

# 电商广告的视觉元素 组合效应研究

21计算广告 骆昕怡

2025年4月19日



# 目录

CONTENTS

1

研究背景

2

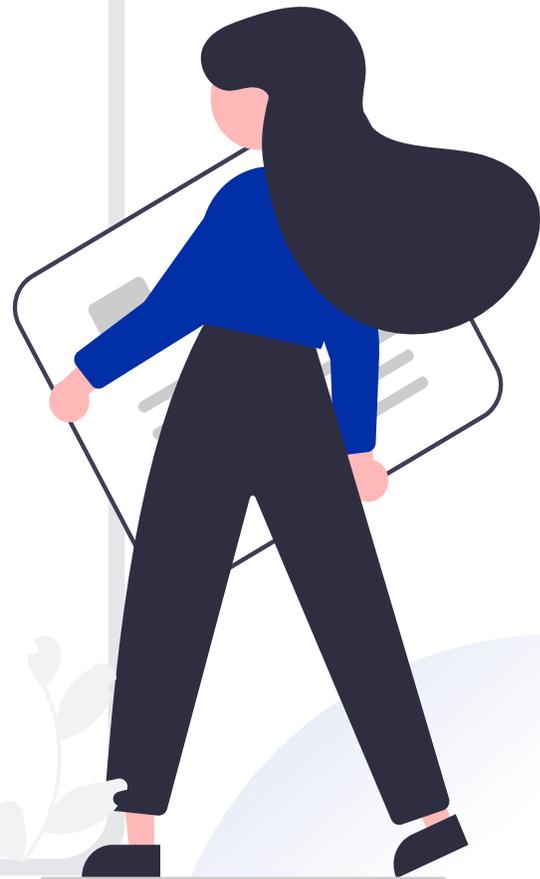
文献综述

3

研究内容

4

结论与建议



01



*PART ONE*

# 研究背景



# 研究背景

## 电商渗透率逐渐攀升，广告视觉呈现方式影响消费者决策

### 消费者

消费者对信息的接收呈现碎片化特征

### 电商平台

流量分发机制倒逼商家兼顾吸引力与转化效率

### 研究现状

当前关于视觉元素组合的研究多聚焦于单一维度，缺乏对多元素组合的量化分析

### 技术层面

深度学习与计算机视觉技术成熟，提供了技术可行性

因此本研究选择以图片素材为切入点，揭示视觉元素组合对消费决策的影响机制

# 研究意义

## 理论意义

现有电商视觉设计研究多聚焦单一元素的独立作用。本研究通过整合图像识别与行为数据，构建“元素组合-行为反馈”关联模型，突破传统单一维度分析局限，弥补了现有文献“定性多、定量少”的短板，揭示多元素组合对消费者决策的影响规律，为电商视觉精细化设计提供了新范式。

## 现实意义

广告素材设计长期面临两难困境：信息密度过高易引发认知负荷，简化设计则难以吸引用户。然而，商家多依赖主观经验或高成本A/B测试，缺乏普适性策略。本研究将通过量化元素组合与CTR关联性，指导商家制定设计策略，帮助平衡广告“吸引力”与“干扰性”，降低试错成本。

02



*PART TWO*

# 文献综述



# 文献综述

## 目前研究多聚焦于单一元素的独立作用

### 商品主体

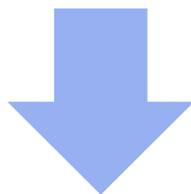
眼动实验发现，主体占比超过50%的图片能显著提升用户凝视时长，而过度放大主体可能导致信息冗余

### 文案设计

促销文案的情绪极性与购买意愿呈正相关，但文案长度存在“信息过载阈值”

### 赠品促销

赠品数量增加可短期刺激点击率，但可能分散用户对核心卖点的关注



因此本研究以电商广告为细分场景，构建“主体-文案-赠品”分析框架，通过整合图像识别与行为数据（CTR），揭示多元素组合的协同机制

03



*PART THREE*

# 研究内容



# 研究设计

## 数据获取

阿里云天池120万张电商广告及对应行为指标（展示量、点击量）

## 数据处理

主体：腾讯优图商品识别API，识别图片主体占比

文案：百度OCR接口，识别文案字数和文案占比

赠品：阿里云商品识别API，识别赠品数量和赠品占比

\*\*剔除脏数据、极端数据

## 提出假设

H1：广告图片中主体占比正向影响点击率

H2：文案字数与点击率呈倒U型关系

H3：赠品/优惠数量正向影响点击率

The image shows a CeraVe advertisement for 'Moisturising Cream' (保湿面霜 15ml). The ad features a blue background with white and yellow text. Key elements include the CeraVe logo, the product name, a price tag of ¥47, and a coupon for 'C乳236ml专享回购券'. Annotations in red boxes and text identify specific areas for analysis: '卖点文案' (Key Selling Point Copy) points to the product name and price; '主体' (Main Subject) points to the product image; '赠品权益' (Gift Benefit) points to the coupon. At the bottom, there is a large yellow banner with '可低至\* 9.9' and another '卖点文案' annotation pointing to the promotional text '抢!随身装30ml享47元券'.

# 研究设计

## 建模分析

通过相关分析及逻辑回归分析，验证假设成立

## 稳健性检验

采用替换模型法进行稳健性检验。Probit模型与逻辑回归模型的关键结论高度一致

$$\text{logit}(CTR) = \beta_0 + \beta_1 \cdot \text{主体占比} + \beta_2 \cdot \text{文案字数} + \beta_3 \cdot \text{文案占比} + \beta_4 \cdot \text{赠品数量} + \beta_5 \cdot \text{赠品占比} + \epsilon$$

变量	回归系数 ( $\beta$ )	标准误	优势比 (OR)	显著性 (p值)
主体占比	0.032**	0.004	1.033	0.002
文案字数	0.018*	0.007	1.018	0.013
文案占比	-0.025***	0.006	0.975	<0.001
赠品数量	0.041***	0.009	1.042	<0.001
赠品占比	-0.009	0.005	0.991	0.074
截距项	-3.721***	0.152	—	<0.001

# 研究创新点

## 研究视角创新

聚焦多元素组合效应，  
突破单一维度分析局限

出将广告素材图片拆解为商品主体、卖点文案、赠品权益三个核心元素，并构建其空间占比、信息密度、数量关系的量化指标体系

## 技术路径创新

跨平台工具整合与轻量化模型构建

通过腾讯优图API定位商品主体与赠品、百度OCR提取营销文案，结合天池公开数据集获取CTR行为指标构建逻辑回归模型

## 应用价值创新

提供可落地的设计策略

本研究通过数据分析提炼实证结论，指导商家针对广告素材特性制定广告视觉设计策略



04



*PART FOUR*

# 结论与建议



# 结论及建议

## 视觉元素的物理属性与空间布局共同影响用户决策

设计维度	推荐参数
主体占比	60%-75%
文案字数	8-15字
赠品数量	≤2项

1. 商品主体的视觉显著性对点击率具有决定性作用
2. 文案设计需平衡信息密度与可读性
3. 赠品权益的展示策略呈现“量效递减”特征

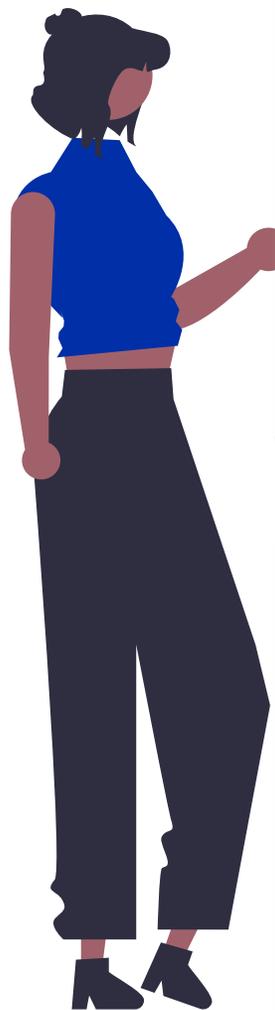
### 对商家的建议

- 主体展示应控制在60%-75%的占比区间
- 文案策略需“短而有力”，8-12字，占比8%以下
- 赠品控制数量（≤2项），切忌喧宾夺主

### 对平台的建议

- 提供自动排版工具
- 流量分配机制增加视觉权重
- 动态监测并引导优化

# 局限与展望



## 局限性

- 视觉元素维度的拆解未纳入颜色对比、动态效果等潜在影响因素
- 用户特征（如年龄、消费习惯）未被纳入模型，不同群体对视觉策略可能有偏好差异

## 未来展望

- 引入更多平台、补充品类划分
- 融合多模态技术，引入饱和度、构图、视觉动线等变量
- 构建行为数据-视觉设计的动态优化模型

05



*PART FIVE*

附录



# 附录

## 通过腾讯优图商品识别API，识别图片主体占比

```
import os
import json
import time
from PIL import Image
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.ticm.v20181127 import ticm_client, models

class ProductDetector:
    def __init__(self, secret_id, secret_key):
        cred = credential.Credential(secret_id, secret_key)
        http_profile = HttpProfile()
        http_profile.endpoint = "ticm.tencentcloudapi.com"
        client_profile = ClientProfile()
        client_profile.httpProfile = http_profile
        self.client = ticm_client.TicmClient(cred, "ap-guangzhou", client_profile)

    def detect_product(self, image_path):
        """
        执行商品检测并返回主体区域坐标
        :param image_path: 图片本地路径
        :return: (x_min, y_min, x_max, y_max) 主体区域坐标
        """
        try:
            # 读取图片尺寸
            with Image.open(image_path) as img:
                width, height = img.size

            # 构建API请求
            req = models.ImageModerationRequest()
            with open(image_path, "rb") as f:
                image_data = f.read()

            params = {
                "FileContent": image_data.decode('latin1'), # Base64编码
                "FileType": "PNG",
                "Size": "normal"
            }
```

## 通过百度OCR接口,识别文案字数和文案占比

```
1 import os
2 import re
3 import time
4 import requests
5 import pandas as pd
6 from PIL import Image
7 from io import BytesIO
8 from concurrent.futures import ThreadPoolExecutor
9
10 # 百度OCR配置 (需自行申请)
11 API_KEY = 'bce-v3/ALTAK-2uMwalesVyU5U8XU04re/65ad448072cd6184a8e898d67bad59bc591910f2'
12 SECRET_KEY = '6594de6aa156457fb3b745c18d0a4559'
13 OCR_URL = "https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic"
14
15 # 数据路径配置
16 IMAGE_DIR = "path/to/unzipped/images" # 解压后的图片目录
17 OUTPUT_FILE = "ocr_analysis_results.csv"
18 LOG_FILE = "processing_errors.log"
19
20 def get_access_token():
21     """获取百度OCR访问令牌"""
22     auth_url = f"https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id={API_KEY}&client_secret={SECRET_KEY}"
23     response = requests.get(auth_url)
24     return response.json().get("access_token")
25
26 def baidu_ocr(image_path, token):
27     """调用百度OCR接口"""
28     try:
29         with open(image_path, 'rb') as f:
30             img = f.read()
31
32         params = {"image": img}
33         headers = {'content-type': 'application/x-www-form-urlencoded'}
34         response = requests.post(
35             OCR_URL + "?access_token=" + token,
36             data=params,
37             headers=headers
```

# 附录

## 通过阿里云商品识别API,识别赠品数量和赠品占比

```
import base64
import pandas as pd
from PIL import Image
from shapely.geometry import box, Polygon
from shapely.ops import unary_union
from concurrent.futures import ThreadPoolExecutor

# 阿里云配置
ACCESS_KEY_ID = '2052316'
ACCESS_KEY_SECRET = 'sk-8b8a2aa28b9d47bc8535f4509c2e6da5'
ENDPOINT = 'imageseg.cn-shanghai.aliyuncs.com'
API_VERSION = '2019-12-30'

# 路径配置
IMAGE_DIR = "path/to/unzipped/images"
OUTPUT_FILE = "gift_analysis_results.csv"
LOG_FILE = "gift_processing_errors.log"

class AliyunAPI:
    def __init__(self):
        self.session = requests.Session()
        self.session.headers.update({
            'Accept': 'application/json',
            'Content-Type': 'application/json',
            'Date': time.strftime('%a, %d %b %Y %H:%M:%S GMT', time.gmtime())
        })

    def sign_request(self, params):
        """阿里云API签名"""
        sorted_params = sorted(params.items())
        canonicalized = '&'.join(['%s=%s' for k, v in sorted_params])
        string_to_sign = f'GET&%2F&{requests.utils.quote(canonicalized)}'
        hmac_key = f'{ACCESS_KEY_SECRET}&'.encode('utf-8')
        signature = base64.b64encode(hmac.new(hmac_key, string_to_sign.encode('utf-8'), hashlib.sha1).digest())
        return signature.decode('utf-8')

    def detect_objects(self, image_path):
```

## 逻辑回归模型

```
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor
from scipy import stats

# 数据加载
data = pd.read_csv("processed_data.csv")

def descriptive_analysis(df):
    """描述性统计分析"""
    desc = df.describe(percentiles=[.25, .5, .75]).T
    desc['skewness'] = df.skew()
    desc['kurtosis'] = df.kurt()
    return desc.round(3)

def correlation_analysis(df):
    """相关性分析矩阵"""
    plt.figure(figsize=(12, 8))
    corr_matrix = df.corr(method='spearman')
    mask = np.triu(np.ones_like(corr_matrix, dtype=bool))
    sns.heatmap(corr_matrix, mask=mask, annot=True, cmap='coolwarm',
                vmin=-1, vmax=1, fmt=".2f")
    plt.title("Spearman Correlation Matrix")
    plt.savefig('correlation_heatmap.png', dpi=300)
    return corr_matrix

def prepare_regression_data(df):
    """数据预处理"""
    # 计算点击率并创建二分类变量
    df['CTR'] = df['clicks'] / df['impressions']
    df['high_CTR'] = (df['CTR'] > df['CTR'].median()).astype(int)

    # 选择特征
    features = ['chinese_chars', 'text_coverage',
               'gift_count', 'gift_coverage']

    # 标准化处理
    scaler = StandardScaler()
```

# 恳请批评指正

21计算广告 骆昕怡

2025年4月19日

