器推理得知. 例如，对于跑步的角色动画，其手部与脚部应呈现周期性的摆动. 基于上述观察，本文定义

$$L_{ee} = \left\| \mathrm{FK}_{ee}\left(m_c^q\right) - \mathrm{FK}_{ee}\left(\tilde{m}_c^q\right)\right\|.$$

其中，$\mathrm{FK}_{ee}(m)$ 表示对动画 m 使用可微分的前向动力学(forward kinematics, FK)[32]，将关节的局部旋转计算得到相应的关节位置坐标，再选取末端执行器对应的关节，本文中选取的关节为左脚趾、右脚趾、左手腕和右手腕.

L_s 旨在让 $\tilde{m}_c^q$ 能够保留所迁移的风格. 由于 $\tilde{m}_c^q$ 所带的风格标签是已知的，因此可以利用该风格标签，使用与式(3)相似的 L_{tri}. 首先取 2 个风格标签为 q 和 p 的风格动画 m_s^q 和 m_{s2}^p，按照与式(4)相同的方式迁移到 m_c 上，分别得到风格化的 $\tilde{m}_c^q$ 和 $\tilde{m}_c^p$；然后利用 L_{tri} 使 $\tilde{m}_c^q$ 的风格潜码与 m_s^q 的风格潜码尽可能相近，$\tilde{m}_c^p$ 的风格潜码与 m_s^q 的风格潜码尽可能远离，则定义

$$L_s = \sum_{i=1}^{3} \max\left(\left\|E_i\left(m_s^q\right) - E_i\left(\tilde{m}_c^q\right)\right\| - \left\|E_i\left(m_s^q\right) - E_i\left(\tilde{m}_c^p\right)\right\| + \delta_2, 0\right).$$

其中，δ_2 表示每一层级 L_{tri} 的间隔.

2.4.3 总损失函数

将上述训练过程的损失函数汇总，定义总损失函数为 $L = \lambda_{con}L_{con} + \lambda_{tri}L_{tri} + \lambda_c L_c + \lambda_{ee}L_{ee} + \lambda_s L_s$. 其中，根据 Aberman 等[3]的超参数设置，设置 $\lambda_{con} = 1.0$，$\lambda_{tri} = 1.0$ 和 $\delta_1 = \delta_2 = 5.0$；根据第2.4.2节的任务设置，设置 $\lambda_c = 0.01$，$\lambda_{ee} = 0.2$ 和 $\lambda_s = 0.1$.

3 实验结果与分析

3.1 实验设计

3.1.1 数据集与数据处理

本文使用BFA动作捕捉数据集[3]作为有标签的

动画数据集, 从中选取 8 个风格类别进行训练与测试. BFA 数据集中, 每个类别包含 2 段长动画序列, 内容为行走、跳跃、过渡等较为单一的内容; 每个类别时长大约为 6 min. 使用 CMU 动作捕捉数据集①作为无标签动画数据集, 它由多段短的动画序列组成, 内容包含行走、拳击、舞蹈、瑜伽、太极和日常活动等更为复杂多样的动作类型, 动画时长共计约为 9 h.

数据预处理时, 动画的每帧都使用与 Holden 等[23]相似的做法预处理, 通过旋转根关节使之面朝 z 轴的正方向进行逐帧对齐; 之后, 选取 CMU 数据集中 90%的动画进行裁剪处理, 作为训练样本, 其余的每段动画不做裁剪处理, 作为测试样本. 裁剪处理过程中, 每段动画都会下采样为 60 帧/s; 然后使用步长为 32 帧的滑动窗口进行裁剪, 得到每一段动画片段的长度 $T=128$ 帧(大约为 2 s), 而少于 128 帧的动画片段则直接舍弃. 对 BFA 数据集, 采用与 CMU 数据集相同的数据预处理与划分方法. 由于 BFA 数据集中连续完整的动画数量少(每个风格类别只有 2 段连续的动画序列), 因此在划分测试集时, 直接从经过裁剪预处理后的动画片段中随机选取 10%作为测试样本, 每个样本为 128 帧的动画片段. 经过数据预处理与数据集划分后, 对训练数据集进行数据增强, 利用人体结构的对称性, 将训练数据每段动画片段的左右身体部分镜像翻转, 得到 2 倍于原始训练数据的动画片段. 数据增强后, BFA 训练集共有约 1 万个片段的训练样本, CMU 训练集共有约 11 万个片段的训练样本.

3.1.2　实验环境与训练过程

本文实验中, 使用 NVIDIA GeForce RTX 2080 GPU, Ubuntu 16.04 操作系统, Python 3.8 编程语言, PyTorch 1.11 深度学习框架. 训练学习率为 0.000 1, 权重衰减为 0.000 1, 使用 RAdam[33]作为优化器, 并设置梯度裁剪系数为 1.0, 选取批次(batch)大小为 24, 在每次迭代中对网络所有参数使用指数移动平均[34]. 经过 15 个时期(epoch)的迭代训练后网络达到收敛, 训练时间约为 10 h. 本文网络的训练时间与 Motion Puzzle[11](约 9 h)较为接近, 比 Aberman 等[3](约 6 h)的用时长, 这是因为本文方法与 Motion Puzzle 都使用图卷积网络, 比 Aberman 等[3]使用的常规一维卷积网络计算更复杂, 所以更耗时. 但是, 图卷积网络可以从动画数据中提取更丰富的特征, 因此生成的动画质量也更高.

3.2　对比实验

3.2.1　动画生成质量

图像合成中经常采用 FID(Fréchet inception distance)指标[35]评估图像生成质量. FID 使用预训练好的 Inception v3 网络[36]提取特征, 通过计算生成图像与真实图片特征分布的差异衡量图像生成质量. 在动画合成中借鉴了类似的思路, 使用 FMD (Fréchet motion distance)[4-5,11]评估动画生成质量. 本文使用 FMD 指标, 并以 Yan 等[17]的网络模型训练一个动作分类器, 使用最后一层池化层的输出计算动画特征.

对 Aberman 等[3], Motion Puzzle[11]和本文方法的动画生成质量进行实验, 从 CMU 测试集与 BFA 测试集中各随机选取一段动画, 分别作为内容动画与风格动画输入并生成动画, 3 种方法生成的风格化动画 FMD 结果如表 1 所示. 为了更准确和公平地评估, 将对比方法进行调整: Aberman 等[3]的方法中设计了投影模块, 能从 3D 或 2D 动画中提取动画风格; 本文方法中除去了网络的 2D 处理模块, 并在 BFA 训练集上重新训练网络; Motion Puzzle 方法需要对角色的不同身体部分分别风格迁移, 而本文方法只关注全体风格迁移, 因此将 Motion Puzzle 方法对不同身体部分风格迁移的设计去除, 然后混合 CMU 训练集与 BFA 训练集(去除风格标签), 在新的数据集上重新训练模型.

表 1　3 种风格迁移方法生成质量对比

方法	FMD
Aberman 等[3]	25.37
Motion Puzzle[11]	11.12
本文	**8.44**

注. 加粗表示在 3 种方法中具有最低的 FMD 值.

从表 1 可以看出, Aberman 等[3]方法的 FMD 值远大于 Motion Puzzle 和本文方法, 原因是其模型只能利用有标签的动画数据集, 而对于训练集分布之外的动画, 生成的结果会趋向于平均化, 因此泛化性较差, 而 Motion Puzzle 与本文方法使用大规模的数据集, 具有更好的泛化性; Motion Puzzle 与本文都使用图卷积网络、StyleNet 融合模块和多级网络架构, 能提高输出动画的质量; 虽然 Motion Puzzle 与本文方法的 FMD 值比较接近, 但

① http://mocap.cs.cmu.edu

前者的 FMD 值稍微更大, 可能在于它只将输入的内容动画与风格动画简单地融合, 而这种融合会导致生成一些不自然的动作姿态, 使得特征分布偏离真实动画的特征分布.

3.2.2 内容保留与风格迁移

通过计算生成动画与真实动画特征分布之间的差异, FMD 指标可用于衡量动画的生成质量, 但 FMD 指标不足以评估生成动画是否保留了语义信息. 例如, 将内容动画“舞蹈”与风格动画“沮丧地行走”作为输入, 生成动画“沮丧地舞蹈”, 需要额外的指标来评估其包含的内容语义“舞蹈”与风格语义“沮丧”. 已有方法[4,11]只在有标签的动画数据集上评估指标, 利用标签训练分类器, 并以分类器的分类准确率来评估保留程度. 但是, 本文需要对分布之外的动画风格化结果进行评估, 而这类数据内容复杂, 缺乏风格标签, 训练类似的分类器并不可行; 并且, 这些无标签的动画所带的风格可能是多种多样的, 难以确切地分为各个风格类别. 因此, 本文使用用户学习的方式评估生成动画语义信息的保留程度. 首先通过手工的方式, 对少量的 BFA 与 CMU 测试数据用文本标记它们的风格与内容; 然后分别使用 Aberman 等[3], Motion Puzzle[11] 和本文方法生成动画.

本文邀请 14 名受试者参与实验评估. 首先在用户学习评估系统中随机地为每位受试者展示一段动画, 动画来源于 3 种方法的输出, 或来源于数据集(BFA 测试集和 CMU 测试集); 然后受试者观看播放的动画, 并从内容描述与风格描述中各选择最符合描述的一项. 选择分为 5 个级别, 对应不同的得分: 完全不符合(1 分)、比较不符合(2 分)、不确定(3 分)、比较符合(4 分)和完全符合(5 分). 用户学习评估系统的操作界面如图 5 所示.

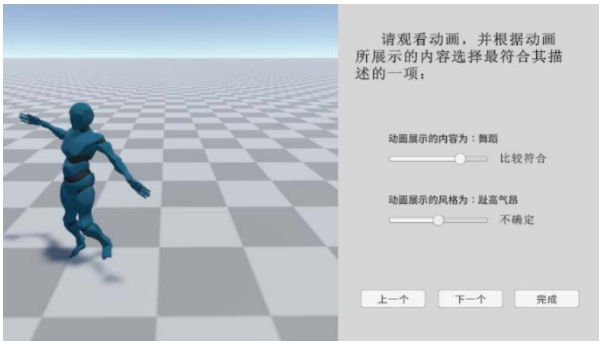


图 5 用户学习评估系统操作界面

本次实验总共收集到 732 份用户学习结果, 将分数统计求和平均转化为百分制表示, 结果如表 2 所示. 实验中, 由于 BFA 测试集的动画内容相对简单, CMU 测试集中的大部分动画风格并不明显, 因此不纳入内容保留得分和风格迁移得分. 可以看出, 在内容保留上, 本文方法明显优于其他 2 种方法, 这归功于本文对大规模动画数据集的利用, 使得模型泛化性强; Motion Puzzle 在内容保留和风格迁移上得分都较低, 可能原因是它只能将风格动画的某些数据特征迁移到内容动画上, 然而这些特征并非人类所能理解的风格特征, 并且还会对原有的动画内容造成影响.

表 2 3 种方法内容保留与风格迁移百分比得分对比 %

评估内容	百分比得分			
	数据集	Aberman 等[3]	Motion Puzzle[11]	本文
内容保留	95.5	68.3	68.9	**86.6**
风格迁移	91.5	66.8	60.7	**70.9**

注. 加粗表示在 3 种方法中具有最高的百分比得分.

3.3 结果展示与分析

3.3.1 风格一致性

合理的风格迁移模型对风格迁移后的动画应该保持风格一致性. 例如, 对于“年迈地跳跃”和“年迈地行走” 2 段风格动画, 模型应当能提取出“年迈”的风格特征, 而不受其内容“跳跃”和“行走”的影响. 对 Motion Puzzle 和本文方法进行风格一致性实验, 验证不同方法的风格迁移结果, 结果如表 3 所示. 可以看出, Motion Puzzle 方法风格迁移的结果并没有很好地保持风格一致性, 对于同一风格不同内容的风格动画, 生成的动画有一定的差异; 本文方法在这 2 种情况下都能生成相似的动画, 保持风格的一致性.

3.3.2 内容保留与风格迁移对比

从定性的角度对 Aberman 等[3], Motion Puzzle[11]和本文方法的内容保留与风格迁移结果进行对比, 3 种方法输出动画的结果如表 4 所示.

可以看出, 3 种方法都能够从示例的风格动画上迁移相应的风格特征, 如风格“年迈”中的弯腰和风格“沮丧”中头部的低垂; 在内容保留上, 本文方法明显优于其他 2 种方法; Aberman 等[3]的方法由于只利用了少量的有标签动画数据集进行训练, 因此在处理更复杂的、分布之外的动画时, 生成动画每帧的姿势趋向于训练集的姿态, 无法很好地保留原有的内容; Motion Puzzle 中, 由于风格动画的内容对输出的结果造成影响, 使得内容的保留较差.

表 3　2 种方法风格一致性结果对比

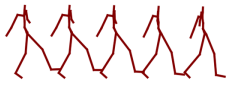
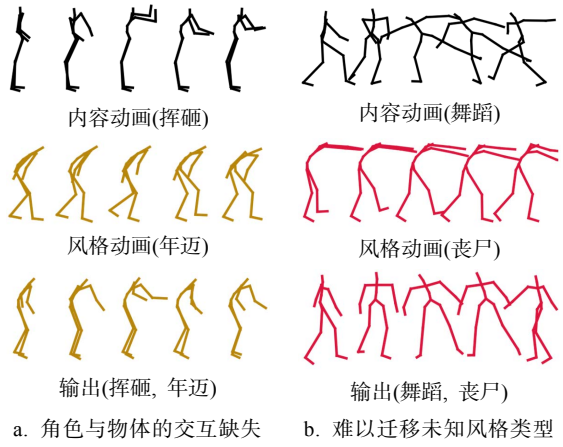
内容动画	风格动画	Motion Puzzle[11]	本文
 (拳击)	 (跳跃，年迈)	 (拳击，年迈)	 (拳击，年迈)
	 (行走，年迈)	 (拳击，年迈)	 (拳击，年迈)
	 (跳跃，趾高气昂)	 (拳击，趾高气昂)	 (拳击，趾高气昂)
	 (跳跃，趾高气昂)	 (拳击，趾高气昂)	 (拳击，趾高气昂)

表 4　3 种方法生成质量结果对比

内容动画	风格动画	Aberman 等[3]	Motion Puzzle[11]	本文
 (舞蹈)	 (年迈)	 (舞蹈，年迈)	 (舞蹈，年迈)	 (舞蹈，年迈)
	 (趾高气昂)	 (舞蹈，趾高气昂)	 (舞蹈，趾高气昂)	 (舞蹈，趾高气昂)
	 (沮丧)	 (舞蹈，沮丧)	 (舞蹈，沮丧)	 (舞蹈，沮丧)
 (站立游泳)	 (年迈)	 (站立游泳，年迈)	 (站立游泳，年迈)	 (站立游泳，年迈)
	 (趾高气昂)	 (站立游泳，趾高气昂)	 (站立游泳，趾高气昂)	 (站立游泳，趾高气昂)
	 (沮丧)	 (站立游泳，沮丧)	 (站立游泳，沮丧)	 (站立游泳，沮丧)
 (运球)	 (年迈)	 (运球，年迈)	 (运球，年迈)	 (运球，年迈)
	 (趾高气昂)	 (运球，趾高气昂)	 (运球，趾高气昂)	 (运球，趾高气昂)
	 (沮丧)	 (运球，沮丧)	 (运球，沮丧)	 (运球，沮丧)

3.4 局限性

尽管本文方法取得了一定的成功，能够将任意动画内容迁移到特定风格上，但仍存在一些局限性. (1) 输出动画无法保持与物体的接触信息. 这是由于在实现风格化过程中没有考虑角色动画与物体之间的交互关系，角色动画原有的交互在风格化后会丢失. 如图 6a 中对内容动画“挥砸”的风格化结果所示，原动画中角色双手靠拢举起物体向下挥砸，然而这种交互关系却在风格化之后缺失. (2) 本文方法难以泛化到未知的风格类别上，如图 6b 中将风格“丧尸”迁移到内容“舞蹈”上的风格化结果所示，输出的动画并不能很好地体现“丧尸”的风格特点. 这是由于在训练过程中使用的风格数据集的风格类别数量有限，无法充分地覆盖所有可能出现的风格类别，模型会过拟合到特定的几个风格类别上. 实验结果表明，当尝试将本文方法应用于未知的风格类别时，可能会导致风格化效果不佳甚至失败.



a. 角色与物体的交互缺失　b. 难以迁移未知风格类型

图 6　本文方法的失败案例

针对以上局限性，未来应进一步探索解决方案. 例如，可以尝试加入额外的交互信息和损失函数，使输出的动画能够保持与物体的交互；或者使用逆向动力学修正角色的关节位置，保持与物体的接触. 其次，针对模型对未知风格泛化性差的现状，除了进一步增加风格数据集的风格种类，还可以借用任意风格迁移的思想，使得方法能够迁移任意风格动画，并生成自然的高质量动画.

4 结　语

针对单一地使用带风格标签或无标签动画数据而导致的泛化性差、风格不一致等问题，本文提出半监督角色动画风格迁移方法. 对带风格标签和无标签的数据分别设计了对应的动画损失函数，能够同时利用有标签和无标签的数据训练网络，提高了模型对未知动画数据的泛化性，又保持了风格一致性. 此外，通过使用图卷积网络、StyleNet 融合模块和多层级网络架构设计等方式，进一步增强网络的表征能力，提高动画生成质量. 通过多种定性和定量实验，证明了本文方法在角色动画风格迁移问题上具有更好的泛化性以及更高的生成质量；最后讨论了其不足之处，并针对现有的问题提出可行的解决方案及未来的工作方向.

参考文献(References):

[1] Xia S H, Wang C Y, Chai J X, *et al*. Realtime style transfer for unlabeled heterogeneous human motion[J]. ACM Transactions on Graphics, 2015, 34(4): Article No.119

[2] Yumer M E, Mitra N J. Spectral style transfer for human motion between independent actions[J]. ACM Transactions on Graphics, 2016, 35(4): Article No.137

[3] Aberman K, Weng Y J, Lischinski D, *et al*. Unpaired motion style transfer from video to animation[J]. ACM Transactions on Graphics, 2020, 39(4): Article No.64

[4] Park S, Jang D K, Lee S H. Diverse motion stylization for multiple style domains via spatial-temporal graph-based generative model[J]. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2021, 4(3): Article No.36

[5] Tao T X, Zhan X H, Chen Z Q, *et al*. Style-ERD: responsive and coherent online motion style transfer[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 6593-6603

[6] Hsu E, Pulli K, Popović J. Style translation for human motion[C] //Proceedings of the ACM SIGGRAPH Papers. New York: ACM Press, 2005: 1082-1089

[7] Huang X, Liu M Y, Belongie S, *et al*. Multimodal unsupervised image-to-image translation[C] //Proceedings of the 15th European Conference on Computer Vision. Heidelberg: Springer, 2018: 172-189

[8] Liu M Y, Huang X, Mallya A, *et al*. Few-shot unsupervised image-to-image translation[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 10551-10560

[9] Zhu J Y, Park T, Isola P, *et al*. Unpaired image-to-image translation using cycle-consistent adversarial networks[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 2223-2232

[10] Park T, Zhu J Y, Wang O, *et al*. Swapping autoencoder for deep image manipulation[C] //Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: Curran Associates Inc, 2020: Article No.604

[11] Jang D K, Park S, Lee S H. Motion puzzle: arbitrary motion

style transfer by body part[J]. ACM Transactions on Graphics, 2022, 41(3): Article No.33

[12] Gatys L A, Ecker A S, Bethge M. A neural algorithm of artistic style[OL]. [2023-06-29]. https://arxiv.org/abs/1508.06576v2

[13] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 1501-1510

[14] Li Y J, Fang C, Yang J M, *et al*. Universal style transfer via feature transforms[C] //Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: Curran Associates Inc, 2017: 385-395

[15] Sheng L, Lin Z Y, Shao J, *et al*. Avatar-net: multi-scale zero-shot style transfer by feature decoration[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 8242-8250

[16] Park D Y, Lee K H. Arbitrary style transfer with style-attentional networks[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 5880-5888

[17] Yan S J, Xiong Y J, Lin D H. Spatial temporal graph convolutional networks for skeleton-based action recognition[C] //Proceedings of the 32th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2018

[18] Liu S H, Lin T W, He D L, *et al*. AdaAttN: revisit attention mechanism in arbitrary neural style transfer[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2021: 6649-6658

[19] Guo C, Zou S H, Zuo X X, *et al*. Generating diverse and natural 3D human motions from text[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 5152-5161

[20] Tevet G, Gordon B, Hertz A, *et al*. MotionCLIP: exposing human motion generation to CLIP space[C] //Proceedings of the 17th European Conference on Computer Vision. Heidelberg: Springer, 2022: 358-374

[21] Peng X B, Guo Y R, Halper L, *et al*. ASE: large-scale reusable adversarial skill embeddings for physically simulated characters[J]. ACM Transactions on Graphics, 2022, 41(4): Article No.94

[22] Tessler C, Kasten Y, Guo Y R, *et al*. CALM: conditional adversarial latent models for directable virtual characters[C] //Proceedings of the ACM SIGGRAPH Conference Proceedings. New York: ACM Press, 2023: Article No.37

[23] Holden D, Saito J, Komura T. A deep learning framework for character motion synthesis and editing[J]. ACM Transactions on Graphics, 2016, 35(4): Article No.138

[24] Holden D, Habibie I, Kusajima I, *et al*. Fast neural style transfer for motion data[J]. IEEE Computer Graphics and Applications, 2017, 37(4): 42-49

[25] Lee D H. Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks[C] //Proceedings of the 30th International Conference on Machine Learning. New York: ACM Press, 2013: 3(2): 896

[26] Zou Y, Yu Z D, Kumar B V K V, *et al*. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training[C] //Proceedings of the 15th European Conference on Computer Vision. Heidelberg: Springer, 2018: 289-305

[27] Arazo E, Ortego D, Albert P, *et al*. Pseudo-labeling and confirmation bias in deep semi-supervised learning[C] //Proceedings of the International Joint Conference on Neural Networks. Los Alamitos: IEEE Computer Society Press, 2020: 1-8

[28] Laine S, Aila T. Temporal ensembling for semi-supervised learning[OL]. [2023-06-29]. https://openreview.net/forum?id=BJ6oOfqge

[29] Sohn K, Berthelot D, Li C L, *et al*. FixMatch: simplifying semi-supervised learning with consistency and confidence[C] //Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: Curran Associates Inc, 2020: Article No.51

[30] Tarvainen A, Valpola H. Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results[C] //Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: Curran Associates Inc, 2017: 1195-1204

[31] Zhou Y, Barnes C, Lu J W, *et al*. On the continuity of rotation representations in neural networks[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 5745-5753

[32] Villegas R, Yang J M, Ceylan D, *et al*. Neural kinematic networks for unsupervised motion retargetting[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 8639-8648

[33] Liu L Y, Jiang H M, He P C, *et al*. On the variance of the adaptive learning rate and beyond[OL]. [2023-06-29]. https://openreview.net/forum?id=rkgz2aEKDr

[34] Karras T, Aila T, Laine S, *et al*. Progressive growing of GANs for improved quality, stability, and variation[OL]. [2023-06-29]. https://openreview.net/forum?id=Hk99zCeAb

[35] Heusel M, Ramsauer H, Unterthiner T, *et al*. GANs trained by a two time-scale update rule converge to a local nash equilibrium[C] //Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: Curran Associates Inc, 2017: 6629-6640

[36] Szegedy C, Liu W, Jia Y Q, *et al*. Going deeper with convolutions[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2015: 1-9