

基于提示工程的程序设计探索与实践^{*}

王兵书¹ 李静怡¹ 雍珊珊² 李树一³

1. 西北工业大学软件学院, 西安 710129
2. 深圳北理莫斯科大学工程系, 深圳 518172
3. 北京工业大学信息学部, 北京 100124

摘要 本文研究了提示工程在程序设计领域的应用与实践, 包括生成代码、验证代码以及添加代码注释, 实现将提示工程用于专业课程编程设计。通过设计具有提示性的文本描述, 引导大语言模型生成符合特定要求的代码, 提高了代码开发的效率和质量。实验结果证明了该方法在代码生成和代码修正方面的有效性, 不仅为教师教学提供了强有力的技术支持, 也有助于推动教育领域的数字化转型。

关键字 提示工程, 代码生成, 代码验证, ChatGPT

The Exploration and Practice of Programming Design Based on Prompt Engineering

WANG Bingshu¹ LI Jingyi¹ YONG Shanshan² LI Shuyi³

1. School of Software, Northwestern Polytechnical University, Xi'an 710129, China, wangbingshu@nwpu.edu.cn
2. Faculty of Engineering, Shenzhen MSU-BIT University, Shenzhen 518172, China, yongss@smbu.edu.cn
3. Department of Information, Beijing University of Technology, Beijing 100124, China, syli2022@bjut.edu.cn

Abstract—This paper investigates the application and practicality of prompt engineering in the field of programming, encompassing code generation, code validation, and code annotation, aiming to integrate prompt engineering into the professional course programming design. By devising instructive textual prompts, the study guides large language models to produce code tailored to specific requirements, thereby enhancing the efficiency and quality of code development. Experimental findings validate the efficacy of this method in both code generation and correction, offering substantial technical support for educators and contributing to the advancement of digital transformation in education.

Keywords—prompt engineering, code generation, code validation, ChatGPT

1 引言

提示工程 (Prompt Engineering, PE) 是指通过设计和编写具有提示性的文本描述, 以引导大语言模型 (Large Language Model, LLM) 生成符合特定要求的答案输出过程, 也是一个对任务情境、用户需求、情报问题、情报分析流程等方面的信息组织过程。目前已在自然语言处理、文本生成、对话系统、信息检索等领域发挥作用。

提示工程作为一项新兴技术, 为教师教学水平的提升提供了强有力的技术支持。在当前的教育体系, 传统的课堂教学模式时间有限, 但教师往往需要讲授大量的知识, 学生可能会错过某些重要信息或概念。

其次, 随着学生人数的增加, 教师难以对每一位同学都提供个性化的指导和答疑。另外, 每位学生的学习目标、风格、行为和偏好等各不相同, 由于师资力量有限, 很难得到及时且针对性的帮助。提示工程可以将自然语言固化为提示词, 根据个性化教学任务需求挖掘 AI 大模型的潜力, 大幅提升 LLM 的总结、转化、分类、拓展等任务能力, 从而获取更加符合需求且精准的答案。因此, 研究一种提示工程引导的新型教学模式对提升高校教师的教学水平具有重要意义。

本文采用循序渐进的提示技术策略, 集成零样本提示、少样本提示、生成知识提示和自动提示工程师, 最终实现了一种智能化的提示生成器, 如图 1 所示。通过结合实际的教学服务场景, 深层次、全方位地挖掘学生需求, 将其转化为具体的情报问题, 再结合大语言模型特征, 利用提示工程技术将情报问题精确转换为提示词, 以获取更加精准的答案, 为学生提供个性化的答疑解惑服务。

^{*} 基金资助: 西北工业大学教育教学改革研究项目(重点): 24GZ11259; 国家自然科学基金青年基金: 62102318, 62306021

^{**} 通讯作者: 李树一 syli2022@bjut.edu.cn。

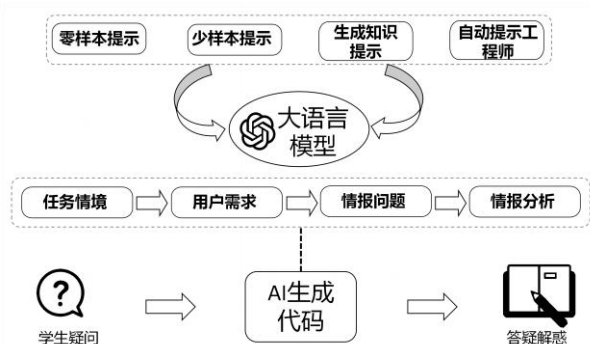


图 1 基于提示工程构建 ChatGPT 生成代码的策略

2 相关研究

人工智能技术在智能问答方面有明显进步，利用 LLM 结合提示工程的方法对程序设计的探索具有启发意义。时宗彬等^[1]以对话形式剖析了从文献中抽取材料的实体和类型的问题，提出了一种易于扩展、准确且高效的有机电池材料抽取方法。该方法利用 LLM，采用人机协作的模式，通过构建层次化提示指令，依次识别单篇论文中给体-受体体系核心主题句、核心材料以及材料类型特征句，然后在特征句基础上识别相应的实例信息。针对查询式摘要数据集稀缺以及科研人员个性化查询需求，商锦铃等^[2]通过构建生成与自我校验提示链，基于 ChatGPT 和提示工程提出一种以大语言模型为“数据标注员”的自动化数据标注方法，构建了自然语言处理领域学术会议记录查询摘要数据集 AMTQFSum。

编程人员的开发和代码审查经验不足会导致智能合约漏洞引发安全问题，可以借助 LLM 对代码进行审查。耿辰等^[3]提出一种在零样本场景下基于提示工程的智能合约漏洞检测方法 Prompt-enhanced ChatGPT，使用标准提示文本的 ChatGPT 为研究对象，通过将传统的漏洞检测任务转化为文本问答任务，利用模型的推理能力进行检测。并且将智能合约源码经过预处理去除冗余信息，设计包含“任务描述”“漏洞描述”“检测步骤”“推理过程”和“答案格式”的提示文本模版，从而生成漏洞检测结果和可解释的分析。

在大语言模型和提示工程的支持下，针对教育领域解决交互质量问题成为新的研究重点。王群等^[4]结合新工科建设方法和学校人才培养目标，研究人才培养模式的改革和创新之路，提高面向新工科的应用型人才培养质量。乔文豹等^[5]提出“一体多翼”的理论和实践教学改革，在非计算机类专业学生中推广开源教育，培养学生解决跨专业技术信息化和智能化问题的能力。焦建利^[6]认为 LLM 能够为师生生成所需要的文本材料如活动方案、教学设计、练习题等。刘明等^[7]以 LLM 的三大技术路线 BERT、T5 和 GPT 系列为主要研究对象，探索其在赋能高阶思维培养、加强阅读理解能

力以及提升写作与数学解题水平等教学应用场景中的潜力，为智能化教学与教育数字化转型提供新思路、新方法以及可持续发展路径。赵晓伟等^[8]提出了教育提示语设计的“CORE”框架，该框架由必备要素（语境、目标）和可选要素（角色、规则、示例）构成，归纳基础提示、示例提示、角色提示、规则提示、组合提示五种提示语类，提供了结构化提示模版设计范例。方海光等^[9]设计了基于三轮对话的 T|BO-LRQET-P&P（简称 LRQET）模型，通过教育提示工程的理论来设计学习提示单以助力师生的教与学，这不仅能满足教师的教学需求，也能满足学生的个性化学习与自主学习的需要。曲海鹏等^[10]分析了大语言模型工具对程序设计类课程教学带来的挑战，以及对于各种不规范的使用提出防范方法和措施。Wang 等^[11]探讨了 AI 工具在实际应用中的价值和局限性，为在科研和教育领域中的使用提供了参考和指导。汪芳等^[12]基于人工智能对话机器人的程序设计教学，利用人工智能技术促进了编程学习与个性化教育，为教师提供了创新的思维与经验。李志刚等^[13]提出了一套基于 ChatGPT 交互问答的“五步法”教学方案，在翻转课堂教学实践中探索了 ChatGPT 交互的创新应用，改进了问题驱动教学模式的实施效果。

在生成式人工智能技术快速发展的背景下，Louie G^[14]认为提示工程为研究人员提供了有效利用大语言模型的关键技能，帮助他们在学术追求中保持利用前沿技术的领先地位。汪彦婷等^[15]分析了当前“科技论文写作”课程教学存在的问题，基于“四步法”模型提出了一系列改革措施，融合生成式人工智能于“三阶段六环节”模型。Jiang 等^[16]提出了一种结合大型语言模型的自规划代码生成方法，通过两个主要阶段——规划阶段和实施阶段，来帮助模型更好地理解复杂意图并降低问题解决的难度。在规划阶段，LLM 结合提示从意图中规划出简洁的解决方案，随后在实施阶段，模型在解决方案步骤的指导下逐步生成代码。通过实验表明，与直接利用 LLM 进行代码生成的方法相比，自规划代码生成在多个代码生成数据集上取得了显著的性能提升。邹勤等^[17]分析人工智能专业的培养要求和授课学生的知识基础，提出“典型算法编程练习激发学生潜力”和“渐进难度实训引导学生提高动手能力”的教学方案，为人工智能本科教学提供了有力支持。Wang 等^[18]对 ChatGPT 辅助教育教学的最新发展进行了研究，发现 LLM 在教育领域的应用给不同受众带来了积极影响，这为未来 ChatGPT 在教育领域的应用提供了坚实的基础。此外，他还设计了一个通用信息素养框架^[19]，为教学过程和能力提升提供了支持。

提示工程在各个应用领域都展现了广泛的适用性和高效性，已经在多个领域取得了显著成果。然而，

在教学领域,进行了很多探索和尝试^[20],目前尚缺乏将提示工程应用于专业课程教学设计方面的研究。因此,需要探索一种基于提示工程的 LLM 教学模式,使得不同背景的教师能够运用自然语言并将其固化为提示词进而挖掘 LLM 最大潜力。本文致力于将提示工程应用到程序设计教学中,提出了一种基于提示工程的代码生成与验证方法。

3 基于提示工程的程序设计

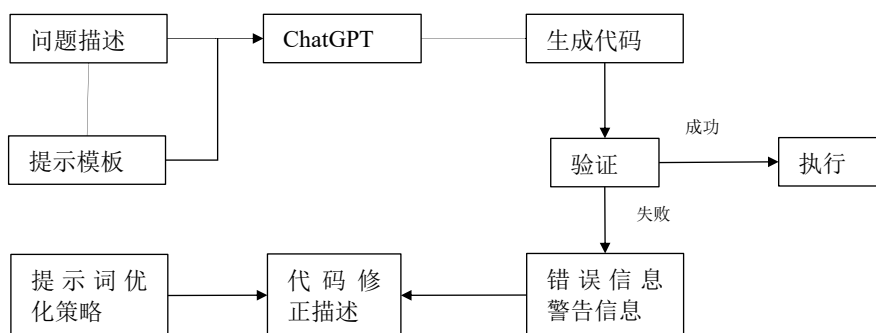


图 2 基于 ChatGPT 和提示工程的代码生成流程图

首先,利用提示模板对问题进行具体阐述,输入到 ChatGPT 中。通过其强大的自然语言处理能力,生成符合要求的代码。随后,对生成的代码进行严格的验证,如果验证通过,对其进行执行。反之,不通过则接受到错误信息与警告信息,将其与提示词优化策略结合进一步对代码修正进行描述,输入到 ChatGPT 中,迭代循环,直到验证通过为止。

3.1 基于提示工程的代码生成与验证

随着软件行业的快速发展,传统的代码编写方式已经无法满足高效、高质量的开发需求。鉴于此,本文提出了一种基于提示工程的代码生成流程,通过清晰的问题描述,利用提示工程智能生成代码,并进行代码的正确性检验和评估。本流程为用户提供了一种准确高效的代码生成方法,有助于提高代码开发的效率和质量,如图 2 所示。

3.2 代码添加注释

本文提出了一种基于提示工程的代码注释生成方法,如图 3 所示。首先,输入一段需要添加注释的代码,利用提示工程设计出相应的提示模板,引导 ChatGPT 深入分析并理解代码的逻辑和结构,生成与该代码相对应的注释。

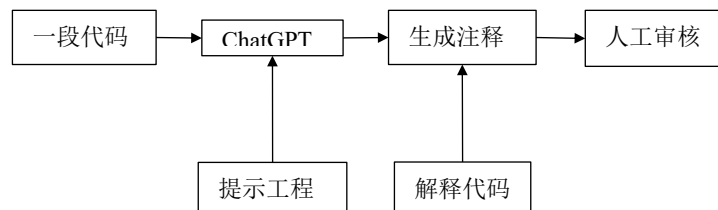


图 3 基于 ChatGPT 和提示工程的代码注释流程图

这些注释能够清晰地解释代码的功能、输入输出、依赖关系等关键信息,增强了代码的可读性与可维护性。为了确保代码注释的准确性和相关性,可以使用其他工具或者人工审核来验证生成的注释。

4 实现效果分析

4.1 代码生成

为了验证基于提示工程的代码自动生成方法,本文选择了六种经典且易于理解的算法作为实验对象,包括二分查找,找两个数的最大公约数、求二维数组中最大元素及下标、用 Heron 公式计算三角形面积、

冒泡排序和汉诺塔求解。通过精心设计的提示模板,引导 ChatGPT 生成相应的 C 语言程序,为学生提供了一种全新的学习途径。

六组典型示例如下:

(1) 二分法查找

```

int BinarySearch(int A[],int n,int find)
{
    int low,upper,mid;
    low=0,upper=n-1;
    while(low<=upper)
    {
        mid=low+(upper-low)/2;
        if(A[mid]<find) low=mid+1;
        else if(A[mid]>find) upper=mid-1;
    }
}
  
```

```

        else return mid;
    }
    return -1;
}
#define N 10
int main()
{
    int A[N]= {8,24,30,47,62,68,83,90,92,95};
    int i,find;
    scanf("%d",&find);
    i=BinarySearch(A,N,find);
    if(i>=0) printf("A[%d]=%d\n",i,find);
    else printf("not found\n");
    return 0;
}

```

(2) 找两个数的最大公约数

```

int gcd(int m,int n);
int main()
{
    int m,n;
    scanf("%d%d",&m,&n);
    printf("%d\n",gcd(m,n));
    return 0;
}
int gcd(int m,int n)
{
    int r;

    while(n!=0)
    {
        r=m%n;
        m=n;
        n=r;
    }

    return m;
}

```

(3) 求二维数组中最大元素及下标

```

int findmax(int A[3][4],int B[2])
{
    int i,j,max,r=0,c=0;
    max=A[r][c];
    for(i=0; i<3; i++)
        for(j=0; j<4; j++)
            if(A[i][j]>max)
                {
                    r=i,c=j;
                    max=A[r][c];
                }
    B[0]=r,B[1]=c;
    return max;
}
int main()
{
    int A[3][4]= {{7,5,-2,4},{5,1,9,7},{3,2,-1,6}};
    int B[2];
    int max;
    max=findmax(A,B);
    printf("max:A[%d][%d]=%d\n",B[0],B[1],max);
    return 0;
}

```

(4) 用 Heron 公式计算三角形面积

```

int main()
{
    double a,b,c;
    printf("input a,b,c:");
    scanf("%lf%lf%lf",&a,&b,&c);
    if(a+b>c&&a+c>b&&b+c>a)
    {
        double s,t;
        t=(a+b+c)/2.0;
        s=sqrt(t*(t-a)*(t-b)*(t-c));
        printf("area=%lf\n",s);
    }
}

```

```

    }
    else printf("error\n");
    return 0;
}

```

(5) 冒泡排序

```

#define N 10
int main()
{
    int A[N],i,j,t;
    for(i=0; i<N; i++)
        scanf("%d",&A[i]);
    for(j=0; j<N-1; j++)
        for(i=0; i<N-1-j; i++)
            if(A[i]>A[i+1])
                {
                    t=A[i];
                    A[i]=A[i+1];
                    A[i+1]=t;
                }
    for(i=0; i<N; i++)
        printf("%d ",A[i]);
    return 0;
}

```

(6) 汉诺塔求解

```

void Hanoi(int n,char A,char B,char C)
{
    if(n==1) printf("%c->%c",A,C);
    else
    {
        Hanoi(n-1,A,C,B);
        printf("%c->%c",A,C);
        Hanoi(n-1,B,A,C);
    }
}
int main()
{
    int n;
    printf("input n:");
    scanf("%d",&n);
    Hanoi(n,'A','B','C');
    return 0;
}

```

基于以上实验，本文设计如下供参考的提示模板：

(1) 任务描述：描述想要生成代码的功能或解决的问题。

(2) 详细要求：代码应该满足的具体要求，如实现的具体功能，性能、安全性或其他非功能性需求。

(3) 编码规范：使用 C 语言进行编程，并添加必要的注释。

(4) 代码实例：提供输入输出示例。

示例如下：

任务目标：编写一个 C 语言函数，用于计算两个整数的最大公约数（GCD）。

详细要求：

(1) 函数接口应设计为接收两个整数作为参数。

(2) 函数应使用有效的算法（如欧几里得算法）来计算这两个整数的最大公约数。

(3) 函数应返回计算得到的最大公约数。

编码规范:

(1) 遵循标准的 C 语言编程规范和命名习惯。

(2) 在代码中添加必要的注释, 解释计算最大公约数的核心逻辑部分。

代码实例:

给定两个整数: 48 和 18

期望的最大公约数: 6

ChatGPT 输出代码如下:

```
#include <stdio.h>
// 计算两个整数的最大公约数 (GCD)
int gcd(int a, int b) {
    int temp;
    // 使用欧几里得算法求最大公约数
    while (b != 0) {
        temp = b;
        b = a % b;
        a = temp;
    }
    // 返回最大公约数
    return a;
}
int main() {
    int num1 = 48, num2 = 18;
    // 调用gcd函数计算最大公约数
    int result = gcd(num1, num2);
    // 输出结果
    printf("给定两个整数: %d 和 %d\n", num1,
num2);
    printf("期望的最大公约数: %d\n", result);
    return 0;
}
```

为确保生成的代码不仅语法正确, 且逻辑上也满足实际需求。将 ChatGPT 生成的代码复制到 C 语言开发环境中, 通过测试验证了正确性。实验结果表明, 基于提示工程的代码自动生成方法能够高效地生成准确、可用的代码, 从而显著提高了代码开发的效率和质量。

4.2 代码修正

对六组典型的示例进行代码生成得到的验证结果均正确, 证明了基于提示工程的代码生成方法有效。本文进一步研究该方法在代码修正方面的应用, 设计出参考的代码修正提示模块如下:

(1) 任务描述: 请对提供的 C 语言代码进行仔细审查, 识别并指出代码中存在的任何错误或潜在问题。在成功识别错误后, 请提供修正建议或直接给出修正后的正确代码段。

(2) 修正要求: 修正所有识别出的语法错误、逻辑错误或潜在的运行时问题。确保修正后的代码能够正确编译并运行, 达到预期的功能。保持代码风格的一致性, 并添加必要的注释以提高代码的可读性。

(3) 基于提示模版, 将错误代码段提供给 ChatGPT, 请求其进行错误分析并提供修正后的代码。

示例如下:

任务描述: 请对提供的 C 语言代码进行仔细审查, 识别并指出代码中存在的任何错误或潜在问题。在成功识别错误后, 请提供修正建议或直接给出修正后的正确代码段。

错误代码如下:

```
int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int i, sum;
    for(i = 0; i <= 5; i++) {
        sum += arr[i];
    }
    printf("Sum of array elements: %d\n", sum);
    return 0;
}
```

修正要求:

(1) 修正所有识别出的语法错误、逻辑错误或潜在的运行时问题。

(2) 确保修正后的代码能够正确编译并运行, 达到预期的功能。

(3) 保持代码风格的一致性, 并添加必要的注释以提高代码的可读性。

ChatGPT 输出修正后的代码如下:

```
#include <stdio.h> // 添加头文件
int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int i, sum = 0; // 初始化 sum 变量为 0
    // 修正循环迭代条件
    for(i = 0; i < 5; i++) {
        sum += arr[i];
    }
    // 输出数组元素的和
    printf("Sum of array elements: %d\n", sum);
    return 0;
}
```

4.3 教学改革与实践效果

为了评估提示工程在编程教育中的实际应用效果, 对学生关于教学改革与实践结果进行了问卷调查, 收集到问卷 90 份。

表 1 展示出学生在使用 ChatGPT 生成代码时, 经常调整提示词的人数占比达到了 43.18%, 有时会调整的人数占比为 42.05%。这表明学生对提示工程有较深理解, 85.23% 的学生会使用提示工程辅助编程, 调试代码, 提升了编码的工作效率。通过提示工程来提升 AI 工具的效果已经为大部分学生接受, 学生对教学中引入提示工程的相关内容有较高的参与度和实践意愿。

表 2 展现出有 80.68% 的学生认为提示工程可以显著提升 AI 工具的效果, 仅有 3.41% 的参与者认为提示

工程对提升效果无关紧要，而 15.91%的参与者还不确定。80.68%的高比例说明提示工程的重要性已经得到广泛认可，在课程中引入提示工程将有助于大多数学生更有效地使用 AI 工具，从而增强学习效果。同时，在当前 AI 技术快速发展的背景下，提升学生在提示工程方面的认知将对其未来的职业发展产生积极影响。

表 1 学生在使用 ChatGPT 时调整提示词的频率

选项	比例
经常会	43.18%
有时会	42.05%
很少会	7.95%
从不会	6.82%

表 2 学生对于提示工程提升 AI 效果的认知

选项	比例
是	80.68%
否	3.41%
不确定	15.91%

表 3 说明了 50%的学生明确表示希望在课程中更深入地学习提示工程技术和方法，42.05%的学生则表现出一定的兴趣但尚未完全确定。在课程设置中，将纳入提示工程是有需求且合理的，能够满足大部分学生的学习需求。在教学过程中，采用提示工程引导学生进行代码生成及验证，不仅能够降低教师辅导学生的压力，而且能够为学生提供个性化的辅导。

表 3 学生对于深入学习提示工程的意愿

选项	比例
希望更深入学习	50%
有一些兴趣，但不确定是否需要	42.05%
不感兴趣	3.41%
是具体情况而定	4.55%

基于以上结果，教师在编程实践类教学中可以对提示工程开展进一步深入的教学改革：

(1) 引入提示词开发和优化技术：在程序设计课程体系中，引入提示工程技术与相关 AI 技术的教学内容。通过增设相关特色课程，确保学生不仅能及时接触到这些前沿技术，还能深入理解其背后的原理与应用场景，从而极大地拓宽知识面和技术视野。

(2) 强化提示工程在编程中应用的实践环节：为了让学生真正掌握并灵活运用提示技术，教学中应增加与“零样本/少样本提示”、“思维链提示”等相关的实践教学环节，提升学生的编程应用能力和问题解决能力。

5 结束语

本文深入探讨了基于提示工程的程序设计理念，通过实践验证了其可行性与正确性。相较于直接寻找代码的方法耗时长且难以符合期望要求，参照提示模版引导代码生成的方法可以实现快速提供符合要求的代码，显著提高了代码开发的效率与准确性，有助于为学生提供个性化的教学服务。未来，我们将继续探索提示工程在程序设计领域的应用，并进一步完善和优化该方法，以推动编程教育改革的持续改革创新。

参考文献

- [1] 时宗彬, 乐小虬. 基于本地大语言模型和提示工程的材料信息抽取方法研究[J]. 数据分析与知识发现, 2024, 1-13.
- [2] 商锦铃, 张建勇. 基于 ChatGPT 和提示工程的查询式摘要数据集 AMTQFSum 构建研究[J]. 数据分析与知识发现, 2024, 1-15.
- [3] 耿辰, 常舒予, 黄海平. 零样本场景下基于提示工程的智能合约漏洞检测研究[J]. 信息对抗技术, 2024, 3(2): 70-81.
- [4] 王群, 李秋丽, 陈蒙. 面向新工科的软件工程应用型人才培养模式研究[J]. 计算机技术与教育学报, 2022, 10(3): 39-42.
- [5] 乔文豹, 李宁, 梁旭, 曾铮. 开源运动和 ChatGPT 引起的高校跨专业计算机教育改革[J]. 计算机技术与教育学报, 2023, 11(4): 61-65.
- [6] 焦建利. ChatGPT: 学校教育的朋友还是敌人? [J]. 现代教育技术, 2023, 33(04): 5-15.
- [7] 刘明, 吴忠明, 廖剑, 等. 大语言模型的教育应用: 原理、现状与挑战——从轻量级 BERT 到对话式 ChatGPT[J]. 现代教育技术, 2023, 33(08): 19-28.
- [8] 赵晓伟, 祝智庭, 沈书生. 教育提示语工程: 构建数智时代的认识论新话语[J]. 中国远程教育, 2023, 43(11): 22-31.
- [9] 方海光, 孔新梅, 刘慧薇, 等. 基于共生理论的人机协同教育主体合作博弈及其优化策略研究[J]. 电化教育研究, 2024, 45(01): 21-27.
- [10] 曲海鹏, 刘培顺, 马慧, 等. 智能语言模型工具对程序设计类课程教学的挑战与应对[J]. 计算机教育, 2024, (05): 107-110+117.
- [11] Wang B, Zhang X, Li S, et al. The Practice of Enhancing Learning and Scientific Innovative Abilities Using LLM-Based AI Tools[C]. 2024 6th International Conference on Computer Science and Technologies in Education (CSTE), 2024.
- [12] 汪芳, 赵左, 王毅航, 等. 人工智能在程序设计教学应用中的探索与实践[J]. 计算机教育, 2023(11): 45-50.
- [13] 李志刚, 杨吉斌, 张睿, 等. 基于 ChatGPT 的程序设计翻转课堂教学方法实践[J]. 计算机技术与教育学报, 2023, 11(2): 125-129.
- [14] Louie G. Prompt Engineering with ChatGPT: A Guide for Academic Writers[J]. Annals of Biomedical Engineering, 2023, 51(12): 2629-2633.

- [15] 汪彦婷, 成研, 王丽. 生成式AI背景下“科技论文写作”课程改革探索与实践[J]. 计算机技术与教育学报, 2024, 12(1): 56-61.
- [16] Jiang X, Dong Y, Wang L, et al. Self-planning Code Generation with Large Language Models[J]. ACM Transactions on Software Engineering and Methodology, 2023.
- [17] 邹勤, 杜博, 陈驰, 周剑. “人工智能程序设计实训”教学探讨[J]. 计算机技术与教育学报, 2024, 12(1): 51-55.
- [18] Wang B, Li S, Dong Y, et al. ChatGPT-Aided Education Teaching[C]. 2024 6th International Conference on Computer Science and Technologies in Education (CSTE), 2024.
- [19] Wang B, Zhang X, Li S, et al. The Practice of Enhancing Learning and Scientific Innovative Abilities Using LLM-Based AI Tools[C]. 2024 6th International Conference on Computer Science and Technologies in Education (CSTE), 2024.
- [20] 谢晓艳, 谢晓巍, 曹伟. 面向能力培养的程序设计基础课程改革实践[J]. 计算机技术与教育学报, 2022, 10(3): 90-93.