中图分类号：TP391　　单位代号：10280
密　　级：公开　　学　　号：22721486

上海大学

硕士学位论文

SHANGHAI UNIVERSITY
MASTER'S DISSERTATION

题目	面向潜空间扩散模型文生图任务的文本渲染改善方法研究

作　　者　陈斌
学科专业　计算机应用技术
导　　师　王昊
完成日期　二〇二五年六月

姓　　名：陈斌　　学号：22721486

论文题目：面向潜空间扩散模型文生图任务的文本渲染改善方法研究

上海大学

本论文经答辩委员会全体委员审查，确认符合上海大学硕士学位论文质量要求。

答辩委员会签名:

主　　席:

委　　员:

导　　师:

答辩日期:　　年　　月　　日

姓　　名：陈斌　　学号：22721486

论文题目：面向潜空间扩散模型文生图任务的文本渲染改善方法研究

上海大学学位论文原创性声明

本人郑重声明: 所呈交的学位论文是本人在导师指导下, 独立进行研究工作所取得的成果。除了文中特别加以标注和致谢的内容外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他研究者对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名:

日期:　　年　　月　　日

上海大学学位论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定，即: 学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

(保密的论文在解密后应遵守此规定)

学位论文作者签名:　　导师签名:

日期:　　年　　月　　日　　日期:　　年　　月　　日

上海大学工学硕士学位论文

面向潜空间扩散模型文生图任务的文本渲染改善方法研究

作　　者: 陈斌
导　　师: 王昊
学科专业: 计算机应用技术

计算机工程与科学学院
上海大学
2025 年 6 月

A Dissertation Submitted to Shanghai University for the
Degree of Master in Engineering

Research on Improving Text Rendering for Latent Diffusion-Based Text-Guided Image Generation

Candidate: Bin Chen
Supervisor: Hao Wang
Major: Technology of Computer Application

School of Computer Engineering and Science
Shanghai University
June, 2025

摘　要

随着扩散模型（Diffusion Models）为代表的新一代生成式模型的快速发展，文本到图像生成（Text-to-Image Generation）技术将计算机视觉与自然语言处理这两大人工智能领域紧密融合，极大地推动了数字艺术创作、广告视觉设计、虚拟内容生产和人机交互界面的创新实践。然而，随着应用场景对图像生成质量要求的不断提升，人们已不再满足于模型仅能准确呈现常规视觉元素（如人物、动物或场景），而进一步追求模型在视觉文本渲染（Visual Text Rendering）方面也具备高度可控的生成能力，以满足标志设计、海报制作、广告创意及社交媒体内容创作等高精度、低容错的视觉设计需求。因此，深入研究并持续改善文本到图像生成模型的文本渲染能力，对进一步推动生成式人工智能技术在产业界和教育界的广泛落地与应用具有重要的现实意义。本文致力于探索文本渲染改善方法，所做的工作如下：

（1）基于混合专家的多属性可控文本渲染改善方法：现有文本渲染专用模型（如AnyText、TextDiffuser 等）通常围绕文本内容控制和文本位置控制进行优化，而在颜色控制与字体控制方面则表现出明显不足。这些模型普遍采用独立的字形控制条件分支来确保文本内容控制的准确性，而对于图片背景、文本颜色及字体等视觉属性的控制则通过自然语言提示统一注入，会导致属性与实体绑定混淆，即控制条件耦合；还会导致难以精准地为不同控制条件分配合适的控制强度，即控制力度分配不佳。为了解决控制条件耦合的问题，本文专门构建了针对文本颜色和字体控制的合成式数据集，分别独立训练了颜色专家与字体专家，以此实现文本颜色与字体控制条件的解耦注入；进一步地，为了克服控制力度分配不佳的问题，本文创新性地引入了一个自适应路由器，能够根据具体输入的特征动态调整各专家之间的协作强度，以实现颜色、字体和拼写控制力度的精准动态分配，从而大幅提升文本渲染的多属性可控性与视觉一致性。在聚焦于文本内容控制的大规模评估数据集上的自动化量化实验表明，提出的方法较好地保留了主干架构的文本内容控制能力；在聚焦于文本颜色与字体控制的合成式评估数据集上的大量人工评估表明，提出的方法显著地提升了主干架构的文本颜色与字体控制能力。

（2）基于推理时扩展与后过滤的文本渲染改善方法：现有文本渲染的评估手段

大多聚焦于文本内容控制维度，忽略了对文本颜色控制和文本字体控制等维度的量化评估，无法系统全面地刻画文本到图像生成模型的真实文本渲染能力。此外，随着新兴的第一梯队通用文本到图像生成模型（如 Stable Diffusion 3、FLUX 等）的快速发展，其架构逐渐多样化、复杂化，传统基于模型架构调整或模型参数微调的文本渲染改善方法难以直接迁移到这些通用模型上。为了解决上述问题，本文首先构建了一套全面覆盖文本内容控制、文本颜色控制、文本字体控制、文本位置控制与图片内容控制的五维自动化文本渲染量化评估框架，实现了对任意文本到图像生成模型文本渲染能力的客观度量；进一步地，本文提出了一种基于推理时扩展与后过滤的文本渲染改善方法，通过在推理阶段对模型进行多次采样扩展，生成更多的候选图片，再利用大型视觉语言模型（LVLM）对候选图片集合进行自动化评分和筛选，以架构无关、无需训练的方式，有效地挖掘和放大不同架构文本到图像生成模型自身既有的文本渲染能力，从而提升生成结果与用户期望的匹配程度。实验结果表明，提出的方法能够在一定程度上改善模型固有的文本渲染能力。

综上所述，本文围绕潜空间扩散模型在文本渲染任务上的发展瓶颈，提出了兼具灵活性与实用性的文本渲染改善方法，构建了系统且全面的文本渲染量化评估框架，为文本到图像生成领域在视觉文本渲染方面的研究提供了新的思路与见解，以期推动生成式人工智能在视觉设计、数字创作与智能媒体生产等实际场景中的进一步落地与应用。

关键词：视觉文本渲染；扩散模型；文本到图像生成

ABSTRACT

With the rapid advancement of a new generation of generative models represented by diffusion models, Text-to-Image Generation technology has closely integrated Computer Vision and Natural Language Processing—two major artificial intelligence fields—significantly driving innovation in digital art creation, advertising visual design, virtual content production, and human-computer interaction interfaces. However, as application scenarios increasingly demand higher quality in generated images, users are no longer satisfied with models that accurately depict only conventional visual elements (such as people, animals, or scenes); they now seek highly controllable capabilities in visual text rendering as well, to meet precise and low-tolerance visual design needs in logo design, poster creation, advertising creativity, and social media content generation. Therefore, conducting in-depth research and continuously improving the text rendering capabilities of Text-to-Image generation models hold significant practical importance for the wider adoption and application of generative AI technologies in industry and education. This paper is devoted to exploring methods for improving text rendering and carries out the following research:

(1) Multi-attribute Controllable Text Rendering Improvement Method based on Mixture-of-Experts: Existing dedicated text rendering models (such as AnyText, TextDiffuser, etc.) typically optimize primarily around controlling text content and positioning, showing notable limitations in controlling text color and fonts. These models commonly use independent glyph-conditioning branches to ensure the accuracy of text content control, while the control of visual attributes like background, text color, and fonts relies on unified injections through natural language prompts. This approach results in attribute-entity binding confusion, causing coupling among control conditions, and leads to difficulty precisely allocating suitable control intensities to various control conditions, known as poor control strength distribution. To address this coupling issue, this paper constructs synthetic datasets specifically for text color and font control and independently trains color and font experts, thus enabling decoupled injections of text color and font control conditions. Furthermore,

to overcome the inadequate distribution of control strength, an innovative adaptive router is introduced, dynamically adjusting collaboration intensities among experts based on input-specific features. This approach achieves precise, dynamic allocation of control strengths for color, font, and spelling controls, significantly enhancing multi-attribute controllability and visual consistency in text rendering. Automated quantitative experiments on a large-scale evaluation dataset focused on text content control demonstrate that the proposed method effectively retains the baseline architecture' s text content control capabilities. Extensive human evaluations on synthetic datasets focused on text color and font control confirm that the proposed method substantially improves these control abilities.

(2) Text Rendering Improvement Method based on Inference-time Scaling and Post-filtering: Existing evaluations of text rendering often concentrate primarily on the dimension of text content control, neglecting quantitative assessments of other dimensions such as text color and font control, thereby failing to systematically and comprehensively characterize the true text rendering capabilities of Text-to-Image generation models. Additionally, as emerging leading general-purpose Text-to-Image generation models (such as Stable Diffusion 3 and FLUX) rapidly evolve, their architectures are increasingly diversified and complex, making traditional methods based on architectural adjustments or fine-tuning challenging to directly transfer onto these general models. To address these issues, this paper first develops a comprehensive, five-dimensional automated text rendering evaluation framework covering text content, text color, text font, text positioning, and image content controls, enabling objective measurement of text rendering capabilities for arbitrary Text-to-Image generation models. Further, an inference-time scaling and post-filtering-based method is proposed to improve text rendering. By generating multiple samples during inference, producing numerous candidate images, and subsequently utilizing Large Vision-Language Models (LVLM) for automated scoring and filtering, this approach exploits and amplifies the inherent text rendering capabilities of diverse architectures effectively in an architecture-agnostic, training-free manner, thereby improving alignment between generated images and user expectations. Experimental results demonstrate that the proposed method can notably enhance the inherent text rendering capabilities of models to a certain extent.

In summary, addressing the developmental bottleneck of latent-space diffusion models

in text rendering tasks, this paper proposes flexible and practical improvement methods for text rendering and constructs a systematic, comprehensive evaluation framework. This provides novel ideas and insights for research in visual text rendering within the Text-to-Image generation domain, aiming to further facilitate the application and adoption of generative AI technologies in practical scenarios such as visual design, digital creation, and intelligent media production.

Keywords: Visual Text Rendering; Diffusion Models; Text-to-Image Generation

目　录

第一章　绪论

1.1　研究背景及意义

随着扩散模型（Diffusion Models）[1]等新一代生成式模型的迅猛发展，图像生成（Image Generation）成为了计算机视觉领域的研究前沿。其中，文本到图像生成（Text-to-Image Generation）则进一步将自然语言处理与计算机视觉这两大人工智能领域深度融合，提供了一种便捷、可控的图像生成范式：用户提供对预期生成图片的自然语言描述，交由文本到图像生成模型进行理解与生成，即可产生符合用户预期的高质量图片。这一跨模态的能力迅速在日常生活甚至工作场景中落地，推动了数字艺术创作、广告视觉设计、虚拟内容生产以及人机交互界面等诸多行业的创新实践，也使得生成式人工智能在产业端展现出了巨大的商业潜力与应用价值。

与图片相对的，文本作为人类在沟通交流中另一种常用的一种信息载体，直接地以字符或汉字组合排列的方式传递和存储语义信息，在生活中可谓无处不在。随着数字化时代的到来，将文本嵌入图片中，即将文本内容转换成视觉文本（Visual Text），进而与其余视觉元素协同呈现，已经是一种司空见惯的、更为高效的信息表达方式，广泛地被应用于标志设计、海报制作和广告创意等专业行业，也在社交媒体上以诸如表情包等形式与人们的日常生活产生交织。视觉文本作为图片中的一部分，自然而然地被图像生成领域的研究者所关注。鉴于文本到图像生成的快速进步，人们不再满足于仅仅控制模型准确地生成人物、动物、场景等常规的视觉内容，而是探索让模型能够可控地生成包含视觉文本的图片，即视觉文本渲染（Visual Text Rendering）[2]。这也标志着计算机视觉与自然语言处理的进一步融合。

然而，与常规的视觉内容不同，视觉文本对生成的精度要求更高，容错率更低。当常规的视觉内容在生成的过程中产生轻微的、非原则性的偏差时，往往不容易被用户所察觉和顾虑；而当这样的情况发生在视觉文本渲染的场景中，用户则通常很轻易地能辨识出文本的错误，影响生成图片的可用性。为此，近期的许多研究工作[3-6]积极地探索基于文本到图像生成的文本渲染技术，在文本内容控制、文本位置控制以及多语言文本内容控制等方面都取得了长足的进步。但是，视觉文本渲染并不仅限于内容的正确性：视觉文本的呈现，天然地与文本颜色和文本字体等视觉属性密不

可分。过往的研究工作大多没有关注到这一点，由此暴露出文本渲染的一系列潜在的改善方向。文本渲染改善研究，将进一步推动生成式人工智能在产业界与教育界的落地与应用。

1.2 研究问题

根据上述的研究背景，本文主要围绕文本渲染改善方法展开研究。本文的研究对象是文本渲染，研究的主体是潜空间文本到图像生成模型，涵盖长期占主导地位的基于 U-Net 架构的扩散模型，以及最近新兴的基于 DiT[7] 架构的扩散模型。目前的文本到图像生成模型通常只具备一定的文本内容控制能力，而不具备可靠的文本颜色与文本字体控制能力。此外，越来越多不同架构的文本到图像生成模型展现出了文本渲染的潜力，但现有的评估手段不足以系统全面地了解其文本渲染能力。最后，针对不同架构的模型，文本渲染改善方法往往无法直接迁移。综上所述，本研究主要关注以下两个研究问题：

（1）如何使潜空间文生图扩散模型具备文本颜色与字体可控的文本渲染能力？

早期的通用潜空间文本到图像生成模型，如 Stable Diffusion 1.5[8] 和 Stable Diffusion 2.1[8] 等，通常不具备令人满意和可用的文本渲染能力：模型能够遵循用户的自然语言提示，忠实地生成指定的常规视觉内容（人物、动物、场景等），但无法在图片中包含正确可读的视觉文本。最近，一些研究工作对通用文本到图像生成模型进行了改善，借助引入额外的字形控制条件等模型架构调整，以及专为文本内容控制而收集构建的数据集，实现了一定程度上可用的文本渲染，表现为能够对文本内容和文本位置等进行控制。

然而，当需要同时准确地控制文本颜色或字体时，现有的文本渲染专用模型仍然展现出相当的局限性：一方面，指定的文本颜色经常被渲染到图片背景区域或仅仅覆盖文本的局部区域，说明文本颜色属性与文本实体的绑定不准确或控制力度不足；另一方面，文本字体几乎没有被系统地关注，生成图片中视觉文本的字体完全依赖于模型的随机呈现，不具备可控性。这两方面的局限，阻碍了模型在更高定制化需求的文本渲染场景中的应用。因此，在不削弱模型既有的文本内容和文本位置控制能力的前提下，如何使模型具备文本颜色与字体可控的文本渲染能力，让生成结果既“写对字”，又“用对颜色、选对字体”，是一个值得探索的问题。

(2)如何对文生图模型的文本渲染能力进行系统全面的评估和即插即用的改善？

随着文本到图像生成技术的飞速演进，最近涌现的一批处于“第一梯队”的通用文本到图像生成模型，如 Stable Diffusion 3[9]和 FLUX[10]等，凭借更加新颖的架构设计以及规模庞大的高质量训练数据，固有地展现出一定的文本渲染能力：不仅能够在某些场景中保持相对可靠的文本内容控制，甚至在文本颜色和字体等细粒度视觉属性上也具备一定的控制能力。然而，系统全面的量化评估手段的缺失，成为了制约模型迭代和横向比较的关键瓶颈。目前公开可用的文本渲染评估框架（如 MARIO-Eval[6]、AnyText-benchmark[5]等）主要聚焦于评估文本内容控制；对于模型的文本颜色和文本字体控制能力则无法进行较好的量化评估。因此，构建一套系统全面的文本渲染量化评估框架，是一个需要解决的问题。

此外，随着文本到图像生成模型架构的不断演化，从基于 U-Net 架构的扩散模型，到基于 DiT 架构（Diffusion Transformer）[7]的扩散模型，再到基于视觉自回归架构（Visual AutoRegressive，VAR）[11]的模型，现有的文本渲染改善方法通常无法在不同架构的模型之间直接迁移。因此，提供一种即插即用、架构无关的文本渲染改善方法，是一个需要探索的问题。

1.3 研究内容

基于上述讨论，本文要解决文本到图像生成模型在文本渲染任务中存在的控制条件耦合、控制力度分配不佳、文本渲染量化评估不全面、文本渲染改善方法不易迁移这四个问题。

（1）针对如何使潜空间文生图扩散模型具备文本颜色与字体可控的文本渲染能力，需要解决两个问题：现有文本渲染专用模型统一用自然语言提示指定图片内容与文本颜色或字体而引发控制条件在语义空间中耦合的问题、多控制条件下控制力度分配不佳的问题。为了解决第一个问题，本文构建了专用的合成式数据集，并在此基础上训练了独立的颜色专家与字体专家，以解耦的方式为主干架构注入视觉文本颜色与字体控制条件，并基于启发式规则协调各专家之间的协作，从而在源头上缓解控制条件耦合的问题。为了解决第二个问题，本文额外构建了自适应路由器，以相对可解释的方式动态协调各专家进行协作，从而规避文本颜色与字体控制力度分配不佳的问题。自动化量化评估结合大量人工量化评估表明，本文提出的方法使文

本到图像生成模型具备了文本颜色与字体可控的文本渲染能力。在这过程中，本文探究了不同的颜色和字体表示方式间的差异，以及文本颜色与字体控制条件的注入对主干架构原始文本内容控制能力的影响。

（2）针对如何对文生图模型的文本渲染能力进行系统全面的评估和即插即用的改善，需要解决两个问题：现有文本到图像生成模型的文本渲染能力无法被全面量化评估的问题、现有文本渲染改善方法难以在不同架构模型间直接迁移的问题。为了解决第一个问题，本文提出了一套专门用于衡量文本渲染效果的自动化量化评估框架，从五个维度全面地评估文本到图像生成模型的文本内容控制能力、文本颜色控制能力、文本字体控制能力、文本位置控制能力以及图片内容控制能力。为了解决第二个问题，本文基于大规模文本渲染量化评估所得的实验观察，提出了一种基于推理时扩展与后过滤的文本渲染改善方法，将推理时扩展的思想引入文本到图像生成模型，并利用大型视觉语言模型对较原先更多的候选图片进行评估和筛选，有效地提升了文本到图像生成模型最终输出文本渲染正确的生成图片的能力。实验结果表明，本文提出的文本渲染量化评估框架能够全面客观地评估文本到图像生成模型的文本渲染能力，由此引申的基于推理时扩展与后过滤的文本渲染改善方法能一定程度上对文本到图像生成模型固有的文本渲染能力进行挖掘和放大，并且无需调整模型架构或微调模型参数。

1.4 研究创新点

本文主要针对文本到图像生成领域的文本渲染任务，从文本渲染中的多属性控制和文本渲染中的多维度评选这两个方面展开研究，图 1.1 展示了本文的创新点。

（1）提出基于混合专家的多属性可控文本渲染改善方法，实现文本颜色与字体可控的文本渲染：现有的文本渲染专用模型通常在文本内容控制和文本位置控制等方面进行专门优化，在文本颜色控制和文本字体控制上则显示出较为明显的局限性。这类文本渲染专用模型通常针对文本内容控制采用独立的控制条件注入分支，而针对图片内容、文本颜色及文本字体的控制条件则往往统一地嵌入在自然语言提示中，导致控制条件耦合和控制力度分配不佳的这两个问题。为了解决控制条件耦合的问题，本文构建了用于文本颜色和字体控制的专用合成式数据集并在此基础上独立训练了颜色专家与字体专家，从而解耦地引入文本颜色控制条件与文本字体控制条件；

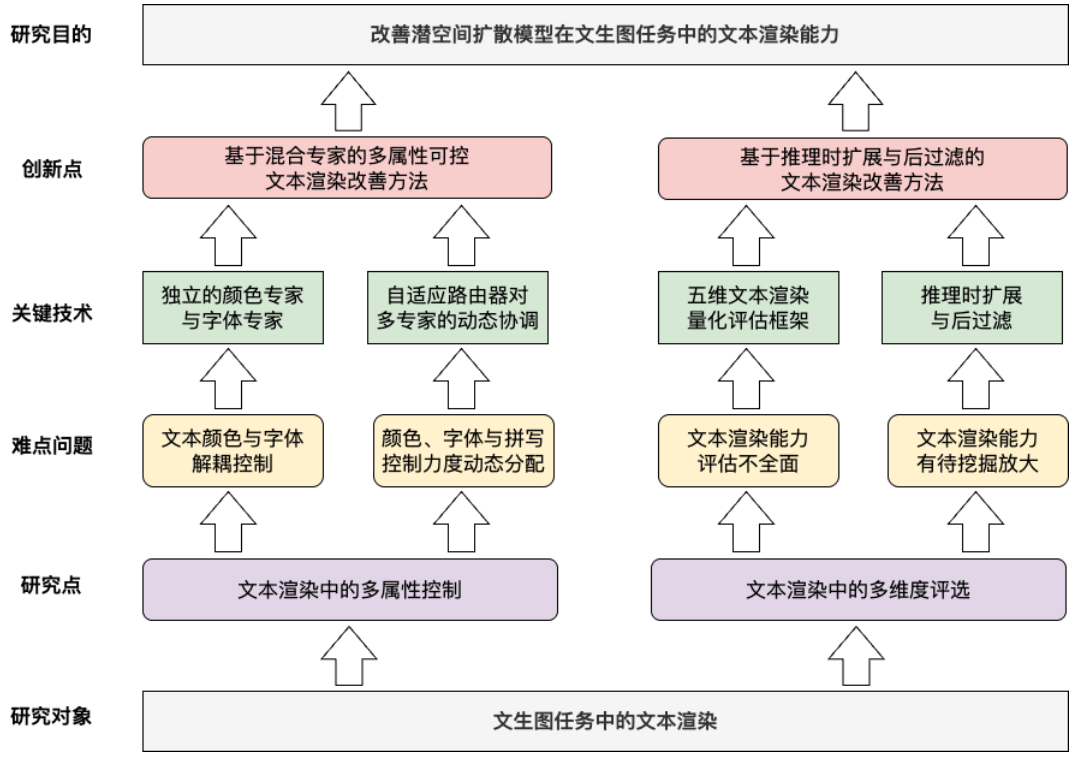


图 1.1　本文的研究创新点

为了解决控制力度分配不佳的问题，本文引入自适应路由器，根据不同的输入，动态地协调各专家间的协作，从而实现颜色、字体与拼写控制力度的动态分配。这两个关键技术构成本文的第一个创新点，即基于混合专家的多属性可控文本渲染改善方法。

（2）提出基于推理时扩展与后过滤的文本渲染改善方法，全面量化评估现有文生图模型的文本渲染能力，并对其进行即插即用的能力挖掘与放大：现有的文本渲染评估大多聚焦于文本内容控制，忽略了文本颜色控制与文本字体控制等维度，无法系统全面地评估文本到图像生成模型的文本渲染能力。另外，越来越多新兴的、位处第一梯队的通用文本到图像生成模型展现出了一定的文本渲染能力。然而，它们的模型架构正在逐渐演化，导致传统的文本渲染改善方法难以直接进行迁移。为了全面地评估文本到图像生成模型的文本渲染能力，本文构建了五维文本渲染量化评估框架，覆盖对文本内容控制、文本颜色控制、文本字体控制、文本位置控制以及图片内容控制的自动化量化评估；为了即插即用地挖掘与放大不同架构的文本到图像生成模型自身既有的文本渲染能力，本文对模型进行推理时扩展与后过滤，采用大型视觉语言模型对经推理时扩展所得的更多候选生成图片进行评估和筛选，以架构无关、无需训练的方式，有效地提升文本到图像生成模型输出符合预期的图片的能

力。这两个关键技术构成本文的第二个创新点，即基于推理时扩展与后过滤的文本渲染改善方法。

1.5 本文的组织结构

本文基于文本到图像生成对文本渲染改善展开深入的研究，首先介绍了图像生成领域中文本渲染的研究背景及意义，随后介绍了文本渲染的相关工作。针对现有的文本到图像生成模型无法准确控制文本颜色与字体的问题，本文研究了多属性可控文本渲染改善方法。为了探究现有文本到图像生成模型的文本渲染能力并架构无关地改善模型的文本渲染能力，本文提出了文本渲染量化评估框架以及基于推理时扩展与后过滤的文本熏染改善方法。最后，本文总结了研究工作并进行展望。

第一章绪论作为全文的开篇，概述了图像生成领域中文本渲染的研究背景和重要性。本章将总览并分析文本渲染所面临的挑战和现有研究的局限性，进而引出本研究旨在解决的问题和研究的创新点。

第二章介绍相关理论与研究方法。首先，本章概述文本到图像生成研究，依次介绍基于生成式对抗网络、自回归模型以及扩散模型的文本到图像生成，对其中具有代表性的研究工作进行阐述。其次，本章介绍文本到图像生成中视觉文本渲染的研究现状，归纳为基于文本编码器优化、额外控制条件以及基座模型优化的视觉文本渲染，对其中具有代表性的研究工作进行说明。

第三章介绍文本提出的基于混合专家的多属性可控文本渲染改善方法。本章将通过构建专用的合成式数据集并在此基础上训练独立的颜色专家与字体专家，再进一步引入自适应路由器，探索如何使潜空间文生图扩散模型具备颜色与字体可控的文本渲染能力。在大规模文本内容控制评估测试集以及合成式颜色与字体控制评估测试集上进行大量自动化评估与人工评估验证方法的有效性。

第四章介绍本文提出的基于推理时扩展与后过滤的文本渲染改善方法。本章将构建文本渲染量化评估框架，系统全面地从五个维度探究现有文本到图像生成模型的文本渲染能力。通过对第三章中人工评估的部分实验进行回溯性验证，证明评估框架的可靠性。基于大规模量化评估实验所得的结论，对现有模型进行推理时扩展与后过滤，从而实现架构无关的文本渲染改善。

第五章将对全文进行总结，并对未来的研究方向进行展望。本章将回顾前述章

节的主要发现和贡献，并讨论研究的局限性和未来改进的可能方向。通过总结，本文旨在为面向文本到图像生成的文本渲染提供新的视角和研究思路。

第二章　相关理论与研究方法

2.1　文本到图像生成的研究概述

文本到图像（Text-to-Image，T2I）生成[8,12-31]，旨在依据用户提供的自然语言提示（prompt），生成语义匹配的高质量图片。此处提到的自然语言提示，是指用户用自然语言形式给出的对预期图片内容的描述。这指示了文本到图像生成是一个跨模态的生成任务，需要兼顾对文本语义和图像分布的深度理解：正确解析自然语言提示中的实体、属性以及关系，将高维、抽象的概念，转换为低维、具象的像素分布。

文本到图像生成是通过生成式模型实现的，生成式模型的目标是学习真实数据的潜在分布，并基于该分布生成全新的、与真实数据相似的样本。近年来，文本到图像生成所基于的生成式模型，发展脉络整体上与无条件图像生成是一致的，即从较早的生成式对抗网络（Generative Adversarial Networks，GAN），到自回归模型（Autoregressive Models，AR），再到近期的扩散模型（Diffusion Models，DM），如图 2.1 所示。

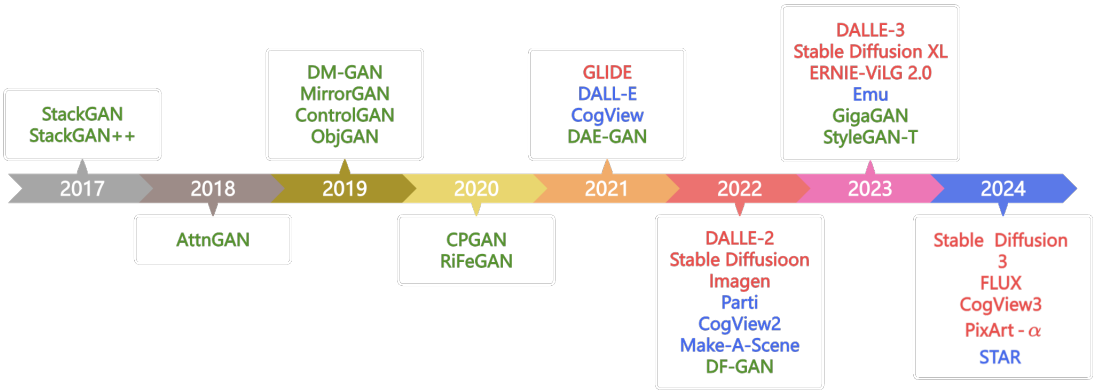


图 2.1　文本到图像生成中的一些代表性研究工作（绿色、蓝色、红色分别代表基于 GAN、基于 AR、基于 DM）

2.1.1　基于生成式对抗网络的文本到图像生成

生成式对抗网络（GAN）在 2014 年由 Ian Goodfellow 等人[32]首次提出，开创性地构建了一种同时包含生成器（Generator）和判别器（Discriminator）在内的对抗训

练框架，如图 2.2 所示。其中，生成器负责尽力由随机噪声产生逼近真实样本的生成样本，而判别器则负责尽力区分真实样本与生成样本。GAN 的训练是极小极大博弈，“极小”指的是训练判别器从而最小化判别出错的可能性，“极大”指的是训练生成器从而最大化判别出错的可能性。生成器和判别器在对抗训练中相互竞争、相互提升，直至判别器无法判断给定的样本是真实样本还是生成样本，即达到纳什均衡（Nash Equilibrium）。过程可以表示为：

$$\min_{G}\max_{D} V(D_\theta, G_\theta) = \mathbb{E}_{x\sim p_{data}(x)}\big[\log D_\theta(x)\big] + \mathbb{E}_{z\sim p_z(z)}\big[\log\big(1 - D_\theta(G_\theta(z))\big)\big] \tag{2.1}$$

其中，x 是真实样本，z 是随机噪声，$D_\theta(\cdot)$ 表示判别器，$G_\theta(\cdot)$ 表示生成器，$\mathbb{E}$ 代表期望。在此基础上，后续的许多改进工作，如著名的 StyleGAN 系列[33-35]，极大地提升了 GAN 的图片生成能力与稳定性，为 GAN 进一步在文本到图像生成中的应用奠定了基础。

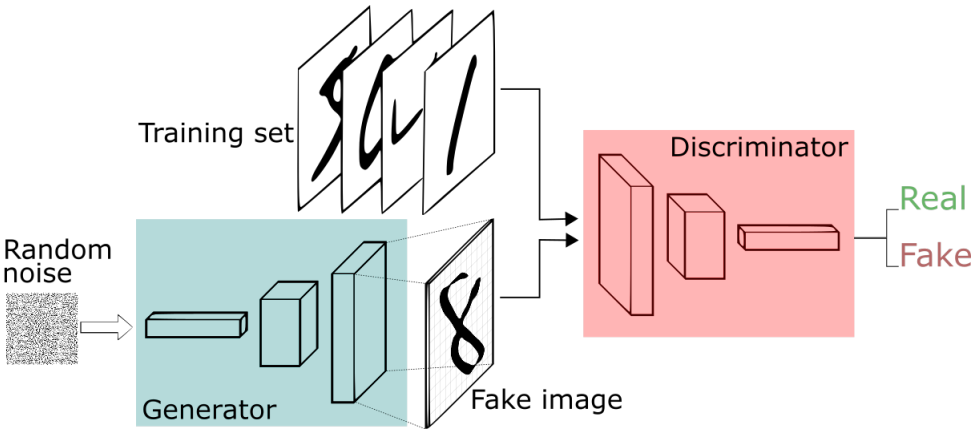


图 2.2　GAN 架构[36]

在文本到图像生成领域，GAN 曾是在扩散模型流行之前的主流基础模型。StackGAN[15]通常被视作是 GAN 在文本到图像生成领域中应用的里程碑式工作之一，其核心创新在于提出了一种由粗到细（coarse-to-fine）的两阶段生成架构（堆叠了两个 GAN 架构）：第一阶段生成低分辨率的图片，形成粗略的轮廓与色块；第二阶段在此基础上进行细化，最终生成高分辨率的图片。这种两阶段生成架构较好地提升了生成图片的分辨率与保真度，而后续的 StackGAN++[37]则进一步推动了多阶段生成架构的发展。在多阶段生成架构的基础上，为了实现更一致的图片与文本间的对齐，AttnGAN[16]首次引入了注意力机制。在这之后的时间里，GAN 被持续地改进从而更好地实现文本到图像生成：ControlGAN[38]允许在文本到图像生成的过程中对某

些视觉属性实施精细控制；DM-GAN[17]提出动态记忆机制用于解决多阶段图像生成架构中初始图片对后续细化过程的不良影响；MirrorGAN[39]采用文本到图像生成与图像到文本生成的镜像操作来优化自然语言提示与生成图片之间的匹配程度；Obj-GAN[40]引入对象级信息用于更好地生成图片；CPGAN[41]通过对自然语言提示与图片内容进行深度解析来实现更好的图文一致性；RiFeGAN[18]探索解决自然语言提示的模糊性与抽象性所造成的影响；DAE-GAN[42]通过引入所谓的方面信息来实现图文更一致的图片生成；DF-GAN[43]优化了经典多阶段图像生成架构的不足。诸如此类基于生成式对抗网络的文本到图像生成工作还有很多，直至自回归模型与扩散模型的相继出现令 GAN 渐渐淡出了主流的视线。尽管 GAN 的训练不稳定性是一个困扰许久的问题，然而 GAN 的推理耗时具有天然的优势，GAN 只需单次前向传播即可生成图片而无需迭代推理。因此，近期关于 GAN 的研究[44-47]始终没有停歇：StyleGAN-T[45]在 StyleGAN-XL[48]的基础上引入 CLIP 文本编码器，实现了速度非常快的图片生成，同时保持了一定的生成质量；GigaGAN[44]通过大规模预训练展现出了与扩散模型相当的图片生成能力，并且在推理耗时上表现突出。这些都表明 GAN 依然具备相当的研究潜力，并可能在未来的某一天再次兴起。

2.1.2 基于自回归模型的文本到图像生成

自回归模型（AR）起源于自然语言处理（NLP）领域，随着 Transformer 架构[49]的提出而逐渐大放异彩。自回归模型的概率分解公式可以形式化为：

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t \mid x_1, x_2, \ldots, x_{t-1}) \tag{2.2}$$

其中，$x_1, x_2, \ldots, x_T$ 代表一个特定序列；x_t 代表当前元素（自回归单元），x_t 只与之前的 $x_1, x_2, \ldots, x_{t-1}$ 相关。在特定数据集上基于上式进行优化，是训练自回归模型的主要思想。在图像生成领域，需要为图片构建对应的、合理的序列，从而将图像生成问题转化为序列预测问题。VQGAN[50]通常被视作是 Transformer 真正被应用于高质量图像生成的一个开创性与奠基性的研究工作。VQGAN 借鉴了 VQVAE[51]思想，在架构中同时包含 GAN 与 Transformer：利用 VQGAN 将图片压缩到一个离散、低维的潜空间中，通过 Transformer 学习该空间中的分布。这一方案为自回归模型在高分辨率图像生成中计算开销过大的问题提供了解决思路。

在文本到图像生成领域，DALL-E[19]通常被视为基于自回归模型的一个里程碑

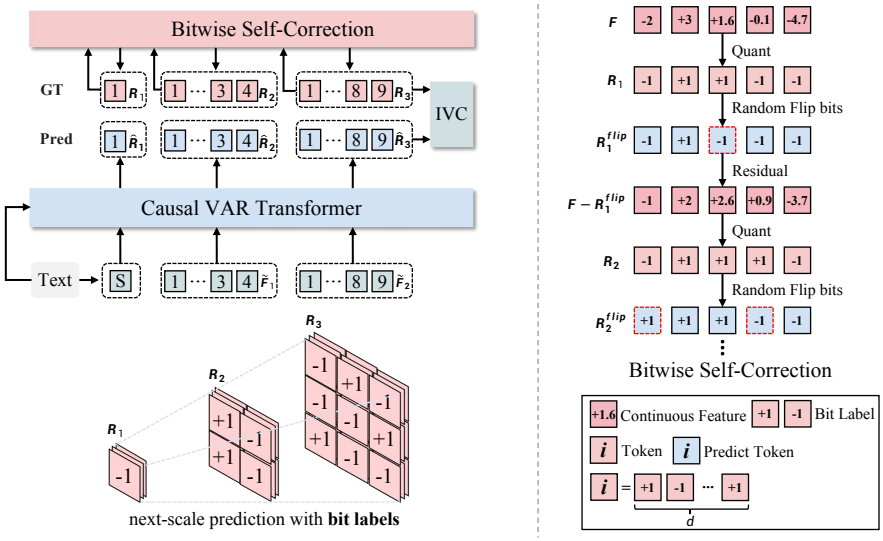


图 2.3 Infinity 的架构[52]（Infinity 是一种基于自回归模型的文本到图像生成方法）

式的工作。DALL-E 是一个两阶段的架构：首先采用一个离散化的变分自编码器（dVAE）对图片进行压缩；再对自然语言提示进行编码，进一步将两种词元（token）拼接用以训练自回归 Transformer 学习联合分布。紧随其后的 CogView[21] 有着与 DALL-E 接近的思想。CogView2[53] 则进一步对 CogView 进行了改进，提出了一种层级式 Transformer 架构和跨模态通用语言模型（CogLM），有效提升了生成速度与质量。Make-A-Scene[54] 在自然语言提示以外，引入一种场景控制机制，实现了更可控的图片生成。Parti[55] 通过对模型参数进行规模化（350M 至 20B），证明了 Scaling Law 在文本到图像生成中的有效性。在这之后的许多研究工作[56-65] 持续地探索与优化基于自回归模型的文本到图像生成。其中，STAR[61] 所基于的 VAR[11]，开创性地提出了有别于传统 “Next-Token-Prediction” 的 “Next-Scale-Prediction” 范式，有望改善自回归图像生成模型困扰已久的生成速度缓慢问题。如图 2.3 所示，VAR 直接对应的文本到图像生成模型 Infinity[52]，展现出了与一些最先进（State-of-the-Art，SOTA）的基于扩散模型的文本到图像生成相当的效果，并在图片生成速度上也具备一定优势，证明了基于自回归模型的文本到图像生成具有巨大研究潜力。

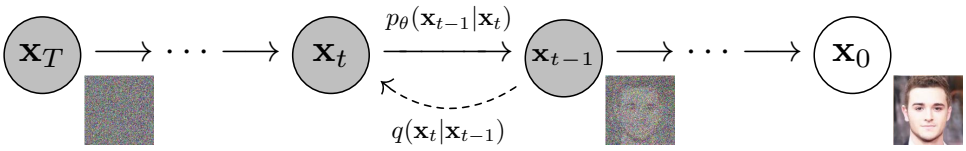


图 2.4 DDPM 示意图[1]

2.1.3 基于扩散模型的文本到图像生成

扩散模型（DM）可谓是当下最主流的图像生成模型，而 DDPM[1]的提出则通常被视为这一领域的一个里程碑式研究工作。DDPM 整体上由两部分组成：前向过程（加噪）与反向过程（去噪）。如图 2.4 所示，前向过程是从原始图片 x_0 到噪声图片 x_T 的过程，期间不断地人为加噪，中间得 $x_1, x_2, \ldots, x_{T-1}$；反向过程是从噪声图片 x_T 到原始图片 x_0 的过程，期间由模型逐步去噪，中间得 $x_{T-1}, x_{T-2}, \ldots, x_1$。对于前向过程，对时间步 $t-1$ 的噪声图片 x_{t-1} 进一步加噪，可以获得时间步 t 的噪声图片 x_t，过程表示为：

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)\mathbf{I}) \tag{2.3}$$

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \tag{2.4}$$

其中，$\alpha_t = 1 - \beta_t$，β_t 为超参数，ϵ_{t-1} 是一个标准正态分布的样本。进一步地，给定来自训练集的原始图片 x_0，对 x_{t-1} 进行倒推直至 x_0，最终可以获得：

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{2.5}$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{2.6}$$

其中，$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$，$\epsilon \sim \mathcal{N}(0, I)$。也就是说，可以直接由原始图片 x_0 求得任意时间步的噪声图片 x_t，从而完成前向过程。对于反向过程，对时间步 t 的噪声图片 x_t 去噪，获得时间步 $t-1$ 的图片 x_{t-1}，可以表示为：

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2.7}$$

其中，θ 是待学习参数。这表明训练模型根据噪声图片 x_t 和时间步 t 预测均值与方差，即可逐步去噪进而完成反向过程，实现图片生成。正是得益于这种逐步去噪的生成机制，基于扩散模型的图像生成往往具有相对较好的可控性与生成质量。之后，DDIM[66]通过构建可跳步的非马尔科夫链式反向过程，在几乎不损失生成质量的前提下，极大地提升了扩散模型生成图片的速度。

在文本到图像生成领域，GLIDE[20]通常被视作是基于扩散模型的首个研究工作。GLIDE 验证了无分类器指导（Classifier-Free Guidance，CFG）[67]在文本到图像生成中的巨大潜力，显著提升了生成图片与自然语言提示的语义一致性，并保持了较高的

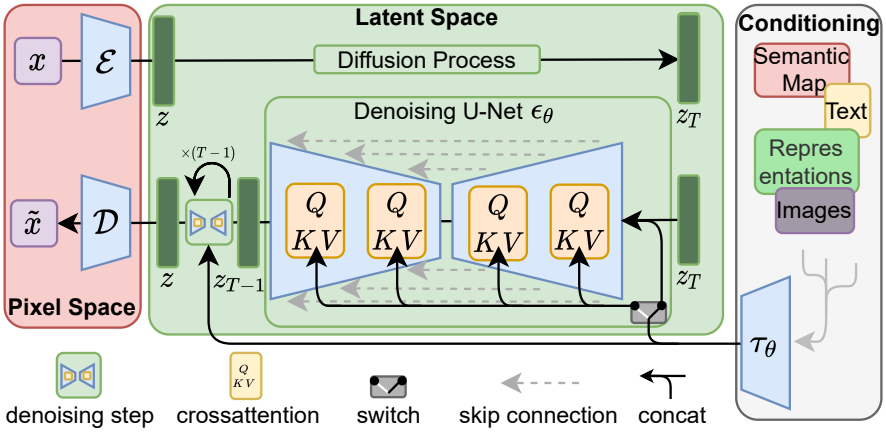


图 2.5　Stable Diffusion 架构图[8]

生成质量。简单来说，CFG 需要在模型训练时以一定的概率将控制条件置为空，让同一个模型同时学习条件生成与无条件生成，从而在推理时通过对条件生成与无条件生成线性外推获得最终的高质量生成图片。CFG 在后续的条件生成扩散模型中被广泛使用。在 GLIDE 之后，Imagen[23]通过预训练的语言模型对自然语言提示进行编码，充分利用了预训练语言模型在大型纯文本语料库中学习到的先验知识，生成了语义匹配度与保真度均较好的图片。

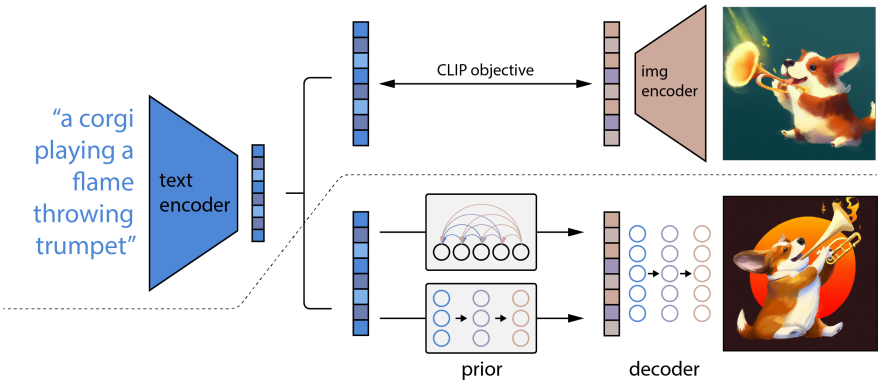
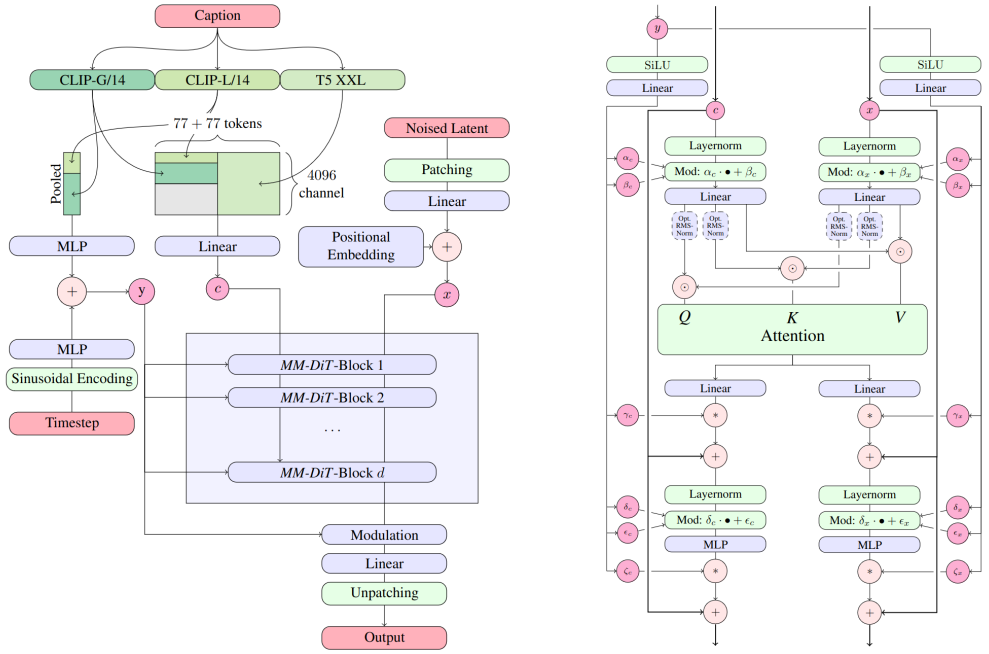


图 2.6　DALL-E 2 架构图[22]

与直接在像素空间（Pixel Space）中应用扩散模型的 GLIDE 和 Imagen 不同，Stable Diffusion[8]开创性地在潜空间（Latent Space）中执行扩散，因而被视为文本到图像生成领域中潜空间扩散模型（Latent Dffusion Model，LDM）的一个里程碑式研究工作。如图 2.5 所示，Stable Diffusion 通过一个预训练的自编码器[68]，将图片从高维的像素空间压缩到低维的潜空间中，并在潜空间中执行扩散与去噪，最终还原至像素空间。潜空间的引入，相比于像素空间，显著地降低了计算开销与资源消耗。此

外，Stable Diffusion 还引入了交叉注意力机制，使得扩散模型可以基于各种控制条件进行图片生成。这两个思路推动了基于扩散模型的文本到图像生成在学术界与工业界的飞速发展，促使了文本到图像生成技术的大规模普及与落地。类似但有所不同地，DALL-E 2[22]借助预训练的 CLIP[69]的潜空间进行基于扩散模型的文本到图像生成。如图 2.6 所示，DALL-E 2 利用 CLIP 文本编码器对自然语言提示进行编码从而获得文本嵌入（text embedding），再通过扩散模型或自回归模型将文本嵌入转换为图片嵌入（image embedding），最终通过扩散模型解码为生成图片。其中的 CLIP 文本编码器与图片编码器在之后的研究工作中被广泛地使用。



(a) Overview of all components.　　(b) One *MM-DiT* block

图 2.7　Stable Diffusion 3 架构图[9]

在 GLIDE、Imagen、Stable Diffusion 和 DALL-E 2 之后，基于扩散模型的文本到图像生成领域发展迅猛。Stable Diffusion XL[70]是 Stable Diffusion 的升级版本，通过扩展模型参数量以及更精细的训练与推理策略，生成图片质量相较于之前的版本有着明显的提升。DALL-E 3[71]通过改善训练数据中的自然语言提示，显著提升了模型对自然语言提示的遵循能力，突出强调了过往训练数据中噪声较多的问题，对之后的研究工作产生了深远的影响。ERNIE-ViLG 2.0[72]作为较早出现的基于扩散模型的大规模中文文本到图像生成模型，通过融入细粒度的文本与视觉知识和在不同去噪阶段采用不同的去噪专家，提升了生成图片的质量。PixArt-α[73]在 DiT（Diffusion

Transformer)[7]中引入了交叉注意力模块从而注入文本条件，并通过相对较低的训练开销获得了具有一定竞争力的文本到图像生成能力。CogView3[74]采用了级联的架构并创新地在文本到图像生成领域引入中继扩散（Relay Diffusion)[75]的思想，在计算资源利用率与图片生成细节上均有提升。Stable Diffusion 3[9]与先前的版本大有不同，通常被视为又一里程碑式的研究工作。如图所示，Stable Diffusion 3 在模型架构上用 DiT 架构取代了在扩散模型发展中沿用已久的 U-Net 架构，在文本编码器上同时使用了 CLIP 和 T5[76]，并且采用了流匹配（Flow Matching)[77]的思想，最终在自然语言提示遵循和视觉文本渲染等方面均有不俗的表现。后续的 FLUX[10]经常被看作是 Stable Diffusion 3 的改进版本，进一步验证了 DiT 架构的有效性，并成为了最近被较多使用的基于扩散模型的文本到图像生成开源模型。值得一提的是，Stable Diffusion 3 和 FLUX 通常被视为当下位处第一梯队的、开源的基于扩散模型的文本到图像生成模型，标志着扩散模型的研究范式正在逐渐从 U-Net 架构向 DiT 架构迁移。相比于 U-Net 架构，DiT 架构具有极佳的可扩展性，无疑会为基于扩散模型的文本到图像生成带来一次新的爆发式发展。

2.2 视觉文本渲染的研究现状

近年来，随着文本到图像生成技术的日趋成熟，诸如 Stable Diffusion[8] 和 Stable Diffusion XL[70] 等通用的文本到图像生成模型，已经在许多应用场景中展现出了卓越的泛化性与可用性。用户仅需输入简洁而明确的自然语言提示，即可生成在视觉保真度和语义一致性等方面均令人满意的动物、人物以及场景等图片内容。然而，当用户希望在生成的图片中同时呈现准确且可读的视觉文本（Visual Text）时，通用的文本到图像生成模型往往表现出相对受限的能力。

视觉文本渲染（Visual Text Rendering)[78-80]在海报、书籍封面、广告和 LOGO 等设计领域具有巨大的商业潜力与应用价值，在社交媒体和数字营销等方面也展现出日益增长的重要性。与一般的图片内容不同，视觉文本对生成偏差的容忍度相对较低：即使是字形偏差这类细微的文本错误也会被用户轻易地观察和识别到，从而影响生成图片的可用性；更不用提字符增加、缺漏和替换等直接影响连贯性与可读性的文本错误。为了缓和甚至解决诸如此类的问题，最近的许多研究工作基于通用文本到图像生成模型进行了改进，从而设计开发针对视觉文本渲染的专用文本到图像

图 2.8 视觉文本渲染示例[2]（第一行为文本内容错误示例，第二行为对应的正确示例）

生成模型。常见地，视觉文本渲染的研究目前可以分为三类：基于文本编码器优化、基于额外控制条件、基于基座模型优化。需要说明的是，在本文中“文本渲染”一词与“视觉文本渲染”将会有相同的指代，均代表文本到图像生成中的视觉文本渲染。

2.2.1 基于文本编码器优化的视觉文本渲染

在文本到图像生成中，文本编码器（Text Encoder）扮演着至关重要的角色，负责将用户提供的自然语言提示转化为文本到图像生成模型能够理解的深层语义表征，即高维的特征向量，通常也称为嵌入（Embedding）。这些语义表征在图片生成的过程中将作为控制条件（Condition），引导文本到图像生成模型最终生成符合预期的图片。例如在扩散模型中指导每一个时间步的去噪操作，从而确保生成的图片内容与自然语言提示在语义上尽可能的匹配。

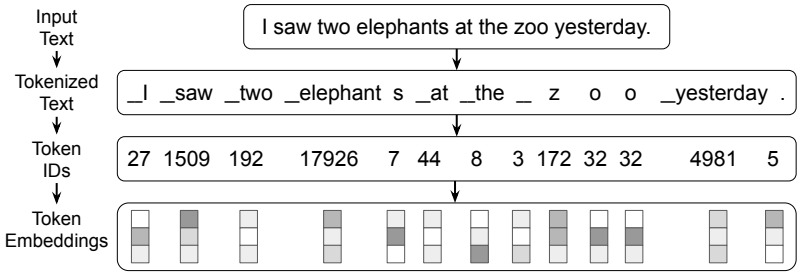


图 2.9 非字符级文本编码器涉及分词的示例[2]

以 Stable Diffusion 1.5[8]为代表的一系列较为主流的通用文本到图像生成模型，通常广泛地采用诸如 CLIP[69]和 T5[76]之类的文本编码器。这类文本编码器的设计初衷往往在于高效地捕获自然语言提示的宏观语义和概念关联，不会直接在字符级别上处理自然语言提示，而是会对自然语言提示进行分词（Tokenization）并以词元（Token）为单位进行处理（如图 2.9 所示），因而可以被认为是非字符级的（Character-Blind）文本编码器。当目标生成图片仅仅涉及人物、动物或场景等常规视觉元素（例如“白色的猫”）时，非字符级的文本编码器通常已经足以为图像生成模型提供稳健且周到的语义表征，从而生成与自然语言提示在语义上高度匹配的图片。然而，当自然语言提示要求目标生成图片中需要同时呈现清晰且可读的视觉文本（例如“WHITE CAT”）时，非字符级的文本编码器潜在地难以提供目标视觉文本的字符级特征，这被视为是通用文本到图像生成模型无法较好地进行视觉文本渲染的因素之一。

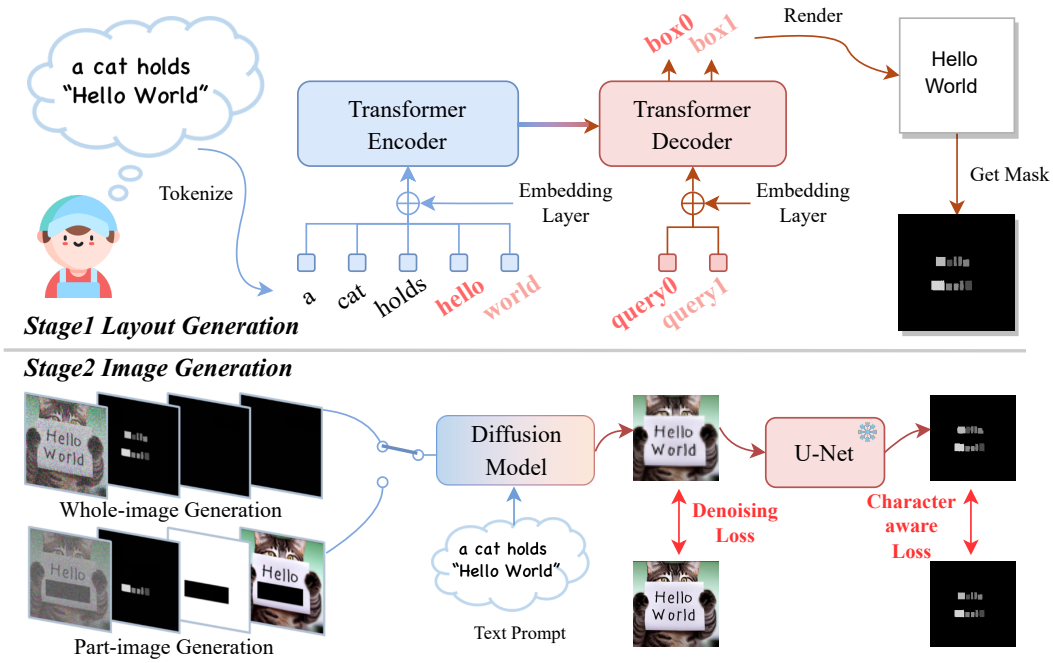
基于此，许多研究工作尝试对文本编码器进行优化，从而改善文本到图像生成模型的视觉文本渲染能力。在视觉文本渲染的早期研究中，Liu 等人[2]证明了采用诸如 ByT5[81]这类字符级（Character-Aware）文本编码器，可以显著提升文本到图像生成中视觉文本渲染的文本内容准确性；然而，完全采用字符级特征又会降低常规图片内容生成时的图文一致性；结合字符级特征和词元级特征则可以较好地达到折中双赢的效果。此外，这项研究工作中还提到了所谓的“拼写奇迹”，即参数规模庞大的非字符级文本编码器同样也能一定程度上进行准确拼写。UDiffText[82]构建了一个字符级文本编码器专用于获取目标视觉文本的稳健的嵌入，从而实现高质量的视觉文本渲染。Glyph-ByT5[83]使用精心构建的字形图片-文本数据集，对字符级文本编码器 ByT5 进行微调，从而弥合字形图片与其对应的文本内容之间的鸿沟；提出了一种区域级交叉注意力机制，将微调后的文本编码器整合到扩散模型中。Glyph-ByT5-v2[84]进一步将前一版本扩展至多语言视觉文本渲染，并更强调生成图片的美感。SceneTextGen[85]通过常规的 CLIP 文本编码器对自然语言提示进行编码，利用额外的字符级文本编码器对目标视觉文本进行编码，通过交叉注意力机制融入到基座扩散模型中，从而改善视觉文本渲染质量。

2.2.2 基于额外控制条件的视觉文本渲染

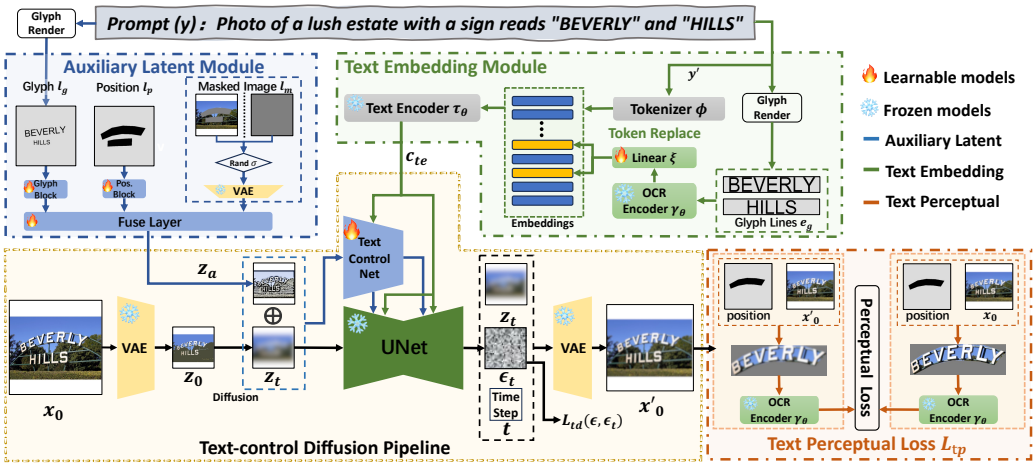
随着文本到图像生成的飞速发展，越来越多的基座模型[8-9,70]能够忠实地遵循用户指定的自然语言提示，生成语义高度一致的图片，并且保持相当高的视觉质量。这

自然而然地促使人们追求可控性更强的文本到图像生成方法[86]，从而满足复杂多样的实际需求。然而，人们逐渐发现自然语言并不容易完全周到细致地描述图片中的所有细节，例如结构、风格以及某些特定的视觉属性。为了克服自然语言的这一固有限制，越来越多的研究工作涌现了出来，尝试在自然语言提示这一单一的控制条件之外，为文本到图像生成模型引入额外的控制条件，从而促成更可控的图像生成。这类研究工作通常都被广义地归于条件生成（Conditional Generation）的范畴。由于近期占主导地位的图像生成方法往往都是基于扩散模型的，所以最近的这类研究也大多是面向扩散模型的，其中不乏一些较为瞩目的工作。ControlNet[87]构建了一个开创性的架构，冻结作为基座的扩散模型的参数，拷贝基座扩散模型中 U-Net 架构的编码器部分并设置为可训练，通过提出的零卷积（Zero Convolution）对这两部分进行整合，从而在学习引入额外控制条件时有效地避免灾难性遗忘与过拟合。ControlNet 架构广泛地被后续针对不同生成任务的可控图像生成研究所采用，充分展现了其普适性与有效性。T2I-Adapter[88]通过可训练的适配器，将外部的额外控制条件整合进基座扩散模型中，用于控制图片的生成过程。Composer[89]强调了控制条件的可组合性，从而实现兼具创意与可控性的图像生成。IP-Adapter[90]通过解耦的交叉注意力机制，将作为额外控制条件的参考图片特征注入冻结的基座扩散模型中，从而达成更可控的图像生成。诸如这样的研究工作还有很多，推动并发展了越来越可控的文本到图像生成。

在视觉文本渲染中，受可控文本到图像生成的启发，许多研究工作尝试引入额外的控制条件，从而改善文本到图像生成模型的视觉文本渲染能力。这类方法通常冻结基座扩散模型，保留其经大规模预训练而获得的先验知识，在此基础上通过可训练模块向基座扩散模型的 U-Net 架构注入额外的、与视觉文本渲染相关的控制条件，从而以相对较低的训练成本，控制模型在生成图片中嵌入正确可读的视觉文本，并保持一定的泛化能力。GlyphDraw[3]将字形图片和字符掩码作为基座扩散模型额外的输入，并将由字形图片编码所得的字形嵌入与由自然语言提示编码所得的文本嵌入进行融合，替换原始的文本条件，从而在生成图片中有效地嵌入视觉文本。GlyphControl[4]用字形图片承载文本字符、文本行以及文本框信息，并借助 ControlNet 架构将这些信息注入到扩散模型中。TextDiffuser[6]设计了一个两阶段的方法，增加了一个布局生成模块，用于获取字符级分割掩码作为扩散模型的额外输入，并引入了字符感知损失，用于监督训练模型正确地渲染视觉文本。AnyText[5]利用字形图片、位置图片和



(a) TextDiffuser 架构图[6]



(b) AnyText 架构图[5]

图 2.10 基于额外控制条件的视觉文本渲染的两个代表性工作

掩码图片作为辅助条件来产生潜在特征图，并在训练过程中引入了文本损失感知，用于监督模型渲染文本的清晰度和完整性。AnyText 探索了多语言的视觉文本渲染，可以一定程度上渲染中文文本。Diff-Text[91]将目标视觉文本的草图图片和边缘图片作为额外的控制条件，并提出了局部注意力约束机制，用于将视觉文本以合理的方式嵌入在生成图片中。Diff-Text 是一个适用于多语言且无需训练的视觉文本渲染架构。CustomText[92]将视觉文本的字体属性引入 TextDiffuser 的字符掩码中，并提出用于字符矫正的一致性解码器。CustomText 是较早且为数不多的考虑到视觉文本属性控制的研究工作之一。GlyphDraw2[93]将字形图片作为额外的控制条件并执行精细设计的

特征注入操作。AnyText2[94]通过类似 ControlNet 的、名为 WriteNet 的模块，同时注入目标视觉文本的字形、位置、字体和颜色特征，并利用提出的 AttnXLayers 将联合文本特征与图片内容进行融合。ControlText[95]引入了像素级文本分割模型，用于在 AnyText 的基础上引入视觉文本的字体特征。AnyText2 和 ControlText 均属于当前少数能同时兼顾文本内容控制与文本属性（颜色、字体）控制的研究工作，可视作与本研究并行的同期工作。

2.2.3 基于基座模型优化的视觉文本渲染

在文本到图像生成的视觉文本渲染中，现有的研究工作大多聚焦在前述的“基于文本编码器优化”和“基于额外控制条件”这两个研究路线上。理想地，直接对基座模型进行优化，使其自身具备良好的文本渲染能力，也是一个潜在的研究方向。随着专为视觉文本渲染而设计的大规模、高质量数据集的出现，微调基座扩散模型从而实现视觉文本渲染已成为可能。如图 2.11 所示，TextDiffuser-2[96]首先微调了一个大语言模型（Large Language Model，LLM），用于执行目标视觉文本的布局规划，从而获取经处理的、纯文本形式的提示。具体来说，该提示在形式上由原始自然语言提示、目标视觉文本拆分后的字符序列（包含空格）、目标视觉文本对应文本框的左上角与右下角相对坐标，以及特定的分隔标记构成。在这之后，TextDiffuser-2 在其构建的专用数据集 MARIO-10M 上微调基座扩散模型的文本编码器与 U-Net，从而实现灵活且可靠的视觉文本渲染。整个过程中，基座扩散模型的原始网络结构保持不变，潜在地可以视作迭代为了自身具备文本渲染能力的、新的基座模型，为后续引入其他控制条件或任务扩展预留了充足的灵活空间。

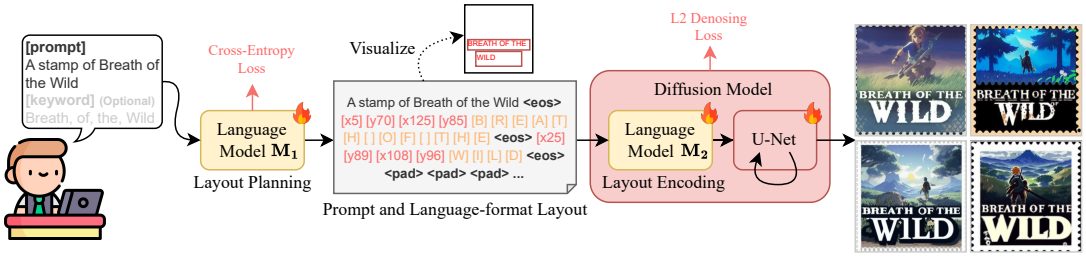


图 2.11 TextDiffuser-2 架构图[96]

第三章　基于混合专家的多属性可控文本渲染改善方法

3.1　研究动机

在包含视觉文本（Visual Text）的图片中，文本内容（Text Content）直接地以字符组合排列的方式传达语义信息，而文本的视觉属性（Visual Attributes），如颜色（Color）和字体（Font），通过感知强化机制，可以拓展文本内容的语义边界，为字面含义隐式地增加情感倾向、社会规约或场景意图等。在这种加持下，简洁的文本内容无需过长的时间来阅读，却能传达更丰富的语义信息。以图 3.1 中所示的停车让行标志为例，“停”的文本内容直接指示行为约束，而红色背景下白色与无衬线加粗字体的组合，通过颜色心理学中的高唤醒度效应和字体设计学中的权威性表征，间接强化了“停止、禁止和限制”等严令遵守的隐含语义。这个例子表明，图片背景、文本内容、文本颜色和文本字体这四者之间存在着紧密的相互关系，共同作用于更全面和高效的信息传达。这种交互作用在标志设计、海报设计和广告设计等领域具有重要的意义。因此，对图片背景、文本内容、文本颜色和文本字体的解耦控制研究具有重要的应用价值。

视觉属性可控的文本渲染任务，广义地可以分为三类：**1）视觉属性可控的风格文本生成**，这类任务不考虑复杂的图片背景，通常针对单一颜色的背景，强调将文本内容以可控的视觉属性进行呈现；**2）视觉属性可控的场景文本编辑**，这类任务需要考虑复杂的图片背景，专注于将原始图片中的文本编辑为视觉属性可控的指定文本，同时维持原有的图片背景不变；**3）视觉属性可控的场景文本生成**，这类任务同样需要考虑图片背景，旨在渲染视觉属性可控的文本，同时生成语义可控的图片背景。其中，第三类为本章的主要关注点。

在场景文本生成领域，最近的工作[3-6,96]基于 U-Net 架构的潜空间扩散模型取得

图 3.1　视觉元素共同作用从而高效传达信息的示例

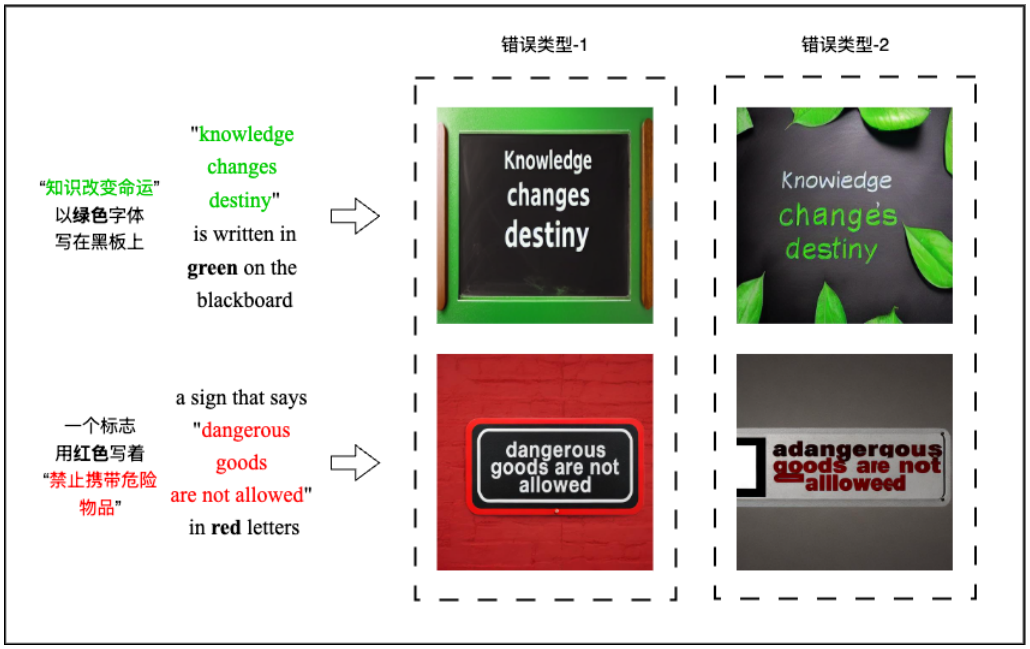


图 3.2　场景文本生成中视觉文本颜色控制不准确的案例

了显著的进展，尤其是在文本内容控制、文本布局控制、多语言文本内容控制等方面。这些工作大多采用专门的字形条件（Glyph Condition）来确保图片中视觉文本内容的准确呈现，利用通用的文本条件（Textual Condition）来控制图片背景和文本的视觉属性。其中，文本条件通常也可以被称作文本提示（Textual Prompt）或自然语言提示（Natural Language Prompt）。然而，这些工作在视觉文本的颜色和字体控制上都面临挑战，表现为有限的颜色控制能力和更少被关注的字体控制能力。关于视觉文本颜色控制，本章首先对具有代表性的场景文本生成方法[4,96]进行了定性探究。为了简洁明了，图 3.2 中仅展示与颜色控制相关的文本条件，忽略与视觉文本内容控制相关的额外条件。通过对生成图片进行定性分析可以发现，指定颜色在图片中均得到了呈现，这表明模型对颜色概念具有良好的理解能力。然而，可以观察到两类视觉文本颜色控制不准确的现象：**1）指定颜色出现在图片背景的像素区域，而非视觉文本的像素区域**；**2）指定颜色出现在视觉文本的局部像素区域，而非视觉文本的全部像素区域**。这两类现象分别提示了两个潜在问题：1）通过文本提示的方式同时指定图片背景与文本颜色，可能导致语义空间中属性与实体的绑定混淆；2）即使颜色属性与视觉文本实体绑定正确，控制力度分配不佳也可能导致颜色未能准确应用到整个文本区域。

基于上述问题，本章首先使用解耦的的方式为主干架构注入视觉文本颜色控制条件，从而规避不同控制条件在语义空间中耦合而导致的属性与实体绑定混淆问题；

其次，鉴于现有模型通常没有字体控制能力，且字体属性与颜色属性共同从属于视觉文本实体，本章使用类似的方式为主干架构注入字体控制条件；最后，本章通过额外的、可解释的方式动态协调各种控制条件的控制力度，从而规避颜色与字体控制力度分配不佳的问题。此外，现有文本渲染数据集缺乏对文本颜色和字体的精确标注，本章为此专门构建了合成式数据集。通过这些措施，本研究改善了文本到图像场景文本生成领域的文本颜色和字体控制问题。在这过程中，本研究探究了不同的颜色和字体表示方式间的差异，以及文本颜色与字体控制条件的注入对主干架构原始文本内容控制能力的影响，为未来构建更精确强大的属性可控的文本到图像场景文本生成模型提供参考。

3.2 提出方法

为了改善文本到图像的场景文本生成（Text-to-Image Scene Text Generation）领域中现有方法无法精确控制文本颜色和字体的不足，本研究受集成学习方法（Ensemble Learning）和混合专家架构（Mixture-of-Experts, MoE）的启发，提出了一种新颖的文本颜色和字体控制增强方法，模型架构如图 3.3 所示。

本研究构建了独立的颜色专家（Color Expert）和字体专家（Font Expert），通过解耦的方式将颜色和字体控制条件分别注入主干架构，从而实现视觉文本颜色和字体控制，并且不与图片内容和视觉文本内容互相干扰。为了术语统一，本研究将主干架构中负责图片内容和视觉文本内容控制的模块称为文本专家（Text Expert）。额外地，本研究构建了一个自适应路由器（Adaptive Router），根据不同的输入，动态协调各个专家之间的协作，从而改善文本颜色或字体的控制力度分配不佳的问题，提供更精确的颜色和字体控制。

本节首先介绍合成式数据集的收集过程，然后依次阐述颜色专家、字体专家及自适应路由器的构建细节。

3.2.1 合成式数据集

在文本到图像的场景文本生成领域中，现有的视觉文本渲染数据集[5-6] 主要包括图片、图片的说明文字（Caption）和图片中视觉文本的 OCR 标注，图片的说明文字通常也被称为文本提示（Text Prompt）。然而，在这些数据集中但并非每个样本都包含精确且一致的文本颜色标注，更不用提难以用自然语言描述的字体标注。因此，

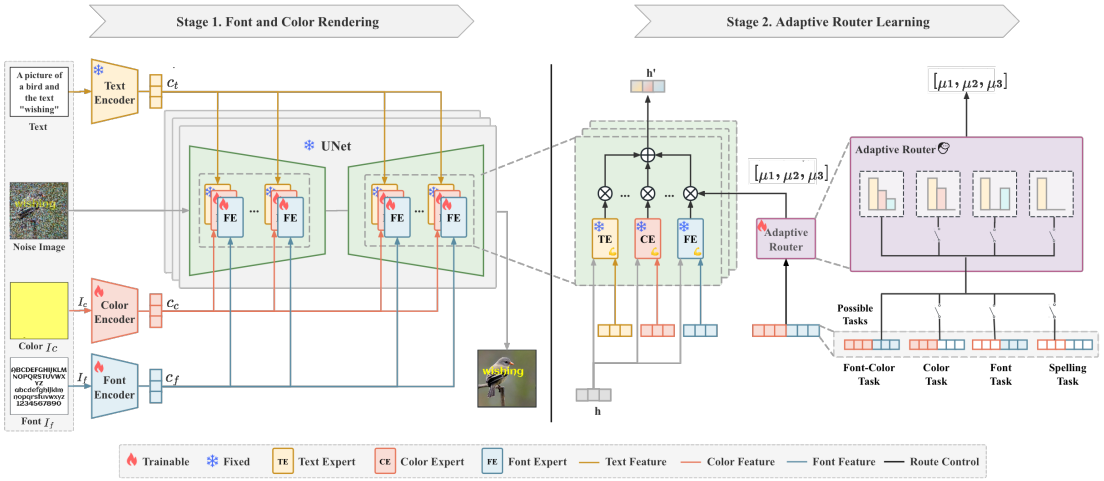


图 3.3 提出方法（FC-Render）的整体框架图

虽然这些数据集能够高效地支持模型学习对视觉文本内容和位置的控制，但不足以有效地为模型构建文本颜色和字体的控制能力。考虑到从真实图片中获取精确且一致的文本颜色和字体标注的难度，本研究选择构建合成式数据集，用于更好地提升模型对文本颜色和字体的控制能力，构建步骤与细节如下：

（1）单词、字体、颜色候选集

本研究收集了一个根据使用频率排序的候选词列表 $\mathcal{W} = [w_1, w_2, ..., w_{20,000}]$、一个常用谷歌字体的候选集合 $\mathcal{F} = \{f_1, f_2, ..., f_{20}\}$ 和一个常见颜色的候选集合 $\mathcal{C} = \{c_1, c_2, ..., c_7\}$。在字体选择上，遵循常用且风格鲜明的原则；在颜色选择上，则参考了光学三原色、印刷三原色及其对应的间色。

（2）背景图片候选集

本研究采用面向通用文生图任务的潜空间扩散模型 Stable Diffusion 1.5[8]和严谨的过滤机制来构建背景图片候选集 $\mathcal{I}_{bg} = \text{Filter}(\text{Generator}(\mathcal{T}))$。为此，本研究首先为背景图片构建了一个规模为 60 的文本提示集合 $\mathcal{T}_{bg}$。其中，每一个文本提示 t_{bg} 都遵循“A picture/photo of a/an/the [OBJECT]”模版，“picture/photo”是随机的，而“a/an/the”则会根据 [OBJECT] 进行正确选择。关于 [OBJECT] 候选集的构建，本研究部分参考了 MS COCO[97]的类别名称并挑选了一些常见动物名称、食物名称以及漫画人物名称，挑选的原则为：该物体名称不得导致生成的背景图片中存在意料之外的视觉文本元素，因此诸如“book”和“stop sign”等物品名称被严格排除在外。基于每一个文本提示 t_{bg}，本研究利用 Stable Diffusion 1.5 进行 100 次图片采样生成，因此获得规模为 6000 的图片集合 $\mathcal{I}$。最后，本研究对 $\mathcal{I}$ 进行了严格的 OCR 初筛和人工复筛，获

得规模为 5863 的背景图片候选集 $\mathcal{I}_{bg}$。其中，人工复筛环节共有 6 名在校研究生参与，每名参与者均充分知晓复筛标准。需要解释的是，由于某些生成的图片背景中存在细小的或不可读的类视觉文本元素，这部分背景图片难以通过 OCR 工具[①]进行过滤，因此人工复筛进一步确保了数据集中所有背景图片均不包含意料之外的视觉文本元素，减少了数据集在后续使用过程中的噪声干扰。

（3）合成式样本

本研究基于上述单词、字体、颜色和背景图片候选集，逐一合成数据集中的样本 $sample = (i_s, p_t, p_c, p_f, p_p)$，其中 i_s 为包含视觉文本的合成图片，p_t、p_c、p_f 和 p_p 分别为该合成图片对应的文本提示和其中视觉文本对应的颜色、字体以及位置提示。首先，本研究按使用频率从高到低的顺序选中一个单词 $w \in \mathcal{W}$，随机选中一个字体 $f \in \mathcal{F}$、一个颜色 $c \in \mathcal{C}$ 和一张背景图片 $i_{bg} \in \mathcal{I}_{bg}$。对于随机选中的背景图片 i_{bg}，取其构建时所用的文本提示 t_{bg} 备用。

本研究基于图片处理工具 PIL[②]，在一张透明背景图片 i_n 的随机位置上，根据字体 f 和颜色 c，将单词 w 渲染为视觉文本。透明背景图片 i_n 的尺寸与前述背景图片 i_{bg} 完全一致。其中，视觉文本的字符大小、字符间距和所在位置由单词 w 的字符数决定，从而确保视觉文本不超出图片范围而导致截断。本研究以文本边界框左上角和右下角的形式，记录视觉文本在图片中的位置，获得当前样本的视觉文本位置提示 p_p。接着，将包含视觉文本的透明背景图片 i_n 覆盖于随机选中的背景图片 i_{bg} 上，获得当前样本所需的包含视觉文本的合成图片 i_s。为了获得合成图片的文本提示 p_t，遵循 “A picture/photo of a/an/the [OBJECT] and/with the text [WORD]” 模版，本研究组合了背景图片 i_{bg} 对应的文本提示 t_{bg} 和单词 w。至此，获得了当前样本所需的合成图片 i_s、合成图片的文本提示 p_t 以及视觉文本的位置提示 p_p。值得一提的是，随机选中的文本颜色和文本位置偶尔会使视觉文本与图片背景的边界略显模糊，但本研究认为适当地保留这类样本可以提升模型的鲁棒性。通过随机采样和人工检查，可以判定这类样本的占比约为 2%，判定标准为无法在 3 秒内辨识合成图片中的视觉文本内容，人工检查的参与者与先前保持一致。

对于视觉文本的颜色提示 p_c 和字体提示 p_f，本研究采用了多种表示方式，包括图片（Image）标注、类别 ID（Classification ID）标注和文本描述（Textual Description）

① https://github.com/JaidedAI/EasyOCR
② https://python-pillow.github.io/

标注。这一探索有助于分析不同表示方式对视觉文本颜色和字体控制能力的影响。对于颜色提示 p_c，本研究首先构建了对应的图片标注，图片的每个像素值为该颜色的 RGB 值。接着，本研究为该颜色分配一个整数值作为类别 ID 标注。最后，采用颜色名称作为文本描述标注。对于字体提示 p_f，类别 ID 标注的构建方式同上。由于字体名称在日常使用过程中的不一致性与复杂性，本研究舍弃了对字体提示 p_f 进行文本描述标注。至于字体提示 p_f 的图片标注，本研究采用图片处理工具 PIL 和该字体对应的 TTF（TrueTypeFont）文件，将全部的大小写英文字符与阿拉伯数字渲染于单张图片中，作为字体提示 p_f 的图片标注。至此，完成对当前样本的完整构建。注意，本研究所采用的颜色提示 p_c 和字体提示 p_f 的标注方式，在文本内容、文本位置、文本颜色和字体这三个维度是互不干扰的，这意味着仅仅通过颜色提示 p_c 并不能推断文本内容、文本位置和字体，仅仅通过字体提示 p_f 也不能推断文本内容、文本位置和颜色。这有助于本研究为主干架构独立地引入视觉文本的颜色或字体控制，同时较少地影响其原生的文本内容和文本位置控制能力。

3.2.2 基于颜色专家的文本颜色控制方法

为了控制视觉文本的颜色，现有的文本到图像场景文本生成方法通常将颜色控制条件嵌入到图片描述中，例如使用类似 “A pillow with the purple word ‘health’ on it” 的文本提示，也即通过全局的控制条件同时指定图片内容与视觉文本的颜色属性。然而，观察表明，这种方法在某些情况下可能导致颜色属性与实体之间的绑定混淆。例如，在这个案例中，生成的图片可能表现出 “紫色的枕头” 而非 “紫色的文本”。其潜在原因在于，模型中的相应控制模块不足以根据全局的控制条件提供解耦的控制能力，导致文本颜色控制错误地作用于图片中其他实体上。另外，原始训练数据中的标注噪声也可能是一个因素。

受到 MoE 架构设计思想的启发，本研究提出一种解耦的视觉文本颜色控制方法，基于颜色专家（Color Expert）为文本颜色属性构建了独立的控制路径，并使用先前收集的带有精确且一致的文本颜色标注的合成式数据集进行训练。通过这种方法，本研究显式地改善了文本颜色属性与不相关实体的错误绑定问题。

（1）颜色表示

本研究探索了三种文本颜色表示方式：颜色图片、颜色类别 ID 和颜色名称。

对于经过标准化等预处理的颜色图片 X_{ci}，本研究采用了预训练的 CLIP 图像编

码器进行编码，过程可以表示为：

$$E_{ci} = \text{ClipImageEncoder}(X_{ci}) \tag{3.1}$$

其中 $E_{ci} \in \mathbb{R}^{B \times D_{ci}}$ 为编码后的颜色图片表示，B 为批次大小，D_{ci} 为 CLIP 图片编码器的输出特征纬度。

对于经过分词等预处理的颜色名称 X_{ct}，本研究采用了预训练的 CLIP 文本编码器进行编码，过程可以表示为：

$$E_{ct} = \text{ClipTextEncoder}(X_{ct}) \tag{3.2}$$

其中 $E_{ct} \in \mathbb{R}^{B \times N_t \times D_{ct}}$ 为编码后的颜色名称表示，B 为批次大小，N_t 为序列长度，D_{ct} 为 CLIP 文本编码器的输出特征纬度。鉴于这两个编码器均在大规模数据集上经过充分预训练从而压缩了大量数据知识，本研究认为它们能够直接提供高质量的颜色图片和颜色名称表示，因此没有对它们进行额外的微调。

对于颜色类别 ID 输入 X_{cid}，本研究采用了可学习的颜色类别 ID 嵌入层进行编码，过程可以表示为：

$$E_{cid} = \text{ColorEmbeddingLayer}(X_{cid}) \tag{3.3}$$

其中 $E_{cid} \in \mathbb{R}^{B \times D_{cid}}$ 为编码后的颜色类别 ID 表示，D_{cid} 为颜色类别 ID 嵌入层的输出特征纬度。

由于 E_{ci}、E_{ct} 和 E_{cid} 分别具有不同的嵌入格式，为了统一并更好地适应后续的颜色专家模块，本研究将 E_{ci}、E_{ct} 和 E_{cid} 调整为统一的嵌入格式并进行层归一化，即得到 $F_{ci} \in \mathbb{R}^{B \times N_c \times D_c}$、$F_{ct} \in \mathbb{R}^{B \times N_c \times D_c}$ 和 $F_{cid} \in \mathbb{R}^{B \times N_c \times D_c}$，过程可以表示为：

$$F_{ci} = LN_{ci}(E_{ci}W_{ci}) \tag{3.4}$$

$$F_{ct} = LN_{ct}(E_{ct}W_{ct}) \tag{3.5}$$

$$F_{cid} = LN_{cid}(E_{cid}W_{cid}) \tag{3.6}$$

其中 W_{ci}、W_{ct} 和 W_{cid} 分别为三个可学习的全连接层权重，$LN_{ci}(\cdot)$、$LN_{ct}(\cdot)$ 和 $LN_{cid}(\cdot)$ 分别为三个可学习的层归一化模块。

（2）颜色专家

现有的方法通常将文本颜色控制条件嵌入在自然语言形式的图片描述中，模型中相应的控制模块在控制生成图片的全局语义的过程中，同时尝试控制生成图片中视觉文本的颜色。然而，在某些情况下，文本颜色属性在全局图片语义中会发生泄漏或被干扰，导致文本颜色属性错误地与图片中其他实体绑定。

基于上述的 F_{ci}、F_{ct} 和 F_{cid} 这三种来源于不同表示方式的文本颜色控制条件，本研究构建了颜色专家（Color Expert），为主干架构以解耦的方式引入文本颜色控制，以此来改善文本颜色属性与生成图片中其他实体错误绑定的问题。需要解释的是，颜色专家实则只需要 F_{ci}、F_{ct} 和 F_{cid} 中的任意一种文本颜色控制条件作为输入。本研究为这三种不同来源的文本颜色控制条件分别构建了三个对应的颜色专家，这是为了探究不同的表示方式对文本颜色控制效果的影响。得益于这三种文本颜色控制条件在上一个环节中已被调整为统一的嵌入格式，本研究得以用完全一致的方式构建三个对应的颜色专家。为了表述清晰，将多种文本颜色控制条件统一记作 $F_c \in \{F_{ci}, F_{ct}, F_{cid}\}$。任意一个颜色专家处理其对应的文本颜色控制条件的过程如下：

$$H_c = \text{Softmax}(\frac{Q^{'} K_c^T}{\sqrt{d}})V_c \tag{3.7}$$

其中 $Q^{'} = H^{'} W_h^Q, K_c = F_c W_c^K, V_c = F_c W_c^V$ 分别为注意力机制中的查询（Query）、键（Key）和值（Value）；W_c^K 和 W_c^V 为可学习的参数矩阵，W_h^Q 为主干架构中被冻结的参数矩阵；$H^{'}$ 为主干架构中来自上一个阶段的携带图片语义信息的隐藏状态，H_c 为进一步注入了文本颜色控制条件的新的隐藏状态。

（3）专家协作

在文本颜色可控的场景文本生成任务中，全局图片内容、视觉文本内容和文本颜色需要被同时控制。因此，本研究所构建的颜色专家需要和主干架构中负责控制全局图片内容和视觉文本内容的模块进行协作。为了术语统一，本研究将主干架构中的相应模块称为文本专家（Text Expert），其具有类似的处理控制条件的过程：

$$H_t^{'} = \text{Softmax}(\frac{Q^{'} K_t^T}{\sqrt{d}})V_t \tag{3.8}$$

其中 $H_t^{'}$ 为融合了图片语义信息、全局图片内容控制条件和视觉文本内容控制条件的隐藏状态。$K_t = F_t W_t^K, V_t = F_t W_t^V$ 分别为文本专家对应的键和值，F_t 为全局图片内容和视觉文本内容的控制条件，W_t^K 和 W_t^V 为主干架构中被冻结的参数矩阵；$Q^{'}$

为文本专家与颜色专家共享的查询。

基于颜色专家和文本专家各自处理所得的隐藏状态，为了融合各种控制条件，两个专家的协作可以表示为：

$$H_{\text{color-control}} = H_t^{'} + H_c \tag{3.9}$$

其中 $H_{\text{color-control}}$ 为融合了图片语义信息、文本颜色、全局图片内容和文本内容控制条件的隐藏状态。

（4）训练策略

基于先前收集的带有文本颜色标注的合成式数据集，本研究对颜色专家及对应的颜色编码器进行训练优化。期间，文本专家及主干架构中的其余部分被冻结以提供基础的图片内容和视觉文本内容控制能力。训练所采用的损失函数可以表示为如下形式：

$$\mathcal{L}_{\text{color-denoising}} = \|\epsilon - \epsilon_\theta(X_T, F_t, F_c, T)\|_2^2 \tag{3.10}$$

其中 ϵ 和 ϵ_θ 分别为采样噪声和预测噪声，$T \in [0, T_{\max}]$ 为时间步，X_T 是原始图片根据时间步 T 加噪后的特征表示。

3.2.3 基于字体专家的文本字体控制方法

现有的场景文本生成方法通常可以将文本颜色控制条件以颜色名称的形式嵌入在全局图像描述中，从而实现有限的文本颜色控制。然而，字体属性比颜色属性更难通过自然语言进行精确且一致的描述，导致现有方法在字体控制方面的表现更为受限。因此，本研究进一步为主干架构引入视觉文本字体控制能力。

鉴于字体与颜色均为文本的视觉属性，两者潜在地存在共性，本研究采用类似的方式构建字体专家（Font Expert），以实现对字体的控制。字体专家与文本专家的协作过程表述如下：

$$H_f = \text{Softmax}(\frac{Q^{''} K_f^T}{\sqrt{d}}) V_f \tag{3.11}$$

$$H_{\text{font-control}} = H_t^{''} + H_f \tag{3.12}$$

其中 $H_{\text{font-control}}$ 为融合了图片语义信息、文本字体、全局图片内容和文本内容控制条件的隐藏状态。H_f 为字体专家的输出，$K_f = F_f W_f^K, V_f = F_f W_f^V$ 分别为字体专家

对应的键和值，W_f^K 和 W_f^V 为可学习的参数矩阵。$F_f \in \{F_{fi}, F_{fid}\}$ 为统一标识，F_{fi} 和 F_{fid} 分别是来源于字体图片和字体类别 ID 的字体控制条件。H_t'' 为文本专家的输出，其计算方式与公式 3.8 基本相同，因此省略了详细推导。唯一的区别在于，文本专家此时与字体专家共享查询 $Q'' = H''W_h^Q$，而文本专家在此之前与颜色专家共享查询 $Q' = H'W_h^Q$。

在字体专家及对应字体编码器的训练过程中，文本专家及主干架构中其余部分均保持冻结，采用的损失函数为：

$$\mathcal{L}_{\text{font-denoising}} = \|\epsilon - \epsilon_\theta(X_T, F_t, F_f, T)\|_2^2 \tag{3.13}$$

在此过程中，先前训练得到的颜色专家及颜色编码器被移除，实现了严格的参数隔离（Parameter Isolation）。通过这种分阶段独立训练的策略，本研究旨在明确划分不同专家的能力边界，从而为主干架构引入解耦的文本字体和颜色控制能力。

3.2.4 基于启发式规则的多专家协作机制

通过颜色专家和字体专家，以及主干架构中原始的文本专家，本研究在推理阶段实现了对视觉文本颜色或字体属性的独立控制，即控制文本颜色的同时保持字体的多样性，反之亦然。这可以表述为：

$$H_{\text{controlled}} = \begin{cases} H_t, & \text{如果仅控制文本内容} \\ H_t' + \alpha H_c, & \text{如果控制文本内容和颜色} \\ H_t'' + \beta H_f, & \text{如果控制文本内容和字体} \end{cases} \tag{3.14}$$

其中 $\alpha, \beta \in [0, 1]$ 分别为颜色和字体控制的强度系数，其具体数值的选择是基于启发式（Heuristic）规则的，即依据人工经验进行估计。在仅控制文本内容的场景下，H_t 的计算方式与 H_t' 或 H_t'' 基本相同，区别在于此时文本专家独享查询 $Q = HW_h^Q$，不与颜色或字体专家共享。需要说明的是，文本专家实际上同时控制了全局图片内容和视觉文本内容，并且在文本颜色或字体可控的场景中始终参与其中。因此，在本章的所有场景中，全局图片内容均是可控的。鉴于本章重点关注的是视觉文本及其属性，为简化表述，省略了对“全局图片内容可控”这一特性的详细讨论。

值得注意的是，尽管颜色专家和字体专家是分阶段独立训练的，但通过构建过程中的细节设计，它们能够在推理阶段无需额外的联合训练就实现协作，从而同时

控制视觉文本的颜色和字体。具体来说，颜色专家（或字体专家）与文本专家在处理各自对应的控制条件时，共享查询（Query）；颜色专家（或字体专家）与文本专家在训练过程中通过输出相加的方式进行协作；颜色专家（或字体专家）的训练优化在潜空间中受到文本专家的约束。因此，尽管训练过程是独立的，颜色专家和字体专家在潜空间（Latent Space）中隐式地建立了关联，从而可以无需额外训练地在推理阶段进行协作。作为公式 3.14 的补充，在推理阶段，如果同时控制文本内容、颜色和字体，可以表述为：

$$H_{\text{controlled}} = H_t^{'''} + \alpha H_c^{'} + \beta H_f^{'} \tag{3.15}$$

其中 α, β 的设置与先前保持一致。$H_t^{'''}$、$H_c^{'}$ 和 $H_f^{'}$ 的计算方式分别与公式 3.8、3.7 和 3.11 基本相同，区别在于此时三个专家共享查询 $Q^{'''} = H^{'''} W_h^Q$。

通过这种基于启发式规则的多专家协作机制，模型能够处理不同类型的场景文本生成任务。在本研究中，将多属性可控的场景文本生成任务划分为：1）文本内容控制任务；2）文本颜色控制任务；3）文本字体控制任务；4）文本颜色-字体控制任务。在每种任务中，启发式地选择和协调能力解耦的不同专家进行协作，从而实现对视觉文本的多属性控制。

3.2.5 基于自适应路由器的多专家动态协作机制

通过能力解耦的文本专家、颜色专家和字体专家，以及基于启发式规则的多专家协作机制，本研究有效地改善了先前模型在语义空间中属性与实体绑定混淆的问题（例如，文本颜色与背景实体而非文本实体绑定），从而使生成图片中的视觉文本呈现为指定的颜色和字体。然而，在基于启发式规则的多专家协作机制中，用于控制文本内容的强度系数在所有场景中均固定为 1 且不可调节；用于控制文本颜色和字体的强度系数 α 和 β，是通过大量但有限次数的实验尝试与人工观察而设定的经验系数，并且通常在所有输入场景下保持固定，或者需要根据实际情况而手动调节。这意味着这种机制缺乏灵活性，难以在所有输入场景下实现合理的控制力度分配。为此，本研究进一步提出了基于自适应路由器（Adaptive Router）的多专家动态协作机制，根据不同的输入场景，动态地为各个专家分配强度系数，从而更好地协调专家之间的协作。这可以有效地改善本章开头提到的先前模型存在的第二个问题，即“控制力度分配不佳而导致文本颜色未能准确应用到整个文本区域”这一问题。

本研究中自适应路由器的构建，借鉴了 MoE 架构的设计思想，并在此基础上进行了针对性的调整。在传统的 MoE 架构中，路由器与所有专家一同进行训练和参数优化。期间，路由器根据不同任务选择前 k 个合适的专家，分配相应的强度系数，从而动态协调专家之间的协作。其中，超参数 k 通常在所有任务中保持不变。路由器与专家的联合训练及优化，会促使专家分化，使得每个专家专精于处理特定的任务。然而，路由器学习所得的协调策略以及各个专家分化所得的专精领域，通常是不可预知且不易解释的。与此不同的是，本研究中的文本专家、颜色专家和字体专家，已预先通过分阶段的训练获得了边界清晰的控制能力。因此，本研究假设各个专家已经显式地完成了分化，从而进行了第一个针对性调整：在训练过程中冻结所有专家及相应模块的参数，仅将自适应路由器的参数设置为可学习状态。此外，本研究中的四种控制任务具有特征鲜明的输入场景（例如在文本颜色控制任务中，颜色控制条件非空而字体控制条件为空）。基于这一点，本研究假设自适应路由器能够以可预知和可解释的方式根据不同的输入场景选择合适的专家，从而进行了第二个针对性调整：根据不同的任务，动态地将 k 切换为 1、2、2 和 3。具体来说，在文本内容控制、文本颜色控制、文本字体控制以及文本颜色-字体控制这四种任务中，自适应路由器应该分别选择 1 个专家（文本专家）、2 个专家（文本专家、颜色专家）、2 个专家（文本专家、字体专家）和 3 个专家（文本专家、颜色专家及字体专家）。这有助于在控制特定文本属性时，完全隔离与任务无关的专家，从而保持该任务下无需控制的文本属性的多样性。同时，自适应路由器在 k 根据任务动态切换的情况下做出正确选择的能力，也可以验证三个专家的控制能力的解耦。本研究构建的自适应路由器是一种门控机制，具有可学习的权重矩阵 W_r。将颜色控制条件 F_c 和字体控制条件 F_t 拼接作为输入 X，给定任务相关的 k 和缩放因子 λ，自适应路由器为三个专家动态分配强度系数的过程如下：

$$X = \text{Concat}(F_c, F_t) \tag{3.16}$$

$$\mu_i = \lambda \cdot \text{Softmax}(\text{Top}_k(X \cdot W_r)) \tag{3.17}$$

根据各个专家的强度系数，基于自适应路由器的多专家动态协作机制可以可以表示为如下形式：

$$H_{\text{adaptively-controlled}} = \mu_1 H_t^{'''} + \mu_2 H_c^{'} + \mu_3 H_f^{'} \tag{3.18}$$

其中，主干架构中文本专家的强度系数此时也是动态调节的，这有助于更灵活地为各个专家分配控制力度，从而改善“控制力度分配不佳而导致文本颜色未能准确应用到整个文本区域”这一问题。为了对自适应路由器进行训练优化，本研究采用 L2 损失。此外，针对先前构建的合成式数据集中的每个样本，本研究为其随机分配一个任务标签。在自适应路由器的训练阶段，该标签用于决定当前任务类型，从而将与该任务无关的控制条件置为空。由此，自适应路由器能够根据输入的控制条件选择合适的专家并动态协调专家之间的协作。

3.3 实验

3.3.1 实验环境与细节

（1）硬件环境

本章实验在基于容器的 GPU 算力服务平台上进行，所使用的 GPU 容器实例详情如表 3.1 所示。

（2）实施细节

本章提出的 FC-Render 模型以预训练的 TextDiffuser-2[96] 作为主干架构。模型主要基于 Diffusers 和 Transformers 库构建，其中变分自编码器（VAE）、调度器（Scheduler）和分词器（Tokenizer）的预训练权重取自 Stable Diffusion 1.5；文本编码器（Text Encoder）和 U-Net 的预训练权重取自 TextDiffuser-2。为了对颜色（或字体）图片、颜色（或字体）类别 ID 和颜色名称这三种表示方式进行编码，实验中分别采用基于 OpenCLIP-ViT-H-14 的图片编码器（Image Encoder）、可学习的嵌入层和主干架构中的文本编码器。其中，颜色（或字体）图片的尺寸为 512×512。颜色专家（及相关颜

表 3.1　实验使用的 GPU 容器实例详情

部件	参数
操作系统	Ubuntu 20.04.5 LTS
CUDA 版本	11.6
Python 版本	3.8.10
Pytorch 版本	1.12.1
GPU	NVIDIA GPU9 型 / 2 卡 / 24G
CUDA 数	10000+ 个
CPU	Intel 8358P @2.6GHz / 20 核
内存	160GB

色编码器）、字体专家（及相关字体编码器）和自适应路由器均在本章构建的合成式数据集上进行训练，训练的 epoch 数分别为 150、150 和 3。训练采用 AdamW 优化器，批量大小设置为 14，学习率设置为 0.0001，权重衰减率设置为 0.01。为了实现基于启发式规则的多专家协作，由于缺少自动化的量化验证方法，本研究通过大量但有限的实验观察，启发式地为文本字体控制任务、文本颜色控制任务、文本颜色-字体控制任务、文本内容控制任务将强度系数 α 和 β 设置为 $(0, 0.6)$、$(0.5, 0)$、$(0.5, 0.6)$ 和 $(0, 0)$，从而在不同任务中实现对字体和颜色看起来足够的控制；为了实现基于自适应路由器的多专家协作，依据对 α 和 β 的经验性设置，本研究为文本字体控制任务、文本颜色控制任务、文本颜色-字体控制任务、文本内容控制任务动态地调整缩放因子 λ 为 1.6、1.5、2.1 和 1。另外，同样是由于缺乏成熟的自动化量化评估手段，本研究只能依赖有限的对比实验和人工观察来探究不同的颜色和字体表示方式之间的效果差异。得出的结论是，不同的颜色或字体表示方式在颜色或字体控制上并没有表现出显著的差异。基于这一结论，本章后续实验统一采用颜色图片和字体图片作为颜色与字体的表示方式。

（3）测试集

为了评估 FC-Render 在文本颜色与字体控制方面的表现，本研究基于提出的合成式数据集构建方法，额外生成了 100 个样本构成测试集。为了保证实验的客观性，测试集中每个样本的视觉文本均未在训练集中出现过。除此之外，为了考察 FC-Render 各模块对拼写准确性的影响，本研究沿用主干架构所使用的 MARIO-Eval[6] 测试集进行实验评估。MARIO-Eval 测试集包含 LAIONEval400、TMDBEval500 和 OpenLibrary500 这三个子集，共包含 5000 个测试样本。

3.3.2 实验结果与分析

（1）字体专家和颜色专家对拼写准确性的影响

如表 3.2 所示，本研究在 MARIO-Eval 这一大规模测试集上探究 FC-Render 的字体专家、颜色专家以及自适应路由器对拼写准确性的影响。需要说明的是，由于 MARIO-Eval 测试集主要是设计用于评估文本拼写准确性的，并没有充分地考虑文本颜色与字体，因而本研究为其中每个测试样本随机分配了文本颜色或字体提示。为了保证实验的公平性，本研究在本章所有实验涉及的推理（图片生成）中都使用了 DDPM 采样器（Denoising Diffusion Probabilistic Models，DDPM），将采样步数设定

表 3.2　FC-Render 在 MARIO-Eval 测试集上的表现

Model	**Font**	**Color**	**Router**	**MARIO-Eval**		
				CLIPScore↑	OCR Acc↑	OCR F1↑
TextDiffuser-2				<u>34.97</u>	<u>21.36</u>	<u>52.14</u>
FC-Render	✓			**35.00**	**22.79**	**53.71**
		✓		34.96	18.54	48.53
	✓	✓		34.72	18.73	48.53
	✓		✓	34.81	20.38	51.02
		✓	✓	34.64	15.44	43.98
	✓	✓	✓	34.74	18.86	48.71

为 50，将 CFG 参数（Classifier-Free Guidance，CFG）设定为 7.5，并将随机种子固定为 0。对于根据测试样本所生成的图片，本研究采用 EasyOCR 进行文本检测与识别，将从中识别到的文本内容与文本内容真值进行比对。表 3.2 的第一行给出了未引入任何额外模块的基线模型结果。表格中加粗和下划线指示了性能最好的结果和性能第二好的结果。

单独引入字体专家后，可以观察到 CLIPScore、OCR Accuracy 和 OCR F1 均有所提升，这表明字体专家在精准控制字体的同时，有助于增强文本拼写准确性。从直观的角度来看，文本拼写的准确与否，取决于字符序列的正确排列以及其中每个字符字形的准确呈现。本研究采用的字体提示图片为模型提供了每个字符的字形特征，从而提高了字符级别的生成精度，最终带来整体拼写准确率的提升。

单独引入颜色专家后，CLIPScore 的变化较小，但文本拼写准确率出现下降，本研究推测原因在于：某些样本中随机指定的文本颜色与背景颜色对比度不足，导致字符边界模糊；本研究所采用的 EasyOCR 对此类“困难样本”的文本检测与识别精度受限，进一步放大了误差。两者叠加，使得颜色专家虽然显著提高了颜色可控性，却间接导致了拼写准确率的下降。

同时引入字体专家和颜色专家后，两者对于拼写准确率的相反作用发生了部分的抵消，表现为结果介于单独引入字体专家和单独引入颜色专家之间，但整体上仍然低于基线模型。本研究推测原因可能在于：文本专家的相对权重被进一步稀释，削弱了对字符序列的约束能力；同时，字体专家对拼写的正向增益不足以完全抵消颜色专家带来的负面影响，最终使拼写准确性未能恢复至基线水平。进一步引入自适应路由器后，整体趋势保持不变。

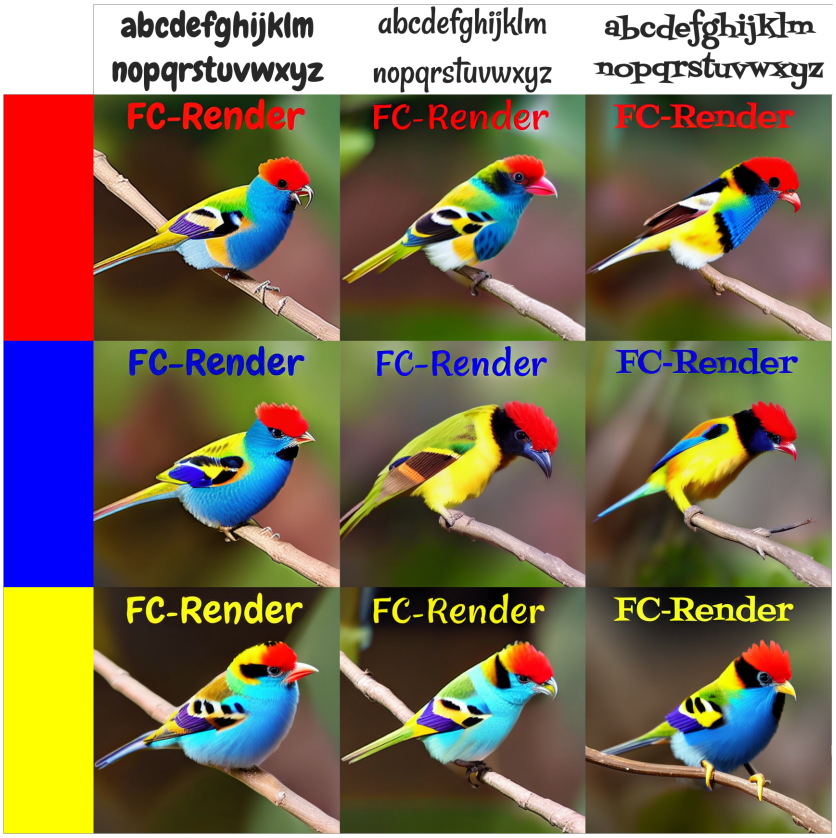


图 3.4　FC-Render 在文本颜色-字体控制任务中的效果展示。最顶端一行展示了字体图片提示，最左侧一列展示了了颜色图片提示。所有生成图片均使用相同的随机种子。

（2）自适应路由器的作用

如表 3.4 所示，本研究在合成式测试集上对拼写准确性和属性控制准确性进行了实验。对于文本颜色控制任务，本研究在基线模型的文本提示中加入了文本颜色说明，其余实验设置均保持一致。在先前的实验中，本研究观察到开源 OCR 工具（如前面所使用的 EasyOCR）在本研究所涉及的样本上存在不稳定性，而这可能会对小规模测试集上的评估结果造成重大干扰。为此，本研究邀请了两名人类评估员对所有生成的图片进行文本检测识别。对于字形错误的字符，本研究规定用星号（*）进行表示，从而确保对拼写准确性的客观评估。文本颜色控制准确性和字体控制准确性

表 3.3　路由器在不同任务中为文本专家、字体专家和颜色专家动态分配的强度系数

Tasks	μ_1	μ_2	μ_3
Spelling	1.0	0.0	0.0
Font	0.84 ± 0.02	0.0	0.76 ± 0.02
Color	0.80 ± 0.01	0.70 ± 0.01	0.0
Font-Color	1.01 ± 0.02	0.49 ± 0.01	0.60 ± 0.02

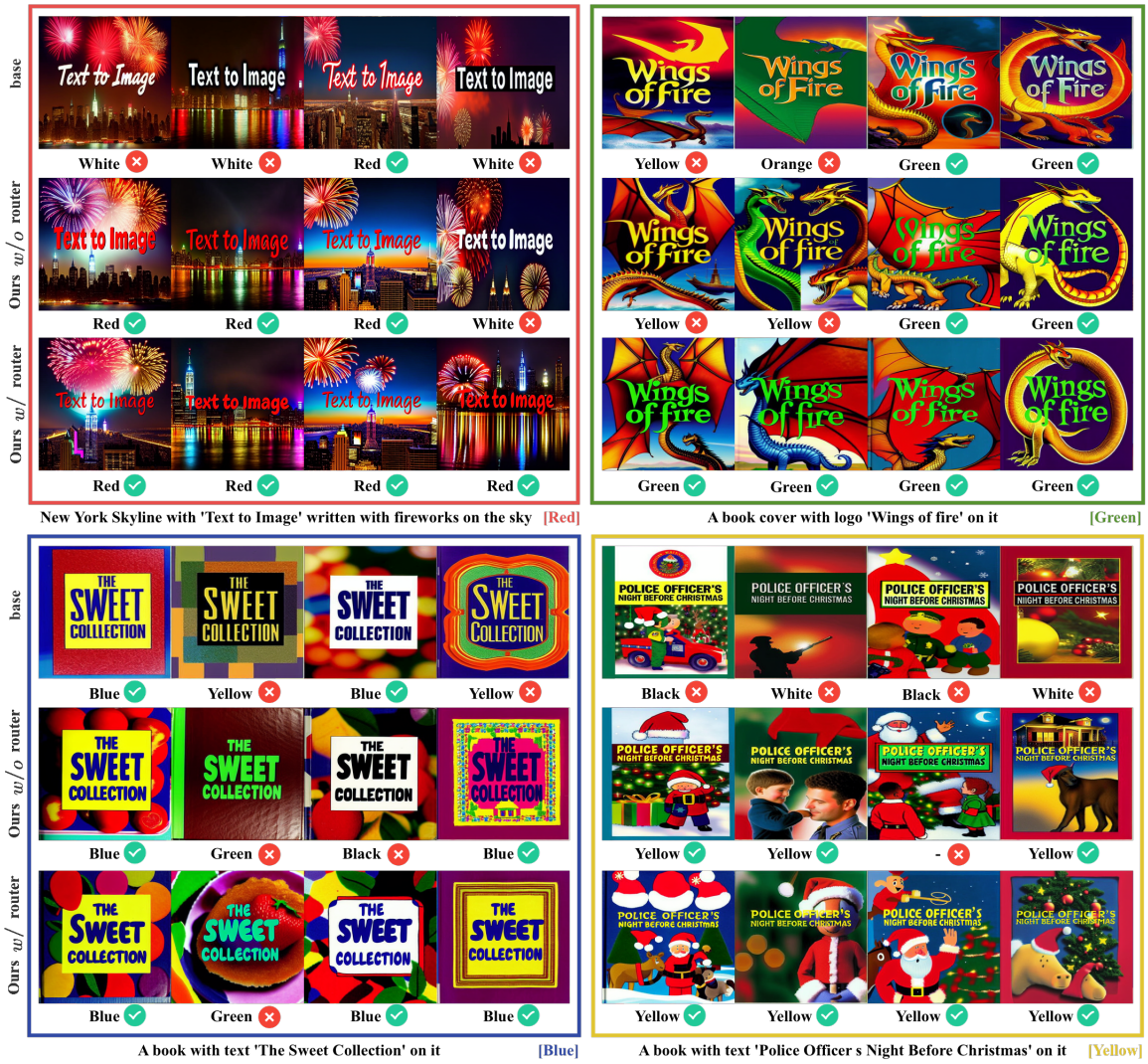


图 3.5　FC-Render 与基线模型在文本颜色控制任务中的效果对比展示

的评估由 11 名人类评估员参与完成。对于文本颜色评估，人类评估员从本研究预先定义的颜色范围中进行选择，在不确定的情况下可以进行多项选择。本研究对某些复杂的颜色情况进行了标准化，与每一位人类评估员进行了统一。当且仅当人类评估员所标注的文本颜色中包含真值时，判定该样本为文本颜色控制正确的样本。对于字体评估，本研究采用 $[0, 1, 2]$ 三级评分，规定人类评估员在文本中有一半的字符与字体提示相匹配时，记为 1 分。

如表 3.3 所示，本研究报告了自适应路由器在合成式训练集上经过训练后，在文本内容控制任务、文本颜色控制任务、文本字体控制任务、文本颜色-字体控制任务中为文本专家、字体专家和颜色专家动态分配强度系数的均值和标准差。可以观察到：在文本字体控制任务和文本颜色控制任务中，自适应路由器降低了文本专家的

表 3.4　FC-Render 在合成式测试集上的表现

Model	**Font**	**Color**	**Router**	**Our Test Set**			
				OCR Acc↑	OCR F1↑	Color Acc↑	Font Acc↑
TextDiffuser-2				**94.00**	**96.75**	38.91	-
FC-Render	✓			91.00	93.84	-	1.56
		✓		92.00	92.75	57.91	-
	✓	✓		92.00	93.55	78.10	1.44
	✓		✓	84.00	86.67	-	**1.60**
		✓	✓	85.00	86.80	**88.82**	-
	✓	✓	✓	92.00	93.55	79.00	1.46

强度系数 μ_1，同时增加了字体专家和颜色专家的强度系数 μ_2 和 μ_3；在文本颜色-字体控制任务中，自适应路由器为三个专家动态分配的强度系数，与本研究中启发式人工固定的强度系数非常接近。

如表 3.4 所示，引入自适应路由器后，文本字体控制任务和文本颜色控制任务中的字体和颜色控制准确率均显著提升，而拼写准确率仅略有下降，整体仍处于较高水平。需要说明的是，在本研究所使用的合成式数据集中，每张图片只包含一个单词，而在基线模型用于训练的数据集和前面实验所使用的 MARIO-Eval 测试集中，每张图片通常包含 1 到 8 个文本。由此，本研究推测：在自适应路由器训练所采用的合成式数据集上，仅需适当的拼写能力即可满足拼写需求，这意味着基线模型中的文本专家在这种情况下展现出了冗余的拼写能力，促使自适应路由器主动降低了文本专家的强度系数，转而加强了对字体与颜色的精细控制。换而言之，自适应路由器在本研究的数据分布上自主地找到了文本专家、颜色专家和字体专家之间合理的协作策略。

如图 3.5 所示，本研究在 MARIO-Eval 测试集上与基线模型进行了文本颜色控制的可视化对比。可以观察到：颜色专家的引入，显著提升了文本颜色控制能力；而自适应路由器的引入，则进一步加强了这一趋势。整体上，拼写准确性也维持在可以接受的水平。值得注意的是，图 3.5 中展示的由 FC-Render 所生成的图片，均在单张图片中包含了多个单词，这表明 FC-Render 具有良好的泛化能力。同时，这也提示：如果在 MARIO 训练集的子集上对自适应路由器进行训练优化，可能会促使其发现新的、针对性的专家协作策略。

如图 3.6 所示，本研究与基线模型进行了文本颜色-字体控制的可视化对比。可

图 3.6　FC-Render 与基线模型在文本颜色-字体控制任务中的效果对比展示

以观察到：自适应路由器为三个专家动态分配的强度系数与本研究中启发式人工固定的强度系数非常接近，因而导致生成图片在视觉上非常相似。然而，在部分案例中，自适应路由器展现出了明显的动态优化能力。图中红色框所展示的情况，正是本章开头所提到的“控制力度分配不佳导致颜色不能准确应用到整个文本区域”的问题，在引入自适应路由器后得到了明显的改善。

3.4　本章小节

本章提出了 FC-Render，用于实现多属性可控的文本渲染。本章首先介绍了专用于文本颜色和字体控制的合成式数据集构建方式，并在此基础上训练了独立的颜色专家和字体专家，再通过启发式规则协调各专家进行协作。进一步地，本章引入了自适应路由器，根据不同的输入，动态地协调各专家间的协作。本章在 MARIO-Eval 这一聚焦于文本内容控制评估的大规模测试集上验证了 FC-Render 的文本内容控制

能力，并在合成式测试集上通过大量人工评估以量化的方式验证了 FC-Render 的文本颜色和字体控制能力。

第四章　基于推理时扩展与后过滤的文本渲染改善方法

4.1　研究动机

文本到图像生成（Text-to-Image，T2I）技术近年来取得了突破性的进展，将自然语言提示转化为符合预期的生成图片，已经成为了视觉生成领域在日常生活中的一种重要应用。与之同步地，基于文本到图像生成的文本渲染也呈现出了水涨船高的发展态势，为标志设计、海报制作、广告创意乃至交互式内容生产等视觉内容创作开辟了低门槛、定制化和高效的全新路径。因此，基于文本到图像生成的文本渲染技术具备较高的研究与应用价值。

在早期的工作中，要在文本到图像生成场景下实现可控的文本渲染，往往需要对通用扩散模型（基座模型）显式地进行架构调整（比如引入额外的字形控制），或需要进行成本较高的参数微调。然而，这些方法通常针对单一维度（大多为文本内容控制）进行专门设计，其余未被重点考虑的维度（如文本颜色和字体）则难以确保可控性。此外，这些方法通常是依赖于基座模型的，基座模型一旦发生改动，就可能需要重新训练甚至重新设计架构，在即插即用的灵活性上存在限制。在最近的工作中，一些新涌现的、位于第一梯队的通用扩散模型，凭借在更精良的训练数据上进行大规模预训练，在无需任何专门适配的情况下就展现出了一定的文本渲染能力，甚至在文本颜色和字体的控制上也具备一定的潜力。发展至今，一个关键的问题逐渐被凸显：针对文本渲染的评估手段目前并不成熟。尽管如 MARIO-Eval[6] 和 AnyText-benchmark[5] 等工作，提供了自动化量化评估框架，可以一定程度上评估模型的文本内容控制能力，但对于文本颜色和文本字体控制能力均无法较好地进行量化评估。这极大地阻碍了人们对文本到图像生成模型的文本渲染能力的全面评估，潜在地会减缓文本渲染技术的进一步发展。

基于上述问题，本章首先提出了文本渲染量化评估框架，旨在同时覆盖文本内容控制、文本颜色控制、文本字体控制、文本位置控制以及图片内容控制这五大维度，对现有文本到图像生成模型的文本渲染能力进行系统全面、自动化的量化评估。大规模量化评估实验表明，现有文本到图像生成模型通常固有地具备一定程度的文本渲染能力。在此基础上，本章提出了基于推理时扩展与后过滤的文本渲染改善方

法，将现有文本到图像生成模型视作黑箱，无需额外地进行架构调整或参数微调，挖掘并放大模型自身既有的文本渲染能力。通过上述贡献，本研究旨在进一步推动文本到图像生成领域中文本渲染技术的发展。

4.2 文本渲染量化评估框架

为了客观、系统且全面地评估文本到图像生成模型的文本渲染能力，本小节以生成图片中的视觉文本与背景内容这两大核心要素为考察点，设计了五个评价维度：文本内容控制（Text Content Control，TCC）、文本颜色控制（Text Color Control，TColC）、文本字体控制（Text Font Control，TFntC）、文本位置控制（Text Position Control，TPosC）以及图像内容控制（Image Content Control，ICC）。其中，TCC 用于评估模型对英文文本拼写的准确性；TColC 和 TFntC 用于评估模型在呈现文本风格（颜色与字体）方面的精确性；TPosC 用于评估模型能否在指定区域精准渲染文本；ICC 用于评估模型能否在渲染文本的同时忠实地渲染图片背景。这五个维度全面地刻画了文本到图像生成模型的文本渲染能力，下面对每个评估维度进行逐一阐述。

4.2.1 文本内容控制

（1）定义

文本内容控制（Text Content Control，TCC）要求文本到图像生成模型能够准确地渲染用户指定的文本内容。在文本渲染任务中，文本内容的准确呈现是最基本的要求，构成文本颜色控制、文本字体控制和文本位置控制的前提。本研究主要关注的是文本到图像生成模型的英文文本内容控制。

在正式的语境下，常见的英文拼写错误可根据字符级别和单词级别进行划分，包含：字符增加（Hello → Helllo）、字符丢失 (Hello → Helo)、字符替换 (Hello → Hollo)、单词增加 (Hello → Hello World)、单词丢失 (Hello World → Hello) 和单词替换（Hello Wolrd → Hi World）。在生成的图片中是否能够规避上述错误从而正确地渲染用户指定的单个或多个单词，可以衡量模型在文本渲染任务中的基本可用性。

在非正式的语境下，存在着一些口语化或网络化但被广泛使用的英文写法，比如：“LET’S GOOOO!”（原始写法：“Let’s go!”）“TL;DR”（原始写法：“Too Long; Didn’t Read”）等。在生成的图片中是否能够正确地渲染用户指定的任意字符串（即便这些字符串是拼写非常规的，甚至是没有语义的），可以衡量模型在文本渲染任务中的进

阶可用性。

（2）评估方法

为了评估模型的文本内容控制（TCC），本研究首先根据不同模型各自的推理方式，利用自然语言提示（Natural Language Prompt）或叠加额外的字形条件（Glyph Condition），为模型引入文本内容控制条件，指导模型生成图片，然后检验所生成的图片是否能够准确地呈现指定的文本内容。

常规的评估方法通常直接对生成图片执行光学字符识别（Optical Character Recognition，OCR），从而检测和识别图片中包含的文本内容，并将识别到的文本内容与用户指定的文本内容真值进行对比。然而，本研究在实践中发现：复杂的图片背景和风格化的文本会降低 OCR 的精度，导致图片中的文本元素被忽略或图片中的非文本元素被错误识别成文本等问题，这会干扰对模型实际的文本内容控制的评估。

为了降低此类评估误差，本研究在 OCR 之前引入像素级文本分割（Pixel-level Text Segmentation），如图 4.1 中的蓝色框所示。像素级文本分割模型被训练用来精确地区分图片中的背景像素和文本像素。受此启发，本研究采用一个预训练的像素级文本分割模型[98]对生成图片进行二值化预处理，滤除图片背景而仅仅保留文本区域，由此产生相应的尺寸不变的掩码图片（Mask Image）。在掩码图片中，前景文本所属的像素通常被置为白色，全部的背景像素则都被置为黑色。获取掩码图片的过程可以表示为：

$$M_{gen} = \text{TextSeg}(I_{gen}) \tag{4.1}$$

其中，I_{gen} 为文本到图像生成模型所生成的图片，TextSeg$(\cdot)$ 代表预训练的像素级文本分割模型，M_{gen} 为生成图片 I_{gen} 所对应的掩码图片。在此之后，本研究将掩码图片 M_{gen} 作为预训练的 OCR 模型的输入。该过程可以表示为：

$$T_{rec} = \text{OCR}(M_{gen}) \tag{4.2}$$

其中，T_{rec} 为检测和识别到的文本内容。通过引入像素级文本分割，本研究期望提升文本检测与识别的可靠性，从而降低对 TCC 的评估误差。

（3）评估指标

为了全面地量化模型的文本内容控制，本研究在单词级别和字符级别两个粒度

表 4.1　混淆矩阵

真实结果	**预测结果**	
	正类	**负类**
正类	TP（真正类）	FN（假负类）
负类	FP（假正类）	TN（真负类）

上选用了三个指标：文本内容准确率（Text Content Accuracy，TCA）、F1 分数（F1 Score，F1）和归一化莱文斯坦相似度[99]（Normalized Levenshtein Similarity，NLS）。其中，文本内容准确率和 F1 分数是较为严格的，而归一化莱文斯坦相似度则能提供相对平滑的补充视角，具体如下：

1）文本内容准确率（Text-Content Accuracy，TCA）： TCA 衡量的是正确呈现指定文本内容的样本占全部样本的比例，计算方式如下：

$$\mathrm{TCA} = \frac{N_{\mathrm{correct}}}{N_{\mathrm{total}}} \tag{4.3}$$

其中，N_{total} 为全部样本的数量，N_{correct} 为正确呈现指定文本内容的样本的数量。当且仅当识别到的文本内容与文本内容真值完全一致时（忽略大小写的差异），判定为正确呈现指定文本内容。TCA 的取值范围为 $[0, 1]$，数值越接近 1，说明模型 TCC 的字面准确性越高。

2）F1 分数（F1 Score，F1）： F1 分数同时考虑了精确率（Precision）和召回率（Recall），是精确率和召回率的调和平均，计算方式如下：

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{4.4}$$

其中，Precision 和 Recall 的计算基于如表 4.1 所示的混淆矩阵（Confusion Matrix）。在本研究中，Precision 衡量的是识别到的与真值完全匹配的文本内容占识别到的全部文本内容的比例，计算方式如下：

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{4.5}$$

在本研究中，Recall 衡量的是识别到的与真值完全匹配的文本内容占全部文本内容真值的比例，计算方式如下：

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{4.6}$$

其中，TP、FP、FN 在本研究中分别代表模型正确渲染的文本内容、模型多余渲染的文本内容以及模型遗漏渲染的文本内容。F1 的取值范围为 $[0, 1]$，数值越接近 1，表明模型多余渲染和遗漏渲染文本内容的情况越少，即反映模型 TCC 的可控性与可靠性越高。

3）归一化莱文斯坦相似度（Normalized Levenshtein Similarity，NLS）：在本研究中，NLS 衡量的是识别到的文本内容与文本内容真值的相似程度。NLS 有别于准确率和 F1 分数，能够在字符级别而非单词级别上反映出文本内容控制的偏差，提供相对平滑和容错的评估视角。NLS 基于莱文斯坦距离（Levenshtein Distance，LD），计算方式如下：

$$\mathrm{NLS} = 1 - \frac{\mathrm{LD}(T_{rec}, T_{gt})}{\max(|T_{rec}|, |T_{gt}|)} \tag{4.7}$$

其中，T_{rec} 为识别到的文本内容，T_{gt} 为文本内容真值。通过将 T_{rec} 和 T_{gt} 的莱文斯坦编辑距离除以二者长度中的较大值，可以实现归一化，从而消除文本长度差异对评估的影响。NLS 的取值范围为 $[0, 1]$，数值越接近 1 表示字符级一致性越高，即模型 TCC 的偏差越小。

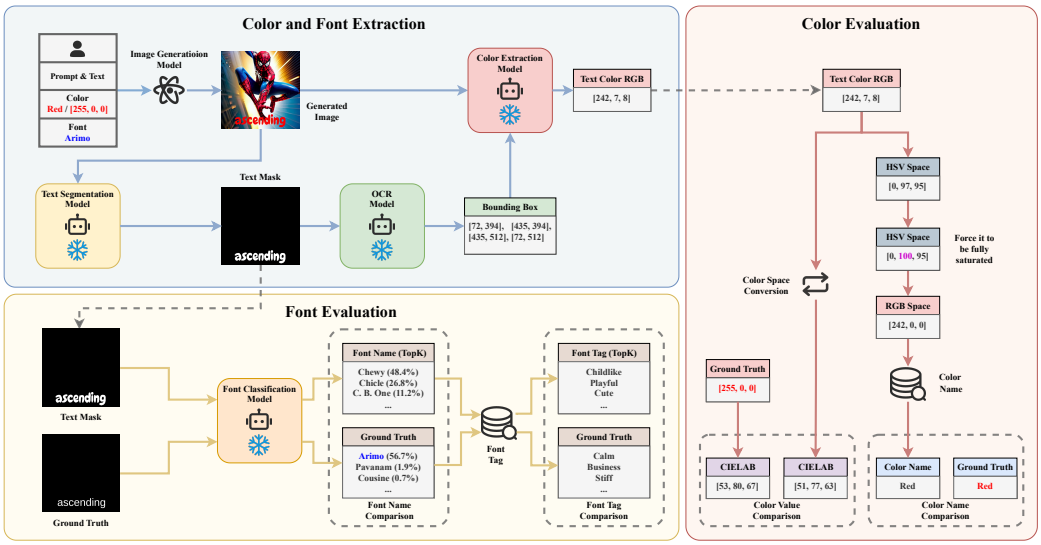


图 4.1　文本内容、文本颜色、文本字体以及文本位置的提取方法

4.2.2　文本颜色控制

（1）定义

文本颜色控制（Text Color Control，TColC）要求文本到图像生成模型能够正确地按照用户指定的颜色对文本内容进行渲染。在文本渲染任务中，准确的文本内容控制（TCC）是前提，可以为用户以直接的方式传达指定的字面信息。更进一步地，如果模型还能够忠实地遵循用户所指定的文本颜色，则可以借助颜色所承载某些约定俗成的情感和语义，起到强调重点、营造氛围和凸显风格等作用，从而更高效地传递信息并显著提升视觉文本的可读性、表达力和感染力。本研究主要关注的是文本到图像生成模型的英文文本颜色控制。

（2）评估方法

为了评估模型的文本颜色控制（TColC），本研究首先利用自然语言提示（Natural Language Prompt）为模型引入文本颜色控制条件，指导模型生成图片，然后检验所生成的图片是否能够以指定的颜色呈现指定的文本内容。

当前对文本颜色控制（TColC）的评估是一个较少被关注和探索的问题，几乎没有直接可用的完善且系统的评估方法和评估指标。评估的难点主要在于如何正确地辨识生成图片中视觉文本的颜色。一种直观的潜在可行的 TColC 评估方式是基于人类受访者的用户调查（Human-based User Study），也就是邀请一组人类受访者观察一组生成图片中的文本颜色，依据预先制定和统一的标准进行评判或打分。尽管人类参与的评估可能是理想的黄金基准，然而这种评估方式受限于难以扩大参与的人类受访者规模和待评判的生成图片规模，而小规模的评估又潜在地存在一定的随机性。同时，基于人类受访者的用户调查往往难以在后续的研究中被复现或复用。进一步地，一种改进的潜在可行的 TColC 评估方式是基于视觉语言模型的用户调查（VLM-based User Study），也就是将人类受访者更换为时下发展迅猛的视觉语言模型（Visual Language Model, VLM）。选用指定的 VLM 并固定相关设置，将预先制定的评判标准与待评测的生成图片作为 VLM 的输入，能够为评估提供较好的可扩展性与可复用性。然而，这种评估方法要求参与的 VLM 具有强大的指令遵循、视觉感知和推理能力，这潜在地要求了相当的计算资源或 API 调用开销。同时，大模型幻觉问题也会影响这种评估方式的可解释性。

为了克服上述两种潜在可行的 TColC 评估方法在可扩展性、可复现性与资源消耗等方面的局限，本研究探索并提出了基于文本颜色数值预测的 TColC 评估框架，如图 4.1 中的蓝色框和红色框所示。正如前面所提到的，TColC 评估的难点主要在于如何正确地识别生成图片中视觉文本的颜色。为了解决这个问题，本研究提出的

TColC 评估框架创新地引入了一个预训练的文本颜色数值预测模型[100]作为核心，并精心设计了颜色空间转换和颜色名称映射流水线，在数值级别（细粒度）和名称级别（粗粒度）这两种粒度下提供 TColC 评估，较好地兼顾了低资源消耗和高可解释性。具体来说，本研究首先沿用 TCC 评估中的文本分割策略，通过一个预训练的像素级文本分割模型对待评估的生成图片 I_{gen} 进行文本分割预处理，从而获得掩码图片 M_{gen}。接着，本研究利用一个预训练的 OCR 模型对掩码图片 M_{gne} 中的文本进行检测和识别，从而获取文本内容以及在 TCC 评估中被忽略的文本位置。获取文本内容和文本位置的过程可以表示为：

$$T_{rec}, B_{rec} = \text{OCR}(M_{gen}) \tag{4.8}$$

其中，T_{rec} 为识别到的文本内容，B_{rec} 为文本内容 T_{rec} 所对应的文本位置边界框坐标。在这之后，将待进行 TColC 评估的生成图片 I_{gen} 与文本位置边界框坐标 B_{rec} 共同作为一个预训练的文本颜色数值预测模型的输入，获得该文本位置边界框内文本的颜色 RGB 数值。获取文本颜色 RGB 数值的过程可以表示为：

$$C_{RGB} = \text{TextColorValuePred}(I_{gen}, B_{rec}) \tag{4.9}$$

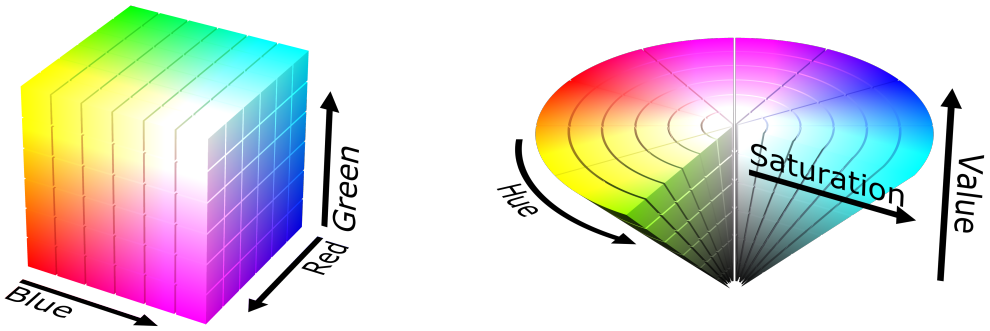
其中，C_{RGB} 为识别所得的文本颜色 RGB 数值，TextColorValuePred$(\cdot)$ 代表预训练的文本颜色预测模型。将识别到的文本内容和其对应的文本颜色配对为 (T_{rec}, C_{RGB})，即解决了 TColC 评估中的文本颜色识别这一难点。为了使 TColC 评估适用于不同精度要求的应用场景，基于获取到的文本颜色 RGB 数值 C_{RGB}，本研究进一步进行颜色空间转换和颜色名称映射，从而支持数值级别和名称级别这两种粒度的 TColC 评估。

1）颜色数值级 TColC 评估：在一些设计相关的专业应用场景中，用户可能会精确地提供颜色数值，要求文本到图像生成模型严格地在颜色数值级别上控制文本颜色。为此，本研究首先提供数值级别的 TColC 评估。基于已经获得的文本颜色 RGB 数值 C_{RGB}，一种容易想到的方式是直接衡量识别到的文本颜色与文本颜色真值在 RGB 颜色空间中的距离。然而，两种颜色在 RGB 颜色空间中的距离并不能较好地反映在人类视觉感知上的颜色差异。考虑到 CIELAB 是一种具有相对完备的感知均匀性的颜色空间，本研究将识别到的文本颜色 RGB 数值与用户指定的文本颜色真值统一地转换到 CIELAB 颜色空间中，从而用于后续的细粒度 TColC 评估指标计算。对

于识别到的文本颜色 C_{RGB}，其颜色空间转换的过程可以表示为：

$$C_{LAB} = \text{rgb2lab}(C_{RGB}) \tag{4.10}$$

其中，rgb2lab$(\cdot)$ 为颜色空间转换规则，C_{LAB} 为识别到的文本颜色所对应的 CIELAB 数值。文本颜色真值的颜色空间转换过程同理。至此，达成进行颜色数值级 TColC 评估指标计算的前提。



(a) RGB[101]　　(b) HSV[102]

图 4.2　不同颜色空间的示意图

2）颜色名称级 TColC 评估：在一些日常的业余应用场景中，用户可能只会粗略地提供颜色名称，要求文本到图像生成模型宽泛地在颜色名称级别上控制文本颜色。为此，本研究进一步提供颜色名称级别的 TColC 评估。核心思想是将已经获得的颜色 RGB 数值 C_{RGB} 映射为对应的颜色名称 C_{NAM}，而这也正是颜色名称级 TColC 评估的主要难点所在。颜色是一种同时受到生理、心理、认知和文化等多种因素影响的复杂感知现象，对于某个 RGB 数值所代表的颜色，需要对其采用一种受到大众认可的颜色名称分类或颜色名称模糊分类方式。为此，本文引入一个现有的颜色调查结果[103]并对其进行补充完善，从而为 RGB 颜色空间构建一种颜色名称分类方式，用于后续的粗粒度 TColC 评估指标计算。如图 4.2(a) 所示的 RGB 颜色空间（RGB 立方体）可以表述为：

$$S_{RGB} = \left\{(r,g,b) \in \mathbb{Z}^3 \mid 0 \leq r,g,b \leq 255\right\} \tag{4.11}$$

图 4.3 对原始的颜色调查结果（Color Survey Results）进行了可视化，展示了颜色名称分类在部分 RGB 颜色空间中的分布。该颜色调查结果的数据来源于一次大规模颜

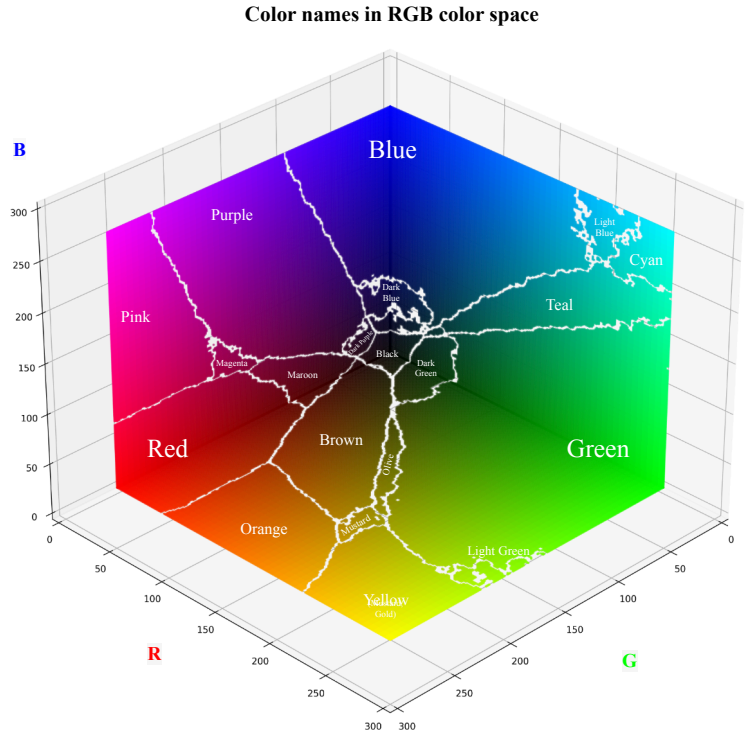


图 4.3 颜色名称分类在部分 RGB 颜色空间中的可视化示意图[103]

色调研（150 万样本），针对 RGB 立方体 S_{RGB} 的三个完全饱和面（Fully-Saturated Faces），即三个至少存在一个 RGB 通道值为 0 的面。该颜色区域可以表述为：

$$\begin{aligned} S_{fs} = &\left\{(0,g,b) \in \mathbb{Z}^3 \mid 0 \le g,b \le 255\right\} \\ &\cup \left\{(r,0,b) \in \mathbb{Z}^3 \mid 0 \le r,b \le 255\right\} \\ &\cup \left\{(r,g,0) \in \mathbb{Z}^3 \mid 0 \le r,g \le 255\right\} \end{aligned} \tag{4.12}$$

本研究进一步对区域 S_{fs} 中的相近颜色名称类别进行了合并，如图 4.4 所示。例如将“Green”、“Light Green”和“Dark Green”合并为“Green”，用以简化和明确主导颜色名称，从而适应颜色名称级粗粒度评估。然而，除了位于区域 S_{fs} 中的颜色外，RGB 立方体 S_{RGB} 中其他区域的颜色至此还没有直接可用的颜色名称分类，至少还明显缺少“Gray”和“White”这两种常用的颜色名称分类。该颜色区域可以表述为：

$$S_g = \{(r,g,b) \in S_{RGB} \mid (r,g,b) \notin S_{fs}\} \tag{4.13}$$

为此，本研究针对颜色区域 C_g 设计了一种基于强制饱和的近似颜色名称映射方法

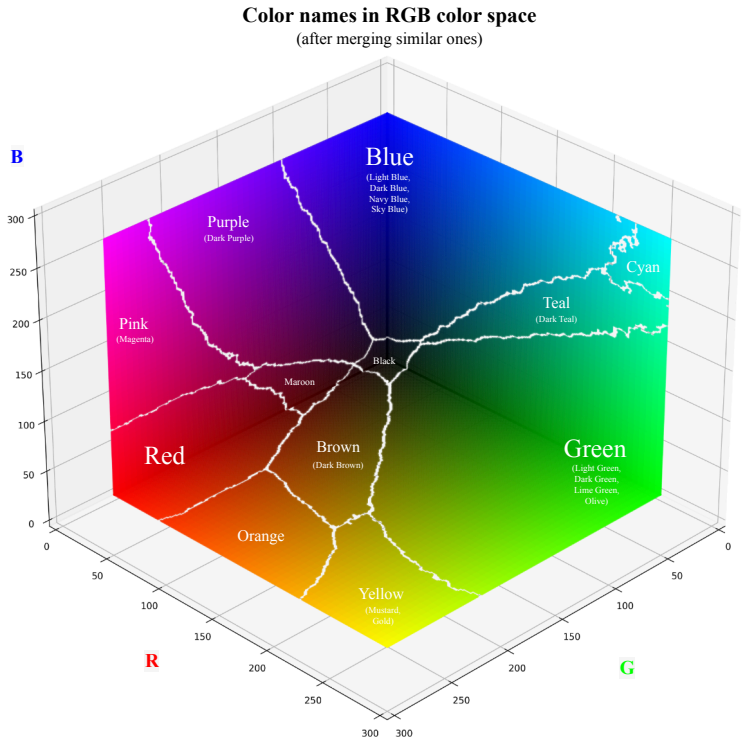


图 4.4　颜色名称分类（合并相近的颜色类别）在部分 RGB 颜色空间中的可视化示意图

（Forced-Saturation-based Approximate Color-Name Mapping Method）并进行了补充颜色调查（Supplementary Color Survey）。具体来说，通过观察公式 4.12 可以发现前述颜色调查结果所针对的颜色区域 S_{fs} 中的所有颜色始终满足：

$$\min(r, g, b) = 0 \tag{4.14}$$

基于这个观察，本研究将前述颜色调查所针对的 RGB 颜色区域 S_{fs} 映射到 HSV 颜色空间。如图 4.2(b) 所示，HSV 颜色空间 S_{HSV} 采用色相 H（Hue）、饱和度 S（Saturation）和亮度 V（Value）三个维度。在 RGB 颜色数值转换为 HSV 颜色数值的公式中，饱和度数值 s 的计算公式如下：

$$s = \begin{cases} 1 - \frac{\min(r,g,b)}{\max(r,g,b)}, & \text{if } \max(r, g, b) \neq 0 \\ 0, & \text{if } \max(r, g, b) = 0 \end{cases} \tag{4.15}$$

观察公式 4.14 和公式 4.15 可以发现，前述颜色调查所针对的 RGB 颜色区域 S_{fs} 中

的每个颜色在 HSV 颜色空间 S_{HSV} 中的饱和度数值始终满足：

$$s_{fs} = \begin{cases} 1, & \text{if } (r,g,b) \neq (0,0,0) \text{ and } (r,g,b) \in S_{fs} \\ 0, & \text{if } (r,g,b) = (0,0,0) \end{cases} \tag{4.16}$$

这意味着，将 RGB 颜色空间 S_{RGB} 中的任意颜色，先映射至 HSV 颜色空间 S_{HSV} 中，再对其进行强制饱和（无论实际饱和度 s 为多少都将其转变为最大值 1），理论上就可以将该任意颜色映射至前述颜色调查所针对的 RGB 颜色区域 S_{fs} 中，从而获取到颜色名称。重要的是，在此过程中保持该任意颜色的色相 s 和亮度 v 不变，可以较大程度地保持颜色本质不变。例如，该任意颜色原先在人眼感知上是某种红色，在经过强制饱和变换后转变为人眼感知上的另一种红色，但都可以被粗略地分类为红色。

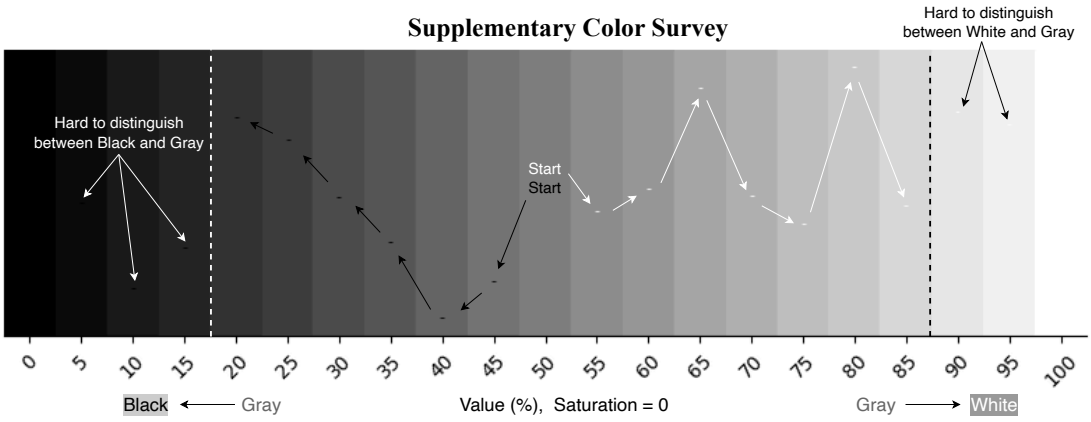


图 4.5 针对黑色、灰色和白色的颜色调研对象

然而，要将前述前述颜色调查结果推广至完整的 RGB 颜色空间仍面临一个问题：当某种颜色的饱和度 s 趋于 0 时，理论上应被人眼感知为黑色、灰色或白色；如果对其进行强制饱和变换，会将该颜色误导至其色相所对应的纯色，从而在颜色名称层面产生显著偏差。例如将色相对应于红色的灰色转变为红色。为此，本研究针对性地设计了补充颜色调查，如图 4.5 所示。该补充颜色调查的目的是相对客观地界定黑色、灰色与白色的亮度阈值。具体来说，本研究首先将 HSV 颜色空间中饱和度为 0 的颜色，按照 5% 的亮度步长依次进行渲染，表现为共 21 个颜色块；接着，本研究在亮度区间为 $0\% - 45\%$ 的每个颜色块中的随机位置各放置一个黑色标记（-），在亮度区间为 $55\% - 100\%$ 的每个颜色块中的随机位置各放置一个白色标记（-）。由于亮度为 0% 的颜色块对应黑色，亮度为 100% 的颜色块对应白色，在黑色块中的黑

算法 4.1: 颜色名称映射

输入: 颜色数值 $C_{RGB} = (r, g, b) \in [0, 255]^3$

输出: 颜色名称 C_{NAM}

$s_{\text{thr}} \leftarrow 0.20$ // 饱和度阈值

$v_{\text{black}} \leftarrow 0.15$ // 黑色亮度阈值

$v_{\text{white}} \leftarrow 0.90$ // 白色亮度阈值

$(h, s, v) \leftarrow \text{RGBtoHSV}(C_{RGB})$

if $s < s_{thr}$ // 饱和度过低则查询补充颜色调查结果

then

  if $v \leq v_{black}$ **then**

    $C_{NAM} \leftarrow$ black

  else if $v \geq v_{white}$ **then**

    $C_{NAM} \leftarrow$ white

  else

    $C_{NAM} \leftarrow$ gray

  end

else

  $(r', g', b') \leftarrow \text{HSVtoRGB}(h,\, 1,\, v)$ // 强制饱和，保持 h, v 不变

  $C_{NAM} \leftarrow \text{RGBtoNAME}(r', g', b')$ // 查询经合并处理后的颜色调查结果

end

return C_{NAM}

色标记和在白色块中的白色标记必然是无法被肉眼分辨的；而在除此之外暂且被统称为灰色的颜色块中，被放置在其中随机位置的黑色或白色标记，理论上是能够被肉眼分辨的。如果此时难以从该颜色块中轻易地观察到黑色（或白色）标记，则可认为该颜色块所呈现的颜色实际在肉眼上无法与黑色（或白色）进行区分，由此将其归为黑色（或白色）的范畴。本研究邀请 11 位无色觉辨认障碍的人类受访者参与补

充颜色调查，调查结果表现为：

$$C_{NAM} = \begin{cases} \text{black}, & \text{if } s = 0 \text{ and } v \leq 15\% \\ \text{gray}, & \text{if } s = 0 \text{ and } 15\% < v < 90\% \\ \text{white}, & \text{if } s = 0 \text{ and } v \geq 90\% \end{cases} \tag{4.17}$$

其中，C_{NAM} 为颜色名称。在此基础上，本研究经验性地设定饱和度阈值为 20%，当颜色饱和度大于等于该阈值时进行强制饱和变换从而采用前述的颜色调查结果，当颜色饱和度小于该阈值时采用本研究获取的补充颜色调查结果。完整流程如算法 4.1 所示。至此，达成进行颜色名称级 TColC 评估指标计算的前提。

（3）评估指标

基于获取到的文本颜色数值和文本颜色名称，本研究选用颜色数值距离（Color Value Distance）和颜色名称准确率（Color Name Accuracy）这两个指标，用于不同精度要求的 TColC 评估。其中，颜色数值距离是较为严格的，颜色名称准确率是相对宽松的，具体如下：

1）颜色数值距离（Color-Value Distance，CVD）：颜色数值距离衡量的是识别到的文本颜色与文本颜色真值在 CIELAB 颜色空间中的欧氏距离（Euclidean Distance），计算方式如下：

$$\text{CVD} = \sqrt{(L^*_{rec} - L^*_{gt})^2 + (a^*_{rec} - a^*_{gt})^2 + (b^*_{rec} - b^*_{gt})^2} \tag{4.18}$$

其中，$(L^*_{rec}, a^*_{rec}, b^*_{rec})$ 为识别到的文本颜色对应的 CIELAB 数值，$(L^*_{gt}, a^*_{gt}, b^*_{gt})$ 为文本颜色真值对应的 CIELAB 数值。CVD 越趋近于 0，表示两种颜色在感知空间中差异越小，即模型 TColC 越精确。

2）颜色名称准确率（Color-Name Accuracy，CNA）：颜色名称准确率衡量的是识别到的文本颜色在名称级别上匹配的样本占全部样本的比例，计算方式如下：

$$\text{CNA} = \frac{N_{match}}{N_{total}} \tag{4.19}$$

其中，N_{total} 为全部样本的数量，N_{match} 为文本颜色名称匹配的样本的数量。当且仅当识别到的文本颜色经过颜色名称映射所得的颜色名称与文本颜色名称真值完全一致时，判定为匹配。CNA 的取值范围为 $[0, 1]$，数值越接近 1，说明模型对用户指定颜色名称的遵从性越高，即 TColC 表现越佳。

4.2.3 文本字体控制

（1）定义

文本字体控制（Text Font Control，TFntC）要求文本到图像生成模型在文本内容控制（TCC）的基础上，忠实地根据用户指定的字体家族名称（Font Family Name）或字体标签（Font Tag）对文本内容进行呈现。与文本颜色类似，文本字体作为一种关键的视觉要素，不仅影响了文本的易读性与可读性，还在潜移默化中塑造信息的情感基调与审美吸引力，可以对阅读认知产生积极的影响。在文本渲染任务中，如果模型能够遵循用户所指定的文本字体，则可以借助字体的特色来突出重点或营造特定氛围。例如，手写风格的字体易于传递亲切与随和的情感，而几何风格的字体常用于表达科技感与理性秩序。本研究主要关注的是文本到图像生成模型的英文文本字体控制。

（2）评估方法

为了评估模型的文本字体控制（TFntC），本研究首先利用自然语言提示（Natural Language Prompt）为模型引入文本字体控制条件，指导模型生成图片，然后检验所生成的图片是否能够以指定的字体家族或字体风格呈现指定的文本内容。

当前对文本到图像生成模型的 TFntC 评估是一个较少被探索的领域，主要的难点在于从复杂的图片背景中识别出前景文本的字体类别。为此，本研究提出一种基于文本字体分类的 TFntC 评估框架，在字体家族名称（Font Family Name）级别和字体标签（Font Tag）级别这两种粒度下提供 TFntC 评估，如图 4.1 中的蓝色框和黄色框所示。受到 Jiang 等人[95]工作的启发，本研究引入一个预训练的文本字体分类模型[104] 作为 TFntC 评估框架的核心，从而获取待评估的生成图片中文本的字体家族名称集合，用于细粒度的 TFntC 评估。这里需要说明的是，之所以采用字体家族名称集合而非单个字体家族名称，是考虑到不同的字体家族名称所代表的字体在视觉感知上实际可能非常相近，以集合的方式将字体分类模型置信度最高的若干个字体家族名称候选进行整体保留，可以大大提升 TFntC 评估的容错性与鲁棒性。此外，本研究额外构建了字体标签映射流水线，从而基于字体家族名称集合获取对应的字体标签集合，用于粗粒度的 TFntC 评估。

1）字体家族名称级 TFntC 评估：为了获取生成图片中文本的字体家族名称集合，本研究仍然沿用 TCC 评估中的像素级文本分割策略，首先获取生成图片 I_{gen} 所

对应的掩码图片 M_{gen}。掩码图片 M_{gen} 在滤除复杂的图片背景的同时较好地保留了文本的笔画和边缘等视觉感知特征，这有助于后续的文本字体分类模型不受干扰和更好地识别文本字体。过程可以表示为：

$$N_f^{rec} = \text{FontClassify}(M_{gen}) \tag{4.20}$$

其中，FontClassify$(\cdot)$ 代表在大规模合成式谷歌字体数据集上经过预训练的字体分类模型，$N_f^{rec} = \{n_1, n_2, ..., n_k\}$ 为字体分类模型输出的 Top-k 个置信度最高的字体家族名称候选所构成的集合。额外地，对于由用户指定的唯一的字体家族名称真值，本研究首先采用图片处理工具 PIL[①]和该字体家族名称真值对应的 TTF 文件，仿照掩码图片 M_{gen} 渲染出一张黑底白字的图片，其中的文本内容与掩码图片 M_{gen} 中的保持一致；接着，将该黑底白字图片输入给字体分类模型，同样获取 Top-k 个置信度最高的字体家族名称候选，作为字体家族名称集合真值 N_f^{gt}。后续，字体家族名称级 TFntC 评估指标将通过比较 N_f^{rec} 和 N_f^{gt} 这两个集合来进行计算，这有助于进一步提升评估的容错性与鲁棒性。

2）字体标签级 TFntC 评估：在业余或日常的应用场景中，用户可能并不需要在字体家族名称级别上精确地要求文本到图像生成模型控制文本字体，而是宽泛地提供一些对期望字体风格的自然语言描述（字体标签）。此时，细粒度的字体家族名称级 TFntC 评估可能过于苛刻，无法客观地反映模型在此类场景下的可用性。为此，本研究对字体分类模型输出的字体家族名称集合 N_f 进行字体标签映射，从而实现粗粒度的字体标签级 TFntC 评估。此时，可以将字体分类模型视作一个字体特征提取器，将其直接输出的字体家族名称集合视作有意义的字体特征的中间表示，而映射所得的字体标签集合才是当前真正预期的输出。该过程可以表示为：

$$T_f = \text{FontTagMapping}(N_f) \tag{4.21}$$

其中，$T_f = \{t_1, t_2, ..., t_i\}$ 为字体标签集合，FontTagMapping$(\cdot)$ 是谷歌字体官方提供的映射规则[105]。本研究中的字体标签，是诸如 “geometric（几何的）”、“calm（冷静的）”、“vintage（复古的）”、“handwritten（手写的）”、“happy（快乐的）”、“cute（可爱的）”、“childlike（天真的）” 这类能够较为到位地描述特定字体风格的自然语言词汇。相比于一个字体家族名称只表示一种字体，一个字体标签可以覆盖多种视觉上

① https://python-pillow.github.io/

相近的字体，而多个字体标签的组合又可细化到更小的风格范围。因此，字体标签级 TFntC 评估，既充分容纳了视觉近似字体带来的自然多样性，又保持了对字体风格的细致表达能力，从而更贴合实际使用场景，能够提高评估的可解释性与鲁棒性。后续，字体标签级 TFntC 评估指标将通过比较 T_f^{rec} 和用户指定的 T_f^{gt} 这两个集合来进行计算。

（3）评估指标

基于获取到的字体家族名称集合和字体标签集合，本研究选用杰卡德系数（Jaccard Similarity Coefficient）作为度量。杰卡德系数常用于比较有限样本集合之间的相似性与差异性，因此适用于本研究的 TFntC 评估。具体如下：

1）字体家族名称集合相似度（Font-Family-Name Set Similarity，FNS）：FNS 衡量的是对待评估的生成图片中文本预测所得的字体家族名称集合与为公平比较而构建的字体家族名称集合真值的交并比，计算方式如下：

$$\text{FNS} = \frac{|N_f^{rec} \cap N_f^{gt}|}{|N_f^{rec} \cup N_f^{gt}|} \tag{4.22}$$

其中，N_f^{rec} 为预测所得的字体家族名称集合，N_f^{gt} 为字体家族名称集合真值。在本研究中，FNS 用于细粒度的 TFntC 评估，取值范围为 $[0, 1]$，数值越高表明模型对指定字体家族的遵循程度越好，。

2）字体标签集合相似度（Font-Tag Set Similarity，FTS）：在用户只需指定宽泛的字体风格而非精确的字体家族的这类评估场景中，将字体分类模型输出的字体家族名称集合视作字体风格特征的中间表示，进一步映射为字体标签集合，再与真值标签集合计算杰卡德系数，得到粗粒度的 FTS 指标，计算方式如下：

$$\text{FTS} = \frac{|T_f^{rec} \cap T_f^{gt}|}{|T_f^{rec} \cup T_f^{gt}|} \tag{4.23}$$

其中，T_f^{rec} 和 T_f^{gt} 分别为预测与真值的字体标签集合。FTS 的取值范围同样为 $[0, 1]$，数值越接近 1，表示模型在字体风格层面与用户期望的契合度越好，能反映模型在日常场景下的 TFntC 可用性。

4.2.4 文本位置控制

（1）定义

文本位置控制（Text Position Control，TPosC）要求文本到图像生成模型能够将指定的文本内容准确地呈现于用户期望的坐标或区域。在文本渲染任务中，文本内容控制（TCC）与文本位置控制（TPosC）的协同，可以显著提升生成图片中文本的可读性与整体排版，进一步强化文本渲染的可控性，为海报设计、广告创意等高度定制化的视觉内容创作提供更大的设计自由度与实现可能性。本研究主要关注的是文本到图像生成模型的英文文本位置控制。

（2）评估方法与指标

目前，大多数文本到图像生成模型并没有针对文本位置进行过专门的优化，支持 TPosC 的只有少数面向文本渲染的专用模型。这类模型通常在自然语言提示之外，还需要额外的文本位置控制条件，比如通过显式的位置坐标或位置掩码输入，又或者隐式地封装在字形控制条件之中。为了客观地评估此类模型的 TPosC，本研究首先按照模型所需的形式为其注入相应的文本位置控制条件，指导模型生成图片，然后检测生成图片中指定文本的实际坐标，与位置真值进行比对，从而判定模型是否能够把文本精准地渲染到目标位置。

在先前的 TColC 评估中，本研究已经能够获取生成图片中的文本内容 T_{rec} 和其文本位置边界框坐标 B_{rec}，由此可以进行 TPosC 评估。具体来说，本文采用目标检测领域常用的交并比（Intersection over Union，IoU）作为 TPosC 的衡量指标。这里值得说明的是，IoU 本质上与 TFntC 评估中的杰卡德系数是同一个概念，衡量的是检测所得的位置边界框与位置边界框真值的重叠程度，计算方式如下：

$$\text{IoU} = \frac{|B_{rec} \cap B_{gt}|}{|B_{rec} \cup B_{gt}|} \tag{4.24}$$

其中，B_{gt} 为位置边界框真值。IoU 的取值范围为 $[0, 1]$，数值越接近 1，表示检测所得的文本位置边界框与文本位置边界框真值的重合度越高，即表明模型的 TPosC 越准确。

4.2.5 图片内容控制

（1）定义

图片内容控制（Image Content Control，ICC）要求文本到图像生成模型在渲染指定文本内容的同时，仍然能够在生成图片中忠实地包含用户期望的非文本元素，如物体、人物和场景等。在文本渲染任务中，如果模型能够一次性端到端地正确渲染

文本与背景，则可以避免将文本与图片内容分阶段生成再组合，或后续对不符预期的文本内容的编辑，从而简化整体流程并提高模型的实用性。

（2）评估方法与指标

为了评估模型的图片内容控制（ICC），本研究遵循通用文本到图像生成领域的一贯做法，衡量生成的图片内容与与其对应的自然语言提示的语义相似性，即计算 CLIP 分数（CLIP Score，CS）。计算方式如下：

$$\text{CS} = \cos(v_t, v_i) \tag{4.25}$$

$$v_t = \text{CLIPTextEncoder}(T_{gen}) \tag{4.26}$$

$$v_i = \text{CLIPImageEncoder}(I_{gen}) \tag{4.27}$$

其中，$\cos(\cdot)$ 表示计算余弦相似度（Cosine Similarity）；v_t 和 v_i 分别为自然语言提示 T_{gen} 和生成图片 I_{gen} 所对应的文本特征向量和图片特征向量；CLIPTextEncoder$(\cdot)$ 和 CLIPImageEncoder$(\cdot)$ 代表成对的 CLIP 文本编码器和 CLIP 图片编码器。CS 数值越大，表示模型 ICC 越佳。

4.3 提出方法

除了系统全面地量化评估现有文本到图像生成模型的文本渲染能力外，本研究进一步提出基于推理时扩展与后过滤的文本渲染改善方法，旨在在不改动模型架构或权重的前提下，挖掘放大模型已有的文本渲染能力。

4.3.1 文本到图像生成模型推理时扩展

受到近期大语言模型（Large Language Model，LLM）领域的启发，本研究探究文本到图像生成模型的推理时扩展（Inference-Time Scaling，ITS）。推理时扩展，广义地可以理解为在推理阶段耗费更多时间与计算资源，从而换取更优的输出结果。作为对比，传统的模型优化通常聚焦于修改模型的架构或微调模型的权重，需要耗费大量的数据资源与训练资源，还需要根据不同模型架构进行调整。然而，随着文本到图像生成领域的发展，模型架构逐渐多样，从基于 U-Net 架构的扩散模型到基于 DiT 架构的扩散模型，再到逐渐兴起的视觉自回归架构的模型。推理时扩展不受模型架构影响，因此具有一定的研究潜力。

本研究采用多采样策略来实现文本到图像生成模型的推理时扩展。具体来说，对于用户指定的自然语言提示，本研究对其用 k 个随机种子进行 k 次采样（固定其余设置），从而生成 k 张候选图片。当模型本身具备一定的文本渲染能力时，这种多采样策略的推理时扩展，可以给模型更多次生成机会；对 k 张候选图片采用进一步的后过滤，最终保留其中的 n 张图片（$n < k$）作为输出，即可达到优化输出的同等效果。这种方法不需要额外的训练，将文本到图像生成模型视作黑箱，推理开销与 k 基本呈线性相关，可以根据实际推理预算进行灵活设置，因此是一种即插即用、架构无关、灵活配置的文本渲染改善方法。

4.3.2 基于大型视觉语言模型的后过滤

本研究采用大型视觉语言模型（Large Vision-Language Model，LVLM）对文本到图像生成模型推理时扩展后所得的生成图片进行后过滤 (Post-Filtering，PF)。具体来说，后过滤分为两个步骤：1）对每张生成图片进行评估；2）在所有生成图片中进行筛选。

（1）评估

为了让 LVLM 对生成图片进行评估，需要为其构建提示（Prompt），并令其运行推理（Inference）。其中，提示通常由指令（Instruction）和查询提示（Query）两部分组成。进一步地，指令则通常包含任务描述、约束条件、上下文范例（可选）、格式化输出范例。

任务描述 (Task Description)：LVLM 具有广泛的能力，为 LVLM 提供详细明确的任务描述，可以引导 LVLM 专注于特定的目标并调用其相关的知识。任务描述应当精心设计，使用清晰合理的自然语言进行增强，使 LVLM 更深入地理解任务需求。对于本研究中的图片评估任务，任务描述 $Instr_t$ 可以设计为：

“你是一位图片评估专家，你会看到一张由文本到图像生成模型所输出的生成图片，以及用户对这张生成图片的预期。你需要仔细观察生成图片，充分理解用户预期，然后评估生成图片是否符合用户预期。

约束条件 (Constraints)：由于 LVLM 的能力广泛，其完成图片评估任务的依据与形式有很多，比如：LVLM 可能会将图片的清晰度作为依据，以自然语言评价的形式执行图片评估任务。为此，本研究为 LVLM 提供约束条件，旨在为其明确评分项（形式）与评分标准（依据）。具体来说，本研究要求 LVLM 按照固定的评分标准

（Rubric），分别为 TCC、TColC、TFntC、ICC 提供对应的拼写分数、颜色分数、字体分数、语义分数。以拼写分数为例，约束条件 $Instr_c$ 中的对应部分可以设计为：

拼写分数：你只关注你在图片中实际看到的文本，不要试图纠正文本的拼写。拼写分数应该是：*1*，如果图片中文本与“*tourists*”完全匹配（不区分大小写，没有多余/缺失/替换的字符）；*0.5*，如果存在轻微的拼写错误，但不影响可读性；*0*，如果文本整体缺失、乱码或完全错误。

格式化输出（Formatting）：为了令 LVLM 的输出格式保持一致和标准化，便于后续的数据处理，提示中需要明确预期的输出格式（通常为 JSON 格式）。在本研究中，格式化输出范例 $Instr_f$ 可以如下：

{“图片中识别到的文本”：“*XXX*”，“图片中文本的颜色”：“*XXX*”，“拼写分数”：*X*，“颜色分数”：*X*，“语义分数”：*X*，“理由”：*[*“*XXX*”，“*XXX*”，“*XXX*”*]}*

查询提示（Query）：查询提示通常由问题对象（QuestionObject）和具体问题（Question）这两部分组成。在本研究中，问题对象是生成图片，对应的具体问题则可以设计如下：

对于生成图片，用户指定的提示是：*A picture of a monkey with the orange text* “*tourists*”；生成图片中的预期文本是：“*tourists*”；生成图片中文本的预期颜色是：“*orange*”。以下是生成图片。请你根据用户预期对生成图片进行评估。

推理（Inference）：在构建完整的提示（指令、查询提示）后，将序列化的提示作为 LVLM 的输入，使其运行推理，最终输出结果。推理过程可以表示为：

$$J = \text{LVLM}(Instr_t, Instr_c, Instr_f, Query) \tag{4.28}$$

其中，J 表示表示 LVLM 推理后输出的结果（JSON 格式）。

（2）筛选

在 LVLM 对所有生成图片完成评估后，每张生成图片即获得了 TCC、TColC、TFntC、ICC 对应的拼写分数、颜色分数、字体分数、语义分数。由于本研究中各个分数的数值范围都设定为 $[0, 1]$，因此本研究直接将各个分数进行累加，作为每张生成图片的最终分数。此时，对于用户指定的自然语言描述，文本到图像生成模型经推理时扩展后得到 k 张候选图片，最终分数最高的 n 张图片被保留作为输出，其余图片则直接丢弃，至此完成后过滤。

4.4 实验

4.4.1 数据集介绍

为了全面客观地量化评估文本内容控制（TCC）、文本颜色控制（TColC）、文本字体控制（TFntC）、文本位置控制（TPosC）以及图片内容控制（ICC），本章基于在第三章中提出的合成式数据集构建方法，额外构建了六个侧重不同的评估数据集子集。这些数据集的统计信息如表 4.2 所示。

表 4.2　评估数据集的统计信息

数据集	覆盖维度	样本量
SingleWordData	TCC, TPosC, ICC	630
DoubleWordData	TCC, TPosC, ICC	315
SingleStringData	TCC, TPosC, ICC	150
ColorData	TCC, TColC, TPosC, ICC	630
FontData	TCC, TFntC, TPosC, ICC	630
ColorFontData	TCC, TColC, TFntC, TPosC, ICC	630
TOTAL		**2,985**

1）**SingleWordData（SWD）**：该数据集侧重于评估正式语境下的 TCC，模拟了最简单的拼写场景。本研究随机挑选了 630 个具有明确语义的英文单词，并嵌入第三章中所采用的文本提示模版中。每个文本提示样本仅包含一个待渲染的英文单词，并且该英文单词与背景物体描述通常没有语义关联，例如：A picture of a tiger and the text “ethnicity’。

2）**DoubleWordData（DWD）**：该数据集同样侧重于评估正式语境下的 TCC，模拟了多词干扰的场景，理论上比 SingleWordData 具有升级的拼写难度。本研究同样采用 SingleWordData 中选定的 630 个具有明确语义的英文单词，但对其进行随机的两两配对。每个文本提示样本中包含两个待渲染的英文单词，例如：A photo of a bear and the words “symbolic” and “benin”。

3）**SingleStringData（SSD）**：为了评估在非正式语境下的 TCC，本研究随机生成了 150 个无语义的英文字符串（长度为 3 至 17 个字符）。每个文本提示样本中包含一个待渲染的英文字符串，例如：A picture of an apple with the text “Borborygaabbcmus”。该数据集理论上模拟了要求更高的拼写场景。

4）**ColorData（CD）**：该数据集侧重于评估 TColC，基于最简单的拼写场景，额外随机指定了文本颜色，例如：A picture of a monkey with the orange text “tourists”。

5）**FontData（FD）**：该数据集侧重于评估 TFntC，在 SingleWordData 数据子集的基础上，额外随机指定了文本字体。在该数据集中，每个样本对应两个版本的文本提示，分别为 “字体家族名称” 版本和其对应的 “字体标签” 版本，例如：A photo of the Iron-Man with the text “complementary”. The text is written in Arimo font 及对应的 A photo of the Iron-Man with the text “complementary”. The text is written in a business, and calm font。

6）**ColorFontData（CFD）**：该数据集侧重于协同评估 TColC 和 TFntC，模拟了较为复杂的、综合的文本渲染场景。由于同样涉及 TFntC，因此该数据集中每个样本类似地对应两个版本的文本提示，例如：A picture of a lion and the green text “maturity”. The text is written in Fontdiner Swanky font 及对应的 A picture of a lion and the green text “maturity”. The text is written in an awkward, excited, happy, loud, playful, vintage, and wacky font。

4.4.2 实验设置

本章实验选用的文本到图像生成模型为 GlyphControl[4]、AnyText[5]、TextDiffuser[6]、Stable Diffusion 3 Medium（SD3）[9]以及 FLUX.1-schnell（FLUX）[10]。其中，GlyphControl、AnyText 和 TextDiffuser 都是基于 U-Net 架构的文本渲染专用扩散模型；SD3 和 FLUX 则是基于 DiT 架构的通用扩散模型。

本章实验采用的 LVLM 为 Qwen2.5-VL-7B-Instruct[106]，其 temperature 参数设定为 $1e-06$，use_fast 参数设定为 True。

4.4.3 实验结果与分析

（1）文本渲染量化评估框架的可靠性验证

为了检验本章提出的文本渲染量化评估框架的可靠性，本章首先对第三章中通过人工评估而获得的结论进行回溯性量化验证。回顾第三章，当时由于缺乏针对 TColC 和 TFntC 的自动量化评估手段，对于 “不同强度系数下不同颜色与字体表示方式的效果差异”，仅通过有限的实验对比与人工观察得出了两个定性结论：1）不同的颜色或字体表示方式在 TColC 或 TFntC 上并无显著的效果差异；2）颜色专家

表 4.3 回溯 FC-Render 中不同强度系数下不同颜色表示方式的效果差异

α	Color Rep.	CS↑	IoU↑	TCA↑	F1↑	NLS↑	CVD↓	CNA↑
0.3	Img	0.3726	0.6932	0.8524	0.8627	0.9611	87.71	0.2731
	ID	0.3729	0.6992	0.8603	0.8692	0.9664	86.88	0.3005
	Nam	**0.3732**	0.7016	0.8810	0.8911	0.9721	90.77	0.2558
0.5	Img	0.3711	0.7111	0.8603	0.8722	0.9670	40.49	0.7241
	ID	0.3729	0.7048	0.8984	0.9005	0.9753	43.89	0.6991
	Nam	0.3730	0.7155	0.8921	0.8963	0.9769	50.21	0.6210
0.6	Img	0.3712	0.7066	0.8889	0.8924	0.9758	25.35	0.8493
	ID	0.3712	0.7077	**0.9016**	**0.9026**	**0.9772**	26.96	0.8310
	Nam	0.3709	**0.7198**	0.8889	0.8899	0.9762	29.48	0.7961
0.8	Img	0.3689	0.7047	0.8524	0.8571	0.9669	7.76	0.9815
	ID	0.3686	0.7059	0.8730	0.8772	0.9712	9.56	0.9657
	Nam	0.3680	0.7118	0.8698	0.8717	0.9746	9.84	0.9672
1.0	Img	0.3650	0.6983	0.7841	0.7894	0.9563	**5.23**	0.9980
	ID	0.3648	0.6984	0.8349	0.8421	0.9664	6.28	0.9906
	Nam	0.3636	0.7008	0.8238	0.8238	0.9629	5.71	**1.0000**

表 4.4 回溯 FC-Render 中不同强度系数下不同字体表示方式的效果差异

β	Font Rep.	CS↑	IoU↑	TCA↑	F1↑	NLS↑	FNS↑	FTS↑
0.3	Img	0.3739	0.7037	0.9048	0.9153	0.9758	0.1322	0.4831
	ID	**0.3746**	0.7006	0.9032	0.9138	0.9770	0.1229	0.4682
0.5	Img	0.3726	**0.7325**	0.9159	0.9180	0.9840	0.2435	0.5761
	ID	0.3726	0.7203	**0.9286**	**0.9307**	**0.9855**	0.2360	0.5732
0.6	Img	0.3727	0.7290	0.9238	0.9259	0.9841	0.2871	0.6005
	ID	0.3719	0.7173	0.9175	0.9185	0.9824	0.2744	0.5936
0.8	Img	0.3707	0.7206	0.8905	0.8957	0.9736	**0.3223**	0.6157
	ID	0.3690	0.7062	0.8889	0.8952	0.9741	0.3132	**0.6189**
1.0	Img	0.3664	0.7023	0.8349	0.8410	0.9619	0.3212	0.6143
	ID	0.3655	0.6890	0.8175	0.8193	0.9576	0.3067	0.6125

和字体专家的强度系数 α 和 β 在 0.5 或 0.6 时表现较好。此外，第三章基于小规模合成式评估数据集，进行了耗费大量人力与时间的人工评估，从而获得了表 3.4 中的量化结果。在本章中，沿用完全相同的评估数据集制作方式，将评估数据集规模由第三章中的 100 个样本扩大为 1400 个样本，并采用本章提出的文本渲染量化评估框架获取新的定量结论与量化结果。由此可以比对本章中量化评估与第三章中人工评估分别获得的结论。

表 4.5 回溯 FC-Render 中自适应路由器的作用

Color	Font	Router	CS↑	IoU↑	TCA↑	F1↑	NLS↑	CVD↓	CNA↑	FNS↑	FTS↑
	✓		**0.3719**	**0.7173**	**0.9175**	**0.9185**	**0.9824**	-	-	0.2744	0.5936
✓			0.3712	0.7077	0.9016	0.9026	0.9772	26.96	0.8310	-	-
✓	✓		0.3676	0.7112	0.8984	0.9045	0.9781	21.78	0.8722	0.2678	0.5887
	✓	✓	0.3675	0.6980	0.8349	0.8349	0.9590	-	-	**0.3008**	**0.6148**
✓		✓	0.3676	0.6948	0.8619	0.8627	0.9642	**11.99**	**0.9539**	-	-
✓	✓	✓	0.3674	0.6991	0.8683	0.8722	0.9732	38.05	0.7164	0.3002	0.6075

如表 4.3 和表 4.4 所示，本章借助新提出的的文本渲染量化评估框架，分别对不同强度系数（0.3、0.5、0.6、0.8、1.0）下不同颜色表示方式（Img、ID、Nam）和不同字体表示方式（Img、ID）的效果差异进行了全面细致的量化评估。表格中加粗表示性能最好的结果。观察表 4.3 和表 4.4 可以发现，最好的结果高度集中在强度系数为 0.5 和 0.6 的范围内：在 TCC 中的 TCA、F1、NLS 指标上以及 TPosC 中的 IoU 指标上表现尤为突出，而 TColC 中的 CVD、CNA 以及 TFntC 中的 FNS、FTS 也在该强度系数范围内取得良好表现。这一量化发现与第三章中基于人工观察所得的“颜色专家和字体专家的强度系数 α 和 β 在 0.5 或 0.6 时表现较好”这一结论完全一致，验证了本章提出的文本渲染量化评估框架的可靠性。同时，对比不同颜色或字体表示方式，可以发现不同表示方式之间的效果差异从 TColC 和 TFntC 的量化指标上来看确实并不显著，与第三章中另一个通过人工观察所得的“不同的颜色或字体表示方式在 TColC 或 TFntC 上并无显著的效果差异”结论相吻合。

如表 4.5 所示，本章同样对第三章表 3.4 中通过人工评估得到的量化结果，用本章提出的文本渲染量化评估框架进行回溯性验证。评估数据集的规模由原先的 100 扩大为了 1400，有助于降低评估结果的偶然性。另外，本章统一采用 ID 作为颜色或字体表示方式，α 与 β 均定为 0.6。观察表 4.5 可以发现，TCC 在单独引入字体专家时为最佳，而 TColC 和 TFntC 则分别在同时引入路由器和颜色或字体专家时为最佳。该自动化量化结论与第三章人工评估观察到的趋势完全一致，再次验证了本章文本渲染量化评估框架的可靠性。

总而言之，使用本章提出的文本渲染量化评估框架对第三章人工评估所得结论进行回溯验证后，可以发现两者所得的实验结论完全一致。这充分有力地验证了文本渲染量化评估框架的可靠性。此外，该评估框架完全无需人类评估员参与，天然地具备较高的可复现性，并在实施成本和操作便捷性上相比于人工评估具有极大的

优势。因此，该量化评估框架具有在未来的文本渲染研究中推广应用的潜力。本章后续均采用该文本渲染量化评估框架进行评估。

(2) 对现有模型文本渲染能力及推理时扩展效应的评估

为了全面地量化评估现有模型的文本渲染能力，并探究推理时扩展的效应，本研究在构建的六个评估数据集上，采用提出的文本渲染量化评估框架开展实验。具体来说，本研究对每个数据子集中的每个样本，令各个模型在固定的推理设置下用 10 个不同的随机种子各生成 1 张图片（共 10 张）；然后按 $k = 1, 2, \ldots, 10$ 逐步评估和计算 k 张图片时的量化指标，得到对应不同采样数量的 10 组量化指标，从而模拟推理时扩展对模型文本渲染表现的影响。

1）TCC 分析。图 4.6 展示了各个模型在 SWD、SSD 和 DWD 这三个评估数据集上的量化评估结果。这三个评估数据集侧重于评估模型在不同 TCC 场景下的表现。需要说明的是，由于 SD3 和 FLUX 无法进行精确的 TPosC，因此本研究将这两个模型在 IoU 指标上的数值标记为 0。从图 4.6 的横向角度看，在 TCC 相关的 NLS、F1 和 TCA 指标上：GlyphControl 和 TextDiffuser 在 SSD 上表现最好，在 SWD 上表现其次，在 DWD 上表现最差；AnyText 在 SSD 和 DWD 上表现相当，在 SWD 上表现最差；SD3 和 FLUX 在 SWD 上表现最好，在 SSD 上表现其次，在 DWD 上表现最差。这表明，文本渲染专用模型比较擅长处理无语义的单个字符串渲染，而通用模型则比较擅长处理有语义的单个单词渲染。潜在的原因在于，文本渲染专用模型的语义理解能力相对受限，无语义的字符串可能不容易导致整体语义混淆，因此对这类模型来说 SSD 是最简单的 TCC 场景；通用模型具有更强的语义理解能力，但未针对 TCC 而专门优化，其 TCC 能力主要来源于庞大而精良的训练数据，有语义的单个单词拼写场景可能在训练数据中出现更多，因此对这类模型来说 SWD 是最简单的 TCC 场景。从图 4.6 的纵向角度看，AnyText 和 TextDiffuser 这两个文本渲染专用模型在 TCC 方面是显著领先的，但 SD3 和 FLUX 这两种通用模型已经能赶上甚至超越作为文本渲染专用模型的 GlyphControl 了（除了无法进行精确的 TPosC），体现出通用模型近年来发展之迅猛。另外，三个文本渲染专用模型的 TPosC 表现没有明显差异；所有模型的 CS 指标都相对接近和稳定，表明各个模型的 ICC 表现没有明显差异。最后，在经推理时扩展后，所有模型的文本渲染表现基本保持稳定，因此线条看起来相对重合。

2）TFntC 分析。图 4.7 展示了各模型在 FD 数据子集上的量化评估结果。FD 数

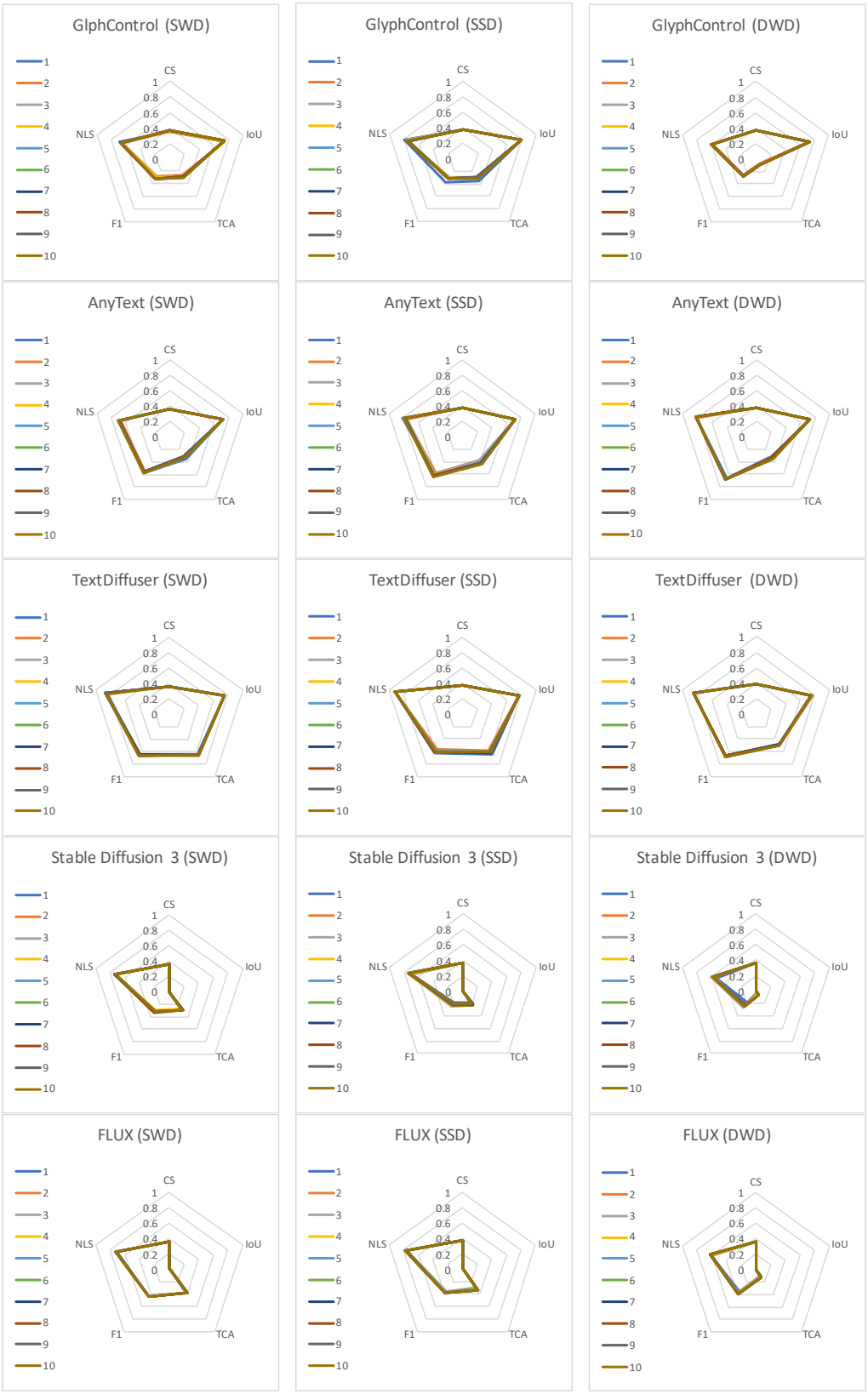


图 4.6 各模型在 SingleWordData、SingleStringData、DoubleWordData 上的量化评估结果

据子集侧重于评估模型的 TFntC 表现。本研究通过额外的实验发现采用字体标签的方式相较于直接采用字体家族名称的方式，能够令各模型更好地进行 TFntC，体现

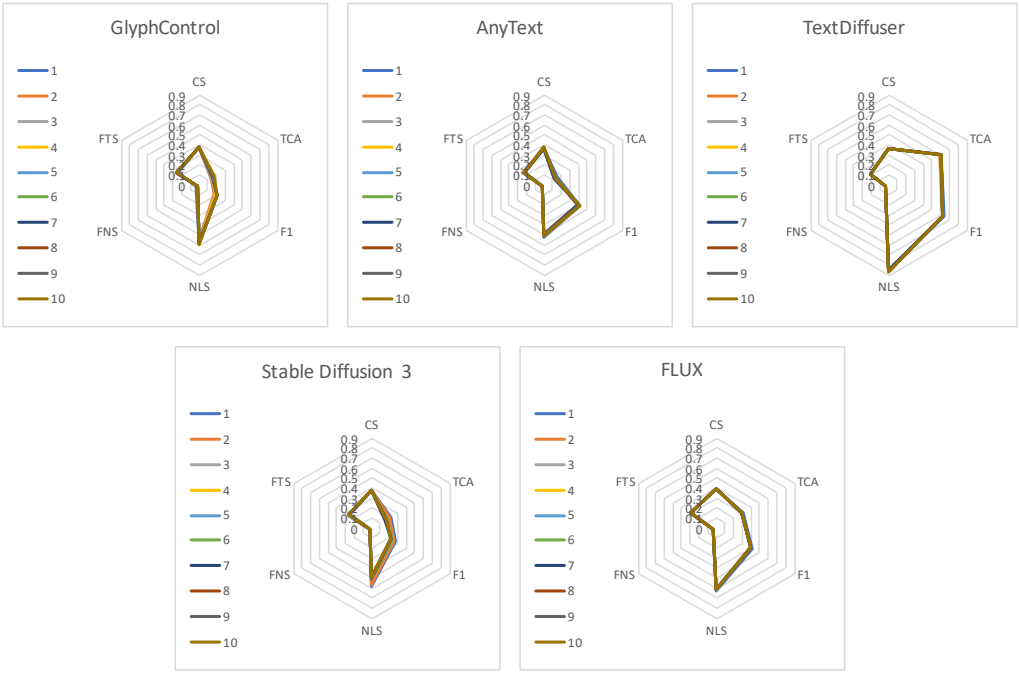


图 4.7　各模型在 FontData 上的量化评估结果（忽略了 TPosC）

为 TFntC 相关的 FNS 和 FTS 指标小幅提升，因此本章后续涉及 TFntC 的评估均统一使用字体标签进行采样。此外，由于观察到三个文本渲染专用模型的 TPosC 表现相当，而另外两个通用模型的 TPosC 被本研究标记为 0；为了更清晰地观察其他维度的表现，暂时对 TPoSC 维度进行忽略。从图 4.7 可见，细粒度字体指标 FNS 在所有模型上几乎趋近于 0；而在粗粒度的 FTS 指标上，FLUX 和 SD3 表现最好，表明这两个通用模型凭借其更强的语义理解能力，能更好地遵循以字体标签形式给出的字体提示。在 CS 指标上各模型表现依旧接近；在 TCC 相关的指标（TCA、F1、NLS）上，表现最好的为 TextDiffuser 和 FLUX，即 FLUX 在语义复杂的 TCC 场景下具有比部分文本渲染专用模型更好的表现。进一步地，对比图 4.6 和图 4.7 可以发现，随着字体条件的引入，各模型的 TCC 指标均出现明显下降，表明 TFntC 对 TCC 产生了干扰。最后，在经推理时扩展后，SD3 的文本渲染表现出现明显下降，其余模型则基本保持稳定。

3）TColC 分析。表 4.6 和表 4.7 展示了各模型在 CD 上的量化评估结果。CD 数据子集侧重于评估模型的 TColC 表现。本研究通过颜色名称的方式为各模型指定文本颜色，该颜色名称真值被进一步转化为对应的颜色数值真值，用于细粒度的 CVD 颜色指标计算。加粗和下划线表示性能最好的结果和性能第二好的结果。由表 4.6 和表 4.7 可见，在 TColC 相关的指标（CVD、CNA）上表现最好的是 FLUX 和 SD3 这

表 4.6　GlyphControl、AnyText 和 TextDiffuser 在 ColorData 上的量化评估结果

Method	SampleNum	CS↑	IoU↑	TCA↑	F1↑	NLS↑	CVD↓	CNA↑
GlyphControl	1	0.3714	0.7377	**0.3302**	**0.3450**	**0.7084**	**87.34**	**0.4148**
	2	0.3715	0.7365	0.2913	0.3162	0.6647	92.60	0.3647
	3	0.3725	0.7354	0.2905	0.3183	0.6708	93.05	0.3550
	4	0.3731	0.7406	0.2869	0.3185	0.6635	93.20	0.3428
	5	0.3732	**0.7423**	0.2883	0.3226	0.6649	91.09	0.3560
	6	0.3735	0.7411	0.2876	0.3221	0.6668	91.45	0.3536
	7	0.3737	0.7398	0.2807	0.3162	0.6611	91.09	0.3526
	8	**0.3738**	0.7407	0.2825	0.3156	0.6640	90.25	0.3556
	9	0.3736	0.7407	0.2825	0.3145	0.6625	90.06	0.3563
	10	**0.3738**	0.7395	0.2897	0.3202	0.6680	90.23	0.3492
AnyText	1	0.3624	0.7273	0.3349	0.5760	0.6913	**72.94**	0.5731
	2	0.3648	**0.7315**	0.3175	0.5589	0.6608	74.54	**0.5764**
	3	0.3647	0.7310	0.3217	0.5654	0.6762	76.35	0.5517
	4	0.3641	0.7312	0.3377	0.5701	0.6888	76.30	0.5488
	5	0.3641	0.7292	0.3422	0.5759	0.6963	76.29	0.5513
	6	0.3649	0.7286	0.3437	0.5775	0.6964	77.07	0.5364
	7	0.3657	0.7298	0.3385	0.5723	0.6884	76.98	0.5350
	8	0.3662	0.7304	0.3444	0.5755	0.6922	76.59	0.5437
	9	0.3672	0.7309	0.3441	0.5763	0.6925	76.73	0.5402
	10	**0.3680**	0.7308	**0.3503**	**0.5805**	**0.6972**	76.25	0.5474
TextDiffuser	1	**0.3663**	**0.7574**	**0.6317**	**0.6354**	**0.8654**	92.62	0.2922
	2	0.3654	0.7557	0.6302	0.6341	0.8642	92.32	0.2811
	3	0.3654	0.7545	0.6291	0.6332	0.8623	92.26	0.2898
	4	0.3652	0.7552	0.6242	0.6296	0.8601	92.71	0.2867
	5	0.3651	0.7553	0.6257	0.6307	0.8597	92.83	0.2789
	6	0.3650	0.7560	0.6265	0.6315	0.8620	92.93	0.2829
	7	0.3652	0.7557	0.6240	0.6291	0.8618	92.46	0.2854
	8	0.3651	0.7559	0.6244	0.6298	0.8613	92.35	0.2885
	9	0.3651	0.7560	0.6243	0.6299	0.8612	92.25	0.2909
	10	0.3651	0.7560	0.6267	0.6323	0.8614	**92.10**	**0.2928**

两个通用模型，大幅领先三个文本渲染专用模型。造成该差距的原因可能在于：这两个通用模型具有更优更多样的训练数据，因此对颜色语义和视觉一致性有着更好的先验；而文本渲染专用模型无论是训练数据还是模型创新都重点关注 TCC，即更关注文本拼写而忽略了文本颜色。在 TCC 相关的指标（TCA、F1、NLS）上，表现最好的是 TextDiffuser。进一步地，对比图 4.6 中各模型在 SWD 上的表现可以发现，除了 SD3 外，TColC 对其余模型的 TCC 干扰甚微。最后，在经推理时扩展后，所有模型的文本渲染表现基本保持稳定，仅有小幅的指标波动。

4）TColC 和 TFntC 协同分析。表 4.8 和表 4.9 展示了各模型在 CFD 上的量化

表 4.7 Stable Diffusion 3 和 FLUX 在 ColorData 上的量化评估结果

Method	SampleNum	CS↑	IoU↑	TCA↑	F1↑	NLS↑	CVD↓	CNA↑
SD3	1	0.3667	-	**0.2302**	0.2419	**0.6515**	38.05	0.8681
	2	0.3665	-	0.2246	0.2387	0.6432	35.31	0.8534
	3	0.3663	-	0.2286	**0.2443**	0.6391	39.29	0.8433
	4	0.3663	-	0.2187	0.2315	0.6330	38.29	0.8454
	5	**0.3674**	-	0.2267	0.2405	0.6502	**34.80**	**0.8717**
	6	0.3669	-	0.2238	0.2385	0.6510	38.15	0.8410
	7	0.3668	-	0.2166	0.2312	0.6440	38.14	0.8379
	8	0.3665	-	0.2135	0.2272	0.6411	37.59	0.8457
	9	0.3659	-	0.2095	0.2231	0.6374	37.75	0.8394
	10	0.3661	-	0.2033	0.2158	0.6330	38.52	0.8350
FLUX	1	**0.3709**	–	0.3778	0.4194	0.7142	39.96	0.8262
	2	0.3704	-	0.3849	0.4247	0.7203	37.22	0.8372
	3	0.3701	-	0.3841	0.4229	0.7193	37.45	0.8377
	4	0.3699	-	0.3825	0.4183	0.7194	37.37	0.8411
	5	0.3698	-	0.3813	0.4166	0.7187	36.18	0.8482
	6	0.3697	-	0.3825	0.4180	0.7184	35.82	0.8507
	7	0.3696	-	0.3859	0.4209	0.7227	35.26	0.8561
	8	0.3696	-	0.3897	0.4248	0.7234	35.06	**0.8568**
	9	0.3695	-	0.3938	**0.4272**	**0.7242**	34.98	0.8514
	10	0.3694	-	**0.3941**	**0.4272**	0.7239	**34.72**	0.8518

评估结果。CFD 数据子集侧重于协同评估模型的 TColC 和 TFntC 表现。由表 4.8 和表 4.9 可见，此时 TColC 表现最好的依次是 SD3 和 FLUX，TFntC 表现最好的依次是 FLUX 和 SD3，TCC 表现最好的依次是 TextDiffuser 和 FLUX，ICC 表现上各模型没有显著差异。进一步地，与先前单独 TFntC 和单独 TColC 时的评估结果对照可见，各模型在协同 TColC 和 TFntC 时，TCC 表现与 TFntC 表现基本与单独 TFntC 时的表现持平，TColC 表现则基本与单独 TColC 时的表现持平。最后，在经推理时扩展后，所有模型的文本渲染表现仍然基本保持稳定。

总而言之，通过在本章提出的六个评估数据集上，基于文本渲染量化评估框架进行全面且大规模的实验，可以得出以下结论：1）文本渲染专用模型（TextDiffuser、AnyText、GlyphControl）专精于 TCC，而通用模型（FLUX、SD3）在 TColC 和 TFntC 上占优，其中 FlUX 表现较为均衡；2）TFntC 会对 TCC 产生较大的干扰，而 TColC 对 TCC 没有明显的影响，TColC 和 TFntC 之间也不存在明显的干扰；3）各个模型的 ICC 与文本渲染专用模型的 TPosC 通常保持稳定；4）在各种场景下，经推理时扩展后，各模型的文本渲染表现基本维持稳定。

表 4.8　GlyphControl、AnyText 和 TextDiffuser 在 ColorFontData 上的量化评估结果

Method	SampleNum	CS↑	IoU↑	TCA↑	F1↑	NLS↑	CVD↓	CNA↑	FNS↑	FTS↑
GlyphControl	1	0.3753	0.7259	**0.1918**	**0.2307**	**0.6181**	**82.40**	**0.4516**	**0.0254**	0.2544
	2	0.3775	0.7342	0.1786	0.2139	0.5901	89.94	0.4046	0.0236	0.2580
	3	0.3796	0.7360	0.1762	0.2110	0.5934	90.77	0.3915	0.0241	**0.2599**
	4	0.3801	0.7390	0.1709	0.2096	0.5828	93.77	0.3553	0.0232	0.2555
	5	0.3803	0.7401	0.1739	0.2137	0.5813	94.79	0.3496	0.0224	0.2544
	6	0.3810	0.7386	0.1803	0.2199	0.5847	93.74	0.3659	0.0225	0.2555
	7	0.3813	0.7382	0.1749	0.2144	0.5790	93.18	0.3677	0.0220	0.2546
	8	0.3811	0.7402	0.1737	0.2124	0.5820	92.79	0.3695	0.0227	0.2551
	9	0.3813	**0.7417**	0.1741	0.2130	0.5808	92.30	0.3690	0.0218	0.2552
	10	**0.3815**	0.7406	0.1788	0.2179	0.5854	92.95	0.3562	0.0219	0.2548
AnyText	1	0.3671	**0.7470**	**0.1531**	0.3981	**0.5222**	**74.74**	**0.5654**	**0.0179**	0.2270
	2	0.3688	0.7446	0.1316	0.3951	0.4981	78.77	0.5147	0.0198	0.2352
	3	0.3691	0.7438	0.1340	0.3951	0.5031	80.45	0.5000	0.0195	**0.2379**
	4	0.3696	0.7429	0.1505	0.4056	0.5178	79.66	0.5229	0.0216	0.2361
	5	0.3689	0.7419	0.1522	**0.4093**	0.5203	79.86	0.5280	0.0213	0.2338
	6	0.3700	0.7415	0.1507	0.4070	0.5195	80.06	0.5231	0.0208	0.2342
	7	0.3703	0.7425	0.1405	0.3961	0.5035	79.66	0.5160	0.0210	0.2351
	8	0.3710	0.7433	0.1441	0.3991	0.5072	79.62	0.5210	**0.0217**	0.2332
	9	0.3718	0.7426	0.1442	0.3994	0.5062	80.13	0.5154	0.0216	0.2340
	10	**0.3727**	0.7428	0.1471	0.4036	0.5115	79.36	0.5190	0.0215	0.2342
TextDiffuser	1	**0.3690**	0.7555	**0.6122**	**0.6238**	**0.8628**	90.43	0.3298	0.0283	0.2110
	2	0.3681	0.7526	0.5980	0.6111	0.8516	90.04	0.3309	0.0285	0.2114
	3	0.3684	0.7540	0.5878	0.6011	0.8495	90.18	0.3373	0.0300	**0.2124**
	4	0.3685	0.7548	0.5888	0.6020	0.8500	89.96	0.3397	0.0307	0.2114
	5	0.3686	0.7558	0.5841	0.5974	0.8494	89.77	0.3341	0.0312	0.2121
	6	0.3687	**0.7560**	0.5881	0.6015	0.8503	90.20	0.3341	0.0322	0.2107
	7	0.3684	0.7553	0.5878	0.6013	0.8506	89.83	0.3361	**0.0329**	0.2111
	8	0.3683	0.7557	0.5852	0.5992	0.8498	**89.72**	0.3395	0.0321	0.2105
	9	0.3681	0.7557	0.5805	0.5946	0.8496	89.93	0.3373	0.0318	0.2107
	10	0.3684	0.7550	0.5808	0.5941	0.8491	90.05	**0.3398**	0.0319	0.2102

(3) 基于推理时扩展与后过滤的文本渲染改善方法验证

为了验证基于推理时扩展与后过滤的文本渲染改善方法的有效性，本研究先对各模型进行推理时扩展，即对每一个自然语言提示进行 10 次采样，生成 10 张候选图片，再进行后过滤，最终保留其中的 4 张。对照组采用先前量化评估实验中正常采样 4 张的结果。需要说明的是，在先前的大规模量化评估实验中，本研究观察到三个文本渲染专用模型（GlyphControl、AnyText、TextDiffuser）的 TPosC 表现相对接近与稳定，而两个通用模型（SD3、FLUX）又无法进行精确的 TPosC。因此本研究在后过滤的过程中未考虑 TPosC，并在本章剩余实验中统一忽略 TPosC 相关的 IoU 指标，重点聚焦于 TCC、TCol、TFntC 以及 ICC。另外，在初步实验中，本研究发现 ColorFontData 子集在后过滤阶段存在异常和不稳定的情况，故本研究暂不在该子集

表 4.9 Stable Diffusion 3 和 FLUX 在 ColorFontData 上的量化评估结果

Method	SampleNum	CS↑	IoU↑	TCA↑	F1↑	NLS↑	CVD↓	CNA↑	FNS↑	FTS↑
SD3	1	**0.3841**	–	0.1327	0.1676	0.4987	47.53	0.7593	0.0149	**0.2803**
	2	0.3837	–	0.1296	**0.1749**	**0.5047**	41.41	0.8017	**0.0163**	0.2705
	3	0.3838	–	0.1156	0.1587	0.4759	40.12	0.8144	0.0147	0.2698
	4	0.3840	–	0.1133	0.1526	0.4705	38.49	0.8293	0.0135	0.2703
	5	0.3837	–	**0.1331**	0.1712	0.4984	**34.94**	**0.8532**	0.0137	0.2711
	6	0.3830	–	0.1323	0.1739	0.4932	38.77	0.8057	0.0145	0.2739
	7	0.3830	–	0.1259	0.1651	0.4821	39.63	0.7974	0.0149	0.2735
	8	0.3825	–	0.1258	0.1632	0.4856	40.08	0.7950	0.0158	0.2751
	9	0.3824	–	0.1227	0.1606	0.4796	40.23	0.7878	0.0152	0.2760
	10	0.3829	–	0.1220	0.1576	0.4789	40.59	0.7888	0.0160	0.2766
FLUX	1	0.3900	–	0.2633	0.3694	0.6153	47.71	0.7604	**0.0306**	**0.3110**
	2	0.3907	–	0.2786	0.3837	0.6217	**47.64**	0.7658	0.0294	0.3066
	3	0.3915	–	0.2878	0.3948	0.6258	50.38	0.7635	0.0286	0.3104
	4	0.3914	–	**0.2980**	**0.3957**	**0.6300**	50.15	0.7686	0.0291	0.3106
	5	**0.3917**	–	0.2939	0.3919	0.6282	49.88	0.7624	0.0283	0.3067
	6	**0.3917**	–	0.2915	0.3903	0.6274	50.12	0.7699	0.0277	0.3049
	7	0.3916	–	0.2915	0.3901	0.6276	49.65	0.7743	0.0283	0.3047
	8	0.3916	–	0.2913	0.3910	0.6275	49.60	0.7752	0.0283	0.3045
	9	0.3915	–	0.2934	0.3932	0.6284	49.45	**0.7791**	0.0280	0.3051
	10	0.3913	–	0.2941	0.3925	0.6292	49.37	0.7781	0.0276	0.3045

表 4.10 在 ColorData 上对各模型采用推理时扩展与后过滤的结果（原始 / 推理时扩展与后过滤）

Method	CS↑	TCA↑	F1↑	NLS↑	CVD↓	CNA↑
GlyphControl	0.3731 / **0.3736**	0.2869 / **0.3452**	0.3185 / **0.3806**	0.6635 / **0.6847**	93.20 / **86.71**	0.3428 / **0.3933**
AnyText	0.3641 / **0.3666**	0.3377 / **0.3944**	0.5701 / **0.6124**	0.6888 / **0.7106**	76.30 / **73.92**	0.5488 / **0.5770**
TextDiffuser	0.3652 / **0.3655**	0.6242 / **0.6627**	0.6296 / **0.6690**	0.8601 / **0.8720**	92.71 / **89.71**	0.2867 / **0.3215**
SD3	0.3663/ **0.3667**	0.2187 / **0.2996**	0.2315 / **0.3151**	0.6330 / **0.6642**	38.29 / **36.56**	**0.8454** / 0.8441
FLUX	0.3699 / **0.3700**	0.3825 / **0.5206**	0.4183 / **0.5556**	0.7194 / **0.8022**	37.37 / **35.21**	0.8411 / **0.8604**

上进一步测试，对其结果留待后续工作深入探讨。

图 4.8 展示了在 SingleWordData、SingleStringData、DoubleWordData 这三个数据子集上对各模型采用推理时扩展与后过滤的结果。可以看到，在 TCC 相关的指标（NLS、F1、TCA）上，推理时扩展与后过滤均带来了指标增益。究其原因，在先前的大规模量化评估中已经能发现各模型经推理时扩展后表现相对稳定，这意味着随着采样数的增加，正确进行文本渲染的生成图片数量也在增加；在后过滤阶段中，LVLM 较好地对全部生成图片进行了 TCC 相关的评估与筛选，因此最终经后过滤的输出结果相比原先有提升。另外，ICC 表现则前后无明显的差异。

表 4.10 展示了在 ColorData 上对各模型采用推理时扩展与后过滤的结果。在 ColorData 上，可以明显地观察到推理时扩展与后过滤对各模型的 TCC 和 TColC 均有提

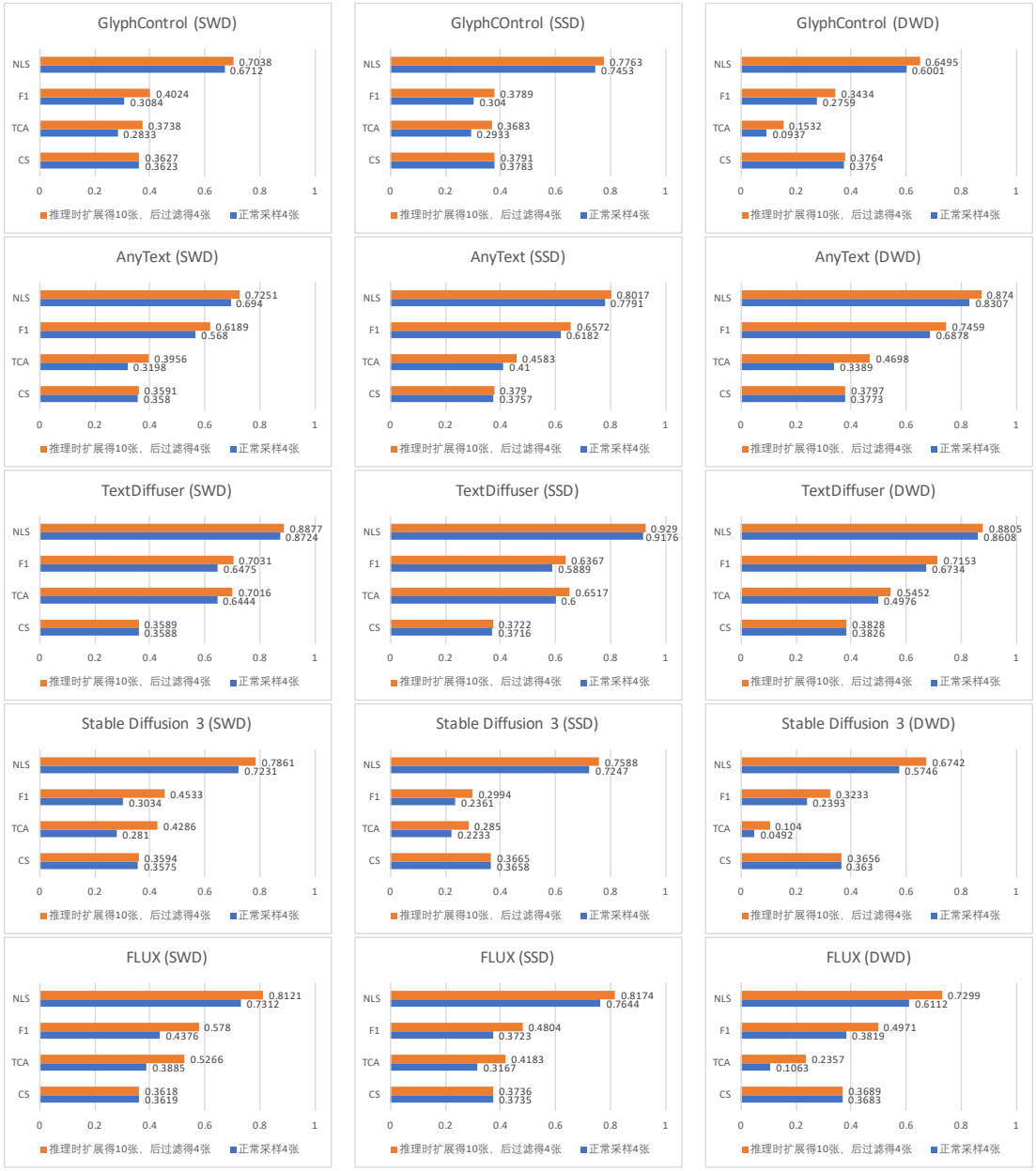


图 4.8 在 SingleWordData、SingleStringData、DoubleWordData 上对各模型采用推理时扩展与后过滤的结果

升，而 TCC 的增幅普遍大于 TColC。造成这一现象的主要原因在于，本研究使用的 LVLM 在文本检测与识别方面表现出色，能较为可靠地筛选出 TCC 正确的候选图片；而在叠加进行文本颜色识别时则偶有“幻觉”（即使本研究进行了多次提示优化仍无法解决），因此推理时扩展与后过滤对 TColC 有增益但相对有限。ICC 表现仍然是前后无明显差异。

表 4.11 展示了在 FontData 上对各模型采用推理时扩展与后过滤的结果。与在 ColorData 上的结果不同，表 4.11 呈现出较为杂散的模式：各模型在 TCC 上总体仍

表 4.11　在 FontData 上对各模型采用推理时扩展与后过滤的结果（原始 / 推理时扩展与后过滤）

Method	CS↑	TCA↑	F1↑	NLS↑	FNS↑	FTS↑
GlyphControl	0.3762 / **0.3778**	0.1485 / **0.1638**	0.1923 / **0.2222**	**0.5802** / 0.5732	**0.0259** / 0.0247	0.2549 / **0.2584**
AnyText	0.3674 / **0.3683**	**0.1347** / 0.1291	**0.4100** / 0.4011	**0.5103** / 0.4948	**0.0239** / 0.0234	0.2331 / **0.2345**
TextDiffuser	0.3649 / **0.3652**	0.6056 / **0.6224**	0.6246 / **0.6453**	0.8627 / **0.8657**	**0.0321** / 0.0313	0.2125 / **0.2134**
SD3	0.3795 / **0.3803**	0.1653 / **0.1995**	0.2295 / **0.2727**	0.4887 / **0.5050**	**0.0246** / 0.0235	0.2613 / **0.2710**
FLUX	0.3835 / **0.3836**	0.2898 / **0.3316**	0.4035 / **0.4649**	0.6180 / **0.6267**	0.0307 / **0.0331**	0.3013 / **0.3067**

有小幅增益，而在 TFntC 相关的指标（FNS、FTS）上未见一致的提升趋势。究其原因，一方面，各模型在 TFntC 上本就能力非常受限，提升空间有限；另一方面，本研究观察到所采用的 LVLM 对字体的评估能力仍有待提升，且叠加字体评估会干扰对 TCC 或 TColC 的评估，带来筛选的不稳定性（在 ColorFontData 上更为明显）。此时，ICC 表现依旧是前后无明显差异。

综上所述，本章提出的基于推理时扩展与后过滤的文本渲染改善方法，可以为各模型的 TCC 与 TColC 带来较为稳定的提升，而对 ICC 基本保持中性，对 TFntC 的增益则受 LVLM 字体评估能力所限，尚未表现出一致优势。本研究认为，在后续工作中若对后过滤所基于的 LVLM 进行专门优化与微调，有望进一步挖掘放大文本到图像生成模型自身既有的文本渲染能力。当然，现有文本到图像生成模型的文本渲染仍有可观的改善空间，尤其是在 TFntC 这一方面。

4.5　本章小结

本章首先提出了文本渲染量化评估框架，旨在从文本内容控制、文本颜色控制、文本字体控制、文本位置控制以及图片内容控制这五个维度，系统全面地探究现有文本到图像生成模型的文本渲染能力。通过对第三章中人工评估的实验进行回溯验证，本章验证了文本渲染量化评估框架的可靠性；通过对现有文本到图像生成模型进行大规模量化评估实验，本章探究了模型文本渲染能力的现状，验证了参与评估的模型在推理时扩展上的稳定性。在此基础上，本章提出了基于推理时扩展与后过滤的文本渲染改善方法，利用 LVLM 对经推理时扩展所得的候选图片进行评估与筛选，从而令被保留并最终输出的生成图片中包含渲染更佳的视觉文本。该方法与模型框架无关且无需额外训练。实验表明，该方法具备一定的有效性，为未来的文本渲染研究提供见解。

第五章　总结和展望

5.1　总结

随着以扩散模型为代表的新一代生成式人工智能技术的迅猛发展，文本到图像生成作为融合计算机视觉与自然语言处理两大领域的交叉研究方向，逐渐成为了学术界和工业界关注的热点技术之一。这种技术的兴起，不仅推动了数字艺术创作、广告视觉设计、虚拟内容生产等产业领域的深刻变革，也在日常生活场景和人机交互界面上展现出巨大的应用潜力。然而，在视觉内容生成要求日益精细化的背景下，人们不再满足于模型仅能准确地生成人物、动物或场景等常规视觉元素，而是进一步追求模型具备高度可控的视觉文本渲染能力，以实现标志设计、海报制作、广告创意及社交媒体内容创作等高精度视觉任务。这也对现有文本到图像生成模型提出了更高的挑战。

本文围绕改善文本到图像生成模型的文本渲染能力展开深入研究。本文的研究工作主要包括以下两个方面：

贡献一：提出基于混合专家的多属性可控文本渲染改善方法，解决文本渲染专用模型在文本颜色与字体控制方面能力不佳的问题。

针对当前文本渲染专用模型通常在文本内容与位置控制上表现突出，而在文本颜色和字体控制方面明显不足的现状，本文分析了这种不足的根本原因，即现有模型通常将图片背景、文本颜色和字体控制条件统一注入在自然语言提示中，导致属性与实体绑定混淆（控制条件耦合）和难以精准分配控制强度（控制力度分配不佳）。为此，本文构建了针对文本颜色和字体控制的专用合成式数据集，分别独立训练得颜色专家与字体专家，有效实现了文本颜色与字体控制条件的解耦。同时，为解决控制力度分配不佳的问题，本文进一步引入了自适应路由器，可以根据输入的特征动态地调整各专家间的协作强度，最终实现了颜色、字体与文本拼写控制力度的精准动态分配。实验结果表明，提出的方法不仅能够较大程度地保留模型原有的文本内容控制能力，还在文本颜色与字体控制能力上表现出显著提升，验证了混合专家机制及自适应路由器的有效性与可行性。

贡献二：构建五维自动化文本渲染量化评估框架，提出基于推理时扩展与后过

滤的文本渲染改善方法，解决现有文本渲染评估不全面以及改善方法难以通用于不同模型架构的问题。

针对当前文本渲染领域缺乏全面系统评估手段，特别是在文本颜色控制和文本字体控制维度上缺乏量化评价体系的现状，本文构建了一套覆盖文本内容控制、文本颜色控制、文本字体控制、文本位置控制和图片内容控制的五维文本渲染量化评估框架，实现了对任意架构模型文本渲染能力的客观、系统的量化评价。进一步地，针对当前通用文本到图像生成模型架构多样化且传统文本渲染改善方法难以直接迁移的问题，本文提出一种无需修改模型架构与参数的推理时扩展与后过滤方法：在模型推理阶段通过多次采样扩展生成更多候选图片，再利用大型视觉语言模型（LVLM）对候选图片集合进行自动化评估筛选，即插即用地挖掘与放大模型自身的文本渲染能力。实验结果验证了该方法在架构无关、无需训练的前提下能够有效提升现有模型的文本渲染表现，具有广泛的适用性和实用性。

5.2 展望

本文针对文本到图像生成模型的文本渲染改善进行了深入的研究并提出了相应的多属性可控文本渲染专用模型和评估方案，但依旧存在诸多不足和可以改进之处：

（1）本文提出的 FC-Render 完全基于合成式数据集进行训练，尽管这种方式相比于标注真实数据会极大地节省开销，却会限制生成图片中视觉文本与图片背景的深度融合。因此，未来可以探索将真实数据与合成数据混合用于训练，从而达到数据高效性与模型泛化性的更好平衡。此外，FC-Render 是从源头解耦控制条件注入的，这某种程度上会造成模型架构复杂和应用时的不便。因此，未来研究中一个潜在能提高这项工作易用性的途径是消除自然语言提示之外的单独控制机制，从而允许用户直接在自然语言提示中指定字体和文本颜色，并生成符合预期的图片。

（2）在构建文本渲染量化评估框架并进行大规模实验的过程中，本文意识到常用的 CLIP Score 可能已不再是能较好地评估图片内容控制的指标了。究其原因，随着文本到图像生成模型的迅猛发展，模型的生成图片肉眼可见地质量更高，但实测的 CLIP Score 却与早先的模型相差不大，无法提供高分辨率的评估视角。因此，未来的研究方向是探索更好的图片内容控制的评估指标。

（3）在采用 LVLM 进行后过滤的实验中，本文发现在要求 LVLM 进行相对复杂

与综合的评估时，LVLM 的输出会存在不稳定性，表现为 LVLM 可以较好地进行单一维度的评估，但随着评估维度的增多而逐渐展现出幻觉。因此，未来的研究方向可以是对 LVLM 进行微调，从而得到专用于生成图片评估与后过滤的模型。

未来，期望本文分析和提出的潜在研究方向与技术方案能够为更多研究者提供有益的启发，并促进后续研究在文本到图像生成领域中文本渲染技术的不断深化，进一步解决文本渲染领域当前尚未完全攻克的技术瓶颈，推动生成式人工智能技术更广泛、更深入地落地于产业界和教育界的实际应用场景。

插图索引

表格索引

参考文献

[1] HO J, JAIN A, ABBEEL P. Denoising diffusion probabilistic models[J]. Advances in neural information processing systems, 2020, 33: 6840-6851.

[2] LIU R, GARRETTE D, SAHARIA C, et al. Character-aware models improve visual text rendering[A]. 2022.

[3] MA J, ZHAO M, CHEN C, et al. Glyphdraw: Seamlessly rendering text with intricate spatial structures in text-to-image generation[A]. 2023.

[4] YANG Y, GUI D, YUAN Y, et al. Glyphcontrol: glyph conditional control for visual text generation[J]. Advances in Neural Information Processing Systems, 2023, 36: 44050-44066.

[5] TUO Y, XIANG W, HE J Y, et al. Anytext: Multilingual visual text generation and editing[A]. 2023.

[6] CHEN J, HUANG Y, LV T, et al. Textdiffuser: Diffusion models as text painters[J]. Advances in Neural Information Processing Systems, 2023, 36: 9353-9387.

[7] PEEBLES W, XIE S. Scalable diffusion models with transformers[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2023: 4195-4205.

[8] ROMBACH R, BLATTMANN A, LORENZ D, et al. High-resolution image synthesis with latent diffusion models[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 10684-10695.

[9] ESSER P, KULAL S, BLATTMANN A, et al. Scaling rectified flow transformers for high-resolution image synthesis[C]//Forty-first international conference on machine learning. 2024.

[10] LABS B F. Flux[EB/OL]. 2024. https://github.com/black-forest-labs/flux.

[11] TIAN K, JIANG Y, YUAN Z, et al. Visual autoregressive modeling: Scalable image generation via next-scale prediction[J]. Advances in neural information processing systems, 2024, 37: 84839-84865.

[12] 曹寅, 秦俊平, 马千里, 等. 文本生成图像研究综述[J/OL]. 浙江大学学报 (工学版), 2024, 58(2): 219. https://www.zjujournals.com/eng/CN/abstract/article_46169.shtml. DOI: 10.3785/j.issn.1008-973X.2024.02.001.

[13] MANSIMOV E, PARISOTTO E, BA J L, et al. Generating images from captions with attention[A]. 2015.

[14] REED S, AKATA Z, YAN X, et al. Generative adversarial text to image synthesis [C]//International conference on machine learning. PMLR, 2016: 1060-1069.

[15] ZHANG H, XU T, LI H, et al. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5907-5915.

[16] XU T, ZHANG P, HUANG Q, et al. Attngan: Fine-grained text to image generation with attentional generative adversarial networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 1316-1324.

[17] ZHU M, PAN P, CHEN W, et al. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 5802-5810.

[18] CHENG J, WU F, TIAN Y, et al. Rifegan: Rich feature generation for text-to-image synthesis from prior knowledge[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10911-10920.

[19] RAMESH A, PAVLOV M, GOH G, et al. Zero-shot text-to-image generation[C]// International conference on machine learning. Pmlr, 2021: 8821-8831.

[20] NICHOL A, DHARIWAL P, RAMESH A, et al. Glide: Towards photorealistic image generation and editing with text-guided diffusion models[A]. 2021.

[21] DING M, YANG Z, HONG W, et al. Cogview: Mastering text-to-image generation via transformers[J]. Advances in neural information processing systems, 2021, 34: 19822-19835.

[22] RAMESH A, DHARIWAL P, NICHOL A, et al. Hierarchical text-conditional image generation with clip latents: Vol. 1[A]. 2022: 3.

[23] SAHARIA C, CHAN W, SAXENA S, et al. Photorealistic text-to-image diffusion models with deep language understanding[J]. Advances in neural information processing systems, 2022, 35: 36479-36494.

[24] 高欣宇, 杜方, 宋丽娟. 基于扩散模型的文本图像生成对比研究综述[J]. 计算机工程与应用, 2024, 60(24): 44-64.

[25] 赖丽娜, 米瑜, 周龙龙, 等. 生成对抗网络与文本图像生成方法综述[J]. 计算机工程与应用, 2023, 59(19): 21-39.

[26] ZHANG C, ZHANG C, ZHANG M, et al. Text-to-image diffusion models in generative ai: A survey[A]. 2023.

[27] 王威, 李玉洁, 郭富林, 等. 生成对抗网络及其文本图像合成综述[J]. 计算机工程与应用, 2022, 58(19): 14-36.

[28] 刘泽润, 尹宇飞, 薛文灏, 等. 基于扩散模型的条件引导图像生成综述[J/OL]. 浙江大学学报（理学版）, 2023, 50(6): 651-667. https://doi.org/10.3785/j.issn.1008-9497.2023.06.001.

[29] ZHANG N, TANG H. Text-to-image synthesis: A decade survey[A]. 2024.

[30] 龚帅, 邓勇, 向金海. 基于扩散模型的图像生成方法研究综述[J/OL]. 武汉大学学报 (工学版), 2025, 58(02): 292-305. DOI: 10.14188/j.1671-8844.2024.0148.

[31] 管凤旭, 张涵宇, 路斯棋, 等. 扩散模型在计算机视觉领域的研究现状[J]. 智能系统学报, 2025, 20(02): 265-282.

[32] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[33] KARRAS T, LAINE S, AILA T. A style-based generator architecture for generative adversarial networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4401-4410.

[34] KARRAS T, LAINE S, AITTALA M, et al. Analyzing and improving the image quality of stylegan[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 8110-8119.

[35] KARRAS T, AITTALA M, LAINE S, et al. Alias-free generative adversarial networks [J]. Advances in neural information processing systems, 2021, 34: 852-863.

[36] SILVA T. An intuitive introduction to generative adversarial networks (gans)[EB/OL]. 2018. https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/.

[37] ZHANG H, XU T, LI H, et al. Stackgan++: Realistic image synthesis with stacked generative adversarial networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(8): 1947-1962.

[38] LEE M, SEOK J. Controllable generative adversarial network[J]. Ieee Access, 2019, 7: 28158-28169.

[39] QIAO T, ZHANG J, XU D, et al. Mirrorgan: Learning text-to-image generation by redescription[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 1505-1514.

[40] LI W, ZHANG P, ZHANG L, et al. Object-driven text-to-image synthesis via adversarial training[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 12174-12182.

[41] LIANG J, PEI W, LU F. Cpgan: Content-parsing generative adversarial networks for text-to-image synthesis[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16. Springer, 2020: 491-508.

[42] RUAN S, ZHANG Y, ZHANG K, et al. Dae-gan: Dynamic aspect-aware gan for text-to-image synthesis[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 13960-13969.

[43] TAO M, TANG H, WU F, et al. Df-gan: A simple and effective baseline for text-to-image synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 16515-16525.

[44] KANG M, ZHU J Y, ZHANG R, et al. Scaling up gans for text-to-image synthesis[C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 10124-10134.

[45] SAUER A, KARRAS T, LAINE S, et al. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis[C]//International conference on machine learning. PMLR, 2023: 30105-30118.

[46] TAO M, BAO B K, TANG H, et al. Galip: Generative adversarial clips for text-to-image synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 14214-14223.

[47] XU Y, ZHAO Y, XIAO Z, et al. Ufogen: You forward once large scale text-to-image generation via diffusion gans[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 8196-8206.

[48] SAUER A, SCHWARZ K, GEIGER A. Stylegan-xl: Scaling stylegan to large diverse datasets[C]//ACM SIGGRAPH 2022 conference proceedings. 2022: 1-10.

[49] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[50] ESSER P, ROMBACH R, OMMER B. Taming transformers for high-resolution image synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 12873-12883.

[51] VAN DEN OORD A, VINYALS O, et al. Neural discrete representation learning[J]. Advances in neural information processing systems, 2017, 30.

[52] HAN J, LIU J, JIANG Y, et al. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis[C]//Proceedings of the Computer Vision and Pattern Recognition Conference. 2025: 15733-15744.

[53] DING M, ZHENG W, HONG W, et al. Cogview2: Faster and better text-to-image generation via hierarchical transformers[J]. Advances in Neural Information Processing Systems, 2022, 35: 16890-16902.

[54] GAFNI O, POLYAK A, ASHUAL O, et al. Make-a-scene: Scene-based text-to-image generation with human priors[C]//European Conference on Computer Vision. Springer, 2022: 89-106.

[55] YU J, XU Y, KOH J Y, et al. Scaling autoregressive models for content-rich text-to-image generation: Vol. 2[A]. 2022: 5.

[56] DAI X, HOU J, MA C Y, et al. Emu: Enhancing image generation models using photogenic needles in a haystack[A]. 2023.

[57] WU C, LIANG J, JI L, et al. Nüwa: Visual synthesis pre-training for neural visual world creation[C]//European conference on computer vision. Springer, 2022: 720-736.

[58] YU L, SHI B, PASUNURU R, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning[A]. 2023.

[59] KOH J Y, FRIED D, SALAKHUTDINOV R R. Generating images with multimodal language models[J]. Advances in Neural Information Processing Systems, 2023, 36: 21487-21506.

[60] LU J, CLARK C, LEE S, et al. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 26439-26455.

[61] MA X, ZHOU M, LIANG T, et al. Star: Scale-wise text-to-image generation via auto-regressive representations[A]. 2024.

[62] TANG H, WU Y, YANG S, et al. Hart: Efficient visual generation with hybrid autoregressive transformer[C]//The Thirteenth International Conference on Learning Representations.

[63] TENG Y, SHI H, LIU X, et al. Accelerating auto-regressive text-to-image generation with training-free speculative jacobi decoding[A]. 2024.

[64] HE W, FU S, LIU M, et al. Mars: Mixture of auto-regressive models for fine-grained text-to-image synthesis[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 39. 2025: 17123-17131.

[65] GU J, WANG Y, ZHANG Y, et al. Dart: Denoising autoregressive transformer for scalable text-to-image generation[A]. 2024.

[66] SONG J, MENG C, ERMON S. Denoising diffusion implicit models[C]// International Conference on Learning Representations.

[67] HO J, SALIMANS T. Classifier-free diffusion guidance[A]. 2022.

[68] KINGMA D P, WELLING M, et al. Auto-encoding variational bayes[M]. Banff, Canada, 2013.

[69] RADFORD A, KIM J W, HALLACY C, et al. Learning transferable visual models from natural language supervision[C]//International conference on machine learning. PmLR, 2021: 8748-8763.

[70] PODELL D, ENGLISH Z, LACEY K, et al. Sdxl: Improving latent diffusion models for high-resolution image synthesis[A]. 2023.

[71] BETKER J, GOH G, JING L, et al. Improving image generation with better captions [J]. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf, 2023, 2(3): 8.

[72] FENG Z, ZHANG Z, YU X, et al. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 10135-10145.

[73] CHEN J, YU J, GE C, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis[A]. 2023.

[74] ZHENG W, TENG J, YANG Z, et al. Cogview3: Finer and faster text-to-image generation via relay diffusion[C]//European Conference on Computer Vision. Springer, 2024: 1-22.

[75] TENG J, ZHENG W, DING M, et al. Relay diffusion: Unifying diffusion process across resolutions for image synthesis[C]//The Twelfth International Conference on Learning Representations.

[76] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. Journal of machine learning research, 2020, 21(140): 1-67.

[77] LIPMAN Y, CHEN R T, BEN-HAMU H, et al. Flow matching for generative modeling[A]. 2022.

[78] SHU Y, ZENG W, ZHAO F, et al. Visual text processing: A comprehensive review and unified evaluation[A]. 2025.

[79] SHU Y, ZENG W, LI Z, et al. Visual text meets low-level vision: A comprehensive survey on visual text processing[A]. 2024.

[80] BAI Y, HUANG Z, GAO W, et al. Intelligent artistic typography: A comprehensive review of artistic text design and generation[J]. APSIPA Transactions on Signal and Information Processing, 2024, 13(1).

[81] XUE L, BARUA A, CONSTANT N, et al. Byt5: Towards a token-free future with pre-trained byte-to-byte models[J]. Transactions of the Association for Computational Linguistics, 2022, 10: 291-306.

[82] ZHAO Y, LIAN Z. Udifftext: A unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models[C]//European Conference on Computer Vision. Springer, 2024: 217-233.

[83] LIU Z, LIANG W, LIANG Z, et al. Glyph-byt5: A customized text encoder for accurate visual text rendering[C]//European Conference on Computer Vision. Springer, 2024: 361-377.

[84] LIU Z, LIANG W, ZHAO Y, et al. Glyph-byt5-v2: A strong aesthetic baseline for accurate multilingual visual text rendering[A]. 2024.

[85] ZHANGLI Q, JIANG J, LIU D, et al. Layout-agnostic scene text image synthesis with diffusion models[C]//2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2024: 7496-7506.

[86] CAO P, ZHOU F, SONG Q, et al. Controllable generation with text-to-image diffusion models: A survey[A]. 2024.

[87] ZHANG L, RAO A, AGRAWALA M. Adding conditional control to text-to-image diffusion models[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2023: 3836-3847.

[88] MOU C, WANG X, XIE L, et al. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models[C]//Proceedings of the AAAI conference on artificial intelligence: Vol. 38. 2024: 4296-4304.

[89] HUANG L, CHEN D, LIU Y, et al. Composer: creative and controllable image synthesis with composable conditions[C]//Proceedings of the 40th International Conference on Machine Learning. 2023: 13753-13773.

[90] YE H, ZHANG J, LIU S, et al. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models[A]. 2023.

[91] ZHANG L, CHEN X, WANG Y, et al. Brush your text: Synthesize any scene text on images via diffusion model[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 38. 2024: 7215-7223.

[92] PALIWAL S, JAIN A, SHARMA M, et al. Customtext: Customized textual image generation using diffusion models[A]. 2024.

[93] MA J, DENG Y, CHEN C, et al. Glyphdraw2: Automatic generation of complex glyph posters with diffusion models and large language models[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 39. 2025: 5955-5963.

[94] TUO Y, GENG Y, BO L. Anytext2: Visual text generation and editing with customizable attributes[A]. 2024.

[95] JIANG B, YUAN Y, BAI X, et al. Controltext: Unlocking controllable fonts in multilingual text rendering without font annotations[A]. 2025.

[96] CHEN J, HUANG Y, LV T, et al. Textdiffuser-2: Unleashing the power of language models for text rendering[C]//European Conference on Computer Vision. Springer, 2024: 386-402.

[97] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context [C]//Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13. Springer, 2014: 740-755.

[98] YE M, ZHANG J, LIU J, et al. Hi-sam: Marrying segment anything model for hierarchical text segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025, 47(03): 1431-1447.

[99] BITEN A F, TITO R, MAFLA A, et al. Scene text visual question answering[C/OL]// 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019: 4290-4300. DOI: 10.1109/ICCV.2019.00439.

[100] ZYDDNYS. manga-image-translator[EB/OL]. 2025. https://github.com/zyddnys/manga-image-translator.

[101] SHARKD. Rgb color solid cube[EB/OL]. 2008. https://commons.wikimedia.org/w/index.php?curid=3375025.

[102] SHARKD. Hsv color solid cone[EB/OL]. 2015. https://commons.wikimedia.org/wiki/File:HSV_color_solid_cone.png.

[103] MUNROE R P. Color survey results[EB/OL]. 2010. https://blog.xkcd.com/2010/05/03/color-survey-results/.

[104] AI S. Font classify[EB/OL]. 2025. https://github.com/Storia-AI/font-classify.

[105] TEAM T G F. Fonts[EB/OL]. 2025. https://github.com/google/fonts.

[106] TEAM Q. Qwen2.5-vl[EB/OL]. 2025. https://qwenlm.github.io/blog/qwen2.5-vl/.

攻读硕士学位期间取得的研究成果

发表论文

[1] FC-Render: Adaptive Font- and Color-Aware Text Diffusion Model. IEEE International Conference on Image Processing (ICIP 2025). （第一作者，CCF-C，第三章）

[2] Enhancing Visual Text Rendering with Fine-Grained Font and Color Control. Neurocomputing. （第一作者，SCI，Under Review，第四章）

致　谢

终于到了落笔毕业论文致谢的时候了，是二十五年来的第二次写，也可能是这辈子的最后一次写。纸短情长，谨向这一阶段所有接纳与承托我的师长、亲友致以最诚挚的谢意。愿以详尽笔录，长抵漫漫岁月对记忆的冲刷与侵蚀。

感谢王教授与朱教授。从论文的选题到定稿，王老师给予了我充分的信任，以渊博的学识与前瞻的眼光为我指点迷津，以严谨的态度教会我做人做事的道理，以包容的胸怀接纳我成长中的愚钝与无知。朱教授讲述自己丰富的经历与阅历，教诲我不要试图复制别人的人生，而要敢于开辟属于自己的道路。一日为师，终身为父。

感谢课题组的同学以及室友。王同学与尹同学作为我的同辈，在我感到困惑时向我伸出援手；陈同学、李同学与张同学作为我的前辈，在我提出问题时对我倾囊相授。一个人可以走得很快，但一群人才能走得更远。愿诸君前程似锦，江湖再见。

感谢我的父母。在低谷，是你们没有放弃我，是你们坚定地相信我。母亲在家中不分日夜的陪伴，父亲在远方打来电话的关切，是这段时间里让我此生难忘的回忆。父母赋予了我生命，也塑造了我的性格。是母亲教会我对任何事都要积极主动和尽心尽力，是父亲让我明白任何时候都要保持情绪稳定与沉稳理智。懵懂时有过顶撞，愧未当面致歉，在此诚挚悔过。对父母的感恩难以诉尽，愿用我的余生守护与回报。

感谢我的伴侣。异地少见但亲切不变，通话仓促但关怀满盈。在煎熬的日子里，是你事无巨细地了解状况并为我出谋划策，是你在我妄图逃避时鼓励我直面困难，是你在我平静讲述挫折时为我心疼而潸然落泪。长路漫漫，若不嫌弃，愿与君同行。

感谢我的猫。许多个挑灯的深夜里，是椅子与柜子上轻微的动静与鼾声，让我心安；是转头间频频撞见的静默凝视，让我不觉孤单。愿悉心照料，常伴左右。

感谢上海大学，七年的庇护让我积淀学识并褪去稚气。此行别后，唯余不舍。

感谢参与我论文评审的学者与专家，你们的宝贵意见让这篇论文得以尽量完善。

最后，感谢这段时光。纵使不愿回首，我仍心怀感激。正是这次淬炼，让我浴火重生。携所得教训与经验，我将踏实前行。谨以此文，献给所有平凡而又不凡的人。

陈斌

2025 年 6 月 25 日