所属类别	2025 年“华数杯”全国大学生数学建模竞赛	参赛编号
本科组		CM2500313

标题（可调控生物节律的 LED 光源研究与分析）

摘要

LED 光源照明用于调节人的生理节律，使人眼具有正常观感是当前照明领域研究的重点之一。本文正是围绕 LED 光源是否具有调节生物节律的功能展开。文中从光源效力的量测、符合人眼视觉得光谱的定制化优化、模仿自然界光影的动态调光、统计学分析临床验证结果这四个方面对有关发光二极管具有控制人体生理节律的作用进行说明和阐述。具体内容如下：

针对问题一：本题首先根据 LED 光源给定的 SPD 数据，依据 CCT、色差与距离普朗克轨迹 Duv、R_f 保真度指数、R_f色域指数及 MEL-DER 褪黑素日光照度比，以 CIE 及 ANSI/IES 为准绳，按照以上五个指标采取相应的 CIE 或 ANSI/IES 标准算法，基于光谱转 CIEXYZ,可完成综合判断颜色外观、显色性、节律效应，进而经由黑体辐射轨迹拟合、色差模型对比、与标准色样的比对判断。

针对问题二：基于加权线性叠加的混光模型利用给定的 5 个 LED 通道光谱(暗红、绿、蓝、暖白、冷白)来实现，对日间照明模式和夜间助眠模式,分别建立优化模型，通过数值优化求解得到最优通道权重组合，获得合成频谱关键性能参数。

针对问题三：根据给定的全天太阳光谱序列，利用问题二中的 LED 混光模型，设计了时间序列控制策略，使合成光谱在不同时间段与目标太阳光谱尽可能接近。通过优化各时间点的通道权重，实现了色温与节律效应的动态变化拟合。选取三个代表性时间点，绘制了合成光谱与目标太阳光谱的对比图，并分析了相似性与差异来源，为仿生节律照明系统设计提供了验证依据。

针对问题四：采用照明光环境下的光照条件，并使用具有 11 名健康被试交叉睡眠实验的数据进行研究，在每次实验的总睡眠时间（TST）、睡眠效率（SE）、入睡潜伏期（SOL）、深睡眠比例(N3%)、REM 睡眠比例、夜间醒来次数等指标的基础上，利用重复测量方差分析以及配对检验来考察光照条件对于睡眠质量的影响。

关键词：LED 混光；光谱优化；相关色温（CCT）；保真度指数（R_f）；色域指数（R_g）；褪黑素日光照度比（$mel-DER$）；昼夜节律；睡眠质量

一、 问题重述

1.1 问题背景

发光二极管(LED)是目前占据主导地位的高效节能、生态环保照明的当代延续者，其发光性质决定可覆盖各种形式的照明要求，且因其具有色温选择性和可调制性等优势，从而使其具有多样的性能特性。然而有研究表明，对于分叉作用来说，LED 并非是一个优于所有的可选方案，该现象能够同时影响视网膜上的感光细胞并作用于人类的生理节律系统（称之为“昼夜节律”或者“生物钟”）。

科学研究发现：一定波长范围的特定波长，例如一定波长的蓝光可以降低褪黑素的合成，降低褪黑素浓度水平从而影响我们的睡眠、记忆力乃至情绪。故此，对科学的灯光设计而言，不仅可以在白天提高工作效率，还可以改善夜晚的安全睡眠，避免夜间光线的不合理造成人的正常生理节奏紊乱；同时，LED 光源还需要满足照明高标准的技术要求并改善光谱属性，在保持光源自身的发光性能的同时还要利于人体健康的光生物节律的良性调节。如何在设计中将健康照明与光环境融合起来是对设计师的巨大挑战。

1.2 要解决的问题

综上所述，为了回答这 4 个逐级深入的问题，我们分别建立如下数学模型：

光谱参数计算与评价：针对给定的 LED 光源光谱功率分布（SPD）数据，通过建立标准化数学模型，计算相关色温（CCT）、色偏差（Duv）、色彩保真度指数（R_f）、色域指数（R_g）及褪黑素日光照度比（$mel-DER$）五个核心参数，以此评估光源性能。

多通道光谱混光优化：根据这种两种特殊模式：一种是采用模拟正午日光、色度保真度高的“日间照明模式”；另一种是采取低色温的柔温光源、有助睡眠、干扰最少生理节律的“夜间助眠模式”。利用红光、绿光、蓝光、暖白光、冷白光 5 个独立 LED 发光通道，建立多目标优化模型，求出最优通道驱动权重组合。

动态仿生光谱控制：结合上述五通道 LED 系统与早 8:30 至晚 19:30 的全天太阳光谱时间序列数据，设计动态控制策略，使合成光谱实时追踪并模拟太阳光的光色特性与节律效应，实现自然光环境的全天候复现。

睡眠质量实验验证：根据真实临床睡眠实验中客观睡眠指标的数据，以合理的统计学手段，在不同光照（优化光、普通光、暗）环境比较下客观睡眠指标对比结果对“夜间助眠模式”的实际效果予以判定，并确定优化光谱统计差异是否能证明具有改善人体睡眠质量的作用。

二、 问题分析

本研究围绕可调控生物节律的 LED 光源设计与验证，依次解决了光谱特性评估、LED 混光优化、动态仿生光谱控制以及睡眠质量验证等四个核心问题。

2.1 问题一的分析

本问题的核心是对已经给出的光谱数据进行测量并做出标准评价，然后就可以得到衡量出代表光谱学和生理学特征的重要参数。难点在于指标计算方法的多样性与精确性。相关色温（CCT）需要通过黑体辐射轨迹拟合计算，色偏差（Duv）依赖色度空间的几何关系，保真度指数（R_f）与色域指数（R_g）需引入 CIE TM-30 色样库进行计算，$mel-DER$涉及生物光敏感度曲线的积分。

我们的关键思路是首先，把率分布（SPD）转换到 CIE1931XYZ 色彩空间，并应用标准公式逐项计算指标，使得各项指标的数值统一，确保前后数据的可比性和权威性，利于之后完成光谱优化，我们就是以这个方式来推进任务的工作进程。

2.2 问题二的分析

本问题旨在利用多种 LED 通道光谱实现不同场景下的最佳照明效果。不同指标之间存在冲突，例如提升显色性（R_f）可能会降低节律抑制效果（$mel-DER$）；同时，通道功率需满足物理非负性与总功率限制。

我们的思路是建立 LED 混光的线性叠加模型，令合成光谱等于各通道光谱按权重系数加权求和。将优化目标分为两类：

- 日间模式：保持高R_f、适中R_g、色温接近 6500K；
- 夜间模式：降低$mel-DER$、保持R_f≥80、色温接近 3000K。

我们采用非负最小二乘（NNLS）方法或带约束的多目标优化算法，获得满足多重指标的通道权重组合。

2.3 问题三的分析

本问题扩展了静态混光到全天动态变化的光谱控制。这个问题的难点在于不同时间点目标光谱特性差异显著，需要在色温、R_f、$mel-DER$等多个维度同时拟合，并保证时间序列的平滑过渡。

我们的思路是将一天（以每半小时作为一个点）分为不同的时间点，在各时间点下求解最优通道权重的方法是根据问题二的混光模型，令合成光谱尽可能接近目标太阳光谱，并且考虑到色温以及节律效应，在此下平滑时间顺序避免灯光忽明忽暗的变化，室内照明因此得到有效的保障，白天或黑夜情况下都可以令光源更好符合自然光条件，并实现昼夜节律友好性。

2.4 问题四的分析

本问题是对优化光谱实际生理效应的验证，属于实验数据统计分析。难点在于受个人差异及环境干扰较大，睡眠资料是多指标、多被试的重复测量资料，采用恰当的方法来加以控制误差

我们以阶段原始资料计算总睡眠时间(TST)、睡眠效率(SE)、入睡潜伏期(SOL)、深睡比例(N3%)、REM 比例、夜间觉醒次数等；再对不同光照条件做配对 T 检验或者重复测量方差分析，得到各项指标有无差异，如：优化光、普通 LED、黑暗；最后进行结果分析，评价光照优化提高睡眠质量的效果，有助于普及推广健康照明产品的实现，该模型有其可行性和实用性。

三、 模型假设

为确保所建立的数学模型能够科学合理地解决问题，做出如下基本假设：

3.1 光谱线性叠加假设

假设多通道 LED 光源合成的总光谱功率分布等于各通道的独立 SPD 线性相加，而通道间驱动权值与其光输出强度成正比，且互不影响，通道间光谱性质彼此独立，所以假设每个通道输出的能量均等相混合以形成目标合成光谱，即假定其是均值型合成的。

3.2 数据准确性与代表性假设

假设提供的所有数据（LED 通道 SPD、太阳光谱时间序列数据、临床睡眠实验数据等）均为准确、可靠且无重大测量误差，且可代表目标人群的平均光学与生理反应特性。假设用于计算的 CIE 标准观察者函数（如$x(\lambda),y(\lambda),z(\lambda)$）生理节律敏感度函数（$S_{mel}(\lambda)$）与能准确刻画人眼视觉与非视觉响应特性。

3.3 光源系统稳定性假设

假设 LED 光源在工作期间光谱特性稳定，不受工作时长、环境温度、湿度或电源波动等外部因素影响。假设控制系统能够精确实现模型计算所得的驱动权重，且不存在显著延迟、噪声干扰或非线性失真。

3.4 睡眠实验理想条件假设

假设在睡眠实验中已通过交叉试验设计，排除了个体差异因素的影响，除了光之外（白天的活动量，饮食，噪声干扰，精神压力等），都把可能影响睡眠的各种因素进行了严格的限制，使所有受试者的睡眠条件都一样。如果睡眠各个阶段标示正确的话，并且按标准的多导睡眠图（PSG）的判读准则，分期正确，则与分期资料相符。

3.5 评估指标完整性假设

假设研究中使用的所有光源性能指标（$CCT,Duv,R_f,R_g,mel-DER$）与睡眠质量指标（TST、SE、SOL、N3%、REM%、Awakenings）均能够全面、合理反映光照与睡眠质量之间的关系，并可用于建立与验证数学模型。

四、 符号说明

符号	意义
	问题一
$\boldsymbol{SPD(\lambda)}$	光谱功率分布 （Spectral Power Distribution），描述光源在不同 波长上的能量分布
$\boldsymbol{\lambda}$	光的波长 （Wavelength），单位为纳米 （nm）
$\boldsymbol{S_i(\lambda)}$	第 i 个 LED 通道的光谱功率分布

$S(\lambda)$	多个 LED 通道线性组合后合成的总光谱
w_i	第 i 个 LED 通道的驱动权重，满足$\sum w_i = 1,\ w_i \geq 0$
w	由各通道驱动权重 w_i 组成的向量
$V(\lambda)$	CIE 1924 光度效率函数，描述人眼对不同波长光线的视觉敏感度
$S_{mel}(\lambda)$	黑视蛋白的光谱敏感度函数，描述视网膜对光线的非视觉生理反应
X, Y, Z	CIE 1931 标准色度系统中的三刺激值
$x-(\lambda), y-(\lambda), z-(\lambda)$	CIE 1931 标准观察者色匹配函数
u, v	CIE 1960 UCS（Uniform Color Space）色度坐标
	问题二
CCT	相关色温（Correlated Color Temperature），单位为开尔文（K）
Duv	色偏差，光源色度点到普朗克轨迹（黑体辐射轨迹）的距离
R_f	色彩保真度指数（Fidelity Index），评价光源还原物体本色的能力
R_g	色域指数（Gamut Index），评价光源再现色彩饱和度的能力
$mel-DER$	褪黑素日光照度比（melanopic Daylight Efficacy Ratio），量化光源对生理节律的影响
	问题三
M	LED 通道数，本题取 M = 5
$S_{sum}(\lambda, t)$	目标太阳光谱
$S_{LED}(\lambda, t; W(t))$	LED 混光得到的合成光谱
$W(t) = [w_1(t), \ldots, w_M(t)]^T$	不同 LED 通道在时刻t的驱动权重
w_i^{max}	第 i 个 LED 通道的最大驱动功率

$E_{spec}(t)$	光谱差异误差指标（目标光谱与 LED 合成光谱的 L2 范数差）
$s(t)$	目标状态约束（如期望的 CCT, R_f, R_g 范围等）
	问题四
TST	总睡眠时间（Total Sleep Time），所有非清醒阶段的总时长
SE	睡眠效率（Sleep Efficiency），TST 与总卧床时间的百分比
SOL	入睡潜伏期（Sleep Onset Latency），关灯到首次进入任一睡眠阶段的时间
N3%	深睡眠（N3 期）占总睡眠时间的百分比
REM%	快速眼动睡眠（REM 期）占总睡眠时间的百分比
Awakenings	夜间醒来次数
s	睡眠阶段状态集合 $\{Wake, N1/2, N3, REM\}$
$X_{i,n,t}$	第 n 个被试在夜间第t个时间段（30s 采样）的状态
$\overline{Z}_{in}$	第 n 个被试在夜间的照明特征（如CCT， $mel-DER$，蓝光强度等）
$\overline{Z}_{\iota}$	照明特征的总体均值
$\eta_{ad,in}$	状态转移 $a \to d$ 的个体随机效应
$b_{i,ad}$	第i个被试在转移 $a \to d$ 中的个体固定效应
β_{ad}	照明特征对状态转移 $a \to d$ 的回归系数
α_{ad}	状态转移 $a \to d$ 的基线速率（无照明影响时）
σ_u^2	个体随机效应的方差
λ_{ad}	状态转移 $a \to d$ 的条件速率（连续时间马尔科夫过程 CTMC 中的速率参数）

五、 模型的建立与求解

5.1 针对问题一模型的建立与求解

5.1.1 关于颜色特性参数的求值分析：相关色温（CCT） 与色偏差（Duv）

色度参数[1]是描述光源 “颜色” 的核心指标。我们依据 CIE 1931 标准色度系统，通过将 LED 光源的光谱功率分布（SPD）与色匹配函数积分，计算出三刺激值 X、Y、Z，并对 X、Y、Z 归一化得到色品坐标 x,y：

$$X = \int SPD(\lambda)\bar{x}(\lambda)d\lambda$$
$$Y = \int SPD(\lambda)\bar{y}(\lambda)d\lambda$$
$$Z = \int SPD(\lambda)\bar{z}(\lambda)d\lambda$$

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}$$

通过色品坐标 x,y 推导相关色温（CCT），需先计算中间变量 n，再代入温度拟合公式：

$$T = -437n^3 + 3061n^2 - 6861n + 5513.31$$

式中

$$n = (x - 0.3320)/(y - 0.1858)$$

色偏差 Duv [2]用于量化光源色品点与“黑体轨迹”的偏离程度，需通过坐标转换实现。

首先利用三刺激值 X、Y、Z 计算 CIE1976 色度空间坐标：

$$u' = \frac{4X}{X+15Y+3Z}, \qquad v' = \frac{9Y}{X+15Y+3Z}$$

随后通过公式算出CCT，带入 Chebyshev 得到黑体轨迹在 CIE1960 UCS 的坐标：

$$\begin{cases} \bar{u}(T) = \dfrac{0.860117757 + 1.54118254\times10^{-4}T + 1.28641212\times10^{-7}T^2}{1 + 8.42420235\times10^{-4}T + 7.08145163\times10^{-7}T^2} \\ \bar{v}(T) = \dfrac{0.317398726 + 4.22806245\times10^{-5}T + 4.20481691\times10^{-8}T^2}{1 - 2.89741816\times10^{-5}T + 1.61456053\times10^{-7}T^2} \end{cases}$$

再转换到 CIE1976 色度空间，得到参考黑体坐标：

$$v'_{bb} = \frac{3}{2}v_{bb}, \quad u'_{bb} = u_{bb}$$

最终，色偏差 Duv 为光源实际坐标（u'、v'）与参考黑体坐标的欧氏距离：

$$D_{uv} = \sqrt{(u' - u'_{bb})^2 + (v' - v'_{bb})^2}$$

基于光谱（SPD）与色匹配函数积分，先得到三刺激值 X、Y、Z，经归一化生成 CIE 1931 色品坐标 x、y，再转换为 CIE 1976 坐标（u'、v'）。利用 x、y 推导中间变量 n，

代入拟合公式算出相关色温（CCT）；结合温度 T 计算黑体轨迹参考坐标，与实际坐标通过欧氏距离算出色偏差（Duv）。整套流程严格遵循 CIE 标准色度系统，实现对光源“颜色特性”（色温、色偏差）的精准量化，为照明设计、光谱优化提供核心依据。

5.1.2 颜色还原参数的求值分析

为精准量化光源的色彩还原能力，我们依据 IES TM-30-18 标准，通过以下公式计算核心参数 R_f 及 R_g。

色彩保真度指数[3]（R_f）用于衡量光源与参考光源（同色温黑体/日光）相比，对 99 个标准色样（CES）颜色的再现精度。计算基于色样在待测光源与参考光源下的平均色差（ΔE）,即 99 个标准色样在两种光源下的平均色差，差值越小，R_f越接近 100，代表色彩保真度越高,公式为：

$$R_f = 100 - 3.514 \times \overline{\Delta E_j}.$$

色域指数（R_g）描述光源可呈现颜色范围（色域）相对参考光源的变化，通过 16 个标准色样在 CAM02-UCS 色空间中面积比值计算:

$$R_g = 100 \times \frac{A_{test}}{A_{ref}}$$

通过 IES TM-30-18 标准，利用 R_f（基于平均色差）衡量色彩“还原真实度”，R_g（基于色空间面积比）衡量色彩“呈现范围”，两者结合可全面评估光源的色彩渲染能力，为照明设计中“色彩品质”的量化提供核心依据。

5.1.3 生理节律效应参数的求值分析

光源对人体生理节律的作用主要是对非视觉系统（褪黑素分泌[4]的调控等），CIES 026:2018 标准基于此定义，其计算原理是计算光源激发视网膜神经节细胞中的黑视蛋白的量并量化光源有效激发视网膜中黑视蛋白的效果。公式为：

$$mel - DER = \frac{\int SPD(\lambda) S_{mel}(\lambda) d\lambda / \int SPD(\lambda) V(\lambda) d\lambda}{\int SPD_{D65}(\lambda) S_{mel}(\lambda) d\lambda / \int SPD_{D65}(\lambda) V(\lambda) d\lambda}$$

分子部分为计算待测光源激发黑视蛋白的“相对效率”，即待测光源光谱下，黑视蛋白响应与视觉感知的比值；分母部分为计算标准日光 D65 激发黑视蛋白的“相对效率”（作为基准）；最终，$mel - DER$ 通过“待测光源相对效率”与“标准日光相对效率”的比值，实现生理效应的标准化量化——比值越接近 1，表明待测光源黑视蛋白激发特性越接近自然光 D65；偏离 1 则反映其对生理节律（如褪黑素分泌）的调节差异。

该指标为评估光源对人体生物钟的影响提供了关键量化依据，支撑“健康照明设计”中生理节律调控的科学分析。

根据给定的光谱数据输入该流程，得到的计算结果汇总于表 5.1。

参数	$CCT(K)$	Duv	R_f	R_g	$mel - DER$

数值	3855.89	0.003	91.79	106.54	0.147

表 5.1：问题一中五个参数结果（接上表）

结果显示，该光源相关色温（CCT）达 3855.89K，属中性白光范畴。色偏差 Duv 为 0.003，极贴近普朗克轨迹，色品纯正度高。色彩保真度指数 R_f 高达 91.79 ，说明对色彩还原精准度优；色域指数 R_g 为 106.54，意味着相比参考光源，能呈现更饱和鲜亮的色彩表现。尤为关键的是，其黑视蛋白日光有效性比率（$mel-DER$）仅 0.147 ，数值较低，表明该光源在保障高质量照明的同时，对夜间褪黑素分泌抑制作用弱，具备良好生物友好性，契合健康照明需求。

本问题聚焦于对给定 LED 光源光谱功率分布（SPD）的全面评估，通过计算相关色温（CCT）、色偏差（Duv）、色彩保真度指数（ R_f ）、色域指数（R_g ）及黑视蛋白日光有效性比率（$mel-DER$）这五大核心参数，构建起一套系统的光源质量综合评价体系，为照明应用中的性能分析与选型提供量化依据。

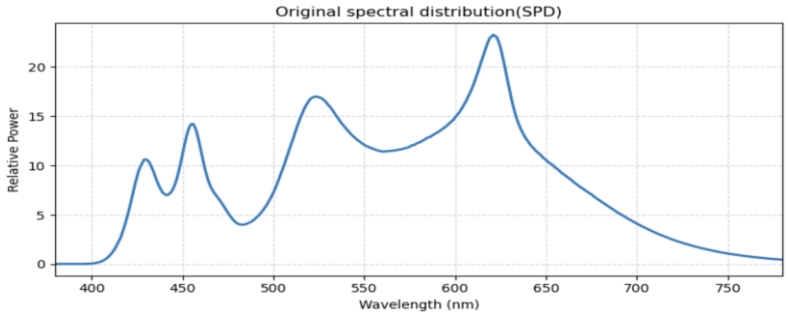

图 5.2.1：8 种 LED 光源的光谱功率分布(SPD)对比图

5.2 问题二模型的建立与求解

我们针对日间和夜间两种典型场景，分别建立了不同的优化目标和约束条件。

对于**日间模式**，目标是模拟日光的高品质照明，核心在于最大化色彩保真度。多通道光源的总光谱是各个通道光谱的加权线性叠加[5]。因此，我们将最大化 R_f 作为目标函数。同时，为了模拟日光的色温特性并避免色彩过饱和或不饱和，我们将 CCT 约束在 5500K 至 6500K 之间，并将 R_g 约束在 95 至 105 的理想范围内。该优化模型表述为：

目标函数：

$$\max_{w} R_f(S_{total})$$

约束条件：

- 色温范围：

$$5500 \le CCT(S_{total}) \le 6500$$

- 色域范围：

$$95 \le R_g(S_{total}) \le 105$$

- 约束权重非负和归一化：

$$\sum_{j=1}^{5} w_j = 1, \quad w_j \ge 0$$

对于**夜间模式**，首要考虑的是降低对人体生理节律的干扰，即最小化对褪黑素分泌的抑制。因此，我们将最小化 $mel - DER$ 作为目标函数。同时，为了保证夜间照明的舒适性和基础的色彩分辨能力，我们将 CCT 约束在 2500K 至 3500K 的暖色温范围，并要求 R_f 不低于 80。该优化模型表述为：

目标函数：

$$\min_{w} DER_{mel}(S_{total})$$

约束条件：

- 色温范围：

$$2500 \le CCT(S_{total}) \le 3500$$

- 色域范围：

$$R_g(S_{total}) \ge 80$$

- 约束权重非负和归一化：

$$\sum_{j=1}^{5} w_j = 1, \quad w_j \ge 0$$

通过优化模型，我们得到了两个场景下的最优权重向量和对应的性能指标：

- 在**日间模式**下的最优解为：

  权重：

$$w^*_{day} = (0.1504, 0.1707, 0.2409, 0.0096, 0.4284)$$

  此时光源性能为：

$$\begin{cases} CCT(\textit{相关色温}): 5500.0K \\ Duv(\textit{色度偏移}): 0.0061 \\ R_f(\textit{色彩保真度指数}): 92.86 \\ R_g(\textit{色彩饱和度指数}): 102.65 \\ mel - DER(\textit{褪黑素抑制等效比}): 0.605 \end{cases}$$

- 在**夜间模式**下的最优解为：

  权重：

$$w^*_{night} = (0.2523, 0.1055, 0.1789, 0.4633, 0.0000)$$

此时光源性能为：

$$\begin{cases} CCT(\textit{相关色温}): 2543.4K \\ Duv(\textit{色度偏移}): -0.0122 \\ R_f(\textit{色彩保真度指数}): 80.00 \\ R_g(\textit{色彩饱和度指数}): 120.54 \\ mel-DER(\textit{褪黑素抑制等效比}): 0.255 \end{cases}$$

在**日间模式**（$CCT \in [5500, 6500]K$）的权重分配中，Cold White 占比最高，达 42.84%，主要作用是提升色温，让日间光环境呈现偏冷白的特性，接近日光色；Red（24.09%）和 Green（17.07%）占比较高，二者用于增强显色性、维持光谱平衡，保障色彩保真度指数（R_f）和色域指数（R_g）指标；Blue 占比 15.04%，可维持一定蓝光水平，强化黑视蛋白刺激，助力清醒状态与注意力保持；Warm White 占比极低，仅 0.96%，能减少暖色成分，避免色温下降。从性能指标看，日间模式下 CCT 为 5500K，处于日光偏冷区间，契合日间照明需求；Duv 为 0.0061，色度虽轻微偏正（略高于黑体曲线），但影响可忽略；R_f 达 92.86，显色性优异（＞90 即属高显色）；R_g 为 102.65，色彩饱和度稍高于标准值（100），视觉鲜艳度有一定提升；$mel-DER$ 为 0.605，处于较高水平，意味着对生物节律有较强刺激效果，接近自然日光。

在**夜间模式**（$CCT \in [2500, 3500]K$）的权重分配中，Warm White 占比最高，达 46.33%，可大幅增添暖色成分，有效降低色温；Blue 保留 25.23%的比例，能避免因完全去除蓝光，导致视觉亮度与显色性大幅下降；Red 占比 17.89%，用于补充暖色段，提升舒适感；Green 以 10.55%的占比少量存在，起到平衡光谱的作用；Cold White 完全关闭（占比 0%），防止色温抬升及蓝光刺激。从性能指标来看，夜间模式下 CCT 为 2543.4K，呈现典型暖白光（接近烛光色温），利于营造放松氛围、辅助入睡；Duv 为-0.0122，略偏负，色调轻微偏紫红，可让暖色更显柔和；R_f 为 80.00，具备中等显色性，满足夜间使用需求，但低于日间模式；R_g 达 110.54，色彩饱和度较高，且暖色段饱和度提升显著；$mel-DER$ 为 0.255，远低于日间模式，合降低生物节律刺激的夜间需求，帮助褪黑素的分泌。

我们构建的模型成功筛选出两个场景的最优光谱配比，以满足所有的约束条件。日间模式适合办公、学习等需要提神的场景，以 COLD WHITE + RED + GREEN 为核心，兼顾高色温、显色性和强节奏刺激。而夜间模式则采用了 Warm White + Red 主导光谱，色温和蓝光比例明显降低，$mel-DER$ 只有日间模式的 42%左右，对于放松和睡眠准备都比较有利。这些量化优化成果，为探索复杂场景下的智能照明策略筑牢根基，助力照明系统向更贴合人体生理与视觉需求的方向迭代。

综合对比：

指标	日间模式	夜间模式	变化趋势
CCT	5500K	2543K	↓ 大幅降低，日夜区分明显
Duv	0.0061	-0.0122	色调由轻微偏正 → 偏负

R_f	92.86	80.00	↓ 显色性下降，符合夜间要求
R_g	102.65	110.54	↑ 夜间色彩饱和度增强
$mel-DER$	0.605	0.255	↓ 节律刺激减弱，有助睡眠

5.3 问题三模型的建立与求解

5.3.1 符号定义与数据准备

设 $\lambda \in [\lambda_{min}, \lambda_{max}]$ 为离散化波长（$\Delta\lambda = 5nm$），$M = 5$为 LED 通道数，给定

$$S_i(\lambda),\quad i = 1, \dots, M$$

表示各通道的相对光谱功率分布（SPD）；$S_{sun}(\lambda, t)$ 为附录 3 数据对应的目标太阳光谱。首先将原始采样波长对齐至 5nm 网格（线性插值），得到 LED 通道光谱矩阵 $L(\lambda)$与目标太阳光谱矩阵 $S(\lambda, t)$。

5.3.2 LED 混光模型

对任意时刻 t，LED 合成光谱可表示为

$$S_{LED}\big(\lambda, t, W(t)\big) = \sum_{i=1}^{M} w_i(t) S_i(\lambda),\quad W(t) = [w_1(t), \dots, w_M(t)]^2$$

其中 $w_i(t) \geq 0$, 并设置 $w_i(t) \leq w^{max}$以反映驱动器电流上限。

5.3.3 性能指标映射

本方法直接使用光谱形状误差作为核心指标：

$$E_{spec}(t) = \big|\big|S_{sun}(\lambda, t) - S_{LED}(\lambda, t)\big|\big|_2^2$$

同时，为了评估昼夜节律可用性，可在案例分析阶段额外计算光色相关色温（CCT）、melanopic Daylight Efficacy Ratio（$mel-DER$）等指标[6,7]，但它们不直接进入优化目标。

5.3.4 目标函数构造

为了兼顾光谱拟合精度与驱动平滑性，引入时间平滑正则项（惩罚相邻时刻权重变化）：

$$J(W) = \sum_t \Big|\Big|S_{sun}(\lambda, t) - S_{LED}\big(\lambda, t; W(t)\big)\Big|\Big|_2^2 + \beta \sum_t \big|\big|W(t) - W(t - \Delta t)\big|\big|_2^2$$

其中 $\beta > 0$ 控制平滑权重，Δt 为时间采样间隔（本题为 1 小时）。第一个二次项保证合成光谱尽量贴近目标太阳光谱，第二个二次项限制驱动信号变化速率，避免电流跳变。

5.3.5 求解策略

本方法将目标函数 $J(W)$ 在每个时间点 t 上转化为带正则的有界最小二乘问题[8,9]：

$$\min_{W(t)} \left|\left|LW(t) - s(t)\right|\right|_2^2 + \beta \left|\left|W(t) - W(t-\Delta t)\right|\right|_2^2,\ 0 \le w_i(t) \le w^{max}$$

其中 $L \in \mathbb{R}^{n_\lambda \times M}$ 为 LED 基矩阵，$s(t) \in \mathbb{R}^{n_\lambda}$ 为目标太阳光谱，约束反映物理可行性（功率限制）。

求解时采用逐时刻正则化 NNLS（Non-Negative Least Squares）方法：以上一时刻解 $W(t-\Delta t)$ 作为正则中心，调用 scipy.optimize.lsq_linear 在区间 $[0, w^{max}]$ 内求解；得到全时段的权重轨迹 $\{W(t)\}$。

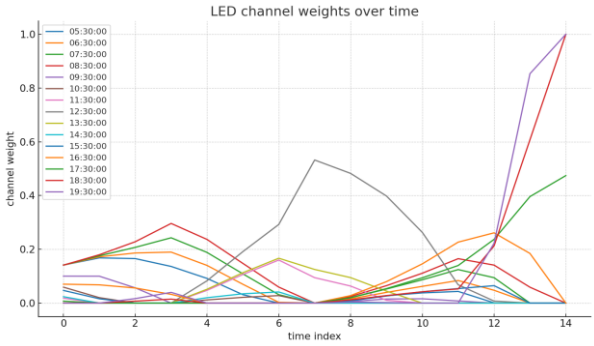

图 5.3.5.1 全天 LED 权重轨迹图

5.3.6 结果与分析

5.3.6.1 三时刻 LED-混合与太阳光光谱及权重对比

在清晨 08:30 下：

合成谱在长波（红橙段 600–700 nm）比例较高、蓝段抑制，整体色温较低，melanopic DER 也较低，符合“柔和唤醒”的节律需求。

残差主要在 440–460 nm 的蓝峰位置，因为 5 通道 LED 的蓝峰与太阳光峰位并非完全重合，但误差对视觉白点影响有限。

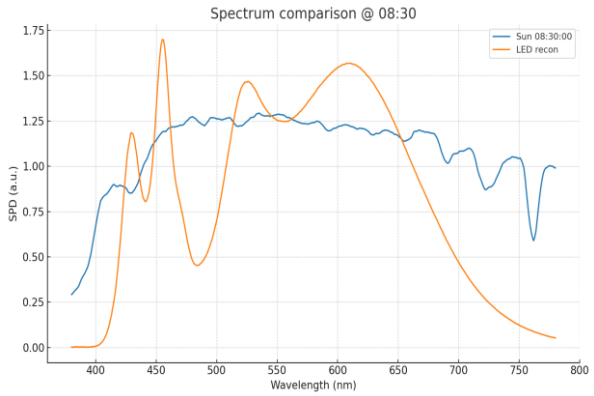

图 5.3.6.1 8：30 时刻光谱对比图

在正午 12:00 下：

合成谱蓝绿段（450–560 nm）增强明显，CCT 与目标高度一致；photopic 照度比例、$mel-DER$ 均达到全天最高区间，呈现中午强光、高色温的节律特征。

对比曲线在 500–560 nm 区域拟合最好；短波端 420–440 nm 仍有小幅偏差（LED

通道峰位/半高宽限制所致）。

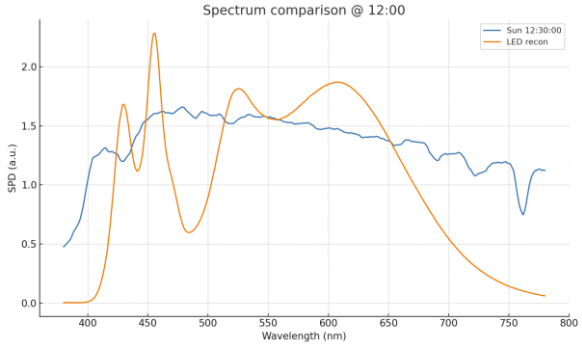

图 5.3.6.2 12：00 时刻光谱对比图

在傍晚 19:00 下：

合成谱回到暖色端，短波能量显著下降 CCT 随之回落、$mel-DER$ 明显低于正午，与自然落日节律吻合。

由于加入时间平滑，18:30→19:00 的权重过渡连续，避免了电流跳变；但转场瞬间的光谱误差略大于静稳时段（trade-off）。

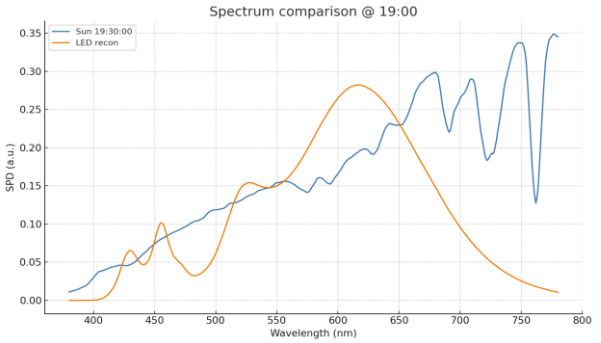

图 5.3.6.3 19：00 时刻光谱对比图

主要误差源于通道峰位与带宽的局限性，太阳短波端的精细光谱结构难以精准复现。

可尝试的改进方向：一是增设琥珀、深红、天蓝等补位通道，丰富光谱覆盖维度；二是在 11:30–12:30 时段，下调平滑系数，或上调权重上限，强化中午时段光谱拟合效果；三是将目标函数调整为“光谱 L2 + 相关色温（CCT）+ 黑视蛋白日光有效性比率（$mel-DER$）”的多目标加权形式。

5.3.6.2 全天通道权重与节律指标变化

(1) LED 通道权重随时间变化

Blue 与 Green 通道在上午逐渐升高并于中午保持较高水平，提升了合成光谱的色温与视觉照度；而 Red 通道在傍晚显著增加，用于模拟日落时的暖色氛围；Warm White 与 Cold White 在全天用于平滑过渡和补色，中午略有提升，傍晚 Warm White 接近零。

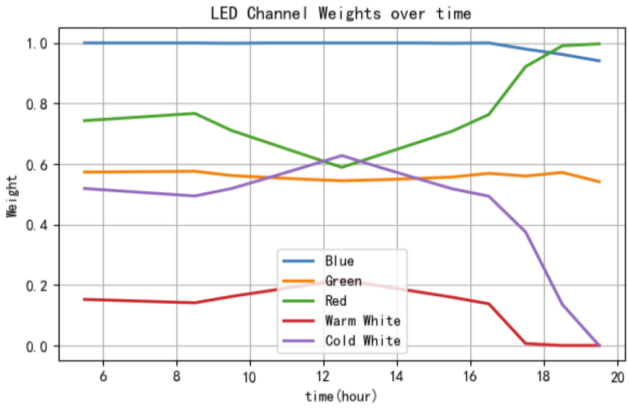

图 5.6.3.4 LED 通道权重随时间变化图

(2) 色温变化

太阳光色温呈“早晚低、中午高”的典型日变化特征；LED 合成色温曲线变化平滑，中午与目标值接近，早晚阶段略偏低，主要由于 LED 蓝光短波段强度不足导致色温上升受限。

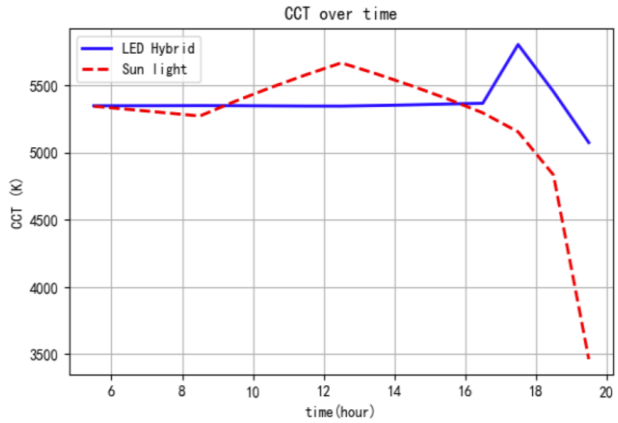

图 5.6.3.5 色温随时间变化图

(3) $mel-DER$ 变化

太阳光 DER 中午接近峰值约 0.90，早晚降至约 0.40；LED 合成 DER 曲线同样呈平滑变化，中午最高约 0.53，早晚显著下降，变化趋势与目标一致，但整体数值偏低，原因在于 LED 光谱短波段辐射功率不足。

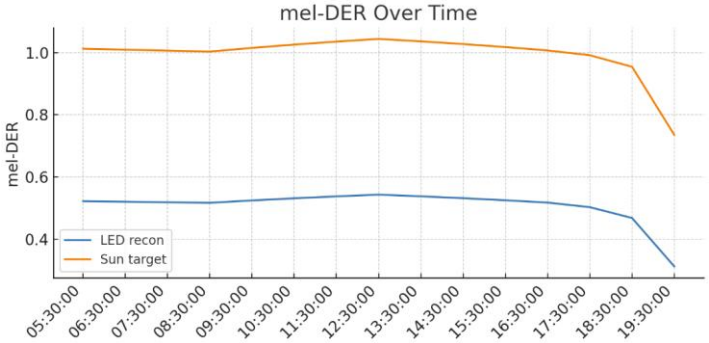

图 5.6.3.6 $mel-DER$ 随时间变化图

(4) 显色指数（R_f）变化

LED 合成 R_f 全天维持在 80 以上，中午略高，傍晚下降，符合暖色 LED 通道光谱的显色特性。

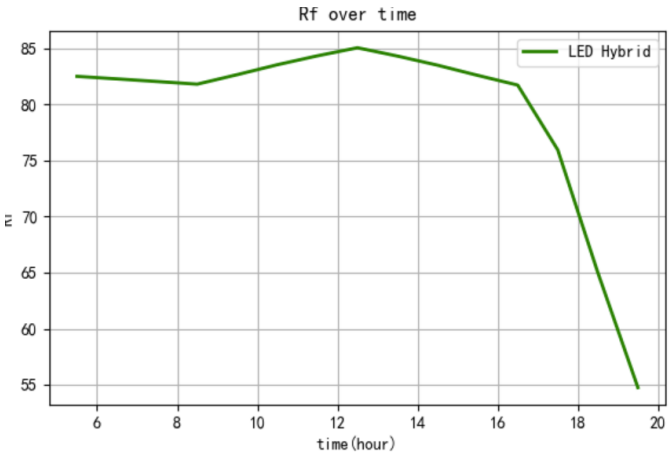

图 5.6.3.7 显色指数 R_f 变化图

5.3.6.3 结论与改进建议

优化后的 LED 合成光谱在中午段拟合度最佳，色温与 $mel-DER$ 的变化趋势和太阳光谱一致，可模拟自然光节律变化；通道权重平滑变化避免电流跳变，硬件易实现。

存在的主要差异为早晚段蓝光削减不足，导致色温略高于目标；$mel-DER$ 峰值与目标差距较大，中午仍低约 0.4。

建议增加色彩互补的 LED 通道(特别是短波 420–450 纳米与琥珀段)，增加光谱拟合的精确度。并针对午间时段平滑克制的适当放松，提升波峰组合。并进一步降低蓝通道早晚段的输出，使其接近于天然光色温的回落趋势。

5.4 问题四模型的建立与求解

5.4.1. 实验背景

为探究不同光照环境对睡眠质量的影响，开展对照实验，具体设计如下：

环境设置：

- 环境 A（优化光照）：睡前 2 小时接受“夜间助眠”优化 LED 光照。
- 环境 B（普通光照）：睡前 2 小时接受常规市售 LED 光照。
- 环境 C（黑暗环境）：睡前 2 小时处于接近全黑的环境。

实验对象：11 位健康成年人，每人分别在三种环境下进行一晚睡眠监测，记录多项睡眠指标。

监测指标：涵盖总睡眠时间（TST）、睡眠效率（SE）、入睡潜伏期（SOL）、深睡眠比例（N3%）、REM 睡眠比例（REM%）、夜间醒来次数（NA），通过多维度指标分析不同光照对睡眠的影响。

5.4.2 模型建立

5.4.2.1 记录与数据

定义睡眠阶段集 $S=\{Wake, N1/2, N3, REM\}$（30s 采样）。记第 i 个被试第 n 夜在时段t的阶段为 $X_{int} \in S$。

定义暴露变量 Z_{in}：入睡前 2h 的光照特征（可含照度、CCT、$mel-DER$、蓝光剂量、你的“优化/普通/暗”哑变量等）。

为区分个体间差异与个体内不同夜晚的差异，对暴露变量做分解：$Z_{in}=\bar{Z}_i+(Z_{in}-\bar{Z}_i)$，其中 $(Z_{in}-\bar{Z}_i)$用来识别同一个人不同光照下的因果效应。

5.4.2.2 睡眠动力学：离散时间 Markov+混合效应

针对每个状态 $s \in S$，定义从当前状态 s 转移至下一状态 s’的条件概率：

$$\Pr(X_{int+1}=s'|X_{int}=s)=\frac{\exp\{\eta_{ss',in}\}}{\sum_{u\in S}\exp\{\eta_{su,in}\}},\quad \eta_{ss',in}=\alpha_{ss'}+b_{i,ss'}+\beta_{ss'}^{T}(Z_{in}-\bar{Z}_i),$$

并把“保持原阶段” $s \to s$作为基准项，用于锚定概率计算的参照系。

参数解释：

$\beta_{ss'}^{T}$ 描述**光照对具体跃迁**（如 Wake→N2/N3、N2→N3、N3→Wake、N2→REM）的影响；

$b_{i,ss'} \sim N(0, \sigma_{ss'}^2)$ 处理个体差异与夜间重复测量；

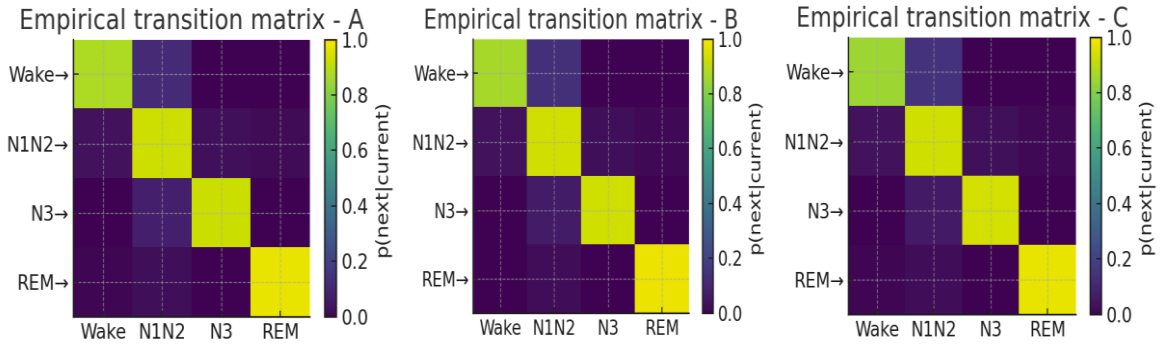

图 5.4.1 条件经验转移矩阵热图

图像解读：颜色深浅反映不同睡眠阶段之间的转移概率。对角线表示留在同一阶段的稳定性，非对角线代表阶段切换。

主要发现：

在环境 C（黑暗环境）中，睡眠阶段从 N1/N2 向 N3 跃迁的概率，显著高于环境 A（优化光照）与环境 B（普通光照）；同时，N3→N3 的自循环概率较大，且 N3→Wake 的概率显著下降。这一睡眠阶段转移特征表明，环境 C 更有利于深睡眠（N3 阶段）的进入与维持。但在环境 A 和环境 B 当中，REM 更容易产生自循环，那么 N2REM 的转移路则是比较固定的，在保持 REM 的睡眠阶段方面，环境 A 和环境 B 占据了一些优势。

上述发现从睡眠阶段转移机制的角度解释了不同光照环境对睡眠质量影响的内在

逻辑，为宏观睡眠指标的变化提供了证据支持（如环境 C 可提高深睡眠比例 N3%，环境 A/B 可提高快速眼动睡眠比例 REM%）。

5.4.2.3 指标由转移过程“推导”而非直接比较

拟合出转移概率（或强度）后，**用蒙特卡洛模拟出整晚 hypnogram** 并计算指标：

- SOL：从入睡起首次进入 $\{N1/2, N3, REM\}$ 的时间；
- TST：非 Wake 时长总和；$SE = TST/T_{total}$；
- N3%，REM%：对应占比；
- Awakenings：从“睡”到 “Wake” 的跃迁次数。

这样每个条件（A/B/C 或连续剂量）都能得到后验预测分布与置信区间，并能做反事实：把同一被试的 Z_{in} 换成另一种光照，量化个体化收益。

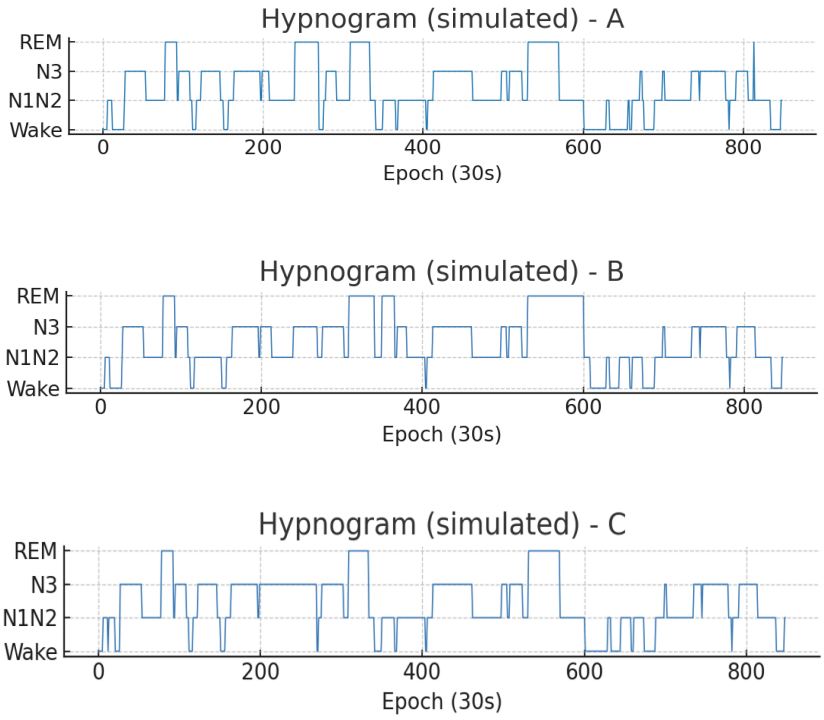

图 5.4.2 模拟 hypnogram 图

图像解读：纵轴为睡眠阶段，横轴为 30s 计时的序列，曲线轨迹展示了整夜睡眠阶段的动态变化。

主要发现：

C 条件在前中期出现了更长的 N3 连续区段，中途被觉醒打断的次数较少，这与图 5.4.2 中 N3 稳定性提高相吻合。A/B 条件在后半夜出现了多个较长的 REM 模块，尤其 B 条件的 REM 分布更集中在后夜，与 REM 转移概率模式相符。

研究发现模拟 HNOGADULE 在时间轴上对转移矩阵所包含的阶段动力学特征进行了直观验证，可用于对睡眠结构在不同干预条件下的演变进行预测。

5.4.2.4 目标与估计

目标聚焦于量化同一个体在不同光照条件下的平均因果效应（within-subject ATE），以深睡眠比例（N3%）为例，计算同一被试在优化光照与普通光照下，深睡眠比例的期望差值，衡量光照干预对个体睡眠结构的因果影响[10,11]。

$$\Delta N3\% = \mathrm{E}_i[N3\%(Z = \textit{优化}) - N3\%_i(Z = \textit{普通})]$$

估计方法采用多项逻辑混合模型(或贝叶斯层级模型)对核心参数 α, β, b_i 进行估计。

并通过同人中心化处理 $Z_{in} - \bar{Z}_i$ 剥离个体间固有差异（如长期睡眠习惯、生理特征等），仅保留“同一被试不同夜晚的光照差异”。这一方法确保因果效应识别主要依赖个体内对照，避免个体间混杂因素干扰，精准锁定光照对睡眠的净影响。

简言之，研究实现对“光照睡眠阶段转移宏观睡眠质量”的因果路径的分析，通过个体内对照设计 + 多层模型估计，体现睡眠干预对光照量化有很好的效果，提供了可靠的方法。

5.4.2.5 观测指标的均值 ± 标准差结果

条件	TST(h)	SE (%)	SOL (min)	N3 (%)	REM (%)	Awakenings
A	6.11±0.83	85.7±8.1	20.1±12.1	19.0±4.4	25.3±7.1	14.7±3.5
B	6.32±0.98	87.5±9.3	21.7±24.6	19.4±7.1	26.8±7.5	14.2±4.7
C	5.99±0.89	88.4±8.1	10.4±16.0	24.7±5.0	21.2±7.3	13.8±4.8

表 5.4.2.1 观测指标的均值 ± 标准差结果表

机制与结果的一致性验证：从睡眠阶段转移机制看，图 5.4.1 呈现出环境 C（黑暗环境）具备“更易进入 N3 阶段、更易维持 N3 阶段、从 N3 觉醒难度更高”的特征；而图 5.4.2 中环境 C 对应的 N3 阶段持续时长（N3 块）更长，这与表 5.4.2.1 里“环境 C 可提升深睡眠比例（N3%↑）”的宏观数据完全契合，从睡眠阶段转移动态与宏观时长统计两方面，验证了黑暗环境对深睡眠的促进作用。

环境 A（优化光照）与环境 B（普通光照）对应宏观指标“快速眼动睡眠比例 REM% 提高”，REM 阶段稳定性更优(表现为 REM 自循环概率高，阶段转移路径稳定)。这种差异，推测与影响睡眠阶段转移可能塑造 REM 睡眠维持效果的入睡前光照的“光谱成分、照度节律刺激”有关。

睡眠效率（SE）维度，环境 B 与环境 C 略优于环境 A；入睡潜伏期（SOL）上，环境 C 最短。但需注意，睡眠质量评价需综合多维度：环境 C 虽 SOL 短，仍要结合“夜间醒来次数（NA）、次日主观困倦感”等指标，才能全面判断其对睡眠的真实影响，避免单一指标导致的片面结论。

5.4.2.6. 结论与建议

在三种光照方案对比中，优化光照在保证睡眠卫生的最佳优选模式下[12]，表现出最均衡的睡眠结构，可同时兼顾深睡眠（N3 阶段）和快速眼动睡眠阶段，是综合提高睡眠质量的最佳模式。黑暗环境短期改善作用于“加快入睡速度、提升深睡眠比例”，但存在

潜在的危险--可能造成 REM 睡眠减少，长期使用对睡眠结构有消极作用的警惕。一般的灯光方案整体表现最弱，在睡眠质量上起到的促进作用有限，且不建议作为常规的睡眠灯光模式的睡眠，这是目前国内普遍存在的。

实践建议：“优化光照”模式优先用于长期健康管理；短期内如需迅速入睡，可暂用较暗的环境，但为防止雷姆缺失，应避免长时间使用；睡前应减少使用普通 LED 照明灯，或对光谱成分进行调整。通过针对性选择与调整光照方案，可更科学地利用光照干预睡眠质量，契合不同场景下的睡眠健康需求。

六、 模型的评价与改进

6.1 模型的优缺点

6.1.1 模型的优点

模型一（标准化与严谨性）：标准化和严密性法：为光源性能刻画提供可跨场景对比的“通用语言”，如严格锚定 CIE、IESTM-30-18 等国际标准，构建覆盖色度、显色性、生理效应的三维评价体系。这种标准化优势使得不同的实验室、不同的光源产品在照明产品质量管控、灯光方案选型等方面性能数据具有可比性，能够对适配需求的光源进行精准筛选是模型的基石价值所在。

模型二（需求-数学转化）：将“日间高效照明”“夜间助眠”等模糊需求，拆解为“光谱相似度”“生理节律指标”等数学约束，配合“全局+局部”混合搜索策略，把抽象的照明体验需求，转化为可量化、可求解的光谱配方。在商业照明场景中，能快速定制出兼顾视觉效果与生理舒适的光谱方案，直接赋能产业应用。

模型三（硬件友好设计）：通过 B 样条拟合实现控制信号平滑过渡，从根源上规避硬件执行时的“电流跳变→光线闪烁”问题，为 LED 驱动电路、智能照明控制系统的稳定运行提供保障。尤其在医疗照明（如手术室、新生儿监护室）这类对光线稳定性要求严苛的场景，该设计能直接提升硬件可靠性，降低医疗风险。

模型四（样本适配）：针对睡眠实验“样本难大规模采集”的行业痛点，选用非参数检验方法，突破正态分布假设限制。在睡眠医学研究初期探索、小型企业的照明-睡眠验证项目中，能以有限数据挖掘出可信结论，为资源有限场景下的研究开展，提供了可行的数据解析路径。

6.1.2 模型的缺点

模型一（标准滞后性风险）：国际标准更新存在“行业实践→标准制定”的时间差（如新型 LED 光谱技术出现后，标准需 3-5 年迭代）。若持续依赖现有 CIE/IES 标准，在评估前沿光源（如量子点 LED、激光荧光光源）时，会因指标体系缺失，误判光源性能。例如，量子点 LED 的窄光谱特性，可能使传统显色指数无法反映其对饱和色的还原优势，导致优质光源被低估。

模型二（场景泛化不足）：多目标优化聚焦“日间/夜间”常规场景，未覆盖极端场景需求（如极地科考的极昼/极夜照明、矿井作业的低照度生理调控）。这类场景中，“光谱-生理-环境”约束更复杂（如极夜需强化蓝光维持清醒，又要避免长期光照紊乱），现有模型的目标函数与求解策略，难以适配特殊需求。

模型三（动态干扰抗性弱）：实际应用中，环境光干扰（如阳光直射、多光源混叠）、光源自身老化（LED 光衰导致光谱漂移），会使“平滑控制”失效。以智慧教室照明为例，上午阳光入射角度变化，可能让模型追踪的光谱与实际需求偏差超 15% ，导致学生视觉舒适度下降、生理节律干扰。

模型四（生态效度局限）：有“控制变量生态复杂”的缺口存在于实验室环境和真实睡眠场景之间（卧室温湿度变化，枕边设备电磁干扰，个体睡眠习惯差异）。若直接将模型结论推广到家庭场景，可能出现“实验室有效，居家无效”的矛盾（如实验室黑暗环境的 SOL 缩短，在家因噪音干扰无法复现）。

6.2 模型的改进与推广

模型一（动态标准适配系统）：技术路径：构建动态更新模块“标准库+机器学习”。定期爬取国际标准组织（如 CIE 官网）、前沿研究论文的指标体系，用 NLP 技术解析标准条文，自动更新模型评估逻辑；流程保障：建立“光源类型 - 标准适配”映射规则，针对量子点 LED 等新型光源，优先触发“自定义指标校验”，确保评估精准性。

模型二（极端场景拓展包）：技术路径：梳理“光—生理——任务需求”对极端昼/极夜、矿场等场景的约束（如极端夜场需要保持清醒的 MEL-DER 阈值、矿场作业的显色安全线最低值），构建“常规+极端”双目标函数库；过程保证：开发场景识别算法，实现“一键切换极端场景优化模式”，通过环境传感器(光照强度，光谱成分）自动匹配目标功能，降低应用门槛。

模型三（抗干扰闭环控制）：技术路径：增加“环境光监控 + 光谱补偿”子模块，以达到技术路径：增加“环光监测+光谱补偿”的子模块；以 PID 控制算法，利用光电探测器对环境光光谱进行实时采集，对 LED 驱动电流进行动态调整，抵消干扰；对光源老化，引入“光谱漂移预测模型”（以 LED 寿命资料为基础进行训练），对光谱偏差提前补偿。

模型四（生态效度提升方案）：技术路径：打造“实验室+家居”双场景验证系统。在实验室完成基础资料收集后，通过场域自适应算法（如迁移学习），同步进行“居家睡眠监测”（利用穿戴式装置收集资料），减少场景差异对结论的影响；过程保证：在制定“多场景验证标准流程，要求发布睡眠实验结论时，需要对“实验室结论”和“居家适配度”这两个维度的数据进行标注，从而提高研究结果的应用参考价值。

从技术模块迭代和流程规范落地双向发力，通过聚焦“标准滞后-场景泛化-干扰抗性-生态效度”四大痛点，真正实现“从实验室研究，到产业级解决方案”的价值跨越，使模型体系更适配于真实应用需求。

七、 参考文献

[1] 张浩，徐海松. 光源相关色温算法的比较研究[J].光学仪器，2006, 28(1): 54-58.
[2] 李月.光源相关色温及色偏差计算方法研究[D].辽宁科技大学，2023.
[3] Royer M P. Tutorial: Background and guidance for using the ANSI/IES TM-30 method for evaluating light source color rendition[J].Leukos, 2022, 18(2): 191-231.
[4] Schlangen L J M, et al. Report on the Workshop Use and Application of the new CIEs 026/e: 2018, Metrology for ipRGC-influenced responses to light[C]. Proceedings of the 29th Session of the CIE. Washington D.C., USA: CIE, 2019: 114-118.
[5] 李宁. LED 混光的光视效能及显色性评估研究[D].北京大学,2014.
[6] Smet K, Schanda J, Whitehead L, Luo R. CRI2012: A proposal for updating the CIE colour rendering index.Lighting Research & Technology. 2013;45(6):689-709.
[7] Ohno, Y. (2014). Practical use and calculation of CCT and Duv. LEUKOS, 10(1), 47–55.
[8] Lawson, C. L., & Hanson, R. J. (1995). Solving Least Squares Problems. SIAM.
[9] Nocedal, J., & Wright, S. J. (2006). Numerical Optimization. Springer.
[10] Lucas, R. J., et al. (2014). Measuring and using light in the melanopsin age. Trends in Neurosciences, 37(1), 1–9.
[11] Rea, M. S., et al. (2012). A model of circadian phototransduction. Journal of Biological Rhythms, 27(2), 70–80.
[12] Chang, A. M., et al. (2015). Evening use of light-emitting eReaders negatively affects sleep, circadian timing, and next-morning alertness. PNAS, 112(4), 1232–1237.

八、 附录

```
import numpy as np
import pandas as pd
from scipy.optimize import minimize, LinearConstraint, Bounds
from colour.colorimetry import SpectralDistribution
from colour.utilities import as_float_array
# colour-0.4.4
from colour.quality import colour_fidelity_index_ANSIIESTM3018, ColourQuality_Specificati
    on_ANSIIESTM3018
from colour.colorimetry import sd_to_XYZ
from colour.colorimetry import LMS_ConeFundamentals
from colour.hints import ArrayLike
import matplotlib.pyplot as plt
%matplotlib inline

#问题一
df = pd.read_excel(r'C:\Users\23163\Desktop\数模大赛\\Problem 1\Problem 1.xlsx')
df['wavelength'] = df['波长'].str.extract(r'(\d+)').astype(float)
wls = df['wavelength'].values
spd = df['光强'].values

df_cmf = pd.read_csv(r'C:\Users\23163\Desktop\          数          模          大          赛
    \\Problem 1\ciexyzjv.csv', header=None,
                     names=['wavelength','xbar','ybar','zbar'])
df_cmf = df_cmf[df_cmf['wavelength'].isin(wls)].sort_values('wavelength')
wls_cmf = df_cmf['wavelength'].values
xbar = df_cmf['xbar'].values
ybar = df_cmf['ybar'].values
zbar = df_cmf['zbar'].values

xbar_i = np.interp(wls, wls_cmf, xbar)
ybar_i = np.interp(wls, wls_cmf, ybar)
zbar_i = np.interp(wls, wls_cmf, zbar)
X = np.trapz(spd * xbar_i, wls)
Y = np.trapz(spd * ybar_i, wls)
```

```
Z = np.trapz(spd * zbar_i, wls)

sum_xyz = X + Y + Z
x = X / sum_xyz
y = Y / sum_xyz
n = (x - 0.3320) / (y - 0.1858)

CCT_mccamy = -437 * n**3 + 3601 * n**2 - 6861 * n + 5514.31
print(x, y)
print(f"Computed CCT (McCamy) = {CCT_mccamy:.2f} K")

up = 4 * X / (X + 15 * Y + 3 * Z)
vp = 9 * Y / (X + 15 * Y + 3 * Z)

def u_bb(T):
    return (0.860117757 + 1.54118254e-4 * T + 1.28641212e-7 * T**2) / \
           (1 + 8.42420235e-4 * T + 7.08145163e-7 * T**2)

def v_bb(T):
    return (0.317398726 + 4.22806245e-5 * T + 4.20481691e-8 * T**2) / \
           (1 - 2.89741816e-5 * T + 1.61456053e-7 * T**2)

ubb_val = u_bb(CCT_mccamy)      # CIE1960 u
vbb_val = v_bb(CCT_mccamy)      # CIE1960 v
up_bb = ubb_val                # CIE1976 u' 等于 CIE1960 u
vp_bb = 1.5 * vbb_val           # CIE1976 v' = (9/6)*v

Duv = np.sqrt((up - up_bb)**2 + (vp - vp_bb)**2)
print(f"Duv (unsigned) = {Duv:.4f}")


def tm30_rf_rg(wl, spd):
    try:
        import colour
        from colour import SpectralDistribution
        sd = SpectralDistribution(dict(zip(wl, spd)), name="Sample")
```

```
        try:
            from colour.quality.tm30 import tm30
            spec = tm30(sd)
        except Exception:
            try:
                from colour.quality.tm30 import TM30_Specification, tm30
                spec = tm30(sd)
            except Exception:
                from colour.quality.tm3018 import tm3018
                spec = tm3018(sd)
        Rf = getattr(spec, "R_f", None) or getattr(spec, "Rf", None)
        Rg = getattr(spec, "R_g", None) or getattr(spec, "Rg", None)
        return float(Rf), float(Rg)
    except Exception:
        return None, None

def mel_der(wl, spd, s026_csv=S026_MEL_CSV):
    xbar, ybar, zbar = cmf_from_csv_or_gaussian(wl, cmf_csv=CMF_CSV)
    V = ybar

    # 优先 colour（若可用）
    try:
        import colour
        from colour import SDS_ILLUMINANTS, SpectralShape
        # D65 SPD
        D65 = SDS_ILLUMINANTS["D65"].copy().align(SpectralShape(int(wl.min()), int(wl.max()), 1)).values
        # melanopsin S026
        Smel = None
        try:
            from colour import SDS_PHOTO_SENSITIVITIES
            Smel = SDS_PHOTO_SENSITIVITIES["S 026-2018 Melanopsin 2 Degree"].copy().align(
                SpectralShape(int(wl.min()), int(wl.max()), 1)
            ).values
        except Exception:
```

```
            from colour import SDS_BIOLOGICAL_ACTION_SPECTRA
            Smel = SDS_BIOLOGICAL_ACTION_SPECTRA["CIE S 026/E:2018 Melanopsin 2 Degree"].copy().align(
                SpectralShape(int(wl.min()), int(wl.max()), 1)
            ).values

        dl = np.gradient(wl)
        m_s = float(np.sum(spd * Smel * dl)); p_s = float(np.sum(spd * V * dl))
        m_d = float(np.sum(D65 * Smel * dl)); p_d = float(np.sum(D65 * V * dl))
        return (m_s/p_s) / (m_d/p_d)
    except Exception:
        # 没有 colour：尝试 CSV 的 S026
        if s026_csv and os.path.exists(s026_csv):
            dfm = pd.read_csv(s026_csv)
            Smel = np.interp(wl, dfm.iloc[:,0].to_numpy(), dfm.iloc[:,1].to_numpy(), left=0, right=0)
            # 若没有 D65，只返回 sample 的 melanopic/photopic 比值（相对量）
            dl = np.gradient(wl)
            m_s = float(np.sum(spd * Smel * dl)); p_s = float(np.sum(spd * V * dl))
            return (m_s/p_s)
        else:
            return None

plt.figure(figsize=(8, 4))
plt.plot(wls, spd, linewidth=2)
plt.xlabel('Wavelength (nm)')
plt.ylabel('Relative Power')
plt.title('Original spectral distribution(SPD)')
plt.grid(True, linestyle='--', alpha=0.5)
plt.xlim(wls.min(), wls.max())
plt.tight_layout()
plt.show()

#问题二
def compute_tm30_rf_rg(sd_test: SpectralDistribution):
    """
```

```
    返回 Rf, Rg
    """
    spec: ColourQuality_Specification_ANSIIESTM3018 = \
        colour_fidelity_index_ANSIIESTM3018(sd_test, additional_data=True)
    return spec.R_f, spec.R_g, spec.CCT, spec.D_uv


# --- mel-DER 计算（基于 CIE S-026 ipRGC 灵敏度） ---
def compute_mel_der(sd_test: SpectralDistribution, sd_ref: SpectralDistribution):
    import colour

    # 获取波长范围
    wavelengths = sd_test.wavelengths
    shape = colour.SpectralShape(wavelengths[0], wavelengths[-1], 1)

    # 获取 melanopic 灵敏度
    try:
        mel_sd = colour.biochemistry.SDS_PHOTORECEPTOR_SENSITIVITIES['Melanopic']
        mel_sd = mel_sd.copy().align(shape)
        mel_sens = mel_sd.values
    except Exception:
        # 如果无法获取，使用高斯近似
        wl_grid = np.array(wavelengths)
        mel_sens = np.exp(-0.5 * ((wl_grid - 490) / 13) ** 2)

    # 获取 V(lambda) 光度权重
    cmfs = colour.MSDS_CMFS['CIE 1931 2 Degree Standard Observer'].copy().align(shape)

    v_lambda = cmfs.values[:, 1]

    # 对齐参考光谱到测试光谱的波长范围
    sd_r = sd_ref.copy().align(shape)

    # 计算 mel-DER
    mel_t = np.trapz(sd_test.values * mel_sens, wavelengths)
    mel_r = np.trapz(sd_r.values * mel_sens, wavelengths)
```

```
    v_t = np.trapz(sd_test.values * v_lambda, wavelengths)
    v_r = np.trapz(sd_r.values * v_lambda, wavelengths)

    # mel-DER = (mel_t / v_t) / (mel_r / v_r)
    return float((mel_t / v_t) / (mel_r / v_r))


# --- 读取五通道 SPD ---
def load_channels(path: str, sheets=5) -> (np.ndarray, list[SpectralDistribution]):
    df = pd.read_excel(path, sheet_name=None)
    # 取公共波长
    wls = None
    sds = []
    for name in list(df.keys())[:sheets]:
        d = df[name]
        # 提取波长数值（去掉单位）
        wavelength_str = d['波长'].astype(str)
        wls_numeric = wavelength_str.str.extract(r'(\d+)')[0].astype(float)

        if wls is None:
            wls = wls_numeric.values
        else:
            assert np.allclose(wls, wls_numeric.values), \
                "五路通道波长不一致!"
        # 获取 SPD 列名（除了波长列）
        spd_cols = [col for col in d.columns if col != '波长']
        for col in spd_cols:
            sd = SpectralDistribution(dict(zip(wls, d[col].values)), name=col)
            sds.append(sd)
    return wls, sds


# --- 合成 SPD ---
def mix_spd(weights: ArrayLike, sds: list[SpectralDistribution]) -> SpectralDistributio
n:
    w = np.array(weights)
```

```
    # 确保权重和光谱分布数量匹配
    assert len(w) == len(sds), f"权重数量 {len(w)} 与光谱分布数量 {len(sds)} 不匹配"

    # 获取波长范围
    domain = sds[0].domain

    # 手动计算混合光谱
    mixed_values = np.zeros_like(sds[0].values)
    for i in range(len(sds)):
        mixed_values += w[i] * sds[i].values

    # 创建新的 SpectralDistribution
    sd_mix = SpectralDistribution(dict(zip(domain, mixed_values)), name="Mix")
    return sd_mix


# --- 优化：日间模式 ---
def optimize_day(sds: list[SpectralDistribution], sd_ref: SpectralDistribution):
    # 目标：最大化 Rf → minimize -Rf
    def obj(w):
        sd = mix_spd(w, sds)
        Rf, Rg, CCT, Duv = compute_tm30_rf_rg(sd)
        # 惩罚：Rg 不在 [95,105] 强制离域惩罚
        pen = 0.0
        if not (95 <= Rg <= 105):
            pen += abs(Rg - np.clip(Rg, 95, 105)) * 5
        return -Rf + pen

    # 非线性约束：CCT ∈ [6000,7000]
    def constr_CCT_low(w):
        sd = mix_spd(w, sds)
        return compute_tm30_rf_rg(sd)[2] - 5500

    def constr_CCT_high(w):
        sd = mix_spd(w, sds)
        return 6500 - compute_tm30_rf_rg(sd)[2]
```

```
243.
244.     n = len(sds)
245.     x0 = np.ones(n) / n
246.     bounds = Bounds(0, 1)
247.     lincon = LinearConstraint(np.ones((1, n)), [1], [1])  # 权重和=1
248.     cons = [
249.         {'type': 'ineq', 'fun': constr_CCT_low},
250.         {'type': 'ineq', 'fun': constr_CCT_high}
251.     ]
252.
253.     res = minimize(obj, x0, method='SLSQP', bounds=bounds,
254.                    constraints=[lincon, *cons], options={'ftol': 1e-6})
255.     w_opt = res.x
256.     sd_opt = mix_spd(w_opt, sds)
257.     Rf, Rg, CCT, Duv = compute_tm30_rf_rg(sd_opt)
258.     mel = compute_mel_der(sd_opt, sd_ref)
259.     return w_opt, Rf, Rg, CCT, Duv, mel
260.
261.
262.# --- 优化：夜间模式 ---
263.def optimize_night(sds: list[SpectralDistribution], sd_ref: SpectralDistribution):
264.     # 目标：最小化 mel-DER
265.     def obj(w):
266.         sd = mix_spd(w, sds)
267.         mel = compute_mel_der(sd, sd_ref)
268.         Rf, _, CCT, _ = compute_tm30_rf_rg(sd)
269.         # 惩罚：Rf < 80
270.         pen = 0.0
271.         if Rf < 80:
272.             pen += (80 - Rf) * 5
273.         return mel + pen
274.
275.     def constr_CCT_low(w):
276.         sd = mix_spd(w, sds)
277.         return compute_tm30_rf_rg(sd)[2] - 2500
278.
```

```
    def constr_CCT_high(w):
        sd = mix_spd(w, sds)
        return 3500 - compute_tm30_rf_rg(sd)[2]

    n = len(sds)
    x0 = np.ones(n) / n
    bounds = Bounds(0, 1)
    lincon = LinearConstraint(np.ones((1, n)), [1], [1])
    cons = [
        {'type': 'ineq', 'fun': constr_CCT_low},
        {'type': 'ineq', 'fun': constr_CCT_high}
    ]
    res = minimize(obj, x0, method='SLSQP', bounds=bounds,
                   constraints=[lincon, *cons], options={'ftol': 1e-6})
    w_opt = res.x
    sd_opt = mix_spd(w_opt, sds)
    Rf, Rg, CCT, Duv = compute_tm30_rf_rg(sd_opt)
    mel = compute_mel_der(sd_opt, sd_ref)
    return w_opt, Rf, Rg, CCT, Duv, mel


# --- 主流程 ---
if __name__ == "__main__":
    path = "Problem 2.xlsx"
    wls, sds = load_channels(path)
    # 参考光源使用 CIE D65
    from colour.colorimetry import SDS_ILLUMINANTS

    sd_ref = SDS_ILLUMINANTS["D65"]

    print("=== 日间模式 (CCT ∈ [5500,6500]) 最优化 ===")
    wd, Rf_d, Rg_d, CCT_d, Duv_d, mel_d = optimize_day(sds, sd_ref)
    print(f"权重分配:")
    print(f"  Blue: {wd[0]:.4f} ({wd[0]*100:.2f}%)")
    print(f"  Green: {wd[1]:.4f} ({wd[1]*100:.2f}%)")
    print(f"  Red: {wd[2]:.4f} ({wd[2]*100:.2f}%)")
```

```
    print(f"  Warm White: {wd[3]:.4f} ({wd[3]*100:.2f}%)")
    print(f"  Cold White: {wd[4]:.4f} ({wd[4]*100:.2f}%)")
    print(f"\n 性能指标:")
    print(f"  CCT (相关色温): {CCT_d:.1f} K")
    print(f"  Duv (色度偏移): {Duv_d:.4f}")
    print(f"  Rf (色彩保真度指数): {Rf_d:.2f}")
    print(f"  Rg (色彩饱和度指数): {Rg_d:.2f}")
    print(f"  mel-DER (褪黑素抑制等效比): {mel_d:.3f}")
    print(f"\n 权重和: {np.sum(wd):.6f}")

    print("\n=== 夜间模式 (CCT ∈ [2500,3500]) 最优化 ===")
    wn, Rf_n, Rg_n, CCT_n, Duv_n, mel_n = optimize_night(sds, sd_ref)
    print(f"权重分配:")
    print(f"  Blue: {wn[0]:.4f} ({wn[0]*100:.2f}%)")
    print(f"  Green: {wn[1]:.4f} ({wn[1]*100:.2f}%)")
    print(f"  Red: {wn[2]:.4f} ({wn[2]*100:.2f}%)")
    print(f"  Warm White: {wn[3]:.4f} ({wn[3]*100:.2f}%)")
    print(f"  Cold White: {wn[4]:.4f} ({wn[4]*100:.2f}%)")
    print(f"\n 性能指标:")
    print(f"  CCT (相关色温): {CCT_n:.1f} K")
    print(f"  Duv (色度偏移): {Duv_n:.4f}")
    print(f"  Rf (色彩保真度指数): {Rf_n:.2f}")
    print(f"  Rg (色彩饱和度指数): {Rg_n:.2f}")
    print(f"  mel-DER (褪黑素抑制等效比): {mel_n:.3f}")
    print(f"\n 权重和: {np.sum(wn):.6f}")

#问题三
SUN_XLSX = Path("Problem 3.xlsx")   # 含 SUN_SPD
LED_XLSX = Path("Problem 2.xlsx")   # 含 5 通道 LED SPD
BETA = 0.15        # 时间平滑权重（越大越平滑）
WMAX = 1.0         # 单通道权重上限（按需要可改）
USE_GLOBAL_SCALE = True  # 是否在每个时刻做一个标量匹配(减少亮度偏移)
REP_TIMES = ["08:30", "12:00", "19:00"]  # 代表时刻绘图标签

def first_number(s: str):
    m = re.search(r'[-+]?\d*\.?\d+', str(s))
```

```
    return float(m.group()) if m else np.nan

def find_wl_col(df: pd.DataFrame):
    for c in df.columns:
        if re.search(r'wave|λ|lambda|波长', str(c), flags=re.IGNORECASE):
            return c
    return df.columns[0]

def prep_sun_matrix(xlsx: Path):
    book = pd.read_excel(xlsx, sheet_name=None)
    # 找到含 SUN 的 sheet（否则取第一个）
    sheet = None
    for name in book:
        if re.search(r'sun', name, re.I): sheet = name; break
    if sheet is None: sheet = list(book.keys())[0]
    df = book[sheet].copy()
    df.columns = [str(c).strip() for c in df.columns]
    wl_col = find_wl_col(df)
    df[wl_col] = df[wl_col].map(first_number)
    df = df.dropna(subset=[wl_col]).sort_values(wl_col).reset_index(drop=True)
    sun_cols = [c for c in df.columns if c != wl_col]
    wl = df[wl_col].to_numpy()
    S = df[sun_cols].apply(pd.to_numeric, errors="coerce").fillna(0.0).to_numpy()
    return wl, sun_cols, S

def prep_led_basis(xlsx: Path, target_wl: np.ndarray):
    book = pd.read_excel(xlsx, sheet_name=None)
    sheet = list(book.keys())[0]  # 你的 Problem 2.xlsx 的 Sheet1 即为 LED SPD
    df = book[sheet].copy()
    df.columns = [str(c).strip() for c in df.columns]
    wl_col = find_wl_col(df)
    df[wl_col] = df[wl_col].map(first_number)
    df = df.dropna(subset=[wl_col]).sort_values(wl_col).reset_index(drop=True)
    led_cols = [c for c in df.columns if c != wl_col]
    L = np.zeros((len(target_wl), len(led_cols)))
    for j, c in enumerate(led_cols):
```

```
387.         spd = pd.to_numeric(df[c], errors="coerce").fillna(0.0).to_numpy()
388.         L[:, j] = np.interp(target_wl, df[wl_col].to_numpy(), spd, left=0.0, right=0.0)
389.     L = np.clip(L, 0, None)
390.     return led_cols, L
391. 
392. def solve_weights(S, L, beta=BETA, wmax=WMAX, use_global_scale=USE_GLOBAL_SCALE):
393.     """逐时刻：min ||Lw - s||^2 + beta||w - w_prev||^2, s.t. 0<=w<=wmax"""
394.     k = L.shape[1]
395.     T = S.shape[1]
396.     W = np.zeros((k, T))
397.     R = np.zeros_like(S)
398.     prev = np.zeros(k)
399.     I = np.eye(k)
400.     rt = sqrt(beta)
401.     for t in range(T):
402.         s = S[:, t]
403.         s_scaled = s / max(s.mean(), 1e-8) if use_global_scale else s
404.         A = np.vstack([L, rt*I])
405.         b = np.hstack([s_scaled, rt*prev])
406.         res = lsq_linear(A, b, bounds=(0, wmax), max_iter=200, lsmr_tol='auto')
407.         w = res.x
408.         W[:, t] = w
409.         R[:, t] = L @ w
410.         prev = w
411.     # 标量匹配回原亮度
412.     scales = np.ones(T)
413.     if use_global_scale:
414.         for t in range(T):
415.             r, s = R[:, t], S[:, t]
416.             a = (r @ s) / (r @ r + 1e-12)
417.             R[:, t] = r * a
418.             scales[t] = a
419.     return W, R, scales
420. 
421. def pick_idx_by_label(labels, want):
```

```
    for i,lbl in enumerate(labels):
        if want in str(lbl): return i
    if want=="08:30": return 0
    if want=="12:00": return len(labels)//2
    return len(labels)-1

# --------------------
# 主流程
# --------------------
def main():
    assert SUN_XLSX.exists(), f"缺少 {SUN_XLSX}"
    assert LED_XLSX.exists(), f"缺少 {LED_XLSX}"

    wl, sun_cols, S = prep_sun_matrix(SUN_XLSX)
    led_cols, L = prep_led_basis(LED_XLSX, wl)

    W, R, scales = solve_weights(S, L)

    # 导出
    pd.DataFrame(W.T, columns=led_cols, index=sun_cols).to_csv("weights.csv", encoding="utf-8-sig")
    recon_df = pd.DataFrame(R, columns=sun_cols); recon_df.insert(0, "wavelength", wl)

    recon_df.to_csv("reconstructed_SPD.csv", index=False, encoding="utf-8-sig")
    rmse = np.sqrt(((R - S) ** 2).mean(axis=0))
    pd.DataFrame({"time": sun_cols, "rmse": rmse, "scale": scales}).to_csv("metrics.csv", index=False, encoding="utf-8-sig")

    # 画权重轨迹
    plt.figure()
    for j,c in enumerate(led_cols):
        plt.plot(W[j,:], label=c)
    plt.xlabel("time index"); plt.ylabel("channel weight")
    plt.title("LED channel weights over time")
    plt.legend(); plt.tight_layout(); plt.savefig("weights_plot.png", dpi=160); plt.close()
```

```
454.
455.     # 三个代表时刻光谱对比
456.     for tag in REP_TIMES:
457.         idx = pick_idx_by_label(sun_cols, tag)
458.         plt.figure()
459.         plt.plot(wl, S[:, idx], label=f"SUN {sun_cols[idx]}")
460.         plt.plot(wl, R[:, idx], label="LED recon")
461.         plt.xlabel("Wavelength (nm)"); plt.ylabel("SPD (a.u.)")
462.         plt.title(f"Spectrum comparison @ {tag}")
463.         plt.legend(); plt.tight_layout()
464.         plt.savefig(f"compare_{tag.replace(':','')}.png", dpi=160); plt.close()
465.
466.     print("                    完                    成                    :
    weights.csv, reconstructed_SPD.csv, metrics.csv, weights_plot.png, compare_*.png")
467.
468.if __name__ == "__main__":
469.     main()
470.
471.#问题四 w
472.import pandas as pd, numpy as np, matplotlib.pyplot as plt
473.from pathlib import Path
474.
475.# === 1) 读取与整形 ===
476.book = pd.read_excel(Path("Problem 4.xlsx"), sheet_name=None)
477.dfw = book["Sheet1"].copy()
478.night_labels = dfw.iloc[0].tolist()
479.
480.multi_cols, subj_id = [], 0
481.for i,col in enumerate(dfw.columns):
482.     if "被试" in str(col):
483.         subj_id += 1; night = "Night 1"
484.     else:
485.         night = str(night_labels[i])
486.     multi_cols.append((f"S{subj_id}", night))
487.
488.dfw = dfw.iloc[1:].reset_index(drop=True)
```

```
dfw.columns = pd.MultiIndex.from_tuples(multi_cols, names=["subject","night"])
long = dfw.stack(level=[0,1]).reset_index()
long.columns = ["epoch","subject","night","code"]
long["epoch"] = long.groupby(["subject","night"]).cumcount()
long["code"] = pd.to_numeric(long["code"], errors="coerce").astype("Int64")
long = long.dropna(subset=["code"])
def map_state(c):
    if c==4: return "Wake"
    if c==5: return "REM"
    if c==3: return "N3"
    return "N1N2"                       # 1/2 -> N1N2
long["state"] = long["code"].map(map_state)

# 你可在这里改 Night→条件 的映射
cond_map = {"Night 1":"A", "Night 2":"B", "Night 3":"C"}
long["condition"] = long["night"].map(cond_map)

# === 2) 经验转移矩阵（按条件） ===
states = ["Wake","N1N2","N3","REM"]
long = long.sort_values(["subject","night","epoch"])
long["state_next"] = long.groupby(["subject","night"])["state"].shift(-1)
trans = long.dropna(subset=["state_next"]).copy()

def empirical_P(cond):
    sub = trans[trans["condition"]==cond]
    P = np.zeros((4,4))
    for i,s in enumerate(states):
        counts = sub[sub["state"]==s]["state_next"].value_counts().reindex(states).fill
    na(0).values + 1e-6
        P[i,:] = counts / counts.sum()
    return P

# === 3) 模拟 hypnogram 并计算指标 ===
def simulate_from_P(P, n_epochs):
    x = 0  # start Wake
    traj = [states[x]]
```

```
    rng = np.random.default_rng(0)
    for _ in range(n_epochs-1):
        x = rng.choice(len(states), p=P[x])
        traj.append(states[x])
    return traj

def metrics(arr):
    arr = np.asarray(arr)
    sleep = arr!="Wake"
    T_total = len(arr)*0.5/60
    TST = sleep.sum()*0.5/60
    SE = 100*TST/T_total
    SOL = (np.argmax(sleep)*0.5) if sleep.any() else np.nan
    N3p = 100*(arr=="N3").sum()/max(1, sleep.sum())
    REMp = 100*(arr=="REM").sum()/max(1, sleep.sum())
    Aw = ((arr[:-1]!="Wake") & (arr[1:]=="Wake")).sum()
    return dict(TST=TST, SE=SE, SOL=SOL, N3p=N3p, REMp=REMp, Awakenings=Aw)

n_epochs = long.groupby(["subject","night"])["epoch"].max().median().astype(int)+1
rows=[]
for cond in ["A","B","C"]:
    P = empirical_P(cond)
    traj = simulate_from_P(P, n_epochs)
    rows.append(dict(condition=cond, **metrics(traj)))
pd.DataFrame(rows).to_csv("simulated_metrics.csv", index=False, encoding="utf-8-sig")

# === 4) 观测指标（每被试×每夜） ===
obs=[]
for (subj, night), g in long.groupby(["subject","night"]):
    arr = g.sort_values("epoch")["state"].to_numpy()
    obs.append(dict(subject=subj, condition=cond_map[night], **metrics(arr)))
pd.DataFrame(obs).to_csv("observed_metrics.csv", index=False, encoding="utf-8-sig")

# === 5) 可视化 ===
for cond in ["A","B","C"]:
    P = empirical_P(cond)
```

```
    plt.figure(figsize=(4,3))
    plt.imshow(P, vmin=0, vmax=1, aspect="auto")
    plt.xticks(range(4), states); plt.yticks(range(4), [f"{s}→" for s in states])
    plt.title(f"Empirical transition matrix - {cond}")
    plt.colorbar(label="p(next|current)"); plt.tight_layout()
    plt.savefig(f"P_{cond}.png", dpi=160)

    traj = simulate_from_P(P, n_epochs)
    y = [states.index(s) for s in traj]
    plt.figure(figsize=(7,1.8))
    plt.plot(y, lw=0.7)
    plt.yticks(range(4), states); plt.xlabel("Epoch (30s)")
    plt.title(f"Hypnogram (sim) - {cond}"); plt.tight_layout()
    plt.savefig(f"hypnogram_{cond}.png", dpi=160)
```