

FontDiffuser：通过多尺度内容聚合和样式对比学习的去噪扩散生成一次性字体

杨振华, 彭德志, 孔玉欣, 张玉怡, 姚聪, 金连文

¹华南理工大学-珠海现代产业创新研究院, 阿里巴巴集团 {eezhyang, pengdezh000, kongyxscut, yuyizhang.scut, yaocong2010}@gmail.com, eelwjn@scut.edu.cn

抽象

自动字体生成是一项模拟任务, 旨在创建一个字体库, 该字体库模仿参考图像的样式, 同时保留源图像中的内容。尽管现有的字体生成方法已经取得了令人满意的性能, 但它们仍然难以处理复杂的字符和较大的样式变化。为了解决这些问题, 我们提出了 FontDiffuser, 这是一种基于扩散的图像到图像的一次性字体生成方法, 它创新性地将字体模仿任务建模为噪声到降噪范式。在我们的方法中, 我们引入了一个多尺度内容聚合 (MCA) 块, 它有效地结合了不同尺度的全局和局部内容线索, 从而增强了复杂字符复杂笔触的保留。此外, 为了更好地管理风格迁移中的大变化, 我们提出了一个风格对比细化 (SCR) 模块, 这是一种用于风格表示学习的新结构。它利用样式提取器将样式与图像解开, 然后通过精心设计的样式对比损失来监督扩散模型。广泛的实验证明了 FontDiffuser 在生成各种字符和样式方面的最新性能。与以前的方法相比, 它在复杂的字符和较大的样式变化方面始终表现出色。该代码可在

<https://github.com/yeungchenwa/FontDiffuser> 获取。

介绍

自动字体生成旨在根据参考图像以所需样式创建新的字体库, 这称为仿制任务。字体生成具有重要的应用, 包括新字体创建、古字符恢复和用于光学字符识别的数据增强。因此, 它具有重要的商业和文化价值。但是, 这种模仿过程既昂贵又耗费大量人力, 特别是对于具有大量字形的语言, 例如中文 (> 90,000)、日语 (> 50,000) 和韩语 (> 11000)。现有的自动方法主要解开样式和内容的表示, 然后将它们集成以输出结果。

尽管这些方法在字体生成方面取得了显著的成功, 但它们仍然受到复杂的字符生成和较大的样式变化迁移的影响, 导致



(a) 我们的方法生成的字符



(b) 复杂字符



(c) 较大的风格变化

图 1: (a) 我们的方法生成的不同复杂度的字符。 (b) (c) 不同方法对复杂字符和大风格变化的结果。'ref' 表示参考图像。 (1) - (4) 分别代表 DG-Font (Xie et al. 2021)、MX-Font (Park et al. 2021b)、CG-GAN (Kong et al. 2022) 和 CF-Font (Wang et al. 2023) 的红色框突出显示其他方法的失败。

到严重的笔触缺失、伪影、模糊、布局错误和样式不一致, 如图 1 (b) (c) 所示。回顾性地, 大多数字体生成方法 (Park 等人, 2021a, b; Xie 等人, 2021 年; Tang 等人, 2022 年; Liu 等人, 2022 年; Kong 等人, 2022 年; Wang et al. 2023) 采用基于 GAN 的框架 (Goodfellow et al. 2014), 由于其对抗性训练的性质, 该框架可能会受到不稳定训练的影响。此外, 这些方法中的大多数仅通过单尺度高级特征来感知内容信息, 而忽略了对保留源内容至关重要的细粒度细节, 尤其是对于复杂字符。也有许多方法 (Cha 等人, 2020 年; Park 等人, 2021a, b; Liu 等人, 2022 年; Kong 等人, 2022 年; He et al. 2022), 利用先验知识来促进字体生成, 例如字符的笔画或组件组成;但是, 对复杂字符进行注释的成本很高。此外, 在以前的文献中, 目标风格通常由简单的分类器或判别器表示, 它们很难学习挪用

目标样式在以前的文献中通常由简单的分类器或判别器表示 ATE 样式，并且以较大的变化 ~~而美庄~~ 迁移。

在本文中，我们提出了 FontDiffuser，这是一种基于扩散的图像到图像的一次性字体生成方法，它将字体生成学习建模为一种噪声到降噪范式，并能够生成看不见的字符和样式。在我们的方法中，我们创新性地引入了多尺度内容聚合

(MCA) 块，它利用各种尺度的全局和本地内容特征。此块通过利用大规模特征包含大量精细信息（笔划或组件），而小规模特征主要封装全局信息（布局）这一事实，有效地保留了复杂字符源图像中的复杂细节。此外，我们引入了一种新的风格表示学习策略，通过应用风格对比细化 (SCR) 模块来增强生成器模拟风格的能力，特别是对于源图像和参考图像之间的较大变化。该模块利用样式提取器将样式与字体解开，然后使用样式对比损失向扩散模型提供反馈。SCR 充当监督者，鼓励我们的扩散模型识别各种样本之间的差异，这些样本具有不同的风格但具有相同的特征。此外，我们设计了一个参考-结构交互 (RSI) 模块，通过利用与参考特征的交叉注意力交互来显式学习结构变形（例如，字体大小）。

为了验证生成不同复杂度的字符的有效性，我们根据字符的笔画数量将字符分为三个复杂度级别（简单、中等和困难），并分别在每个级别测试我们的方法。广泛的实验表明，我们提出的 FontDiffuser 在三个复杂程度的字符上优于最先进的字体生成方法。值得注意的是，如图 1 (a) 所示，FontDiffuser 在生成复杂字符和大型样式变化方面始终表现出色。此外，我们的方法可以应用于跨语言生成任务，展示了 FontDiffuser 的跨域泛化能力。

我们将我们的主要贡献总结如下。

- 我们提出了 FontDiffuser，这是一种新的基于扩散的图像到图像的一次性字体生成框架，可在生成复杂字符和处理大型样式变化方面实现最先进的性能。
- 为了增强对复杂字符复杂笔画的保留，我们提出了一个多层次内容聚合 (MCA) 块，利用内容编码器不同尺度的全局和局部特征。
- 我们提出了一种新的风格表示学习策略，并精心设计了一个风格对比细化 (SCR) 模块，该模块使用风格对比损失来监督扩散模型，从而能够有效地处理大的风格变化。
- FontDiffuser 在生成简单、中等和困难复杂度级别的角色时，表现出优于现有方法的性能，展示了对看不见的角色和样式的强大泛化能力。此外，我们的方法可以扩展到跨语言一代，例如中文到韩语。

图像到图像的翻译

图像到图像 (I2I) 转换任务是将图像从源域转换为目标域。以前，图像到图像方法 (Isola 等人, 2017 年;Liu、Breuel 和 Kautz 2017;Zhu 等人, 2017 年;Liu et al. 2019) 通常通过 GAN 来解决 (Goodfellow et al. 2014)。例如，Pix2pix (Isola et al. 2017) 是第一个 I2I 转换框架。FUNIT (Liu et al. 2019) 利用 AdaIN (Huang and Belongie 2017) 来组合编码内容图像和类图像。最近，有许多方法 (Choi 等人, 2021 年;Sasaki、Willcocks 和 Breckon 2021;Saharia 等人, 2022a) 利用扩散模型来解决图像到图像的翻译任务。例如，ILVR (Choi et al. 2021) 仅使用参考图像仅基于经过训练的 DDPM (Ho, Jain and Abbeel 2020) 生成高质量图像。Palette (Saharia et al. 2022a) 提出了一个简单的图像到图像扩散模型，其性能优于 GAN 和回归基线。

Few-shot 字体生成

早期字体生成方法 (Chang 等人, 2018 年;Lyu 等人, 2017 年;田 2017;江 et al. 2017;Sun、Zhang 和 Yang 2018 年) 将字体生成任务视为图像到图像的翻译问题，但他们无法生成看不见的样式字体。为了解决这个问题，SA-VAE (Sun et al. 2017) 和 EMD (Zhang、Zhang 和 Cai 2018) 通过解开风格和内容表示来生成看不见的字体。为了使生成器能够捕获本地样式特征，一些方法 (Wu、Yang 和 Hsu 2020;Huang et al. 2020;Cha 等人, 2020 年;Park 等人, 2021a, b;Liu 等人, 2022 年;Kong et al. 2022) 利用先验知识，例如中风和成分。例如，LF-Font (Park et al. 2021a)、MX-Font (Park et al. 2021b) 和 CG-GAN (Kong et al. 2022) 采用基于组件的学习策略来增强本地风格表示学习的能力。XMP-Font (Liu et al. 2022) 利用预训练策略来促进风格和内容的解开。Diff-Font (He et al. 2022) 采用笔画信息来支持采样，但未能生成看不见的字符。但是，对于复杂字符，描边和组件的注释成本很高。一些无先验方法 (Xie 等人, 2021 年;Tang 等人, 2022 年;Wang et al. 2023) 已被提出。DG-Font (Xie et al. 2021) 以无监督的方式取得了有希望的性能。Fs-Font (Tang et al. 2022) 旨在发现内容图像和样式图像之间的空间对应关系，以了解局部样式细节，但其参考选择策略对结果质量很敏感。CF-Font (Wang et al. 2023) 融合了不同字体的各种内容特征，并引入了迭代样式-向量优化策略。但是，这些方法仍然难以生成复杂的字符和处理样式迁移中的大变化。

扩散模型

最近，扩散模型在视觉生成任务中取得了快速发展。已经开发了几个突出的条件扩散模型 (Nichol 等人。

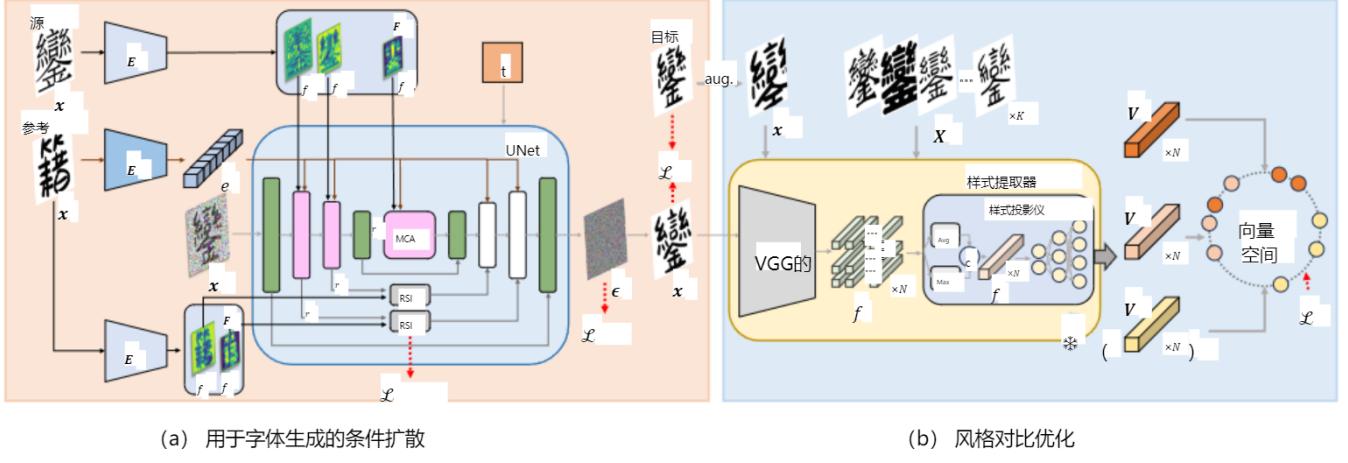


图 2：我们提出的方法概述。 (a) 条件扩散模型是一个基于 UNet 的网络，由内容编码器 E 和样式编码器 E 组成。参考图像 x_{ref} 分别通过样式编码器 E 和内容编码器 E，得到样式嵌入 e ，结构映射 F 。源图像由内容编码器 E 编码。为了获得多尺度特征 F ，我们从 E 的不同层中得出输出，并通过我们提出的 MCA 块注入每个层。RSI 块用于从参考结构特征 F 进行空间变形。 (b) Style Contrastive Refinement 模块是将不同的风格与图像分开，并为扩散模型提供指导。

2021;Ramesh 等人, 2022 年;Saharia 等人, 2022b;Rombach 等人, 2022 年;Zhang 和 Agrawala 2023;Ruiz 等人, 2023 年)。例如, LDM (Rombach 等人, 2022 年) 提出了一种交叉注意力机制, 将条件纳入 UNet 并处理潜在空间中的扩散过程。在文本图像生成中, (Luhman 和 Luhman 2020;Gu 等人, 2023 年;Nikolaïdou 等人, 2023 年) 应用扩散模型来生成手写字符并展示其有前途的效果。CTIG-DM (Zhu et al. 2023) 设计了图像、文本和样式作为条件, 并在扩散模型中引入了四种文本图像生成模式。与一般图像生成相比, 字体生成需要精细级别的不同笔触细节和复杂的结构特征。这促使我们利用多尺度的内容特征, 并提出一种创新的风格对比学习策略。

方法论

如图 2 所示, 我们提出的方法由一个 Conditional Diffusion 模型和一个 Style Contrastive Refinement 模块组成。在条件扩散模型中, 给定一个源图像 x 和一个参考图像 x_{ref} , 我们的目标是训练一个条件扩散模型, 其中最终输出图像不仅应该具有与 x 相同的内容, 而且还应该与参考样式一致。风格对比优化模块旨在从一组图像中解开不同的风格, 并通过风格对比损失为扩散模型提供指导。

用于字体生成的条件扩散

基于 DDPM (Ho、Jain 和 Abbeel 2020), 我们基于扩散的图像到图像字体生成方法的一般思路是设计一个正向过程, 逐步将噪声添加到目标分布 $x \sim q(x)$,

而去噪过程涉及学习反向映射。去噪过程旨在以 T 步将噪声 $x \sim (0, I)$ 转换为目标分布。

具体来说, FontDiffusers 的正向过程是一个马尔可夫链, 噪声添加过程可以总结如下:

$$x = \sqrt{\frac{1}{\alpha}} x + \sqrt{1 - \frac{1}{\alpha}} \epsilon, \quad (1)$$

其中 $t \in [0, T]$ ϵ 是添加的高斯噪声。 $\alpha = 1 - \beta$, $\bar{\alpha} = 1 - \beta$ 是固定的方差超参数。在反向过程中, 可以通过模型近似反向映射来预测噪声 $\epsilon(x, t, x, x)$, 然后得到 x_{as} , 如下所示:

$$x_{as} = \sqrt{\frac{1}{\alpha}} (x - \sqrt{\frac{1 - \alpha}{1 - \bar{\alpha}}} \epsilon(x, t, x, x)) + \sigma z, \quad (2)$$

其中 σ 是超参数, 噪声 $z \sim (0, I)$ 。我们使用条件扩散模型预测噪声 $\epsilon(x, t, x, x)$ 。具体来说, 为了增强复杂字符的保留, 我们采用多尺度内容聚合 (MCA) 块将全局和本地内容线索注入到模型的 UNet 中。此外, 采用参考-结构交互 (RSI) 块来促进参考特征的结构变形。

多尺度内容聚合 (MCA) 生成复杂角色一直是一项具有挑战性的任务, 许多现有方法仅依赖于单一尺度的内容特征, 而忽略了笔触和组件等复杂细节。如图 3 所示, 大规模特征保留了大量详细信息, 而小规模特征则缺乏这些信息。

因此, 我们采用了多尺度内容聚合 (MCA) 块, 将不同尺度的全局和本地内容特征注入到我们扩散模型的 UNet 中。

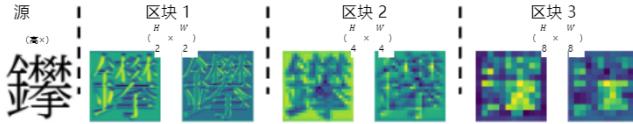


图 3：各种块中的内容特征。

具体来说，源图像 x_{is} 首先由内容编码器 E 嵌入，从不同层获取多尺度内容特征 $F = \{f_1, f_2, f_3\}$ 。与样式编码器 E 编码的样式嵌入一起，每个内容特征 f 分别通过三个 MCA 模块注入到 UNet 中。如图 4 所示，内容特征 f 与前面的 UNet 块特征 r 连接在一起，从而产生通道信息特征 I 。为了增强自适应选择性通道融合的能力，我们对 I 应用了通道注意力（胡、Shen 和 Sun 2018），其中采用了平均池化、两个 1×1 卷积和一个激活函数。注意力产生一个全局通道感知向量 W ，用于对通道信息特征 I_{via} 通道乘法进行加权。然后，在残差连接之后，我们采用 1×1 卷积来减少 I 的通道数，得到输出 I 。最后，我们应用一个交叉注意力模块来插入样式嵌入 e ，其中 e_{is} 用作 Key 和 Value，而 l_{is} 用作 Query。

参考-结构交互 (RSI) 源图像和目标图像之间存在结构差异（例如，字体大小）。为了解决这个问题，我们提出了一个参考-结构交互 (RSI) 块，它采用可变形卷积网络

(DCN) (Dai et al. 2017) 在 UNet 的跳跃连接上进行结构变形。与 (Xie et al. 2021) 相比，我们的条件模型直接从参考特征中提取结构信息，以获得为 DCN 设置的变形偏移 δ_{of} 。

具体来说，参考图像 x_{is} 首先通过内容编码器 E ，得到结构映射 $F = \{f_1, f_2\}$ ，每个 f 分别作为两个 RSI 模块的输入。UNet 特征和参考特征之间的空间位置存在错位。因此，我们没有应用 CNN 来获得传统 DCN 中设置的偏移量 δ_{of} ，而是引入了交叉注意力来实现远距离交互。交互过程可以用公式 3 来概括： ris UNet 特征。此过程的基本要素涉及在 softmax 操作中利用 UNet 功能 $rand$ 结构映射 f ，该操作主要计算相对于每个查询位置的兴趣区域。

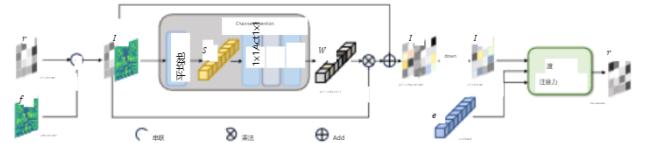


图 4：多尺度内容聚合。

其中 Φ_c , Φ_s , Φ_v 是线性投影，FFN 表示前馈网络。I 是 RSI 的输出。

风格对比优化

字体生成的一个目的是实现预期的样式模仿效果，而不管源和引用之间的样式如何变化。一种新颖的策略是找到合适的样式表示并进一步为我们的模型提供反馈。因此，我们提出了一个风格对比优化 (SCR) 模块，这是一个字体风格表示学习模块，它从一组样本图像中解开风格，并结合风格对比损失来监督我们的扩散模型，确保生成的风格与目标在全局和局部层面保持一致。

SCR 的架构如图 2 的右侧所示，它由一个样式提取器组成。受 (Zhang et al. 2022) 的启发，采用 VGG 网络将字体图像嵌入到提取器中。为了有效地捕捉全局和局部风格特征，我们从 VGG 网络中选择了 N 层特征图 $F = \{f_1, f_2, \dots, f_N\}$ ，将它们用作风格投影仪的输入。投影机应用平均池化和最大池化来分别提取不同的全局通道特征，然后在通道上将它们连接起来，得到特征 $F = \{f_1, f_2, \dots, f_N\}$ 。最后，经过几次线性投影后，得到样式向量 $V = \{v_1, v_2, \dots, v_N\}$ 。

样式向量 V 可以为扩散模型提供监督信号，并指导其模仿样式。因此，我们采用了一种对比学习策略，其中我们利用预先训练的 SCR 并结合风格对比损失 L 来监督生成的样本 x_{is} 的风格是否与目标风格一致，并与负面风格区分开来。为了确保内容不相关和风格相关，我们选择目标图像作为正样本，并选择 K 个风格不同但内容相同的负样本，而不是直接将所选目标样本的其余部分视为负片。因此，SCR 的监督可以总结如下：

$$\begin{aligned}
 S &\in R = f \text{ 拉滕 } (f), \\
 S &\in R = f \text{ flatten } (r), \\
 Q &= \Phi(S), \quad K = \Phi(S), \quad V = \Phi(S), \\
 F &= \text{soft tmax } \left(\sqrt{\frac{QK}{d}} \right) V, \quad \delta_{of} \text{ set} = \text{FFN}(F), \\
 I &= \text{DCN}(r, \delta_{of} \text{ set}),
 \end{aligned} \tag{3}$$

$$L = - \sum_{i=0}^{K-1} \log \frac{\exp(p_i \cdot v_i / \tau)}{\exp(v_i \cdot v_i / \tau) + \sum_{i=1}^{K-1} \exp(v_i \cdot v_i / \tau)}, \tag{4}$$

其中 Extrac 表示样式提取器。K 是负样本数。 V 、 V 和 V 分别表示生成的正负样本的样式向量。 v 、 v 、 v 分别表示生成的第一个正层和负层向量。 τ 是温度超参数，设置为 0.07。SCR 的预训练细节见附录。

为了增强风格模仿的鲁棒性，我们对正目标样本应用了增强策略，其中包括随机裁剪和随机调整大小。

培训目标

我们的培训采用从粗到细的两阶段策略。

第一阶段 在第一阶段，我们主要使用标准的 MSE 扩散损失来优化 FontDiffuser，不包括 SCR 模块。这确保了我们的生成器获得了字体重建的基本能力：

$$L = L + \lambda L + \lambda L_{\text{off}} f \text{ 集}, \quad (5)$$

其中，

$$L = \| \varepsilon - \varepsilon(x, t, x, x) \|, \quad (6)$$

$$L = \sum_{l=1}^L \| VGG(x) - VGG(x) \|, \quad (7)$$

$$L_{\text{off}} f \text{ 集} = \text{mean}(\| \delta_{\text{off}} f \text{ set} \|), \quad (8)$$

其中 L 表示阶段 1 中的总损失。 $VGG(\cdot)$ 是由 VGG 编码的层特征， L 是所选层的数量。 L_{is} 用于惩罚生成的 x 的 VGG 特征与相应的 x_{target} 目标特征之间的内容错位。偏移损失 $L_{\text{off}} f \text{ set}$ 用于约束 RSI 模块中的偏移量， mean 是平均过程。 $\lambda = 0.01$ 和 $\lambda = 0.5$ 。

第 2 阶段 在第 2 阶段，我们实现了 SCR 模块，结合了风格对比损失，为全局和局部层面的扩散模型提供风格模仿指导。因此，我们在第 2 阶段的条件扩散模型通过以下方式进行了优化：

$$L = L + \lambda L + \lambda L_{\text{off}} f \text{ 集} + \lambda L, \quad (9)$$

其中 L 表示阶段 2 中的总损失。超参数 $\lambda = 0.01$ 、 $\lambda = 0.5$ 和 $\lambda = 0.01$ 。

实验

数据集和评估指标

我们收集了 424 种字体的中文字体数据集。我们随机选取 400 种字体（简称“现字”），其中 800 个汉字（简称“现字”）作为训练集。我们在两个测试集上评估方法：一个包括 100 个随机选择的可见字体，其中包含 272 个在训练过程中看不到的字符（称为“SFUC”），另一个测试集由 24 个不可见的字体和 300 个不可见的字符（称为“UFUC”）组成。三个复杂程度的分类细节见附录。此外，我们还对 24 种未见过的字体和 800 种可见的字符（称为“UFSC”）进行了比较。

对于定量评估，我们采用 FID、SSIM、LPIPS 和 L1 损失指标。像素级指标 SSIM 和 L1 损失用于测量生成的样本和目标样本之间的每像素一致性。此外，LPIPS (Zhang et al. 2018) 和 FID (Heusel et al. 2017) 是更接近人类视觉感知的感知指标。

此外，我们进行了一项用户研究以评估图像的主观质量。我们从 SFUC 中随机选择了 30 种可见的字体，从 UFUC 中随机选择了 20 种未见过的字体。在每种字体中，我们随机选择 6 个字符（每种复杂度 2 个字符）。总共有 25 名参与者被要求从所有方法的结果中选择最好的。

实现细节

我们使用 AdamW 优化器训练 FontDiffuser， $\beta_1 = 0.9$ 和 $\beta_2 = 0.999$ 。图像大小设置为 96。此外，按照 (Ho 和 Salimans 2022)，我们只需以 0.1 的概率丢弃源图像和参考图像。在第 1 阶段，我们训练了批量大小为 16 且总步长为 440000 的模型。学习率设置为线性计划的 $1e-4$ 。在第 2 阶段，学习率设置为 $1e-5$ 并固定为常数。我们使用 16 个批次大小、30000 个总步长和 16 个负样本进行训练。实验在单个 RTX 3090 GPU 上进行。

在采样过程中，我们采用无分类器的指导策略 (Ho 和 Salimans 2022) 来放大条件 x 和 x 的影响。我们将无条件内容图像和无条件样式图像设置为像素 255 作为 \emptyset ，我们的采样策略可以表述为：

$$\varepsilon(x, t, x, x) = (1 - s) \varepsilon(x, t, \emptyset, \emptyset) + s \varepsilon(x, t, x, x), \quad (10)$$

其中 s 是指导量表，在实验中设置为 7.5。为了加快采样速度，我们使用了 DPM-Solver++ 采样器 (Lu et al. 2022)，只有 20 个推理步骤。

与最先进的方法的比较

我们将我们的方法与七种最先进的方法进行了比较：一种图像到图像的转换方法 (FUNIT (Liu 等人, 2019 年)) 和六种中文字体生成方法 (LFFont (Park 等人, 2021a)、MX-Font (Park 等人, 2021b)、DGFont (Xie 等人, 2021 年)、CG-GAN (Kong 等人, 2022 年)、Fs-Font (Tang 等人, 2022 年) 和 CF-Font (Wang 等人, 2023 年))。此外，我们还与 Diff-Font (He et al. 2022) 比较了 Unseen Font Seen Character (UFSC)。为了公平地进行比较，我们使用 Song 的字体作为源，所有方法都是根据其官方代码进行训练的。

定量比较 定量结果如表 1 所示。FontDiffuser 在所有矩阵中均达到最佳性能，与 SFUC 和 UFUC 上的其他方法相比，存在显著差距。它指示 FontDiffuser 可以生成在视觉上更接近人类感知的字体。在简单和中等级别，尽管 SFUC 中的 FID 排名第二，但 FontDiffuser 在其余指标上优于其他方法，尤其是感知矩阵 LPIPS。在困难层面上，我们的方法在 SFUC 中表现最好，在 UFUC 中取得了最好的 FID 和 LPIPS 分数。需要注意的是，SSIM 和 L1 损失是像素级指标，可能无法直接反映整体性能。例如，令人印象深刻的视觉效果可能无法与目标像素完全匹配。硬级别结果演示了 FontDiffuser 在生成复杂字符方面的优势。此外，如表 2 所示，FontDiffuser 在 UFSC 上实现了最先进的性能。值得注意的是，Diff-Font (He et al.

		简单 中等 困难 平均 用户														
		FID↓	SSIM↑	LPIPS↓	L1↓	LF↓	字体	ICCV2019	18.6368	0.4823					(%)	
SFUC		FUNIT	ICCV2019	11.3390	0.4342	0.1985	0.3888	11.0158	0.3516	0.2140	0.4474	17.7055	0.3271	0.2374	0.4648	9.6681
		0.2049	0.4184	39.6788	0.3444	0.2349	0.2349	0.7198	0.3830	0.3444	0.2349	0.7196	0.3833	0.3444	0.2349	0.7196
UFUC		FUNIT	ICCV2019	14.5517	0.4507	0.1839	0.3720	16.0900	0.3495	0.2045	0.4484	25.9712	0.2963	0.2403	0.4918	13.1425
		0.1997	0.4257	59.4416	0.3071	0.3071	0.3071	0.4116	0.4116	0.3071	0.3071	0.3071	0.3071	0.3071	0.3071	0.3071

表 1: SFUC 和 UFUC 的定量结果。 “User” 表示用户研究。 “平均” , 用户研究针对三个复杂度级别的所有特征进行评估。粗体表示最先进的技术, 下划线表示次佳。

模型	地点	FID↓	SSIM↑	LPIPS↓	L1↓	LF-字体	ICCV2019	18.6368	0.4823
0.1688	0.8400	DG-字体	C/PR2021	19.19.8079	0.4532	0.2047	0.3646		
MX-字体	CCV2021	9.3239	0.4605	1.6103	0.3571	fs-字体	CVPR2022		
31.3986	0.4270	0.2160	0.3855	CG-GAN	CVPR2022	7.7232	0.4655		
0.1721	0.8544	CF-字体	CVPR2023	14.2027	0.4396	0.2139	0.3713	差异	
字库 - 12.0809	0.4192	0.2022	0.3877	我们的	-7.6708	0.4942	0.1426		
0.3279									

表 2: UFSC 的定量结果。

ref	源		译	
	中	韩	中	韩
脊	脊	脊	脊	脊
瘤	瘤	瘤	瘤	瘤
鳞	鳞	鳞	鳞	鳞
癌	癌	癌	癌	癌

图 5: 跨语言生成 (中文到韩语)。

2022) 只能生成看到的字符, 而我们的方法也大大优于它。

定性比较 在图 7 中, 我们提供了 SFUC 和 UFUC 结果的可视化, 直观地反映了不同方法的视觉效果。FontDiffuser 始终如一地生成高质量的结果, 并且与其他最先进的方法相比, 在内容保留、样式一致性和结构正确性方面表现更好。特别是, 我们的方法在生成复杂字符和处理样式迁移中的大变化方面表现出显著的优势, 而其他方法仍然存在诸如笔触缺失、伪影、模糊、布局错误和样式不一致等问题。我们还在图 5 中展示了一些跨语言的生成样本 (中文到韩文), 这些样本是由我们的方法生成的。它表明 FontDiffuser 在为其他语言生成方面是灵活的, 并且表现出跨域能力, 尽管我们的模型是由中文数据集训练的。

M R S	FID↓	SSIM↑	LPIPS↓	L1↓
✗ ✗ ✗	8.1153	0.4112	0.1526	0.3955
✓ ✗ ✗	7.8419	0.4114	0.1511	0.3954
✓ ✓ ✗	8.4427	0.4137	0.1506	0.3925
✓ ✓ ✓	8.5352	0.4206	0.1496	0.3870

表 3: 不同模块的有效性。M、R 和 S 分别代表 MCA、RSI 和 SCR。第一行表示基线。

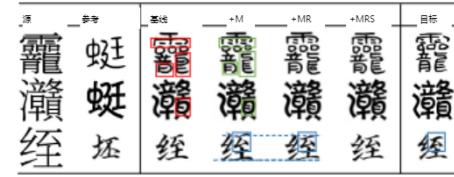


图 6: 不同模块的可视化。M、R 和 S 分别代表 MCA、RSI 和 SCR。红色框表示缺失的笔触, 而绿色框表示相应的改进。蓝色表示结构性提升。

gies. 实验在平均级别的 unseen 字体 unseen characters (UFUC) 上进行了测试。

不同模块的有效性 我们将提议的 MCA、RSI 和 SCR 分开, 并逐步将它们添加到基线中。基线将内容图像与 UNet 的输入 x_{as} 连接起来。表 3 显示, 除 FID 外, 这三个模块的定量结果在 SSIM、LPIPS 和 L1 损失方面都有所改善。此外, 这些模块还有助于增强视觉效果, 如图 6 所示。例如, 在图 6 的第一行中, 通过合并 MCA 模块缓解了基线中缺少笔画的问题。

增强策略在 SCR 中的有效性 我们研究了拟议的增强策略在 SCR 中的优势, 其中 FontDiffuser 在训练阶段 2 中使用和不使用增强策略进行训练。如表 4 所示, 它清楚地表明, 增强策略在 SSIM、LPIPS 和 L1 损失方面提高了生成性能。

消融研究

在本节中, 我们进行了几项消融研究, 以分析我们提出的模块和策略的性能。

源	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
FUNIT	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
MX 字体	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
LF 字体	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
SFUC	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
DG 字体	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
CG-甘	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
Fs-字体	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
CF 字体	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
Ours	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟
目标	攸蚌睹霸躰	狗尻瞳照冀	朋燧麓鑑	灝霹鶴醺囊	涣靦罐髓麟

源	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
FUNIT	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
MX 字体	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
LF 字体	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
UFUC	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
DG 字体	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
CG-甘	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
Fs-字体	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
CF 字体	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
Ours	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒
目标	萱鵠嘶聾鑷	盹叟識趨鼈	鰈瀨榔榔體	蠅嬖鍾鍾鑑	咷戲鶯鶯巒

图 7: SFUC 和 UFUC 的定性比较。红色框突出显示其他方法的失败。

方法	FID↓	SSIM↑	LPIPS↓	L1↓
无增强	8.1758	0.4172	0.1504	0.3900
增强	8.5352	0.4206	0.1496	0.3870
0.1496	0.3870			

表 4: SCR 中增强策略的有效性。

方法	FID↓	SSIM↑	M1	LPIPS↓	L1↓
美国有线电视新闻网	9.1659	0.4130	0.1537	0.3932	
交叉注意力	8.5352	0.4206	0.1496	0.3870	

表 5: 交叉注意力和 CNN 之间的比较。

RSI 中交叉注意力交互与 CNN 的比较 我们对 RSI 中的交叉注意力交互和 CNN 交互进行了比较分析。表 5 中的结果表明，RSI 中的交叉注意力交互在所有矩阵中都优于基于 CNN 的交互，展示了我们提出的方法的优越性。

其他此外，我们在附录中进一步讨论了更多的消融研究，包括负样本对风格对比损失的影响、VGG 层特征在 SCR 中的影响以及引导量表的影响。

SCR 对比评分的可视化

我们在图 8 中提供了 SCR 对比评分的可视化，这表明 SCR 可以有效地将目标与一组样本区分开来，即使其中一些样本表现出相似的风格。通过将 SCR 与风格对比损失相结合，我们观察到 SCR 可以通过对比学习的方式来优化生成的风格。



图 8: SCR 对比评分的可视化。左列表示生成的样本。每行对应于所选样本。红色框突出显示目标，而蓝色框突出显示类似于生成的样式的样本。彩条中较深的颜色表示较大的对比分数，而较亮的颜色表示较小的对比分数。

结论

在本文中，我们提出了一种基于扩散的图像到图像字体生成方法，称为 FontDiffuser，它擅长生成复杂字符和处理样式迁移中的大变化。具体来说，我们提出了 MCA 块，将多尺度内容特征注入我们的扩散模型中，增强了复杂字符的保留。此外，我们提出了一种新的风格表示学习策略，它实现了 SCR 模块并使用风格对比损失来监督我们的扩散模型。此外，RSI 块用于使用参考特征促进结构变形。大量实验表明，FontDiffuser 在三个复杂程度的字符上优于 state-of-the-art 方法。此外，FontDiffuser 展示了其对跨语言字体生成任务（例如，中文到韩文）的适用性，突出了其有前途的跨域能力。

确认

这项研究得到了国家重点研发计划（2022YFC3301703）和阿里巴巴创新研究基金（编号 20210975）的部分支持。我们感谢阿里巴巴华南理工大学联合研究生教育项目的支持。

引用

查, J.;Chun, S.;李, G.;李, B.;金, S.和 Lee, H. 2020 年。具有双重内存的 Few-shot 组合字体生成。在 Proc. ECCV, 735–751 中。斯普林格。张 J.;顾 Y.;张 Y.;王 Y.-F.;和 创新, C. 2018 年。具有分层生成对抗网络的中文笔迹模仿。在 Proc. BMVC, 290.

崔, J.;金, S.;郑 Y.;Gwon, Y.;和 Yoon, S. 2021 年。ILVR: 去噪扩散概率模型的调节方法。在 Proc. ICCV, 14367–14376 中。

戴, J.;齐, H.;熊 Y.;李英;张 G.;胡 H.;和 Wei, Y. 2017 年。可变形卷积网络。在 Proc. ICCV, 764–773 页。

古德费罗, I.;Pouget-Abadie, J.;米尔扎, M.;徐 B.;Warde-Farley, D.;奥泽尔, S.;库尔维尔, A.;和 Bengio, Y. 2014. 生成对抗网络。NeurIPS 论文集, 27。桂 D.;陈 K.;丁 H.;和 Huo, Q. 2023。使用去噪扩散概率模型生成训练数据, 用于手写汉字识别。arXiv 预印本 arXiv: 2305.15660。

他, H.;陈 X.;王 C.;刘 J.;杜, B.;陶 D.;和 Qiao, Y. 2022 年。diff-font: 用于生成稳健 OneShot 字体的扩散模型。arXiv 预印本 arXiv: 2212.05895。赫塞尔, M.;拉姆绍尔, H.;Unterthiner, T.;内斯勒, B.;和 Hochreiter, S. 2017 年。由两个时间尺度更新规则训练的 GAN 收敛到局部纳什均衡。程序

神经 IPS, 30。

Ho, J.;Jain, A.;和 Abbeel, 第 2020 页。去噪扩散概率模型。NeurIPS, 33: 6840–6851. Ho, J.;和 Salimans, T. 2022 年。无分类器扩散引导。arXiv 预印本 arXiv: 2207.12598。胡 J.;沈 L.;和 Sun, G. 2018 年。挤压和激励网络。在 CVPR 程序中, 7132–7141。黄 X.;和 Belongie, S. 2017 年。通过自适应实例规范化实时进行任意样式传输。在 Proc. ICCV, 1501–1510 年。

黄 Y.;他, M.;金, L.;和 Wang, Y. 2020 年。RD-GAN: 通过根式分解和渲染进行少量/零镜头的中文字符样式传输。在 Proc. ECCV, 156–170

斯普林格。

伊索拉, P.;朱, JY.;周, T.;和 Efros, AA. 2017 年。使用条件对抗网络进行图像到图像的翻译。在 CVPR 程序中, 1125–1134。

江 Y.;连, Z.;唐 Y.;和 Xiao, J. 2017 年。DCFont: 端到端的深度中文字体生成系统。在 SIGGRAPH Asia 技术简报中, 1–1

孔, Y.;罗 C.;马, W.;朱 Q.;朱 S.;袁 N.;和 Jin, L. 2022 年。仔细观察以更好地监督: 通过基于组件的鉴别器生成一次性字 * 在 Proc. CVPR, 13482–13491。

刘 M.-Y.;布罗伊尔, T.;和 Kautz, J. 2017 年。无监督图像到图像翻译网络。NeurIPS 论文集, 30。刘 M.-Y.;黄 X.;Mallya, A.;卡拉斯, T.;艾拉, T.;Lehtinen, J.;和 Kautz, J. 2019 年。少镜头无监督图像到图像的转换。在 Proc. ICCV, 10551–10560 中。刘 W.;刘 F.;丁 F.;他, Q.;和 Yi, Z. 2022 年。XMPFont: 用于 fewshot 字体生成的自我监督跨模态预训练。在 CVPR 程序中, 7905–7914。

卢, C.;周, Y.;鲍, F.;陈 J.;李 C.;和 Zhu, J. 2022 年。DPM-Solver++: 用于扩散概率模型引导采样的快速求解器。arXiv 预印本 arXiv: 2211.01095。卢曼, T.;和 Luhman, E. 2020 年。用于手写生成的扩散模型。arXiv 预印本 arXiv: 2011.06704。

柳, P.;白, X.;姚 C.;朱 Z.;黄 T.;和 Liu, W. 2017 年。用于中国书法合成的自动编码器引导 GAN。ICDAR 论文集, 第 1 卷, 1095–1100。IEEE 的。尼科尔, A.;达里瓦尔, P.;拉梅什, A.;夏姆, P.;米什金, P.;麦格鲁, B.;萨茨克弗, I.;和 Chen, M. 2021 年。GLIDE: 使用文本引导扩散模型生成和编辑逼真的图像。arXiv 预印本 arXiv: 2112.10741。

尼古拉杜, K.;雷齐纳斯, G.;克里斯莱因, V.;Seuret, M.;斯菲卡斯, G.;史密斯, EB.;莫凯德, H.;和 Liwicki, M. 2023 年。WordStylist: 使用 Latent Diffusion Models 生成样式化逐字手写文本。arXiv 预印本 arXiv: 2303.16576。

帕克, S.;Chun, S.;查, J.;李, B.;和 Shim, H. 2021a。使用本地化样式表示和分解生成 Few-shot 字体。在 Proc. AAAI, 第 35 卷, 2393–2402 中。

帕克, S.;Chun, S.;查, J.;李, B.;和 Shim, H. 2021b。多个头比一个头好: 由多个本地化专家生成 Few-shot 字体。在 Proc. ICCV, 13900–13909 中。

拉梅什, A.;达里瓦尔, P.;尼科尔, A.;朱 C.;和 Chen, M. 2022 年。分层文本条件图像生成

剪辑潜在值。arXiv 预印本 arXiv: 2204.06125。

罗姆巴赫, R.;布拉特曼, A.;洛伦茨, D.;埃塞尔, P.;和 Ommer, B. 2022 年。使用潜在扩散模型进行高分辨率图像合成。在 CVPR 论文集, 10684–10695。

鲁伊斯, N.;李英;詹帕尼, V.;普里奇, Y.;鲁宾斯坦, M.;和 Aberman, K. 2023 年。DreamBooth: 微调文本到图像扩散模型。在 Proc. CVPR, 22500–22510。

撒哈里亚, C.;陈, W.;张 H.;李, C.;Ho, J.;萨利曼斯, T.;弗利特, D.;和 Norouzi, M. 2022a. 调色板: 图像到图像扩散模型。在 SIGGRAPH 程序中, 1–10。

撒哈里亚, C.;陈, W.;萨克塞纳, S.;李, L.;黄, J.;丹顿, EL;加塞米普尔, K.;贡蒂霍·洛佩斯 (Gontijo Lopes, R.) ;卡拉戈尔·阿扬, B.;萨利曼斯, T.;等人, 2022b。具有深度语言理解的逼真文本到图像扩散模型。NeurIPS, 35: 36479–36484

佐佐木, H.;威尔科克斯, CG;和布雷肯, TP 2021。UNITDDPM: 具有去噪扩散概率模型的未配对图像翻译。arXiv 预印本 arXiv: 2104.05358。孙 D.;任, T;李 C.;苏, H.;和 Zhu, J. 2017 年。通过阅读一些示例来学习书写程式化的汉字。arXiv 预印本 arXiv: 1712.06424。

孙 D.;张 Q.;和 Yang, J. 2018 年。用于自动生成字体的 Pyramid 嵌入式生成对抗网络。在 Proc. ICPR, 976-981 中。IEEE 的。

唐 L.;蔡 Y.;刘 J.;洪, Z.;龚, M.;范, M.;韩, J.;刘 J.;丁 E.;和 Wang, J. 2022 年。通过学习细粒度的局部样式来生成 Few-shot 字体。在 CVPR 程序中, 7895-7904。

田, Y. 2017 年。zi2zi: 掌握具有条件对抗网络的中国书法。<http://github.com/kaonashityc/zi2zi>.

王 C.;周, M.;葛, T.;江 Y.;鲍 H.;和 Xu, W. 2023 年。CF-Font: 用于生成 Few-shot 字体的内容融合。在 CVPR 论文集, 1858-1867 年。

吴, SJ;杨 C.-Y.;和 Hsu, JY-j.2020. CalliGAN: 风格和结构感知的中国书法字符生成器。arXiv 预印本 arXiv: 2005.12500。

谢 Y.;陈 X.;孙, L.;和 Lu, Y. 2021 年。DG-Font: 用于无监督字体生成的可变形生成网络。在 CVPR 程序中, 5130-5140。

张 L.;和 Agrawala, M. 2023 年。向文本到图像扩散模型添加条件控制。arXiv 预印本 arXiv: 2302.05543。

张 R.;伊索拉, P.;埃夫罗斯, AA;谢特曼, E.;和 Wang, O. 2018 年。深度特征作为感知指标的不合理有效性。在 CVPR 程序中, 586-595。

张 Y.;唐 F.;董, W.;黄 H.;马, C.;李, TY;和 Xu, C. 2022 年。通过对比学习进行域增强的任意图像样式迁移。在 SIGGRAPH 程序中, 1-8。张 Y.;张 Y.;和 Cai, W. 2018 年。将样式和内容分开, 以便进行通用样式传输。在 CVPR 程序中, 8447-8455。

朱, JY;帕克, T.;伊索拉, P.;和 Efros, AA 2017 年。使用周期一致的对抗网络的未配对图像到图像转换。在 Proc. ICCV, 2223-2232 中。

朱 Y.;李 Z.;王 T.;他, M.;和 Yao, C. 2023 年。使用扩散模型生成条件文本图像。在 CVPR 程序中, 14235-14245。

方法详细信息

用于字体生成的条件扩散

在本节中，我们将介绍条件扩散模型的更多详细信息，该模型以源图像 x 和单个参考图像 x 为条件，并预测增加的噪声 ϵ 。我们的扩散模型由一个内容编码器 E 、一个样式编码器 E 和一个 UNet 组成。

内容编码器 E and Style Encoder E 在我们的扩散模型中，我们采用了 CG-GAN 的内容编码器和样式编码器 (Kong et al. 2022)。具体来说，我们只接受内容编码器中的前三个块作为我们的块。

UNet 如表 6 所示，FontDiffuser 中的 UNet 由 Conv 块、Down 块、Up 块、多尺度内容聚合 (MCA) 块和样式插入 (SI) 块组成。样式插入 (SI) 块使用 crossattention 模块将样式嵌入 e 插入到 UNet 中。Down 块和 Up 块分别表示 downsample 和 upsample 块。Conv 块是卷积块。UNet 的输入是 $x \in \mathbb{R}$ ，输出是 $\epsilon \in \mathbb{R}$ 。

块	块数	输入形状	输出形状
Conv 块	1	$3 \times H \times W$	$64 \times \text{高} \times \text{宽}$
下块	2	$64 \times \text{高} \times \text{宽}$	$64 \times \text{高} \times \text{宽}$
MCA 块	2	$64 \times \text{高} \times 128 \times \text{宽}$	$64 \times \text{高} \times 128 \times \text{宽}$
MCA 块	2	$128 \times \text{高} \times 256 \times \text{宽}$	$128 \times \text{高} \times 256 \times \text{宽}$
下块	2	$256 \times \text{高} \times 512 \times \text{宽}$	$256 \times \text{高} \times 512 \times \text{宽}$
MCA 块	1	$512 \times \text{高} \times 512 \times \text{宽}$	$512 \times \text{高} \times 512 \times \text{宽}$
向上块	3	$512 \times \text{高} \times 256 \times \text{宽}$	$512 \times \text{高} \times 256 \times \text{宽}$
SI 块	3	$256 \times \text{高} \times 256 \times \text{宽}$	$256 \times \text{高} \times 256 \times \text{宽}$
SI 块	3	$256 \times \text{高} \times 128 \times \text{宽}$	$256 \times \text{高} \times 128 \times \text{宽}$
向上块	3	$128 \times \text{高} \times 64 \times \text{宽}$	$64 \times \text{高} \times \text{宽}$
Conv 块	1	$64 \times \text{高} \times \text{宽}$	$3 \times H \times W$

表 6: UNet 架构。样式插入 (SI) 块采用交叉注意力模块将样式嵌入 e 插入到 UNet 中。Down 块和 Up 块分别表示 downsample 和 upsample 块。Conv 块是卷积块。

风格对比优化

for SCR 风格的计算对比精炼 (SCR) 模块用于监督我们的扩散模型生成的样本 x_{is} 的风格是否与目标风格一致。具体来说，我们在模型预测噪声后计算原始样本 x_{at} 时间步 t

$\epsilon(x, t, x, x)$ 为：

$$x = \sqrt{\frac{1}{\alpha}}(x - \sqrt{1 - \alpha}\epsilon(x, t, x, x)). \quad (11)$$

在训练过程中，在每一步 t 中， x_{is} 都习惯于以下 SCR 模块来计算对比损失。

实验详情

三个复杂度级别的字符的分类

为了验证对不同复杂度的角色的有效性，我们根据角色的笔画数量将角色分为三个复杂度级别（简单、中等和困难）。如表 7 所示，我们将笔画数在 6 到 10 之间的字符分类为易级字符，将笔画数在 11 到 20 之间的字符分类为中等字符，将笔画数大于 21 的字符分类为困难级字符。图 9 显示了几个分类示例。

复杂程度	笔画数 M
Easy	$6 \leq M \leq 10$
Medium	$11 \leq M \leq 20$
Hard	$M \geq 21$

表 7: 三个复杂度级别的分类。

有更串 噇雕臚 麪巖鑾
芷识芷 磔璮愷 檻巓醜
𧈧𠂊𧈧 褵𢂑 漱巓巓

(a) 简单

(b) 中等

(c) 硬

图 9: 三个复杂度级别的示例。

实现细节

我们的培训程序采用从粗到细的两阶段策略。在第 2 阶段，我们聘请了一名经过预先培训的 SCR 作为主管。在本节中，我们提供了 SCR 的预训练详细信息。

SCR 的预训练 我们通过 AdamW 优化器预训练风格对比精炼 (SCR) 模块，其中 $lr = 1e-4$ ，1000 个预热步骤和线性学习率时间表。预训练期间的负样本数设置为 48，图像大小设置为 96。训练集包括 400 种字体和 800 个字符（与我们实验的中文字体生成的训练数据相同）。SCR 受风格对比损失 L 的监督

$$L_{sc} = -\sum_{i=1}^{N_{p-1}X} \log \frac{\exp(v \cdot v/\tau)}{\exp(v \cdot v/\tau) + \sum_{j=1}^p \exp(v_j \cdot v/\tau)}, \quad (12)$$

其中 v_{den} 表示目标图像。 v 和 \bar{v} 表示正样本（增强的目标图像）和负样本（样式不同但字符相同）。正片图像的增强包括随机裁剪和随机调整大小。 K 是所选负样本的数量，在预训练期间设置为 48。在预训练期间， N_{IS} 所选 VGG 层特征的数量，我们选择特征 $F = \{f_1, f_2, f_3, f_4, f_5\}$ (f_i 是第 i 个 VGG 卷积块的 ReLU 输出)。

更多消融研究

负样本对 Lin 期 2 的影响 我们进一步讨论了负样本数量对 L 的影响，如表 8 所示。 $K = 16$ 和 $K = 32$ 的结果是可比的，由于减少了训练时间，我们在所有实验中都采用了 $K = 16$ 设置。

阴性样本 K	FID↓ SSIM↑ LPIPS↓ L1↓
8	8.5900 0.4148 0.1501 0.3919
16	8.5352 0.4206 0.1496 0.3870
32	8.0692 0.4174 0.1487 0.3897
48	8.2454 0.4172 0.1495 0.3899

表 8: L 的阴性样本数量的影响。粗体表示最先进的技术，下划线表示次佳。

VGG 层特征在 SCR 中的影响 我们进一步讨论了第 2 阶段 VGG 层特征 $F = \{f_1, f_2, \dots, f_5\}$ 在 SCR 中的影响 (f_i 是第 i 个 VGG 卷积块的 ReLU 输出)。如表 9 所示，采用多尺度 VGG 特征可以有效地提高性能，设置 $F = \{f_1, f_2, f_3\}$ 可以获得我们这一代的最佳质量。

层特点 F	FID↓ SSIM↑ LPIPS↓ L1↓
F	9.2220 0.4170 0.1527 0.3890
F, F	8.2554 0.4167 0.1499 0.3902
F, F, F	<u>8.2173</u> 0.4166 0.1505 0.3906 F, F, F

表 9: 阶段 2 中 VGG 层特征 F 的影响。

指导量表的影响 我们进一步讨论了指导量表 s 在抽样过程中的影响。如表 10 所示，设置 $s = 7.5$ 可实现最佳性能。

局限性

虽然我们采用了高效的采样器 DPM-Solver++ (Lu et al. 2022)，但我们的方法仍然需要像大多数基于扩散的生成方法一样分几个步骤生成样本（速度比基于 GAN 的方法慢）。

指导量 s	FID↓ SSIM↑ LPIPS↓ L1↓
1	7.1447 0.3826 0.1696 0.4223
3.5	8.5504 0.4137 0.1504 0.3929
5.5	8.4842 0.4188 0.1496 0.3885
7.5	8.5352 0.4206 0.1496 0.3870
9.5	8.8995 0.4198 0.1503 0.3873
11.5	9.6069 0.4201 0.1514 0.3873
15	12.2369 0.4194 0.1532 0.3880
20	18.2550 0.4175 0.1581 0.3899
30	45.3899 0.4087 0.1790 0.3964

表 10: 指导量表的影响 s 。

结果的更可视化

在本节中，我们提供了 FontDiffuser 生成的结果的更多可视化效果。如图 10 所示，中文字体生成结果包括 Seen Font Unseen Character (SFUC) 和 Unseen Font Unseen Character (UFUC) 上生成的三个复杂度级别 (easy、medium 和 hard) 的字符。此外，我们还通过 FontDiffuser 提供了更多跨语言生成（中文到韩文）的可视化，如图 11 所示。

SFUC

尻狼痴臭精矿捐凌徇烂
转骸剑村犷个癸哈拐杓
笔伦财币臣吃驳蚌抱剥
专争殊犹畀竽姬阵惄站
独钫蚩𠵼碑貽甫但导传
纹倭旺迎邢恶徇转业项

UFUC

嵒聿休莫字胤诩麻有帙
旅猢咅奄又傍笔咅欸昭
疽肫帶的经客胎串大社
莫倧块白砌聿字有秩晨
客胎社肫坟昉的经坟佛
更坟坟甫始桧奋拐妓优

(a) 容易复杂程度的字符

SFUC

嬖鰐繁鯷嗔嫖簸薄澶辨
嗔嬖鯷簸嫖端嫖簪辨澶
赖簪嫖嫖蹀辩储鯷嬖嬖
遽憔鷗翥壞檄鞣燧黢痼
赋馥辉盥鮑衡鰐霍灌
麓鯢鰌鯈羸鳌砨貌鷗醉

UFUC

闔漱雕零盡黎轉簇揜鷁
簇鷁轉竈體瓶鷁臻暭
榔唔歛蹠鷁嚙體槧瓶銳
識憲擅膳榭鑣膳掀櫟鷁
覩蓐悅靚情磉鴻慄歛槧
戲臙闔鷁嚙鷁雕鞶乾暭

(b) 中等复杂度的特征

SFUC

躋黯鱣罐醺霸犀囊燿躉
夔犀躉鶴燭蠹麟囊蠹體
禳鐫糖罐靡蠹氍鬟露鱣
赣嬖髑燿鱣癩癩瓠鐫鬻
蹠麝鑾蘸鼈燭趨鼈禳
蠹瓢鑾霹躉羶黯鼈鑾鑾

UFUC

鼈讚饗櫻籜錚灑嚴趨鼈
鼈鼈類箇續麝鼈櫻灑鑾
櫻讓讚箇鑷類廳鼈鼈灑
錚顛餽鼈鼈鼈鼈鼈鼈
鼈鼈鼈鼈鼈鼈鼈鼈鼈鼈
鼈鼈鼈鼈鼈鼈鼈鼈鼈鼈

(c) 复杂度较高的字符

图 10: FontDiffuser 的结果可视化。

图 11：跨语言生成（中文到韩语）的可视化。